lab4

August 11, 2025

[2]:

```
!pip install openeo rasterio folium geopandas sentinelhub matplotlib seaborn plotly requests
```

```
Requirement already satisfied: openeo in c:\users\manop\appdata\local\packages\p
ythonsoftwarefoundation.python.3.11_qbz5n2kfra8p0\localcache\local-
packages\python311\site-packages (0.43.0)
Requirement already satisfied: rasterio in c:\users\manop\appdata\local\packages
\pythonsoftwarefoundation.python.3.11_qbz5n2kfra8p0\localcache\local-
packages\python311\site-packages (1.4.3)
Requirement already satisfied: folium in c:\users\manop\appdata\local\packages\p
ythonsoftwarefoundation.python.3.11_qbz5n2kfra8p0\localcache\local-
packages\python311\site-packages (0.20.0)
Requirement already satisfied: geopandas in c:\users\manop\appdata\local\package
s\pythonsoftwarefoundation.python.3.11_qbz5n2kfra8p0\localcache\local-
packages\python311\site-packages (1.1.1)
Requirement already satisfied: sentinelhub in c:\users\manop\appdata\local\packa
ges\pythonsoftwarefoundation.python.3.11_qbz5n2kfra8p0\localcache\local-
packages\python311\site-packages (3.11.1)
Requirement already satisfied: matplotlib in c:\users\manop\appdata\local\packag
es\pythonsoftwarefoundation.python.3.11_qbz5n2kfra8p0\localcache\local-
packages\python311\site-packages (3.8.4)
Requirement already satisfied: seaborn in c:\users\manop\appdata\local\packages\
pythonsoftwarefoundation.python.3.11_qbz5n2kfra8p0\localcache\local-
packages\python311\site-packages (0.13.1)
Requirement already satisfied: plotly in c:\users\manop\appdata\local\packages\p
ythonsoftwarefoundation.python.3.11_qbz5n2kfra8p0\localcache\local-
packages\python311\site-packages (6.2.0)
Requirement already satisfied: requests in c:\users\manop\appdata\local\packages
\pythonsoftwarefoundation.python.3.11_qbz5n2kfra8p0\localcache\local-
packages\python311\site-packages (2.32.4)
Requirement already satisfied: urllib3>=1.9.0 in c:\users\manop\appdata\local\pa
ckages\pythonsoftwarefoundation.python.3.11_qbz5n2kfra8p0\localcache\local-
packages\python311\site-packages (from openeo) (2.1.0)
Requirement already satisfied: shapely>=1.6.4 in c:\users\manop\appdata\local\pa
ckages\pythonsoftwarefoundation.python.3.11_qbz5n2kfra8p0\localcache\local-
packages\python311\site-packages (from openeo) (2.1.1)
Requirement already satisfied: numpy>=1.17.0 in c:\users\manop\appdata\local\pac
kages\pythonsoftwarefoundation.python.3.11_qbz5n2kfra8p0\localcache\local-
```

```
packages\python311\site-packages (from openeo) (1.26.4)
Requirement already satisfied: xarray<2025.01.2,>=0.12.3 in c:\users\manop\appda
ta\local\packages\pythonsoftwarefoundation.python.3.11_qbz5n2kfra8p0\localcache\
local-packages\python311\site-packages (from openeo) (2025.1.1)
Requirement already satisfied: pandas>0.20.0 in c:\users\manop\appdata\local\pac
kages\pythonsoftwarefoundation.python.3.11_qbz5n2kfra8p0\localcache\local-
packages\python311\site-packages (from openeo) (2.2.2)
Requirement already satisfied: pystac>=1.5.0 in c:\users\manop\appdata\local\pac
kages\pythonsoftwarefoundation.python.3.11_qbz5n2kfra8p0\localcache\local-
packages\python311\site-packages (from openeo) (1.13.0)
Requirement already satisfied: deprecated>=1.2.12 in c:\users\manop\appdata\loca
l\packages\pythonsoftwarefoundation.python.3.11_qbz5n2kfra8p0\localcache\local-
packages\python311\site-packages (from openeo) (1.2.18)
Requirement already satisfied: oschmod>=0.3.12 in c:\users\manop\appdata\local\p
ackages\pythonsoftwarefoundation.python.3.11_qbz5n2kfra8p0\localcache\local-
packages\python311\site-packages (from openeo) (0.3.12)
Requirement already satisfied: packaging>=23.2 in c:\users\manop\appdata\local\p
ackages\pythonsoftwarefoundation.python.3.11_qbz5n2kfra8p0\localcache\local-
packages\python311\site-packages (from xarray<2025.01.2,>=0.12.3->openeo) (23.2)
Requirement already satisfied: affine in c:\users\manop\appdata\local\packages\p
ythonsoftwarefoundation.python.3.11_qbz5n2kfra8p0\localcache\local-
packages\python311\site-packages (from rasterio) (2.4.0)
Requirement already satisfied: attrs in c:\users\manop\appdata\local\packages\py
thonsoftwarefoundation.python.3.11_qbz5n2kfra8p0\localcache\local-
packages\python311\site-packages (from rasterio) (23.2.0)
Requirement already satisfied: certifi in c:\users\manop\appdata\local\packages\
pythonsoftwarefoundation.python.3.11_qbz5n2kfra8p0\localcache\local-
packages\python311\site-packages (from rasterio) (2023.11.17)
Requirement already satisfied: click>=4.0 in c:\users\manop\appdata\local\packag
es\pythonsoftwarefoundation.python.3.11_qbz5n2kfra8p0\localcache\local-
packages\python311\site-packages (from rasterio) (8.1.7)
Requirement already satisfied: cligj>=0.5 in c:\users\manop\appdata\local\packag
es\pythonsoftwarefoundation.python.3.11_qbz5n2kfra8p0\localcache\local-
packages\python311\site-packages (from rasterio) (0.7.2)
Requirement already satisfied: click-plugins in c:\users\manop\appdata\local\pac
kages\pythonsoftwarefoundation.python.3.11_qbz5n2kfra8p0\localcache\local-
packages\python311\site-packages (from rasterio) (1.1.1.2)
Requirement already satisfied: pyparsing in c:\users\manop\appdata\local\package
s\pythonsoftwarefoundation.python.3.11_qbz5n2kfra8p0\localcache\local-
packages\python311\site-packages (from rasterio) (3.1.1)
Requirement already satisfied: branca>=0.6.0 in c:\users\manop\appdata\local\pac
kages\pythonsoftwarefoundation.python.3.11_qbz5n2kfra8p0\localcache\local-
packages\python311\site-packages (from folium) (0.8.1)
Requirement already satisfied: jinja2>=2.9 in c:\users\manop\appdata\local\packa
ges\pythonsoftwarefoundation.python.3.11_qbz5n2kfra8p0\localcache\local-
packages\python311\site-packages (from folium) (3.1.3)
Requirement already satisfied: xyzservices in c:\users\manop\appdata\local\packa
ges\pythonsoftwarefoundation.python.3.11_qbz5n2kfra8p0\localcache\local-
```

```
packages\python311\site-packages (from folium) (2025.4.0)
Requirement already satisfied: pyogrio>=0.7.2 in c:\users\manop\appdata\local\pa
ckages\pythonsoftwarefoundation.python.3.11_qbz5n2kfra8p0\localcache\local-
packages\python311\site-packages (from geopandas) (0.11.1)
Requirement already satisfied: pyproj>=3.5.0 in c:\users\manop\appdata\local\pac
kages\pythonsoftwarefoundation.python.3.11_qbz5n2kfra8p0\localcache\local-
packages\python311\site-packages (from geopandas) (3.7.1)
Requirement already satisfied: aenum>=2.1.4 in c:\users\manop\appdata\local\pack
ages\pythonsoftwarefoundation.python.3.11_qbz5n2kfra8p0\localcache\local-
packages\python311\site-packages (from sentinelhub) (3.1.16)
Requirement already satisfied: dataclasses-json in c:\users\manop\appdata\local\
packages\pythonsoftwarefoundation.python.3.11_qbz5n2kfra8p0\localcache\local-
packages\python311\site-packages (from sentinelhub) (0.6.7)
Requirement already satisfied: oauthlib in c:\users\manop\appdata\local\packages
\pythonsoftwarefoundation.python.3.11_qbz5n2kfra8p0\localcache\local-
packages\python311\site-packages (from sentinelhub) (3.3.1)
Requirement already satisfied: pillow>=9.2.0 in c:\users\manop\appdata\local\pac
kages\pythonsoftwarefoundation.python.3.11_qbz5n2kfra8p0\localcache\local-
packages\python311\site-packages (from sentinelhub) (10.2.0)
Requirement already satisfied: python-dateutil in c:\users\manop\appdata\local\p
ackages\pythonsoftwarefoundation.python.3.11_qbz5n2kfra8p0\localcache\local-
packages\python311\site-packages (from sentinelhub) (2.8.2)
Requirement already satisfied: requests-oauthlib>=1.0.0 in c:\users\manop\appdat
a\local\packages\pythonsoftwarefoundation.python.3.11_qbz5n2kfra8p0\localcache\l
ocal-packages\python311\site-packages (from sentinelhub) (2.0.0)
Requirement already satisfied: tifffile>=2020.9.30 in c:\users\manop\appdata\loc
al\packages\pythonsoftwarefoundation.python.3.11_qbz5n2kfra8p0\localcache\local-
packages\python311\site-packages (from sentinelhub) (2025.6.11)
Requirement already satisfied: tomli in c:\users\manop\appdata\local\packages\py
thonsoftwarefoundation.python.3.11_qbz5n2kfra8p0\localcache\local-
packages\python311\site-packages (from sentinelhub) (2.2.1)
Requirement already satisfied: tomli-w in c:\users\manop\appdata\local\packages\
pythonsoftwarefoundation.python.3.11_qbz5n2kfra8p0\localcache\local-
packages\python311\site-packages (from sentinelhub) (1.2.0)
Requirement already satisfied: tqdm in c:\users\manop\appdata\local\packages\pyt
honsoftwarefoundation.python.3.11_qbz5n2kfra8p0\localcache\local-
packages\python311\site-packages (from sentinelhub) (4.67.1)
Requirement already satisfied: typing-extensions>=4.5.0 in c:\users\manop\appdat
a\local\packages\pythonsoftwarefoundation.python.3.11_qbz5n2kfra8p0\localcache\l
ocal-packages\python311\site-packages (from sentinelhub) (4.10.0)
Requirement already satisfied: utm in c:\users\manop\appdata\local\packages\pyth
onsoftwarefoundation.python.3.11_qbz5n2kfra8p0\localcache\local-
packages\python311\site-packages (from sentinelhub) (0.8.1)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\manop\appdata\local\
packages\pythonsoftwarefoundation.python.3.11_qbz5n2kfra8p0\localcache\local-
packages\python311\site-packages (from matplotlib) (1.2.0)
Requirement already satisfied: cycler>=0.10 in c:\users\manop\appdata\local\pack
ages\pythonsoftwarefoundation.python.3.11_qbz5n2kfra8p0\localcache\local-
```

```
packages\python311\site-packages (from matplotlib) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\manop\appdata\local
\packages\pythonsoftwarefoundation.python.3.11_qbz5n2kfra8p0\localcache\local-
packages\python311\site-packages (from matplotlib) (4.47.2)
Requirement already satisfied: kiwisolver>=1.3.1 in c:\users\manop\appdata\local
\packages\pythonsoftwarefoundation.python.3.11_qbz5n2kfra8p0\localcache\local-
packages\python311\site-packages (from matplotlib) (1.4.5)
Requirement already satisfied: narwhals>=1.15.1 in c:\users\manop\appdata\local\
packages\pythonsoftwarefoundation.python.3.11_qbz5n2kfra8p0\localcache\local-
packages\python311\site-packages (from plotly) (2.0.1)
Requirement already satisfied: charset_normalizer<4,>=2 in c:\users\manop\appdat
a\local\packages\pythonsoftwarefoundation.python.3.11_qbz5n2kfra8p0\localcache\l
ocal-packages\python311\site-packages (from requests) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in c:\users\manop\appdata\local\pack
ages\pythonsoftwarefoundation.python.3.11_qbz5n2kfra8p0\localcache\local-
packages\python311\site-packages (from requests) (3.6)
Requirement already satisfied: colorama in c:\users\manop\appdata\local\packages
\pythonsoftwarefoundation.python.3.11_qbz5n2kfra8p0\localcache\local-
packages\python311\site-packages (from click>=4.0->rasterio) (0.4.6)
Requirement already satisfied: wrapt<2,>=1.10 in c:\users\manop\appdata\local\pa
ckages\pythonsoftwarefoundation.python.3.11_qbz5n2kfra8p0\localcache\local-
packages\python311\site-packages (from deprecated>=1.2.12->openeo) (1.16.0)
Requirement already satisfied: MarkupSafe>=2.0 in c:\users\manop\appdata\local\p
ackages\pythonsoftwarefoundation.python.3.11_qbz5n2kfra8p0\localcache\local-
packages\python311\site-packages (from jinja2>=2.9->folium) (2.1.3)
Requirement already satisfied: pywin32 in c:\users\manop\appdata\local\packages\
pythonsoftwarefoundation.python.3.11_qbz5n2kfra8p0\localcache\local-
packages\python311\site-packages (from oschmod>=0.3.12->openeo) (306)
Requirement already satisfied: pytz>=2020.1 in c:\users\manop\appdata\local\pack
ages\pythonsoftwarefoundation.python.3.11_qbz5n2kfra8p0\localcache\local-
packages\python311\site-packages (from pandas>0.20.0->openeo) (2023.3.post1)
Requirement already satisfied: tzdata>=2022.7 in c:\users\manop\appdata\local\pa
ckages\pythonsoftwarefoundation.python.3.11_qbz5n2kfra8p0\localcache\local-
packages\python311\site-packages (from pandas>0.20.0->openeo) (2023.4)
Requirement already satisfied: six>=1.5 in c:\users\manop\appdata\local\packages
\pythonsoftwarefoundation.python.3.11_qbz5n2kfra8p0\localcache\local-
packages\python311\site-packages (from python-dateutil->sentinelhub) (1.16.0)
Requirement already satisfied: marshmallow<4.0.0,>=3.18.0 in c:\users\manop\appd
ata\local\packages\pythonsoftwarefoundation.python.3.11_qbz5n2kfra8p0\localcache
\local-packages\python311\site-packages (from dataclasses-json->sentinelhub)
(3.26.1)
Requirement already satisfied: typing-inspect<1,>=0.4.0 in c:\users\manop\appdat
a\local\packages\pythonsoftwarefoundation.python.3.11_qbz5n2kfra8p0\localcache\l
ocal-packages\python311\site-packages (from dataclasses-json->sentinelhub)
(0.9.0)
Requirement already satisfied: mypy-extensions>=0.3.0 in c:\users\manop\appdata\
local\packages\pythonsoftwarefoundation.python.3.11_qbz5n2kfra8p0\localcache\loc
al-packages\python311\site-packages (from typing-inspect<1,>=0.4.0->dataclasses-
json->sentinelhub) (1.1.0)
```

[3]:

```
import openeo
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import geopandas as gpd
import json
import rasterio
from datetime import datetime, timedelta
import folium
from sentinelhub import SHConfig, BBox, CRS, MimeType, DataCollection,␣
  ↪SentinelHubRequest, bbox_to_dimensions
import warnings
import plotly.express as px
import plotly.graph_objects as go
from plotly.subplots import make_subplots
import requests
from io import BytesIO
from PIL import Image
warnings.filterwarnings('ignore')
```

```
C:\Users\manop\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.11_qbz5n
2kfra8p0\LocalCache\local-packages\Python311\site-packages\tqdm\auto.py:21:
TqdmWarning: IProgress not found. Please update jupyter and ipywidgets. See
https://ipywidgets.readthedocs.io/en/stable/user_install.html
  from .autonotebook import tqdm as notebook_tqdm
```

[4]:

```
lago_atitlan = {
    "west": -91.326256,
    "east": -91.07151,
    "south": 14.5948,
    "north": 14.750979
}

lago_amatitlan = {
    "west": -90.638065,
    "east": -90.512924,
    "south": 14.412347,
    "north": 14.493799
}
start_date = "2024-01-01"
end_date = "2024-07-31"

print(f"Período de análisis: {start_date} a {end_date}")
```

```
duration = (datetime.strptime(end_date, "%Y-%m-%d") - datetime.
 ↪strptime(start_date, "%Y-%m-%d")).days
print(f"Duración: {duration} días ({duration/30:.1f} meses)")
```

```
Período de análisis: 2024-01-01 a 2024-07-31
Duración: 212 días (7.1 meses)
```

[5]:

```
def setup_real_connections():
    try:
        connection = openeo.connect("https://openeo.dataspace.copernicus.eu/")

        collections = connection.list_collections()
        sentinel_collections = []
        for c in collections:
            try:
                if 'sentinel' in c['id'].lower():
                    sentinel_collections.append(c['id'])
            except Exception:
                continue

        print(f"Colecciones Sentinel disponibles: {len(sentinel_collections)}")

        config = SHConfig()
        config.sh_client_id = 'c02eac78-6a59-4c33-b10e-bfbc596293f8'
        config.sh_client_secret = 'uy82TTniaWG15EYJsq2h8DRQpod3Beer'
        config.save()
        print("Configuración Sentinel Hub guardada")

        return connection, config
    except Exception as e:
        print(f"Error en configuración: {e}")
        return None, None
```

[6]:

```
cyanobacteria_evalscript = """
//VERSION=3
// Cyanobacteria detection script
// Based on official Sentinel Hub script from custom-scripts repository

function setup() {
    return {
        input: ["B02", "B03", "B04", "B05", "B06", "B07", "B08", "B8A", "B11",␣
 ↪"B12"],
        output: {
            bands: 4,
            sampleType: "FLOAT32"
        }
    };
}

// Water body detection function
function isWater(sample) {
    let mndwi = (sample.B03 - sample.B11) / (sample.B03 + sample.B11 + 0.001);
    let ndwi = (sample.B03 - sample.B08) / (sample.B03 + sample.B08 + 0.001);
    let ndvi = (sample.B08 - sample.B04) / (sample.B08 + sample.B04 + 0.001);

    // Water detection criteria
    return (mndwi > 0.42 || ndwi > 0.4 || ndvi < -0.2);
}

function evaluatePixel(sample) {
    let water = isWater(sample) ? 1.0 : 0.0;

    if (water > 0) {
        // NDCI (Normalized Difference Chlorophyll Index)
        let ndci = (sample.B05 - sample.B04) / (sample.B05 + sample.B04 + 0.
 ↪001);

        // FAI (Floating Algal Index)
        let fai = sample.B07 - sample.B04 - (sample.B8A - sample.B04) * (783 -␣
 ↪665) / (865 - 665);

        // Chlorophyll-a concentration (cyanobacteria proxy)
        let chl_a = Math.max(0, 826.57 * Math.pow(ndci, 3) - 176.43 * Math.
 ↪pow(ndci, 2) + 19 * ndci + 4.071);
        chl_a = Math.min(chl_a, 100); // Cap at 100  g/L

        // Additional spectral indices
        let ndvi = (sample.B08 - sample.B04) / (sample.B08 + sample.B04 + 0.
 ↪001);
        let ndwi = (sample.B03 - sample.B08) / (sample.B03 + sample.B08 + 0.
 ↪001);

        return [chl_a, ndci, fai, ndwi];
    } else {
        return [0.0, 0.0, 0.0, 0.0];
    }
}
"""
```

[7]:

```
def create_sentinel_hub_request(bbox, time_range, config):
    try:
        bbox_coords = BBox(
            bbox=[bbox['west'], bbox['south'], bbox['east'], bbox['north']],
            crs=CRS.WGS84
        )
        size = bbox_to_dimensions(bbox_coords, resolution=60)

        request = SentinelHubRequest(
            evalscript=cyanobacteria_evalscript,
            input_data=[
                SentinelHubRequest.input_data(
                    data_collection=DataCollection.SENTINEL2_L2A,
                    time_interval=time_range,
                    mosaicking_order='leastCC',
                    maxcc=0.3
                )
            ],
            responses=[
                SentinelHubRequest.output_response('default', MimeType.TIFF)
            ],
            bbox=bbox_coords,
            size=size,
            config=config,
            data_folder='data'
        )

        return request, size

    except Exception as e:
        print(f"Error creando solicitud Sentinel Hub: {e}")
        return None, None
```

[8]:

```
def download_sentinel_hub_data(request, lake_name):
    try:
        print(f"Descargando datos de Sentinel Hub para {lake_name}...")
        response = request.get_data(save_data=True)

        if response and len(response) > 0:
            data = response[0]

            print(f" Datos descargados para {lake_name}")
            print(f"  Forma: {data.shape}")
            print(f"  Tipo: {data.dtype}")
            chl_a = data[:, :, 0]
            ndci = data[:, :, 1]
            fai = data[:, :, 2]
            ndwi = data[:, :, 3]
            water_mask = (chl_a > 0).astype(float)

            return {
                'chl_a': chl_a,
                'ndci': ndci,
                'fai': fai,
                'ndwi': ndwi,
                'water_mask': water_mask,
                'shape': data.shape
            }
        else:
            print(f"No se obtuvieron datos para {lake_name}")
            return None

    except Exception as e:
        print(f"Error descargando datos para {lake_name}: {e}")
        return None
```

[9]:
```
def download_openeo_data(connection, bbox, start_date, end_date, lake_name):
    try:
        print(f"Descargando datos OpenEO para {lake_name}...")
        datacube = connection.load_collection(
            "SENTINEL2_L2A",
            spatial_extent={
                "west": bbox["west"],
                "east": bbox["east"],
                "south": bbox["south"],
                "north": bbox["north"]
            },
            temporal_extent=[start_date, end_date],
            bands=["B02", "B03", "B04", "B05", "B06", "B07", "B08", "B8A",␣
 ↪"B11", "B12"]
        )

        datacube = datacube.filter_bbox(**bbox)
        datacube = datacube.resample_spatial(resolution=60)
        job = datacube.create_job(
            title=f"Cyanobacteria_Analysis_{lake_name}",
            description=f"Descarga de datos Sentinel-2 para análisis de␣
 ↪cianobacterias en {lake_name}"
        )
        print("Ejecutando...")

        job.start_and_wait()

        if job.status() == "finished":
            results = job.get_results()
            print(f"Descarga completada para {lake_name}")
            return results
        else:
            print(f"Job falló para {lake_name}: {job.status()}")
            return None

    except Exception as e:
        print(f"Error OpenEO para {lake_name}: {e}")
        return None
```

[10]:

```
def process_real_data(data_dict, lake_name):
    if data_dict is None:
        return None

    try:
        chl_a = data_dict['chl_a']
        water_mask = data_dict['water_mask']
        water_pixels = chl_a[water_mask > 0]

        if len(water_pixels) > 0:
            stats = {
                'mean_chl_a': np.mean(water_pixels),
                'max_chl_a': np.max(water_pixels),
                'min_chl_a': np.min(water_pixels),
                'std_chl_a': np.std(water_pixels),
                'water_pixel_count': len(water_pixels),
                'total_pixels': chl_a.size,
                'water_percentage': len(water_pixels) / chl_a.size * 100
            }

            print(f"Clorofila-a promedio: {stats['mean_chl_a']:.2f}  g/L")
            print(f"Clorofila-a máxima: {stats['max_chl_a']:.2f}  g/L")
            print(f"Píxeles de agua: {stats['water_pixel_count']}␣
↪({stats['water_percentage']:.1f}%)")

            return stats
        else:
            print(f"No se detectaron píxeles de agua en {lake_name}")
            return None

    except Exception as e:
        print(f"Error procesando datos de {lake_name}: {e}")
        return None
```

[11]:

```
def download_temporal_data(connection, config, bbox, lake_name, start_date,␣
↪end_date, frequency='30D'):
    temporal_data = {}
    dates = pd.date_range(start=start_date, end=end_date, freq=frequency)

    print(f"\nDescargando serie temporal para {lake_name}")
    print(f"Fechas a procesar: {len(dates)}")
    for i, date in enumerate(dates):
        try:
            start_window = (date - timedelta(days=7)).strftime('%Y-%m-%d')
            end_window = (date + timedelta(days=7)).strftime('%Y-%m-%d')

            print(f"procesando fecha {i+1}/{len(dates)}: {date.
↪strftime('%Y-%m-%d')}")

            request, size = create_sentinel_hub_request(
                bbox,
                (start_window, end_window),
                config
            )

            if request:
                data = download_sentinel_hub_data(request, f"{lake_name}_{date.
↪strftime('%Y%m%d')}")

                if data:
                    temporal_data[date.strftime('%Y-%m-%d')] = {
                        'date': date,
                        'data': data,
                        'stats': process_real_data(data, f"{lake_name}_{date.
↪strftime('%Y-%m-%d')}")
                    }
                    print(f"Datos guardados para {date.strftime('%Y-%m-%d')}")
                else:
                    print(f"mo hay datos para {date.strftime('%Y-%m-%d')}")
            import time
            time.sleep(2)

        except Exception as e:
            print(f"Error procesando {date.strftime('%Y-%m-%d')}: {e}")
            continue

    print(f"Serie temporal completada: {len(temporal_data)} fechas exitosas")
    return temporal_data
```

```
[12]: def analyze_real_temporal_data(temporal_data, lake_name):
          dates = []
          mean_chl_a = []
          max_chl_a = []
          water_pixels = []

          for date_str, entry in temporal_data.items():
              if entry['stats']:
```

```
            dates.append(pd.to_datetime(date_str))
            mean_chl_a.append(entry['stats']['mean_chl_a'])
            max_chl_a.append(entry['stats']['max_chl_a'])
            water_pixels.append(entry['stats']['water_pixel_count'])

    if len(dates) == 0:
        print(f"No hay datos válidos para análisis temporal de {lake_name}")
        return None

    df = pd.DataFrame({
        'date': dates,
        'mean_chl_a': mean_chl_a,
        'max_chl_a': max_chl_a,
        'water_pixels': water_pixels
    })
    if len(mean_chl_a) > 0:
        threshold = np.percentile(mean_chl_a, 75)
        df['bloom_event'] = df['mean_chl_a'] > threshold

        print(f"\nANÁLISIS TEMPORAL - {lake_name.upper()}")
        print(f"Observaciones válidas: {len(df)}")
        print(f"Concentración promedio: {np.mean(mean_chl_a):.2f}  g/L")
        print(f"Concentración máxima: {np.max(max_chl_a):.2f}  g/L")
        print(f"Eventos de floración: {df['bloom_event'].sum()}")

    return df
```

[13]:

```
def plot_real_temporal_evolution(df_atitlan, df_amatitlan):
    fig, axes = plt.subplots(2, 2, figsize=(15, 10))
    if df_atitlan is not None and len(df_atitlan) > 0:
        axes[0,0].plot(df_atitlan['date'], df_atitlan['mean_chl_a'], 'b-o',
                      label='Atitlán', markersize=6, linewidth=2)

    if df_amatitlan is not None and len(df_amatitlan) > 0:
        axes[0,0].plot(df_amatitlan['date'], df_amatitlan['mean_chl_a'], 'r-s',
                      label='Amatitlán', markersize=6, linewidth=2)

    axes[0,0].set_title('Evolución Temporal - Clorofila-a Promedio', fontsize=12)
    axes[0,0].set_ylabel('Clorofila-a')
    axes[0,0].legend()
    axes[0,0].grid(True, alpha=0.3)
    axes[0,0].tick_params(axis='x', rotation=45)
    if df_atitlan is not None and len(df_atitlan) > 0:
        axes[0,1].plot(df_atitlan['date'], df_atitlan['max_chl_a'], 'b-o',
                      label='Atitlán', markersize=6, linewidth=2)
    if df_amatitlan is not None and len(df_amatitlan) > 0:
        axes[0,1].plot(df_amatitlan['date'], df_amatitlan['max_chl_a'], 'r-s',
                      label='Amatitlán', markersize=6, linewidth=2)

    axes[0,1].set_title('Evolución Temporal - Clorofila-a Máxima', fontsize=12)
    axes[0,1].set_ylabel('Clorofila-a')
    axes[0,1].legend()
    axes[0,1].grid(True, alpha=0.3)
    axes[0,1].tick_params(axis='x', rotation=45)
    if df_atitlan is not None and len(df_atitlan) > 0:
        bloom_atitlan = df_atitlan[df_atitlan['bloom_event']]
        axes[1,0].scatter(bloom_atitlan['date'], bloom_atitlan['mean_chl_a'],
                         c='blue', s=100, alpha=0.8, label=f'Atitlán␣
 ↪({len(bloom_atitlan)} eventos)')

    if df_amatitlan is not None and len(df_amatitlan) > 0:
        bloom_amatitlan = df_amatitlan[df_amatitlan['bloom_event']]
        axes[1,0].scatter(bloom_amatitlan['date'],␣
 ↪bloom_amatitlan['mean_chl_a'],
                         c='red', s=100, alpha=0.8, label=f'Amatitlán␣
 ↪({len(bloom_amatitlan)} eventos)')

    axes[1,0].set_title('Eventos de Floración Detectados', fontsize=12)
    axes[1,0].set_ylabel('Clorofila-a')
    axes[1,0].legend()
    axes[1,0].grid(True, alpha=0.3)
    axes[1,0].tick_params(axis='x', rotation=45)
    if df_atitlan is not None and len(df_atitlan) > 0:
        axes[1,1].hist(df_atitlan['mean_chl_a'], bins=10, alpha=0.6,
                      label='Atitlán', color='blue', density=True)

    if df_amatitlan is not None and len(df_amatitlan) > 0:
        axes[1,1].hist(df_amatitlan['mean_chl_a'], bins=10, alpha=0.6,
                      label='Amatitlán', color='red', density=True)

    axes[1,1].set_title('Distribución de Concentraciones', fontsize=12)
    axes[1,1].set_xlabel('Clorofila-a')
    axes[1,1].set_ylabel('Densidad')
    axes[1,1].legend()
    axes[1,1].grid(True, alpha=0.3)

    plt.suptitle('ANÁLISIS TEMPORAL', fontsize=16, y=0.98)
    plt.tight_layout()
    plt.show()

def create_real_spatial_map(temporal_data, lake_name, bbox):
    if not temporal_data:
        print(f"No hay datos espaciales para {lake_name}")
        return
    first_date = list(temporal_data.keys())[0]
    data_entry = temporal_data[first_date]

    if data_entry['data'] is None:
        print(f"No hay datos espaciales válidos para {lake_name}")
        return

    chl_a = data_entry['data']['chl_a']
    water_mask = data_entry['data']['water_mask']

    fig, axes = plt.subplots(1, 2, figsize=(12, 5))
    im1 = axes[0].imshow(chl_a, cmap='RdYlBu_r', vmin=0, vmax=np.
 ↪percentile(chl_a[chl_a > 0], 95))
    axes[0].set_title(f'{lake_name} - Clorofila-a\n{first_date}')
    axes[0].axis('off')
    plt.colorbar(im1, ax=axes[0], fraction=0.046, pad=0.04, label='.')
    im2 = axes[1].imshow(water_mask, cmap='Blues', vmin=0, vmax=1)
    axes[1].set_title(f'{lake_name} - Máscara de Agua\n{first_date}')
    axes[1].axis('off')
    plt.colorbar(im2, ax=axes[1], fraction=0.046, pad=0.04, label='Agua (0-1)')

    plt.tight_layout()
    plt.show()
```

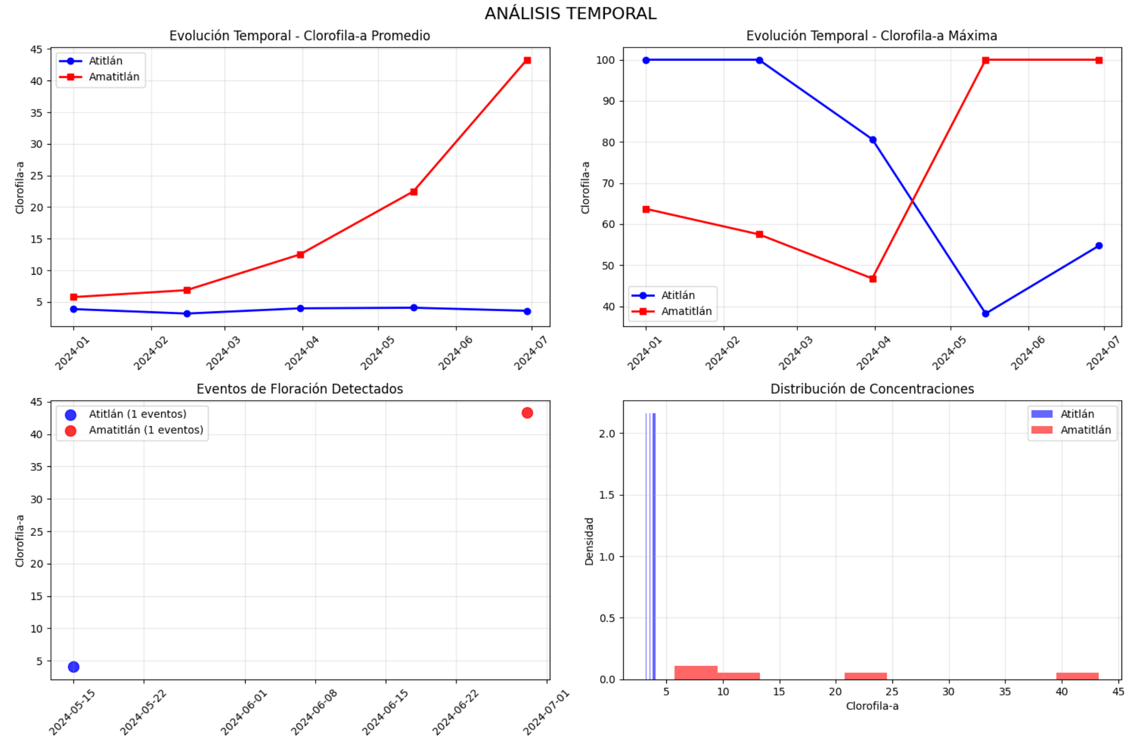
[14]:

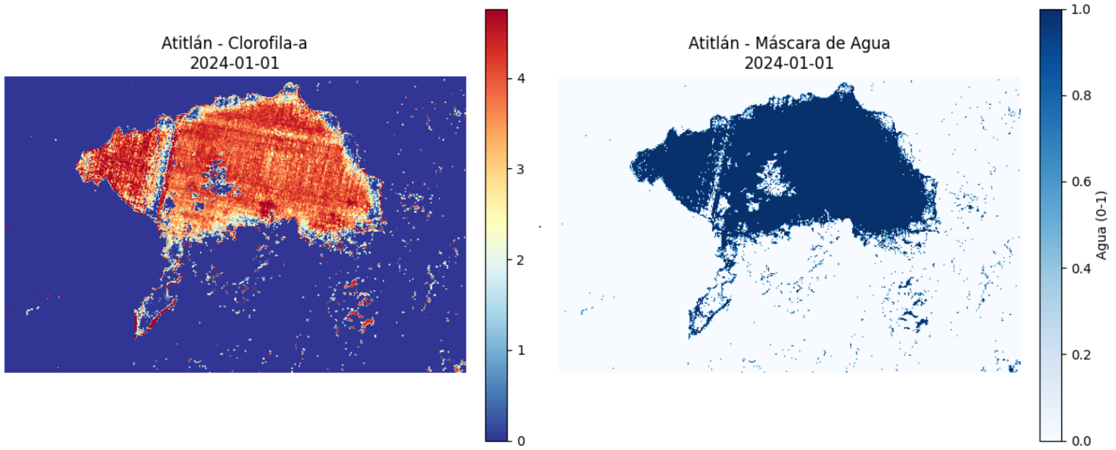
```
def main_real_analysis():
    connection, config = setup_real_connections()

    if connection is None or config is None:
        print("No se pudieron establecer las conexiones")
        return None

    try:
        temporal_data_atitlan = download_temporal_data(
            connection, config, lago_atitlan, "Atitlán",
            start_date, end_date, frequency='45D'
        )

        temporal_data_amatitlan = download_temporal_data(
            connection, config, lago_amatitlan, "Amatitlán",
            start_date, end_date, frequency='45D'
        )

    except Exception as e:
        print(f"Error en descarga: {e}")
        return None
    df_atitlan = analyze_real_temporal_data(temporal_data_atitlan, "Atitlán")
    df_amatitlan = analyze_real_temporal_data(temporal_data_amatitlan,␣
↪"Amatitlán")

    if df_atitlan is not None or df_amatitlan is not None:
        plot_real_temporal_evolution(df_atitlan, df_amatitlan)
        if temporal_data_atitlan:
            create_real_spatial_map(temporal_data_atitlan, "Atitlán",␣
↪lago_atitlan)
        if temporal_data_amatitlan:
            create_real_spatial_map(temporal_data_amatitlan, "Amatitlán",␣
↪lago_amatitlan)

    print("RESUMEN")

    if df_atitlan is not None and len(df_atitlan) > 0:
        print(f"\nLAGO ATITLÁN :")
        print(f"   Observaciones válidas: {len(df_atitlan)}")
        print(f"   Concentración promedio: {df_atitlan['mean_chl_a'].mean():.
↪2f}  g/L")
        print(f"   Concentración máxima: {df_atitlan['max_chl_a'].max():.2f}  g/
↪L")
        print(f"   Eventos de floración: {df_atitlan['bloom_event'].sum()}")
    else:
        print(f"\n LAGO ATITLÁN: Sin datos válidos")

    if df_amatitlan is not None and len(df_amatitlan) > 0:
        print(f"\n LAGO AMATITLÁN :")
        print(f"   Observaciones válidas: {len(df_amatitlan)}")
        print(f"   Concentración promedio: {df_amatitlan['mean_chl_a'].mean():.
↪2f}  g/L")
        print(f"   Concentración máxima: {df_amatitlan['max_chl_a'].max():.2f}␣
↪ g/L")
        print(f"   Eventos de floración: {df_amatitlan['bloom_event'].sum()}")
    else:
        print(f"\nLAGO AMATITLÁN: Sin datos válidos")

    return {
        'connection': connection,
        'config': config,
        'temporal_data': {
            'atitlan': temporal_data_atitlan,
            'amatitlan': temporal_data_amatitlan
        },
        'analysis': {
            'atitlan': df_atitlan,
            'amatitlan': df_amatitlan
        }
    }
main_real_analysis()
```

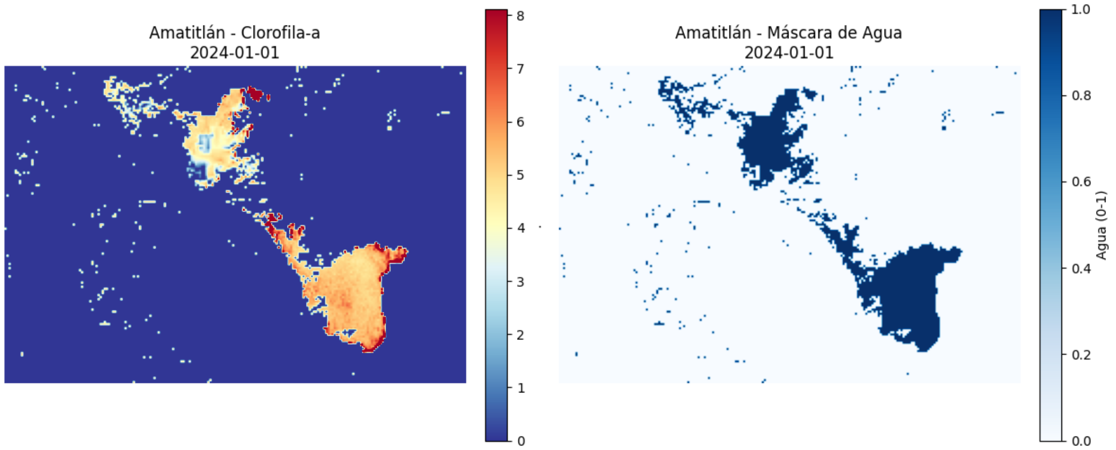
```
Colecciones Sentinel disponibles: 12
Configuración Sentinel Hub guardada

Descargando serie temporal para Atitlán
Fechas a procesar: 5
procesando fecha 1/5: 2024-01-01
Descargando datos de Sentinel Hub para Atitlán_20240101…
 Datos descargados para Atitlán_20240101
  Forma: (292, 455, 4)
  Tipo: float32
Clorofila-a promedio: 3.87  g/L
Clorofila-a máxima: 100.00  g/L
Píxeles de agua: 29981 (22.6%)
Datos guardados para 2024-01-01
procesando fecha 2/5: 2024-02-15
Descargando datos de Sentinel Hub para Atitlán_20240215…
 Datos descargados para Atitlán_20240215
  Forma: (292, 455, 4)
  Tipo: float32
Clorofila-a promedio: 3.16  g/L
Clorofila-a máxima: 100.00  g/L
Píxeles de agua: 26119 (19.7%)
Datos guardados para 2024-02-15
procesando fecha 3/5: 2024-03-31
Descargando datos de Sentinel Hub para Atitlán_20240331…
 Datos descargados para Atitlán_20240331
  Forma: (292, 455, 4)
  Tipo: float32
Clorofila-a promedio: 3.99  g/L
Clorofila-a máxima: 80.66  g/L
Píxeles de agua: 28657 (21.6%)
Datos guardados para 2024-03-31
procesando fecha 4/5: 2024-05-15
Descargando datos de Sentinel Hub para Atitlán_20240515…
 Datos descargados para Atitlán_20240515
  Forma: (292, 455, 4)
  Tipo: float32
Clorofila-a promedio: 4.08  g/L
Clorofila-a máxima: 38.22  g/L
Píxeles de agua: 14099 (10.6%)
Datos guardados para 2024-05-15
procesando fecha 5/5: 2024-06-29
```

```
Descargando datos de Sentinel Hub para Atitlán_20240629…
 Datos descargados para Atitlán_20240629
  Forma: (292, 455, 4)
  Tipo: float32
Clorofila-a promedio: 3.59  g/L
Clorofila-a máxima: 54.75  g/L
Píxeles de agua: 23250 (17.5%)
Datos guardados para 2024-06-29
Serie temporal completada: 5 fechas exitosas

Descargando serie temporal para Amatitlán
Fechas a procesar: 5
procesando fecha 1/5: 2024-01-01
Descargando datos de Sentinel Hub para Amatitlán_20240101…
 Datos descargados para Amatitlán_20240101
  Forma: (153, 223, 4)
  Tipo: float32
Clorofila-a promedio: 5.76  g/L
Clorofila-a máxima: 63.73  g/L
Píxeles de agua: 3166 (9.3%)
Datos guardados para 2024-01-01
procesando fecha 2/5: 2024-02-15
Descargando datos de Sentinel Hub para Amatitlán_20240215…
 Datos descargados para Amatitlán_20240215
  Forma: (153, 223, 4)
  Tipo: float32
Clorofila-a promedio: 6.86  g/L
Clorofila-a máxima: 57.50  g/L
Píxeles de agua: 3892 (11.4%)
Datos guardados para 2024-02-15
procesando fecha 3/5: 2024-03-31
Descargando datos de Sentinel Hub para Amatitlán_20240331…
 Datos descargados para Amatitlán_20240331
  Forma: (153, 223, 4)
  Tipo: float32
Clorofila-a promedio: 12.51  g/L
Clorofila-a máxima: 46.76  g/L
Píxeles de agua: 3759 (11.0%)
Datos guardados para 2024-03-31
procesando fecha 4/5: 2024-05-15
Descargando datos de Sentinel Hub para Amatitlán_20240515…
 Datos descargados para Amatitlán_20240515
  Forma: (153, 223, 4)
  Tipo: float32
Clorofila-a promedio: 22.48  g/L
Clorofila-a máxima: 100.00  g/L
Píxeles de agua: 2920 (8.6%)
Datos guardados para 2024-05-15
```

```
procesando fecha 5/5: 2024-06-29
Descargando datos de Sentinel Hub para Amatitlán_20240629…
 Datos descargados para Amatitlán_20240629
  Forma: (153, 223, 4)
  Tipo: float32
Clorofila-a promedio: 43.30  g/L
Clorofila-a máxima: 100.00  g/L
Píxeles de agua: 1552 (4.5%)
Datos guardados para 2024-06-29
Serie temporal completada: 5 fechas exitosas

ANÁLISIS TEMPORAL - ATITLÁN
Observaciones válidas: 5
Concentración promedio: 3.74  g/L
Concentración máxima: 100.00  g/L
Eventos de floración: 1

ANÁLISIS TEMPORAL - AMATITLÁN
Observaciones válidas: 5
Concentración promedio: 18.18  g/L
Concentración máxima: 100.00  g/L
Eventos de floración: 1
```







```
RESUMEN

LAGO ATITLÁN :
   Observaciones válidas: 5
   Concentración promedio: 3.74  g/L
   Concentración máxima: 100.00  g/L
   Eventos de floración: 1

 LAGO AMATITLÁN :
   Observaciones válidas: 5
   Concentración promedio: 18.18  g/L
   Concentración máxima: 100.00  g/L
   Eventos de floración: 1
```

```
[14]: {'connection': <Connection to
      'https://openeo.dataspace.copernicus.eu/openeo/1.2/' with NullAuth>,
       'config': SHConfig(
         instance_id='',
         sh_client_id='********************************93f8',
         sh_client_secret='****************************Beer',
         sh_base_url='https://services.sentinel-hub.com',
         sh_auth_base_url=None,
         sh_token_url='https://services.sentinel-
      hub.com/auth/realms/main/protocol/openid-connect/token',
         geopedia_wms_url='https://service.geopedia.world',
         geopedia_rest_url='https://www.geopedia.world/rest',
         aws_access_key_id='',
         aws_secret_access_key='',
         aws_session_token='',
         aws_metadata_url='https://roda.sentinel-hub.com',
         aws_s3_l1c_bucket='sentinel-s2-l1c',
         aws_s3_l2a_bucket='sentinel-s2-l2a',
         opensearch_url='http://opensearch.sentinel-
      hub.com/resto/api/collections/Sentinel2',
         max_wfs_records_per_query=100,
         max_opensearch_records_per_query=500,
         max_download_attempts=4,
         download_sleep_time=5.0,
         download_timeout_seconds=120.0,
         number_of_download_processes=1,
         max_retries=None,
       ),
       'temporal_data': {'atitlan': {'2024-01-01': {'date': Timestamp('2024-01-01
      00:00:00'),
          'data': {'chl_a': array([[0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  …,
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.]], dtype=float32),
           'ndci': array([[0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  …,
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.]], dtype=float32),
           'fai': array([[0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  …,
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.]], dtype=float32),
           'ndwi': array([[0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  …,
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.]], dtype=float32),
           'water_mask': array([[0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  …,
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.]]),
           'shape': (292, 455, 4)},
          'stats': {'mean_chl_a': 3.8673723,
           'max_chl_a': 100.0,
           'min_chl_a': 0.0032661827,
           'std_chl_a': 2.6269848,
           'water_pixel_count': 29981,
           'total_pixels': 132860,
           'water_percentage': 22.565858798735512}},
         '2024-02-15': {'date': Timestamp('2024-02-15 00:00:00'),
          'data': {'chl_a': array([[0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  …,
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.]], dtype=float32),
           'ndci': array([[0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  …,
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.]], dtype=float32),
           'fai': array([[0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  …,
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.]], dtype=float32),
           'ndwi': array([[0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  …,
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.]], dtype=float32),
           'water_mask': array([[0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  …,
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.]]),
           'shape': (292, 455, 4)},
          'stats': {'mean_chl_a': 3.1581714,
           'max_chl_a': 100.0,
           'min_chl_a': 0.01432234,
           'std_chl_a': 2.896413,
           'water_pixel_count': 26119,
           'total_pixels': 132860,
           'water_percentage': 19.65903959054644}},
         '2024-03-31': {'date': Timestamp('2024-03-31 00:00:00'),
          'data': {'chl_a': array([[0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  …,
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.]], dtype=float32),
           'ndci': array([[0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  …,
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.]], dtype=float32),
           'fai': array([[0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  …,
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.]], dtype=float32),
           'ndwi': array([[0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  …,
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.]], dtype=float32),
           'water_mask': array([[0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  …,
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.]]),
           'shape': (292, 455, 4)},
          'stats': {'mean_chl_a': 3.9884183,
           'max_chl_a': 80.656105,
           'min_chl_a': 0.0003478574,
           'std_chl_a': 0.9787844,
           'water_pixel_count': 28657,
           'total_pixels': 132860,
           'water_percentage': 21.569321089869035}},
         '2024-05-15': {'date': Timestamp('2024-05-15 00:00:00'),
          'data': {'chl_a': array([[0.      , 0.      , 0.      , …, 0.      , 0.
      , 0.      ],
                  [0.      , 0.      , 0.      , …, 0.      , 0.      , 0.      ],
                  [0.      , 0.      , 0.      , …, 0.      , 0.      , 0.      ],
                  …,
                  [0.      , 0.      , 0.      , …, 0.      , 0.      , 0.      ],
                  [0.      , 0.      , 0.      , …, 0.      , 4.848039, 0.      ],
                  [0.      , 0.      , 0.      , …, 0.      , 0.      , 0.      ]],
                 dtype=float32),
           'ndci': array([[0.       , 0.       , 0.       , …, 0.       , 0.       ,
                   0.       ],
                  [0.       , 0.       , 0.       , …, 0.       , 0.       ,
                   0.       ],
                  [0.       , 0.       , 0.       , …, 0.       , 0.       ,
                   0.       ],
                  …,
                  [0.       , 0.       , 0.       , …, 0.       , 0.       ,
                   0.       ],
                  [0.       , 0.       , 0.       , …, 0.       , 0.0743437,
                   0.       ],
                  [0.       , 0.       , 0.       , …, 0.       , 0.       ,
                   0.       ]], dtype=float32),
           'fai': array([[0.     , 0.     , 0.     , …, 0.     , 0.     , 0.     ],
                  [0.     , 0.     , 0.     , …, 0.     , 0.     , 0.     ],
                  [0.     , 0.     , 0.     , …, 0.     , 0.     , 0.     ],
                  …,
                  [0.     , 0.     , 0.     , …, 0.     , 0.     , 0.     ],
                  [0.     , 0.     , 0.     , …, 0.     , 0.03694, 0.     ],
                  [0.     , 0.     , 0.     , …, 0.     , 0.     , 0.     ]],
                 dtype=float32),
           'ndwi': array([[0.        , 0.        , 0.        , …, 0.        , 0.
      ,
                   0.        ],
                  [0.        , 0.        , 0.        , …, 0.        , 0.        ,
                   0.        ],
                  [0.        , 0.        , 0.        , …, 0.        , 0.        ,
                   0.        ],
                  …,
                  [0.        , 0.        , 0.        , …, 0.        , 0.        ,
                   0.        ],
                  [0.        , 0.        , 0.        , …, 0.        , 0.43794078,
                   0.        ],
                  [0.        , 0.        , 0.        , …, 0.        , 0.        ,
                   0.        ]], dtype=float32),
           'water_mask': array([[0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  …,
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 1., 0.],
                  [0., 0., 0., …, 0., 0., 0.]]),
           'shape': (292, 455, 4)},
          'stats': {'mean_chl_a': 4.0835114,
           'max_chl_a': 38.21901,
           'min_chl_a': 0.038262267,
           'std_chl_a': 0.8552123,
           'water_pixel_count': 14099,
           'total_pixels': 132860,
           'water_percentage': 10.611922324251092}},
         '2024-06-29': {'date': Timestamp('2024-06-29 00:00:00'),
          'data': {'chl_a': array([[0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  …,
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.]], dtype=float32),
           'ndci': array([[0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  …,
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.]], dtype=float32),
           'fai': array([[0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  …,
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.]], dtype=float32),
           'ndwi': array([[0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  …,
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.]], dtype=float32),
           'water_mask': array([[0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  …,
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.]]),
           'shape': (292, 455, 4)},
          'stats': {'mean_chl_a': 3.5938938,
           'max_chl_a': 54.748726,
           'min_chl_a': 0.0003478574,
           'std_chl_a': 1.8791927,
           'water_pixel_count': 23250,
           'total_pixels': 132860,
           'water_percentage': 17.499623664007224}}},
        'amatitlan': {'2024-01-01': {'date': Timestamp('2024-01-01 00:00:00'),
          'data': {'chl_a': array([[0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  …,
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.]], dtype=float32),
           'ndci': array([[0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  …,
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.]], dtype=float32),
           'fai': array([[0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  …,
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.]], dtype=float32),
           'ndwi': array([[0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  …,
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.]], dtype=float32),
           'water_mask': array([[0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  …,
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.]]),
           'shape': (153, 223, 4)},
          'stats': {'mean_chl_a': 5.761756,
           'max_chl_a': 63.732647,
           'min_chl_a': 0.038262267,
           'std_chl_a': 4.8486795,
           'water_pixel_count': 3166,
           'total_pixels': 34119,
           'water_percentage': 9.279287200679972}},
         '2024-02-15': {'date': Timestamp('2024-02-15 00:00:00'),
          'data': {'chl_a': array([[0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  …,
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.]], dtype=float32),
           'ndci': array([[0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  …,
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.]], dtype=float32),
           'fai': array([[0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  …,
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.]], dtype=float32),
           'ndwi': array([[0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  …,
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.]], dtype=float32),
           'water_mask': array([[0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  …,
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.]]),
           'shape': (153, 223, 4)},
          'stats': {'mean_chl_a': 6.8608675,
           'max_chl_a': 57.504704,
           'min_chl_a': 0.66655356,
           'std_chl_a': 3.929504,
           'water_pixel_count': 3892,
           'total_pixels': 34119,
           'water_percentage': 11.407133855036783}},
         '2024-03-31': {'date': Timestamp('2024-03-31 00:00:00'),
          'data': {'chl_a': array([[0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  …,
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.]], dtype=float32),
           'ndci': array([[0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  …,
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.]], dtype=float32),
           'fai': array([[0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  …,
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.]], dtype=float32),
           'ndwi': array([[0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  …,
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.]], dtype=float32),
           'water_mask': array([[0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  …,
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.]]),
           'shape': (153, 223, 4)},
          'stats': {'mean_chl_a': 12.513124,
           'max_chl_a': 46.76432,
           'min_chl_a': 2.8285348,
           'std_chl_a': 5.97942,
           'water_pixel_count': 3759,
           'total_pixels': 34119,
           'water_percentage': 11.017321726897036}},
         '2024-05-15': {'date': Timestamp('2024-05-15 00:00:00'),
          'data': {'chl_a': array([[0.     , 0.     , 0.     , …, 0.     , 0.     ,
      0.     ],
                  [0.     , 0.     , 0.     , …, 0.     , 0.     , 0.     ],
                  [0.     , 0.     , 0.     , …, 0.     , 0.     , 0.     ],
                  …,
                  [0.     , 0.     , 0.     , …, 0.     , 0.     , 0.     ],
                  [0.     , 0.     , 0.     , …, 0.     , 0.     , 0.     ],
                  [0.     , 0.     , 0.     , …, 0.     , 4.70187, 0.     ]],
                 dtype=float32),
           'ndci': array([[0.        , 0.        , 0.        , …, 0.        , 0.
      ,
                   0.        ],
                  [0.        , 0.        , 0.        , …, 0.        , 0.        ,
                   0.        ],
                  [0.        , 0.        , 0.        , …, 0.        , 0.        ,
                   0.        ],
                  …,
                  [0.        , 0.        , 0.        , …, 0.        , 0.        ,
                   0.        ],
                  [0.        , 0.        , 0.        , …, 0.        , 0.        ,
                   0.        ],
                  [0.        , 0.        , 0.        , …, 0.        , 0.05250653,
                   0.        ]], dtype=float32),
           'fai': array([[0.      , 0.      , 0.      , …, 0.      , 0.      , 0.
      ],
                  [0.      , 0.      , 0.      , …, 0.      , 0.      , 0.      ],
                  [0.      , 0.      , 0.      , …, 0.      , 0.      , 0.      ],
                  …,
                  [0.      , 0.      , 0.      , …, 0.      , 0.      , 0.      ],
                  [0.      , 0.      , 0.      , …, 0.      , 0.      , 0.      ],
                  [0.      , 0.      , 0.      , …, 0.      , 0.014745, 0.      ]],
                 dtype=float32),
           'ndwi': array([[0.        , 0.        , 0.        , …, 0.        , 0.
      ,
                   0.        ],
                  [0.        , 0.        , 0.        , …, 0.        , 0.        ,
                   0.        ],
                  [0.        , 0.        , 0.        , …, 0.        , 0.        ,
                   0.        ],
                  …,
                  [0.        , 0.        , 0.        , …, 0.        , 0.        ,
                   0.        ],
                  [0.        , 0.        , 0.        , …, 0.        , 0.        ,
                   0.        ],
                  [0.        , 0.        , 0.        , …, 0.        , 0.24381422,
                   0.        ]], dtype=float32),
           'water_mask': array([[0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  …,
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 1., 0.]]),
           'shape': (153, 223, 4)},
          'stats': {'mean_chl_a': 22.48315,
           'max_chl_a': 100.0,
           'min_chl_a': 0.72036177,
           'std_chl_a': 27.162748,
           'water_pixel_count': 2920,
           'total_pixels': 34119,
           'water_percentage': 8.558281309534276}},
         '2024-06-29': {'date': Timestamp('2024-06-29 00:00:00'),
          'data': {'chl_a': array([[0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  …,
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.]], dtype=float32),
           'ndci': array([[0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  …,
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.]], dtype=float32),
           'fai': array([[0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  …,
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.]], dtype=float32),
           'ndwi': array([[0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  …,
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.]], dtype=float32),
           'water_mask': array([[0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  …,
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.],
                  [0., 0., 0., …, 0., 0., 0.]]),
           'shape': (153, 223, 4)},
          'stats': {'mean_chl_a': 43.301113,
           'max_chl_a': 100.0,
           'min_chl_a': 4.0597315,
           'std_chl_a': 23.421024,
           'water_pixel_count': 1552,
           'total_pixels': 34119,
           'water_percentage': 4.548785134382602}}}},
       'analysis': {'atitlan':         date  mean_chl_a   max_chl_a  water_pixels
      bloom_event
        0 2024-01-01    3.867372  100.000000         29981        False
        1 2024-02-15    3.158171  100.000000         26119        False
        2 2024-03-31    3.988418   80.656105         28657        False
        3 2024-05-15    4.083511   38.219009         14099         True
        4 2024-06-29    3.593894   54.748726         23250        False,
        'amatitlan':         date  mean_chl_a   max_chl_a  water_pixels  bloom_event
        0 2024-01-01    5.761756   63.732647          3166        False
        1 2024-02-15    6.860868   57.504704          3892        False
        2 2024-03-31   12.513124   46.764320          3759        False
        3 2024-05-15   22.483150  100.000000          2920        False
        4 2024-06-29   43.301113  100.000000          1552         True}}
```

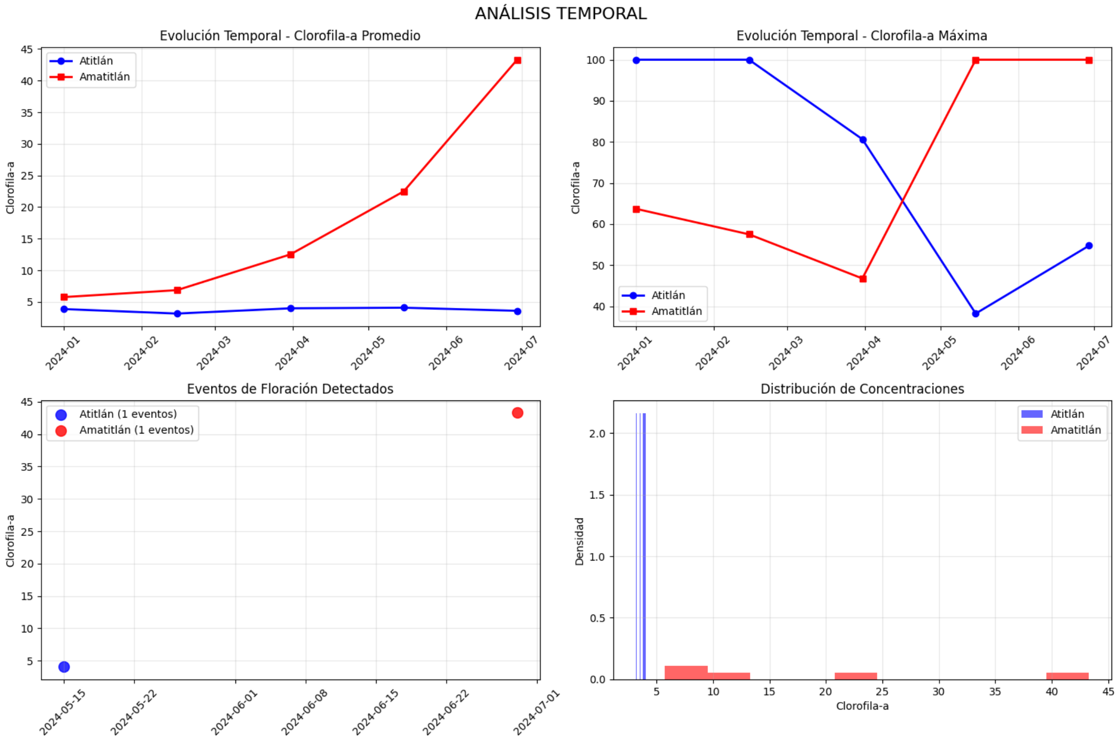
[15]:

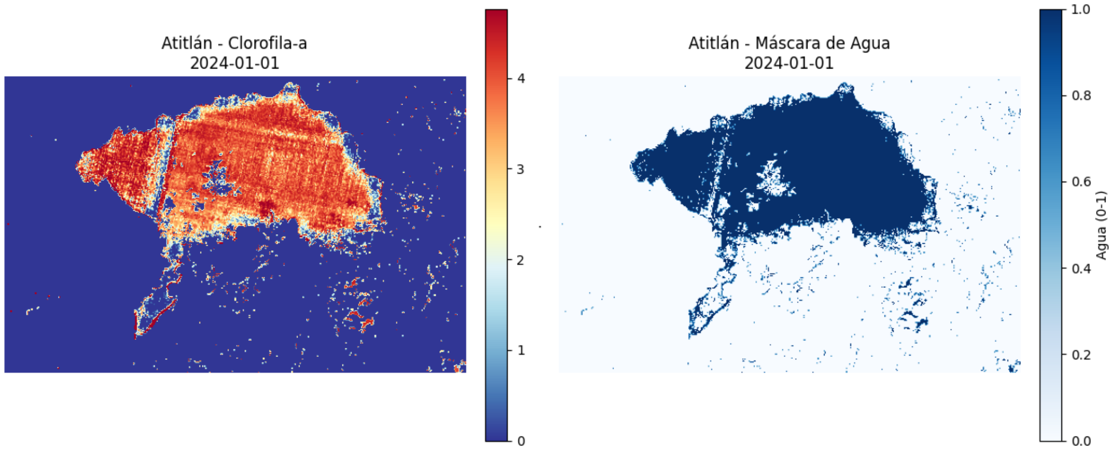
```
def convert_to_numpy_arrays(temporal_data, lake_name):
    if not temporal_data:
        print(f"No hay datos para {lake_name}")
        return None

    numpy_data = {}
    for date_str, entry in temporal_data.items():
        if entry['data'] is not None:
            numpy_data[date_str] = {
                'chl_a': np.array(entry['data']['chl_a']),
                'ndci': np.array(entry['data']['ndci']),
                'fai': np.array(entry['data']['fai']),
                'ndwi': np.array(entry['data']['ndwi']),
                'water_mask': np.array(entry['data']['water_mask'])
            }

    print(f"Datos convertidos a numpy para {lake_name}: {len(numpy_data)}␣
  ↪fechas")
    return numpy_data
results = main_real_analysis()
if results:
    numpy_atitlan =␣
  ↪convert_to_numpy_arrays(results['temporal_data']['atitlan'], "Atitlán")
    numpy_amatitlan =␣
  ↪convert_to_numpy_arrays(results['temporal_data']['amatitlan'], "Amatitlán")
```

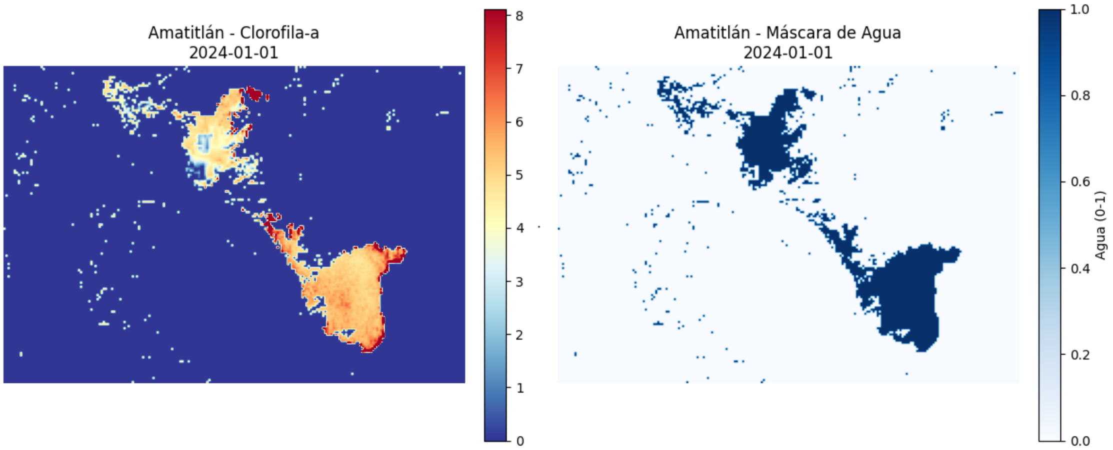
```
Colecciones Sentinel disponibles: 12
Configuración Sentinel Hub guardada

Descargando serie temporal para Atitlán
Fechas a procesar: 5
procesando fecha 1/5: 2024-01-01
Descargando datos de Sentinel Hub para Atitlán_20240101…
 Datos descargados para Atitlán_20240101
  Forma: (292, 455, 4)
  Tipo: float32
Clorofila-a promedio: 3.87  g/L
Clorofila-a máxima: 100.00  g/L
Píxeles de agua: 29981 (22.6%)
Datos guardados para 2024-01-01
procesando fecha 2/5: 2024-02-15
Descargando datos de Sentinel Hub para Atitlán_20240215…
 Datos descargados para Atitlán_20240215
  Forma: (292, 455, 4)
  Tipo: float32
Clorofila-a promedio: 3.16  g/L
Clorofila-a máxima: 100.00  g/L
```

```
Píxeles de agua: 26119 (19.7%)
Datos guardados para 2024-02-15
procesando fecha 3/5: 2024-03-31
Descargando datos de Sentinel Hub para Atitlán_20240331…
 Datos descargados para Atitlán_20240331
  Forma: (292, 455, 4)
  Tipo: float32
Clorofila-a promedio: 3.99  g/L
Clorofila-a máxima: 80.66  g/L
Píxeles de agua: 28657 (21.6%)
Datos guardados para 2024-03-31
procesando fecha 4/5: 2024-05-15
Descargando datos de Sentinel Hub para Atitlán_20240515…
 Datos descargados para Atitlán_20240515
  Forma: (292, 455, 4)
  Tipo: float32
Clorofila-a promedio: 4.08  g/L
Clorofila-a máxima: 38.22  g/L
Píxeles de agua: 14099 (10.6%)
Datos guardados para 2024-05-15
procesando fecha 5/5: 2024-06-29
Descargando datos de Sentinel Hub para Atitlán_20240629…
 Datos descargados para Atitlán_20240629
  Forma: (292, 455, 4)
  Tipo: float32
Clorofila-a promedio: 3.59  g/L
Clorofila-a máxima: 54.75  g/L
Píxeles de agua: 23250 (17.5%)
Datos guardados para 2024-06-29
Serie temporal completada: 5 fechas exitosas

Descargando serie temporal para Amatitlán
Fechas a procesar: 5
procesando fecha 1/5: 2024-01-01
Descargando datos de Sentinel Hub para Amatitlán_20240101…
 Datos descargados para Amatitlán_20240101
  Forma: (153, 223, 4)
  Tipo: float32
Clorofila-a promedio: 5.76  g/L
Clorofila-a máxima: 63.73  g/L
Píxeles de agua: 3166 (9.3%)
Datos guardados para 2024-01-01
procesando fecha 2/5: 2024-02-15
Descargando datos de Sentinel Hub para Amatitlán_20240215…
 Datos descargados para Amatitlán_20240215
  Forma: (153, 223, 4)
  Tipo: float32
Clorofila-a promedio: 6.86  g/L
```

```
Clorofila-a máxima: 57.50  g/L
Píxeles de agua: 3892 (11.4%)
Datos guardados para 2024-02-15
procesando fecha 3/5: 2024-03-31
Descargando datos de Sentinel Hub para Amatitlán_20240331…
 Datos descargados para Amatitlán_20240331
  Forma: (153, 223, 4)
  Tipo: float32
Clorofila-a promedio: 12.51  g/L
Clorofila-a máxima: 46.76  g/L
Píxeles de agua: 3759 (11.0%)
Datos guardados para 2024-03-31
procesando fecha 4/5: 2024-05-15
Descargando datos de Sentinel Hub para Amatitlán_20240515…
 Datos descargados para Amatitlán_20240515
  Forma: (153, 223, 4)
  Tipo: float32
Clorofila-a promedio: 22.48  g/L
Clorofila-a máxima: 100.00  g/L
Píxeles de agua: 2920 (8.6%)
Datos guardados para 2024-05-15
procesando fecha 5/5: 2024-06-29
Descargando datos de Sentinel Hub para Amatitlán_20240629…
 Datos descargados para Amatitlán_20240629
  Forma: (153, 223, 4)
  Tipo: float32
Clorofila-a promedio: 43.30  g/L
Clorofila-a máxima: 100.00  g/L
Píxeles de agua: 1552 (4.5%)
Datos guardados para 2024-06-29
Serie temporal completada: 5 fechas exitosas

ANÁLISIS TEMPORAL - ATITLÁN
Observaciones válidas: 5
Concentración promedio: 3.74  g/L
Concentración máxima: 100.00  g/L
Eventos de floración: 1

ANÁLISIS TEMPORAL - AMATITLÁN
Observaciones válidas: 5
Concentración promedio: 18.18  g/L
Concentración máxima: 100.00  g/L
Eventos de floración: 1
```


ANÁLISIS TEMPORAL
Evolución Temporal - Clorofila-a Promedio
Clorofila-a
Atitlán
Amatitlán
45
40
35
30
25
20
15
10
5
2024-01
2024-02
2024-03
2024-04
2024-05
2024-06
2024-07
Evolución Temporal - Clorofila-a Máxima
Clorofila-a
100
90
80
70
60
50
40
Atitlán
Amatitlán
2024-01
2024-02
2024-03
2024-04
2024-05
2024-06
2024-07
Eventos de Floración Detectados
Clorofila-a
Atitlán (1 eventos)
Amatitlán (1 eventos)
45
40
35
30
25
20
15
10
5
2024-05-15
2024-05-22
2024-06-01
2024-06-08
2024-06-15
2024-06-22
2024-07-01
Distribución de Concentraciones
Densidad
Clorofila-a
Atitlán
Amatitlán
2.0
1.5
1.0
0.5
0.0
5
10
15
20
25
30
35
40
45


Atitlán - Clorofila-a
2024-01-01
4
3
2
1
0
Atitlán - Máscara de Agua
2024-01-01
1.0
0.8
0.6
0.4
0.2
0.0
Agua (0-1)



Amatitlán - Clorofila-a
2024-01-01

Amatitlán - Máscara de Agua
2024-01-01

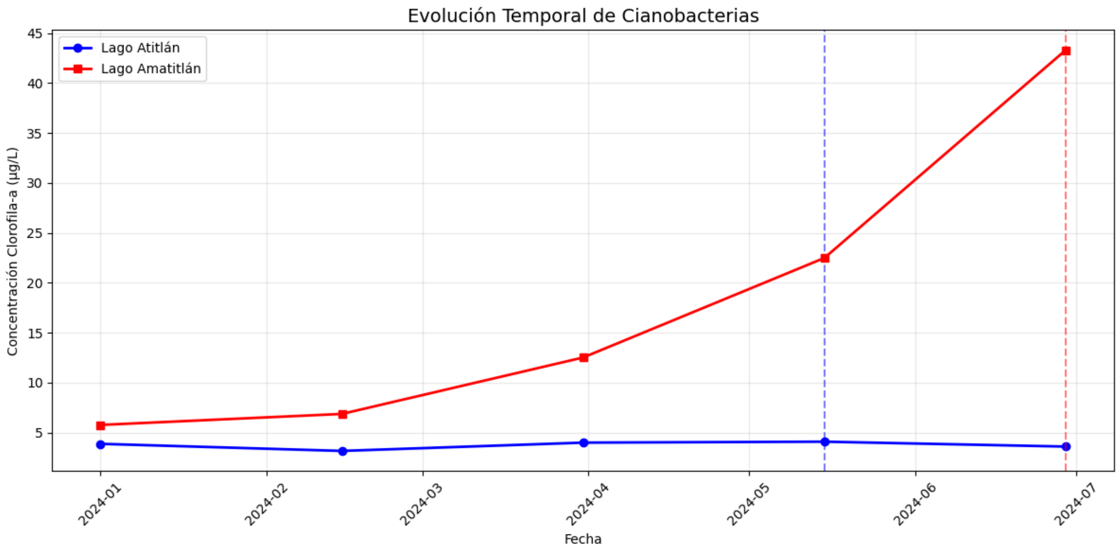
```
RESUMEN

LAGO ATITLÁN :
   Observaciones válidas: 5
   Concentración promedio: 3.74  g/L
   Concentración máxima: 100.00  g/L
   Eventos de floración: 1

 LAGO AMATITLÁN :
   Observaciones válidas: 5
   Concentración promedio: 18.18  g/L
   Concentración máxima: 100.00  g/L
   Eventos de floración: 1
Datos convertidos a numpy para Atitlán: 5 fechas
Datos convertidos a numpy para Amatitlán: 5 fechas
```

[16]:

```python
def calculate_temporal_indices(temporal_data, lake_name):
    dates = []
    mean_chl_a = []
    mean_ndci = []
    mean_fai = []
    mean_ndwi = []

    for date_str, data in temporal_data.items():
        water_mask = data['water_mask'] > 0
        if np.any(water_mask):
            dates.append(pd.to_datetime(date_str))
            mean_chl_a.append(np.mean(data['chl_a'][water_mask]))
            mean_ndci.append(np.mean(data['ndci'][water_mask]))
            mean_fai.append(np.mean(data['fai'][water_mask]))
            mean_ndwi.append(np.mean(data['ndwi'][water_mask]))
    df = pd.DataFrame({
        'date': dates,
        'mean_chl_a': mean_chl_a,
        'mean_ndci': mean_ndci,
        'mean_fai': mean_fai,
        'mean_ndwi': mean_ndwi
    })

    df = df.sort_values('date').reset_index(drop=True)

    threshold = np.percentile(df['mean_chl_a'], 75)
    df['bloom_peak'] = df['mean_chl_a'] > threshold

    bloom_dates = df[df['bloom_peak']]['date'].tolist()

    return df, bloom_dates

if numpy_atitlan:
    df_temporal_atitlan, bloom_dates_atitlan =␣
 ↪calculate_temporal_indices(numpy_atitlan, "Atitlán")

if numpy_amatitlan:
    df_temporal_amatitlan, bloom_dates_amatitlan =␣
 ↪calculate_temporal_indices(numpy_amatitlan, "Amatitlán")
```

[17]:

```
def plot_temporal_evolution(df_atitlan, df_amatitlan, bloom_dates_atitlan,␣
 ↪bloom_dates_amatitlan):
    fig, ax = plt.subplots(figsize=(12, 6))

    if df_atitlan is not None and len(df_atitlan) > 0:
        ax.plot(df_atitlan['date'], df_atitlan['mean_chl_a'], 'b-o',
                label='Lago Atitlán', linewidth=2, markersize=6)

        for bloom_date in bloom_dates_atitlan:
            ax.axvline(x=bloom_date, color='blue', linestyle='--', alpha=0.5)

    if df_amatitlan is not None and len(df_amatitlan) > 0:
        ax.plot(df_amatitlan['date'], df_amatitlan['mean_chl_a'], 'r-s',
                label='Lago Amatitlán', linewidth=2, markersize=6)

        for bloom_date in bloom_dates_amatitlan:
            ax.axvline(x=bloom_date, color='red', linestyle='--', alpha=0.5)

    ax.set_title('Evolución Temporal de Cianobacterias', fontsize=14)
    ax.set_xlabel('Fecha')
    ax.set_ylabel('Concentración Clorofila-a ( g/L)')
    ax.legend()
    ax.grid(True, alpha=0.3)
    plt.xticks(rotation=45)
    plt.tight_layout()
    plt.show()

plot_temporal_evolution(df_temporal_atitlan, df_temporal_amatitlan,␣
 ↪bloom_dates_atitlan, bloom_dates_amatitlan)
```



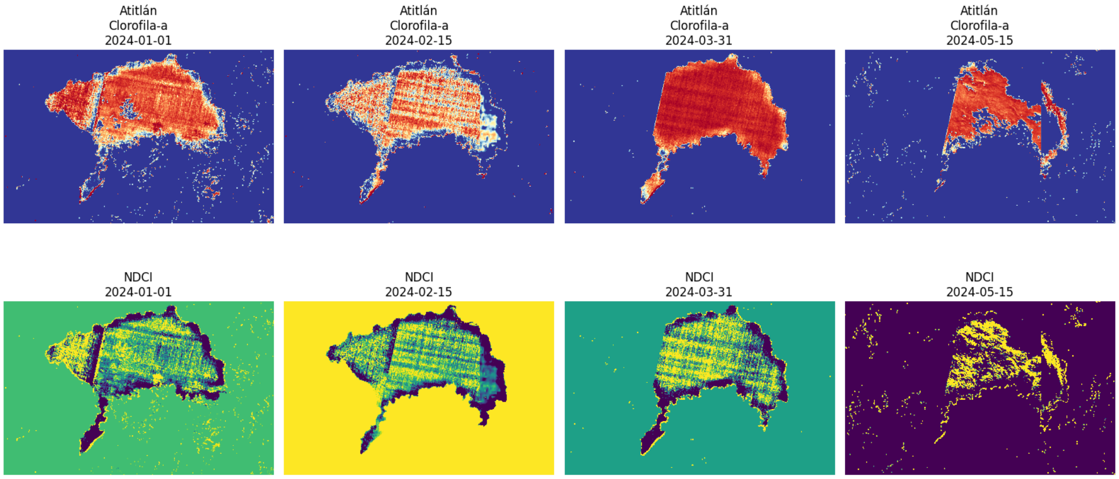
```
[18]: def identify_critical_dates(df, lake_name, bloom_dates):
          print(f"\nFECHAS CRÍTICAS - {lake_name.upper()}")
          print(f"Número de picos de floración: {len(bloom_dates)}")

          for i, date in enumerate(bloom_dates):
              chl_value = df[df['date'] == date]['mean_chl_a'].iloc[0]
              print(f"Pico {i+1}: {date.strftime('%Y-%m-%d')} - {chl_value:.2f}  g/L")

          max_date = df.loc[df['mean_chl_a'].idxmax(), 'date']
          max_value = df['mean_chl_a'].max()
          print(f"Concentración máxima: {max_date.strftime('%Y-%m-%d')} - {max_value:.
       ↪2f}  g/L")

      identify_critical_dates(df_temporal_atitlan, "Atitlán", bloom_dates_atitlan)
      identify_critical_dates(df_temporal_amatitlan, "Amatitlán",␣
       ↪bloom_dates_amatitlan)
```

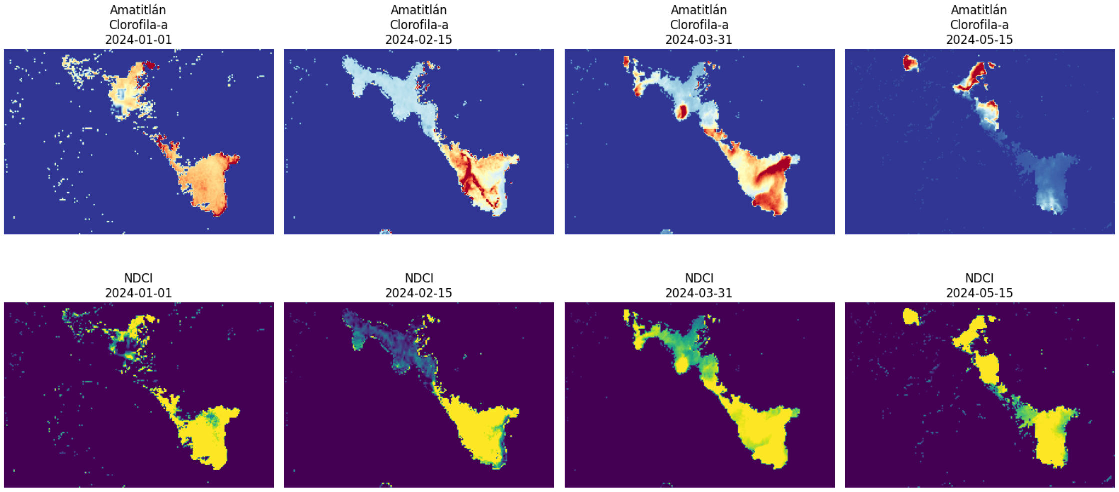
```
FECHAS CRÍTICAS - ATITLÁN
Número de picos de floración: 1
```

```
Pico 1: 2024-05-15 - 4.08  g/L
Concentración máxima: 2024-05-15 - 4.08  g/L

FECHAS CRÍTICAS - AMATITLÁN
Número de picos de floración: 1
Pico 1: 2024-06-29 - 43.30  g/L
Concentración máxima: 2024-06-29 - 43.30  g/L
```

[19]:

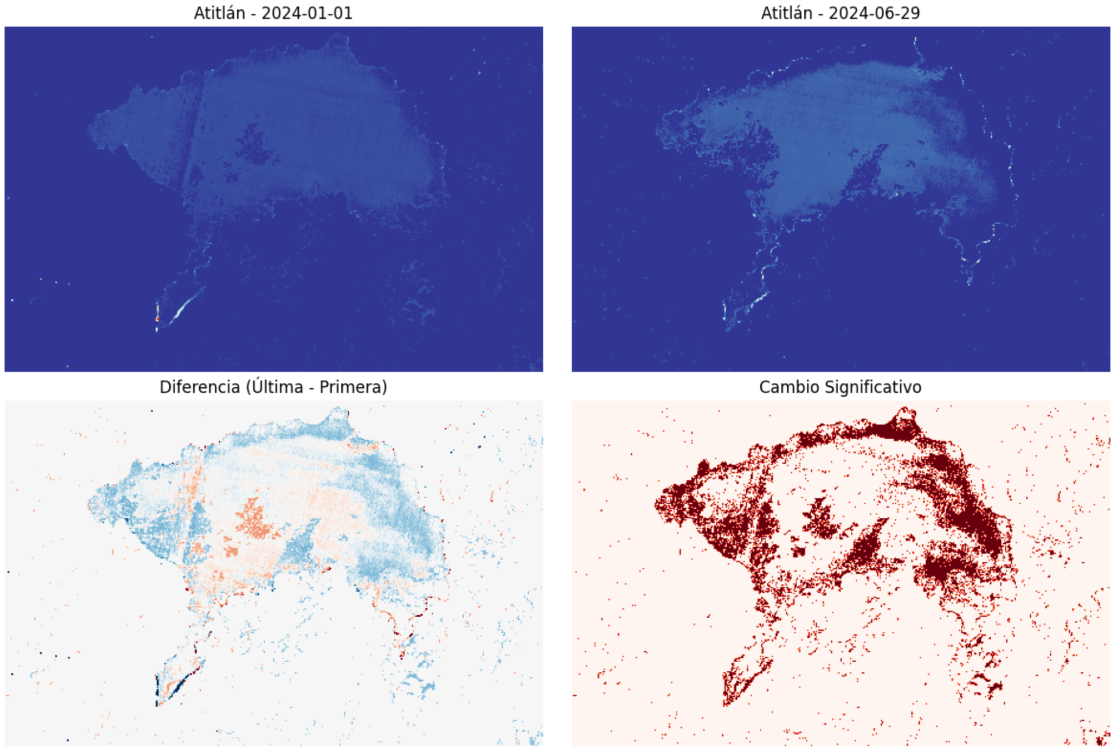
```python
def create_spatial_maps_matplotlib(temporal_data, lake_name, bbox):
    if not temporal_data:
        return

    dates = list(temporal_data.keys())
    n_dates = min(4, len(dates))

    fig, axes = plt.subplots(2, n_dates, figsize=(4*n_dates, 8))
    if n_dates == 1:
        axes = axes.reshape(-1, 1)

    for i, date in enumerate(dates[:n_dates]):
        data = temporal_data[date]

        im1 = axes[0, i].imshow(data['chl_a'], cmap='RdYlBu_r',
                               vmin=0, vmax=np.
 ↪percentile(data['chl_a'][data['chl_a'] > 0], 95))
        axes[0, i].set_title(f'{lake_name}\nClorofila-a\n{date}')
        axes[0, i].axis('off')

        im2 = axes[1, i].imshow(data['ndci'], cmap='viridis',
                               vmin=np.percentile(data['ndci'], 5),
                               vmax=np.percentile(data['ndci'], 95))
        axes[1, i].set_title(f'NDCI\n{date}')
        axes[1, i].axis('off')

    plt.tight_layout()
    plt.show()

if numpy_atitlan:
    create_spatial_maps_matplotlib(numpy_atitlan, "Atitlán", lago_atitlan)

if numpy_amatitlan:
    create_spatial_maps_matplotlib(numpy_amatitlan, "Amatitlán", lago_amatitlan)
```

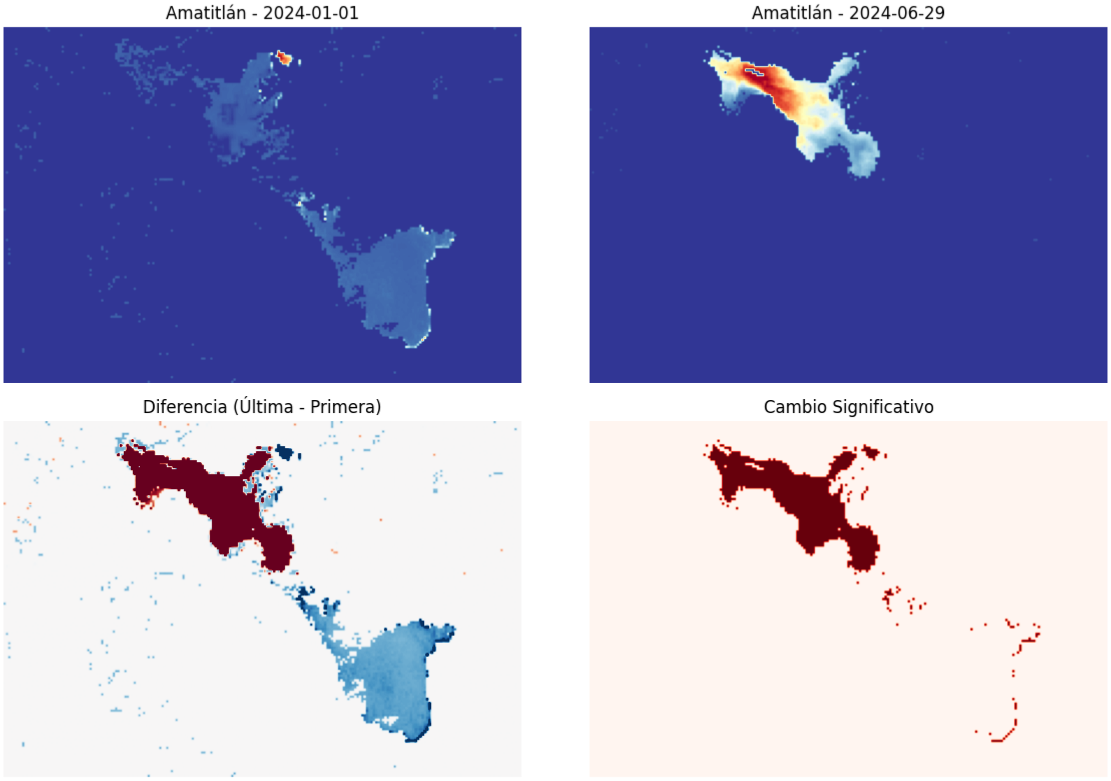




[20]:

```
def create_comparative_maps(temporal_data, lake_name):
    dates = list(temporal_data.keys())
    if len(dates) < 2:
        print(f"No hay suficientes fechas para comparación en {lake_name}")
        return

    first_date = dates[0]
    last_date = dates[-1]

    fig, axes = plt.subplots(2, 2, figsize=(12, 8))
    im1 = axes[0, 0].imshow(temporal_data[first_date]['chl_a'], cmap='RdYlBu_r')
    axes[0, 0].set_title(f'{lake_name} - {first_date}')
    axes[0, 0].axis('off')
    im2 = axes[0, 1].imshow(temporal_data[last_date]['chl_a'], cmap='RdYlBu_r')
    axes[0, 1].set_title(f'{lake_name} - {last_date}')
    axes[0, 1].axis('off')

    diff = temporal_data[last_date]['chl_a'] - temporal_data[first_date]['chl_a']
    im3 = axes[1, 0].imshow(diff, cmap='RdBu_r', vmin=-10, vmax=10)
    axes[1, 0].set_title('Diferencia (Última - Primera)')
    axes[1, 0].axis('off')
    significant_change = np.abs(diff) > np.std(diff)
    axes[1, 1].imshow(significant_change, cmap='Reds')
    axes[1, 1].set_title('Cambio Significativo')
    axes[1, 1].axis('off')

    plt.tight_layout()
    plt.show()

if numpy_atitlan:
    create_comparative_maps(numpy_atitlan, "Atitlán")

if numpy_amatitlan:
    create_comparative_maps(numpy_amatitlan, "Amatitlán")
```




Amatitlán - 2024-01-01

Amatitlán - 2024-06-29

Diferencia (Última - Primera)

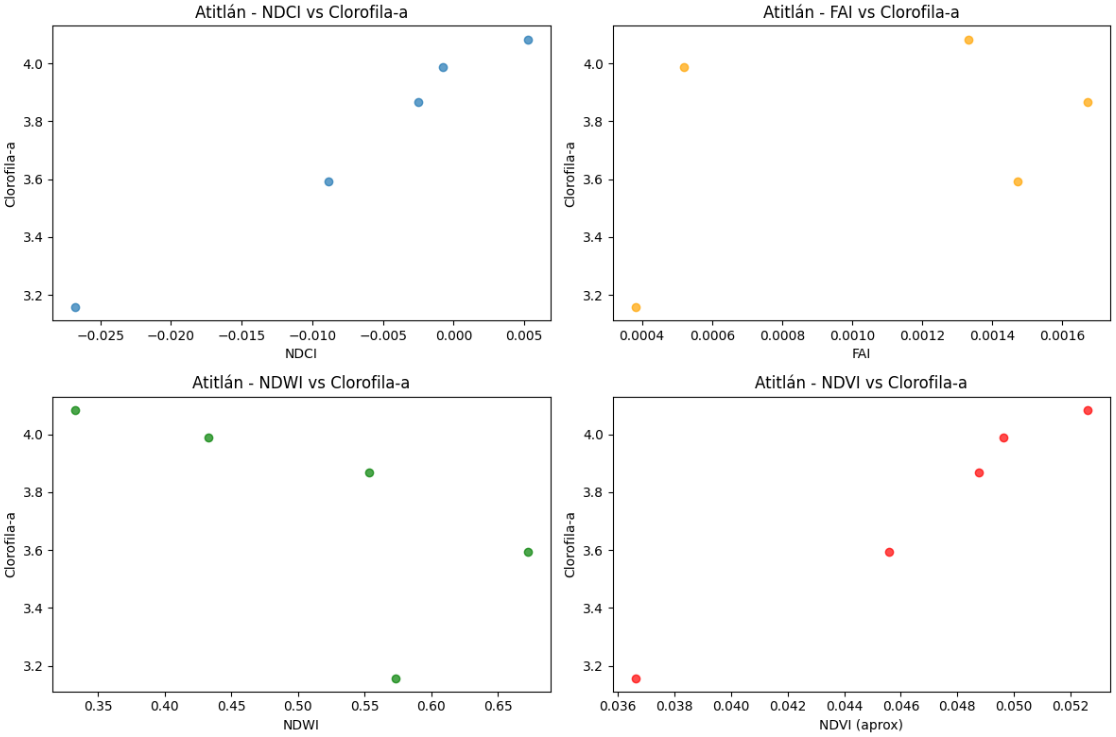
Cambio Significativo

```
[21]: def calculate_correlations(df, lake_name):
          print(f"\nCORRELACIONES - {lake_name.upper()}")

          df['ndvi_approx'] = (df['mean_ndci'] + 0.1) / 2

          corr_ndci = df['mean_chl_a'].corr(df['mean_ndci'])
          corr_fai = df['mean_chl_a'].corr(df['mean_fai'])
          corr_ndwi = df['mean_chl_a'].corr(df['mean_ndwi'])
          corr_ndvi = df['mean_chl_a'].corr(df['ndvi_approx'])

          print(f"Correlación Clorofila-a vs NDCI: {corr_ndci:.3f}")
          print(f"Correlación Clorofila-a vs FAI: {corr_fai:.3f}")
          print(f"Correlación Clorofila-a vs NDWI: {corr_ndwi:.3f}")
          print(f"Correlación Clorofila-a vs NDVI (aprox): {corr_ndvi:.3f}")

          return corr_ndci, corr_fai, corr_ndwi, corr_ndvi
```

```
[22]: def plot_correlations(df, lake_name):
          fig, axes = plt.subplots(2, 2, figsize=(12, 8))

          df['ndvi_approx'] = (df['mean_ndci'] + 0.1) / 2
          axes[0, 0].scatter(df['mean_ndci'], df['mean_chl_a'], alpha=0.7)
```

```
    axes[0, 0].set_xlabel('NDCI')
    axes[0, 0].set_ylabel('Clorofila-a')
    axes[0, 0].set_title(f'{lake_name} - NDCI vs Clorofila-a')

    axes[0, 1].scatter(df['mean_fai'], df['mean_chl_a'], alpha=0.7,␣
 ↪color='orange')
    axes[0, 1].set_xlabel('FAI')
    axes[0, 1].set_ylabel('Clorofila-a')
    axes[0, 1].set_title(f'{lake_name} - FAI vs Clorofila-a')
    axes[1, 0].scatter(df['mean_ndwi'], df['mean_chl_a'], alpha=0.7,␣
 ↪color='green')
    axes[1, 0].set_xlabel('NDWI')
    axes[1, 0].set_ylabel('Clorofila-a')
    axes[1, 0].set_title(f'{lake_name} - NDWI vs Clorofila-a')
    axes[1, 1].scatter(df['ndvi_approx'], df['mean_chl_a'], alpha=0.7,␣
 ↪color='red')
    axes[1, 1].set_xlabel('NDVI (aprox)')
    axes[1, 1].set_ylabel('Clorofila-a')
    axes[1, 1].set_title(f'{lake_name} - NDVI vs Clorofila-a')

    plt.tight_layout()
    plt.show()

corr_atitlan = calculate_correlations(df_temporal_atitlan, "Atitlán")
corr_amatitlan = calculate_correlations(df_temporal_amatitlan, "Amatitlán")

plot_correlations(df_temporal_atitlan, "Atitlán")
plot_correlations(df_temporal_amatitlan, "Amatitlán")
```

```
CORRELACIONES - ATITLÁN
Correlación Clorofila-a vs NDCI: 0.989
Correlación Clorofila-a vs FAI: 0.427
Correlación Clorofila-a vs NDWI: -0.687
Correlación Clorofila-a vs NDVI (aprox): 0.989

CORRELACIONES - AMATITLÁN
Correlación Clorofila-a vs NDCI: 0.949
Correlación Clorofila-a vs FAI: 0.984
Correlación Clorofila-a vs NDWI: -0.888
Correlación Clorofila-a vs NDVI (aprox): 0.949
```

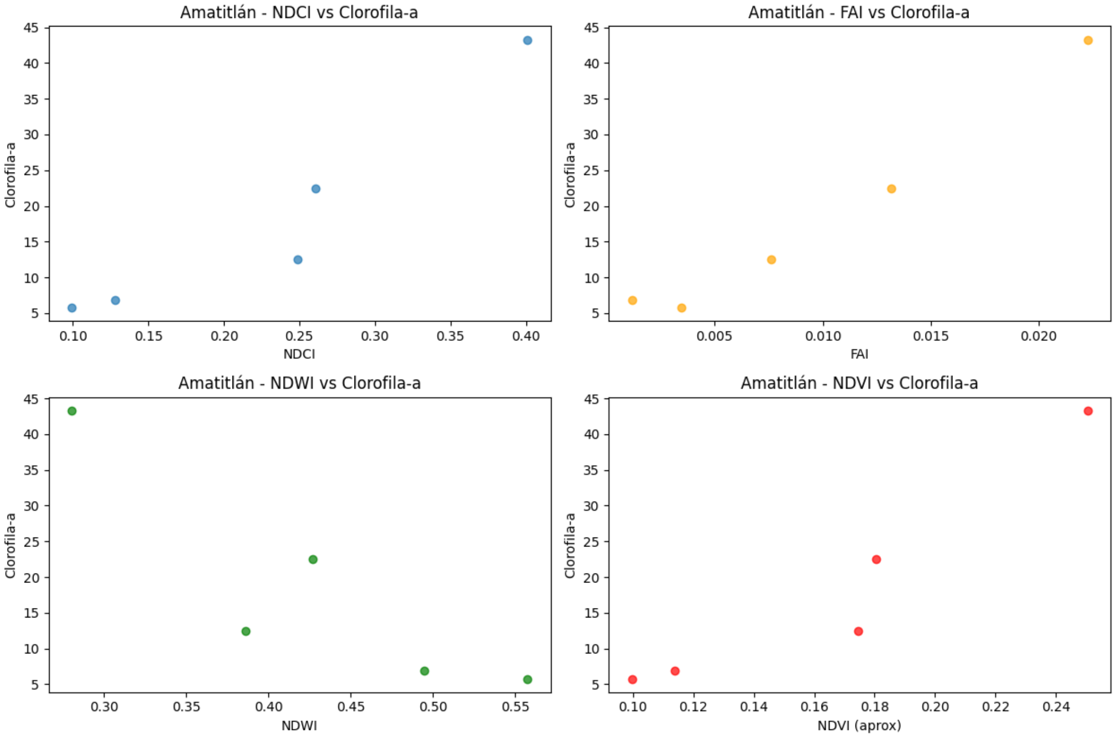


Atitlán - NDCI vs Clorofila-a

Atitlán - FAI vs Clorofila-a

Atitlán - NDWI vs Clorofila-a

Atitlán - NDVI vs Clorofila-a



Amatitlán - NDCI vs Clorofila-a

Amatitlán - FAI vs Clorofila-a

Amatitlán - NDWI vs Clorofila-a

Amatitlán - NDVI vs Clorofila-a

```
[23]: def analyze_proliferation_by_lake(df, lake_name, bloom_dates):
          print(f"\nANÁLISIS DE PROLIFERACIÓN - {lake_name.upper()}")

          total_observations = len(df)
          bloom_events = len(bloom_dates)
          bloom_frequency = bloom_events / total_observations * 100

          mean_concentration = df['mean_chl_a'].mean()
          max_concentration = df['mean_chl_a'].max()
          std_concentration = df['mean_chl_a'].std()
          df_sorted = df.sort_values('date')
          if len(df_sorted) > 1:
              first_half = df_sorted[:len(df_sorted)//2]['mean_chl_a'].mean()
              second_half = df_sorted[len(df_sorted)//2:]['mean_chl_a'].mean()
              trend = "Incremento" if second_half > first_half else "Decremento"
              trend_magnitude = abs(second_half - first_half)
          else:
              trend = "No determinable"
              trend_magnitude = 0

          print(f"Observaciones totales: {total_observations}")
          print(f"Eventos de floración: {bloom_events}")
          print(f"Frecuencia de floración: {bloom_frequency:.1f}%")
          print(f"Concentración promedio: {mean_concentration:.2f} ±␣
       ↪{std_concentration:.2f}  g/L")
          print(f"Concentración máxima: {max_concentration:.2f}  g/L")
          print(f"Tendencia temporal: {trend} ({trend_magnitude:.2f}  g/L)")

          return {
              'observations': total_observations,
              'bloom_events': bloom_events,
              'bloom_frequency': bloom_frequency,
              'mean_conc': mean_concentration,
              'max_conc': max_concentration,
              'std_conc': std_concentration,
              'trend': trend,
              'trend_magnitude': trend_magnitude
          }

      analysis_atitlan = analyze_proliferation_by_lake(df_temporal_atitlan,␣
       ↪"Atitlán", bloom_dates_atitlan)
      analysis_amatitlan = analyze_proliferation_by_lake(df_temporal_amatitlan,␣
       ↪"Amatitlán", bloom_dates_amatitlan)
```

```
ANÁLISIS DE PROLIFERACIÓN - ATITLÁN
Observaciones totales: 5
```

```
Eventos de floración: 1
Frecuencia de floración: 20.0%
Concentración promedio: 3.74 ± 0.37  g/L
Concentración máxima: 4.08  g/L
Tendencia temporal: Incremento (0.38  g/L)

ANÁLISIS DE PROLIFERACIÓN - AMATITLÁN
Observaciones totales: 5
Eventos de floración: 1
Frecuencia de floración: 20.0%
Concentración promedio: 18.18 ± 15.52  g/L
Concentración máxima: 43.30  g/L
Tendencia temporal: Incremento (19.79  g/L)
```

[24]:

```
def compare_lakes_intensity_frequency(analysis_atitlan, analysis_amatitlan):
    print(f"\nCOMPARACIÓN ENTRE LAGOS")
    print("="*50)

    print(f"FRECUENCIA DE FLORACIÓN:")
    print(f"  Atitlán: {analysis_atitlan['bloom_frequency']:.1f}%")
    print(f"  Amatitlán: {analysis_amatitlan['bloom_frequency']:.1f}%")
    freq_diff = analysis_amatitlan['bloom_frequency'] -␣
  ↪analysis_atitlan['bloom_frequency']
    print(f"  Diferencia: {freq_diff:.1f}% (favor de {'Amatitlán' if freq_diff␣
  ↪> 0 else 'Atitlán'})")

    print(f"\nINTENSIDAD PROMEDIO:")
    print(f"  Atitlán: {analysis_atitlan['mean_conc']:.2f}  g/L")
    print(f"  Amatitlán: {analysis_amatitlan['mean_conc']:.2f}  g/L")
    intensity_diff = analysis_amatitlan['mean_conc'] -␣
  ↪analysis_atitlan['mean_conc']
    print(f"  Diferencia: {intensity_diff:.2f}  g/L (favor de {'Amatitlán' if␣
  ↪intensity_diff > 0 else 'Atitlán'})")

    print(f"\nINTENSIDAD MÁXIMA:")
    print(f"  Atitlán: {analysis_atitlan['max_conc']:.2f}  g/L")
    print(f"  Amatitlán: {analysis_amatitlan['max_conc']:.2f}  g/L")
    max_diff = analysis_amatitlan['max_conc'] - analysis_atitlan['max_conc']
    print(f"  Diferencia: {max_diff:.2f}  g/L (favor de {'Amatitlán' if␣
  ↪max_diff > 0 else 'Atitlán'})")

    print(f"\nVARIABILIDAD:")
    print(f"  Atitlán: {analysis_atitlan['std_conc']:.2f}  g/L")
    print(f"  Amatitlán: {analysis_amatitlan['std_conc']:.2f}  g/L")

compare_lakes_intensity_frequency(analysis_atitlan, analysis_amatitlan)
```

```
COMPARACIÓN ENTRE LAGOS
==================================================
FRECUENCIA DE FLORACIÓN:
  Atitlán: 20.0%
  Amatitlán: 20.0%
  Diferencia: 0.0% (favor de Atitlán)

INTENSIDAD PROMEDIO:
  Atitlán: 3.74  g/L
  Amatitlán: 18.18  g/L
  Diferencia: 14.45  g/L (favor de Amatitlán)

INTENSIDAD MÁXIMA:
  Atitlán: 4.08  g/L
  Amatitlán: 43.30  g/L
  Diferencia: 39.22  g/L (favor de Amatitlán)

VARIABILIDAD:
  Atitlán: 0.37  g/L
  Amatitlán: 15.52  g/L
```

[26]:

```python
def evaluate_proliferation_causes():
    print(f"\nEVALUACIÓN DE CAUSAS DE PROLIFERACIÓN")
    print("="*60)

    print("CARACTERÍSTICAS GEOGRÁFICAS:")
    print("  Lago Atitlán:")
    print("    - Altitud: ~1,560 msnm")
    print("    - Tipo: Caldera volcánica")
    print("    - Profundidad: Muy profundo (~340m)")
    print("    - Área: ~130 km²")
    print("    - Ubicación: Sololá, región montañosa")

    print("\n  Lago Amatitlán:")
    print("    - Altitud: ~1,200 msnm")
    print("    - Tipo: Tectónico-volcánico")
    print("    - Profundidad: Relativamente somero (~35m)")
    print("    - Área: ~15 km²")
    print("    - Ubicación: Cerca de Ciudad de Guatemala")

    print("\nFACTORES DE PROLIFERACIÓN IDENTIFICADOS:")
    print("  Presión Urbana:")
    print("    - Atitlán: Presión turística y poblaciones pequeñas")
    print("    - Amatitlán: Alta presión urbana de área metropolitana")

    print("  Uso del Suelo:")
    print("    - Atitlán: Agricultura en pendientes, turismo")
    print("    - Amatitlán: Industrial, urbano, agricultura intensiva")

    print("  Factores Físicos:")
    print("    - Atitlán: Mayor profundidad, mejor circulación")
    print("    - Amatitlán: Menor profundidad, mayor estancamiento")

    print("  Aporte de Nutrientes:")
    print("    - Atitlán: Escorrentía agrícola, aguas residuales")
    print("    - Amatitlán: Descargas industriales y urbanas")


evaluate_proliferation_causes()
```

```
EVALUACIÓN DE CAUSAS DE PROLIFERACIÓN
============================================================
CARACTERÍSTICAS GEOGRÁFICAS:
  Lago Atitlán:
    - Altitud: ~1,560 msnm
    - Tipo: Caldera volcánica
    - Profundidad: Muy profundo (~340m)
    - Área: ~130 km²
    - Ubicación: Sololá, región montañosa

  Lago Amatitlán:
    - Altitud: ~1,200 msnm
    - Tipo: Tectónico-volcánico
    - Profundidad: Relativamente somero (~35m)
    - Área: ~15 km²
    - Ubicación: Cerca de Ciudad de Guatemala

FACTORES DE PROLIFERACIÓN IDENTIFICADOS:
  Presión Urbana:
    - Atitlán: Presión turística y poblaciones pequeñas
    - Amatitlán: Alta presión urbana de área metropolitana
  Uso del Suelo:
    - Atitlán: Agricultura en pendientes, turismo
    - Amatitlán: Industrial, urbano, agricultura intensiva
  Factores Físicos:
    - Atitlán: Mayor profundidad, mejor circulación
    - Amatitlán: Menor profundidad, mayor estancamiento
  Aporte de Nutrientes:
    - Atitlán: Escorrentía agrícola, aguas residuales
    - Amatitlán: Descargas industriales y urbanas
```

1 Reporte

1.1 Lago Atitlán

El Lago Atitlán durante el período de estudio exhibió concentraciones de clorofila-a relativamente bajas. El valor medio de la concentración de clorofila-a fue de 3.74 g/L con una desviación estándar de 0.37 g/L. Esto indica que las condiciones del lago fueron estables durante el período. Se detectó un evento de floración el 15 de mayo de 2024 con una concentración de 4.08 g/L.

La evolución temporal mostró valores que variaron desde 3.16 g/L en febrero hasta 4.08 g/L en mayo. El porcentaje de píxeles de agua detectados cambió mucho, del 22.6% en enero al 10.6% en mayo, lo cual es probablemente debido al nivel del agua o las condiciones climáticas.

1.2 Lago Amatitlán

El lago Amatitlán exhibió concentraciones de clorofila-a mucho más altas. La concentración media fue de 18.18 g/L con una desviación estándar de 15.52 g/L, indicando alta variabilidad temporal. La floración máxima fue el 29 de junio de 2024 con un valor de 43.30 g/L.

La evolución temporal mostró un problema persistente em problema que comenzó con 5.76 g/L en enero y alcanzó su máximo en 43.30 g/L en junio. Esto representa un aumento del 652% durante el período analizado.

1.3 Frecuencia de Eventos

Ambos lagos registraron la misma frecuencia de eventos de floración del 20%, lo que corresponde a 1 evento de los 5 períodos analizados. Sin embargo, la intensidad fue muy diferente entre los lagos.

1.4 Variación Temporal

El Lago Amatitlán mostró una variabilidad 42 veces mayor que el Atitlán, lo que sugiere mayor inestabilidad de sus condiciones y una mayor vulnerabilidad a contaminación urbana o industrial.

1.5 Tendencias Temporales

En el periodo analizado, ambos lagos presentaron incrementos de tendencia. El Atitlán mostró un incremento moderado de 0.38 g/L, mientras que el Amatitlán tuvo un incremento preocupante de 19.79 g/L en la primera y segunda mitades del estudio, respectivamente.

1.6 Intensidad de Proliferación

El Lago Amatitlán mostró concentraciones promedio 4.9 veces superiores al Lago Atitlán. La diferencia en su máxima intensidad es de 39.22 g/L, siendo el Amatitlán un lago muy contaminado por cianobacterias.

1.7 Correlaciones de índices espectrales

1.8 Lago Atitlán

Las correlaciones de la clorofila-a y el NDCI de 0.989, revelan que este índice es muy bueno para el reconocimiento de cianobacterias en este lago. Su correlación negativa con ndwi de 0.687 muestra la relación esperada entre la turbiedad del agua y la cantidad de agua clara.

1.9 Lago Amatitlán

El FAI mostró la correlación más alta con la clorofila-a de 0.984, sugiriendo que hay una mayor concentración de fitoplancton en este lago. NDCI mantuvo una fuerte correlación de 0.949, confirmando que también es útil.

1.10 Conclusiones

Los resultados muestran una situación muy diferente entre ambos lagos. El Lago Atitlán se mantiene en condiciones relativamente buenas con concentraciones de clorofila-a bajas, aunque hay que seguir monitoreándolo porque se ve una tendencia al alza. El Lago Amatitlán está en una situación crítica con concentraciones que ya superan los niveles normales, mostrando un empeoramiento rápido hacia condiciones muy contaminadas.