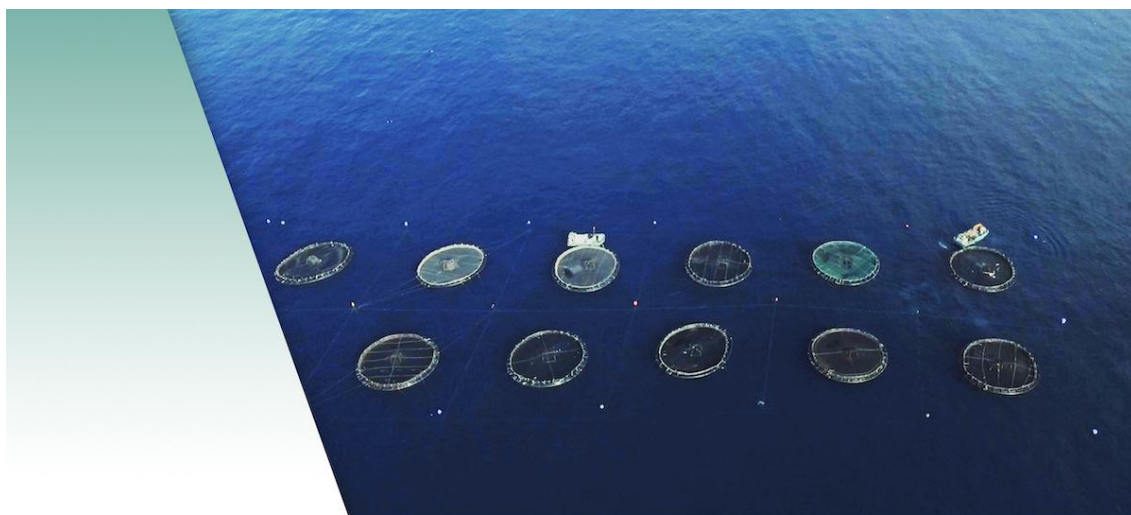




Proyecto SIRENA:

Sistema Remoto Integral de Monitorización, Detección y Predicción de Riesgos de Eventos Marinos Potencialmente Nocivos de Origen Natural o Antrópico en Áreas Off-Shore Destinadas a la Acuicultura



Informe de Resultados / Fuente de Verificación

R1.1 Script de descarga automática de imágenes de satélite

Fecha: 23/05/2025

Entidad Responsable: Universidad de Las Palmas de Gran Canaria (ULPGC)

Este proyecto se desarrolla con la colaboración de la Fundación Biodiversidad del Ministerio para la Transición Ecológica y el Reto Demográfico, a través del Programa Pleamar, y se cofinancia por la Unión Europea por el FEMPA (Fondo Europeo Marítimo, de Pesca y de Acuicultura).





Índice

1. Introducción	3
2. Script de descarga	4

Las opiniones y documentación aportadas en esta publicación son de exclusiva responsabilidad de la persona o personas que ostenten su autoría y no reflejan necesariamente los puntos de vista de las entidades que apoyan económicamente el proyecto.



1. Introducción

En el marco del **Resultado R1.1** del proyecto **SIRENA**, se ha desarrollado un **script en Python** que permite la descarga automática de imágenes satelitales del programa **Copernicus – Sentinel**. Esta herramienta ha sido diseñada y programada por el **Instituto Universitario ECOAQUA de la Universidad de Las Palmas de Gran Canaria (ULPGC)** con el objetivo de facilitar la recopilación sistemática y actualizada de datos satelitales como parte de la estrategia de monitorización ambiental del proyecto.

El script constituye un entregable clave del proyecto, ya que automatiza una tarea crítica para el análisis posterior de la dinámica oceánica, la acuicultura y otros procesos costeros monitorizados en SIRENA.



2. Script de descarga

A continuación, se presenta el código fuente que constituye la **fuentes de verificación** de este resultado.

```
# =====  
# SCRIPT DE DESCARGA CREADO PARA EL PROYECTO SIRENA  
# POR INSTITUTO UNIVERSITARIO ECOAQUA - ULPGC  
# =====  
  
from sentinelsat import SentinelAPI, geojson_to_wkt  
from shapely.geometry import box  
from datetime import datetime  
import os  
  
# =====  
# CONFIGURACIÓN DEL USUARIO  
# =====  
  
# Coordenadas del área (bounding box)  
lonmin = -15.557  
latmin = 27.429  
lonmax = -15.657  
latmax = 27.629  
  
# Fechas en formato 'YYYYMMDD'  
start_date = '20240101'  
end_date = '20240131'
```



```
# Tipo de escena: 'S1' para Sentinel-1 o 'S2' para Sentinel-2
tipo_escena = 'S1'
```

```
# Directorio de descarga
download_dir = r'C:\Sirena\Descargas'
```

```
# Credenciales del Copernicus Open Access Hub
username = 'usuarioSIRENA'
password = 'passSIRENA'
```

```
# =====
# LÓGICA DEL SCRIPT
# =====
```

```
# Crear el polígono de búsqueda a partir de la bounding box
area = box(lonmin, latmin, lonmax, latmax)
footprint = geojson_to_wkt({'type': 'Feature', 'geometry':
area.__geo_interface__})
```

```
# Conexión a la API de Copernicus
api = SentinelAPI(username, password,
'https://scihub.copernicus.eu/dhus')
```

```
# Definir los parámetros según el tipo de escena
```

```
if tipo_escena.upper() == 'S1':
    platformname = 'Sentinel-1'
    extra_params = {
        'producttype': 'GRD',
        'sensoroperationalmode': 'IW'
    }
elif tipo_escena.upper() == 'S2':
```



```
platformname = 'Sentinel-2'
extra_params = {
    'producttype': ' S2MSI2A ', # Nivel 2A (con corrección
atmosférica)
    'cloudcoverpercentage': (0, 30) # Opcional: puedes cambiar el
límite de nubosidad
}
else:
    raise ValueError("tipo_escena debe ser 'S1' o 'S2'.")

# Realizar búsqueda
products = api.query(
    footprint,
    date=(start_date, end_date),
    platformname=platformname,
    **extra_params
)

print(f"Se encontraron {len(products)} productos para {platformname}.")

# Crear carpeta si no existe
os.makedirs(download_dir, exist_ok=True)

# Descargar productos
for product_id, product_info in products.items():
    print(f"Descargando: {product_info['title']}")
    api.download(product_id, directory_path=download_dir)
```