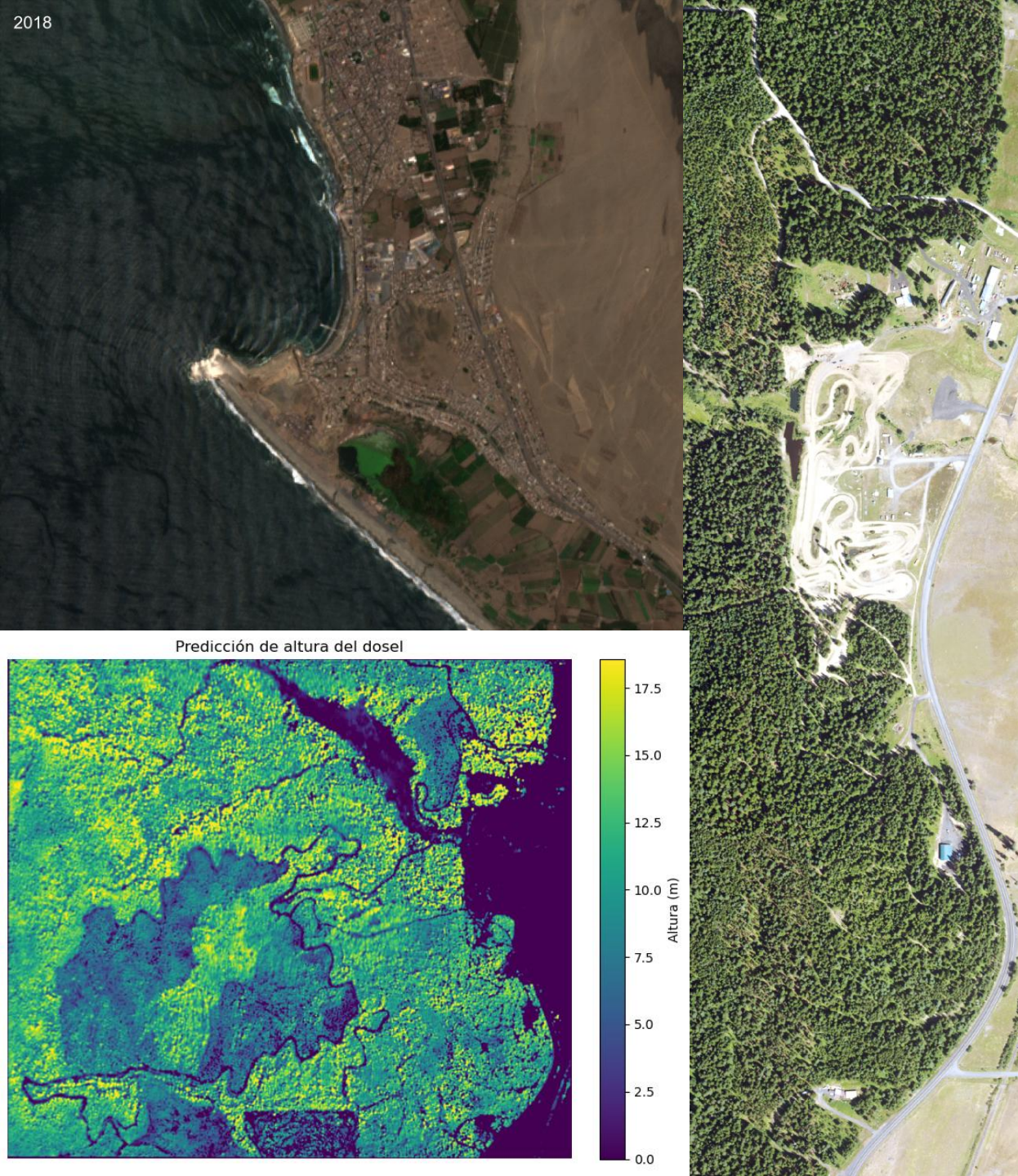




# Del GIS tradicional al GeoAI

Ing. Javier More Sanchez

[jmore@ieee.org](mailto:jmore@ieee.org)

20 de febrero de 2026

# GIS: Sistemas de Información Geográfica

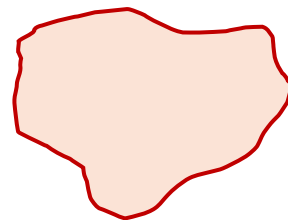
## Datos vectoriales:



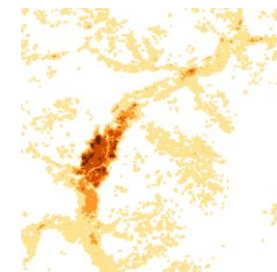
Puntos



Líneas



Polígonos



## Datos ráster:

Ráster

Un Sistema de Información Geográfica (GIS) es una plataforma que integra y analiza datos espaciales y sus atributos para apoyar la **toma de decisiones**

# Software GIS

## Python GIS stack



## R GIS stack



## Desktop GIS



ArcGIS Pro

## Cloud GIS



ArcGIS Online



Google Earth Engine

## DB GIS

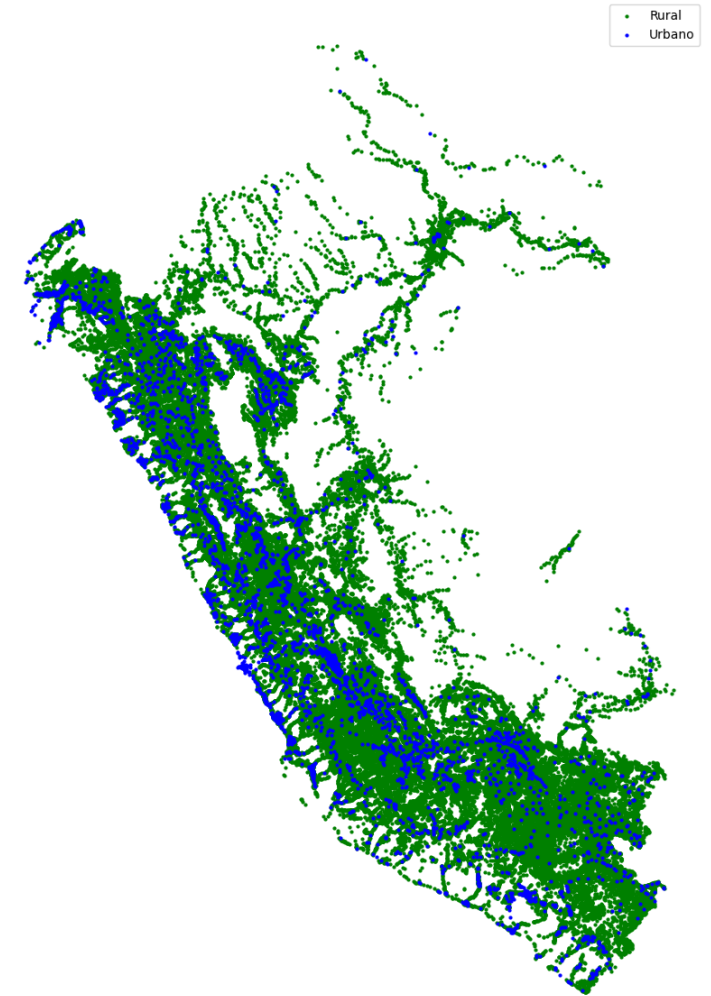


# Mapa con Python

```
#### Extracción de información de CCPP de Perú V1.0
#### By: Ing. Javier More
import pandas as pd
import requests
import zipfile
import io
import geopandas as gpd #De ser necesario se debe instalar desde el CMD: pip install geopandas
import matplotlib.pyplot as plt
import contextily as ctx
#Se indica el URL en el que se encuentra el archivo.
url = "https://cdn.www.gob.pe/uploads/document/file/4544636/Anexos.zip?v=1683816217"
#Se le indica cual es el nombre del archivo
archivo_CCPP = "informe001-ce-ccpp-2023-anexo-04.pdf.xlsb"
#Usamos REQUEST para descargar el archivo ZIP
consulta = requests.get(url)
zip = zipfile.ZipFile(io.BytesIO(consulta.content))
#Se descomprime todos los archivos contenidos en ZIP
zip.extractall()
#Creamos un Dataframe cuando se contiene el archivo_CCPP
df = pd.read_excel(archivo_CCPP, engine="pyxlsb")
df['COORD_X'] = df['COORD_X'].apply(lambda x: -x if x > 0 else x) #Se corrige un error en la coordenada.
#df = df.sort_values(by='COORD_X', ascending=False) #Esta línea solo es para verificar que no se escapen valores positivos en las coordenadas.
#Algunos CCPP estan catalogados como NA, se sugiere reemplazarlo como RURAL
df['CLASIFICACION DE AREA'] = df['CLASIFICACION DE AREA'].replace('NA', 'RURAL')
df.columns = ['UBIGEO', 'UBIGEO_ANT', 'DPTO', 'PROV', 'DIST', 'CCPP', 'CATEGORIA', 'CATEGORIA_2', 'CAP', 'VIVIENDAS', 'HABITANTES', 'Y', 'X', 'TIPO']
#####
# El siguiente código es para eliminar los archivos que se guardaron previamente en el disco duro (Luego de descomprimir). Este paso es OPCIONAL.
import os
for archivo in zip.namelist():
    ruta_archivo = os.path.join(archivo)
    os.remove(ruta_archivo)
#####
df #Permite visualizar el DF.

gdf = gpd.GeoDataFrame(df, geometry=gpd.points_from_xy(df['X'], df['Y']))
gdf.crs = 'epsg:4326' #Se define un sistema de referencia.
gdf_urbano = gdf[gdf['TIPO'] == 'URBANO']
gdf_rural = gdf[gdf['TIPO'] == 'RURAL']
fig, ax = plt.subplots(figsize=(15, 15))
#ctx.add_basemap(ax, crs=gdf.crs.to_string(), source=ctx.providers.Stamen.Terrain)
gdf_rural.plot(ax=ax, color='green', markersize=4, label='Rural')
gdf_urbano.plot(ax=ax, color='blue', markersize=4, label='Urbano')
ax.legend(fontsize='medium')
ax.set_title('CCPP Urbanos y Rurales', fontsize=20, fontweight='bold', color = '#101b6f')
#Para una mejor visualización se sugiere eliminar los ejes:
ax.set_axis_off()
plt.show()
```

CCPP Urbanos y Rurales



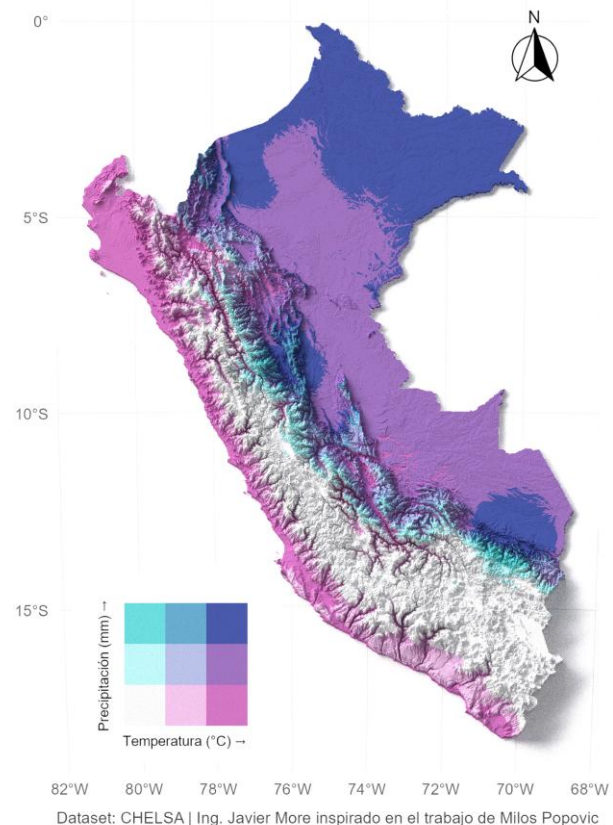
# Mapas con R

## Piura: ríos y quebradas

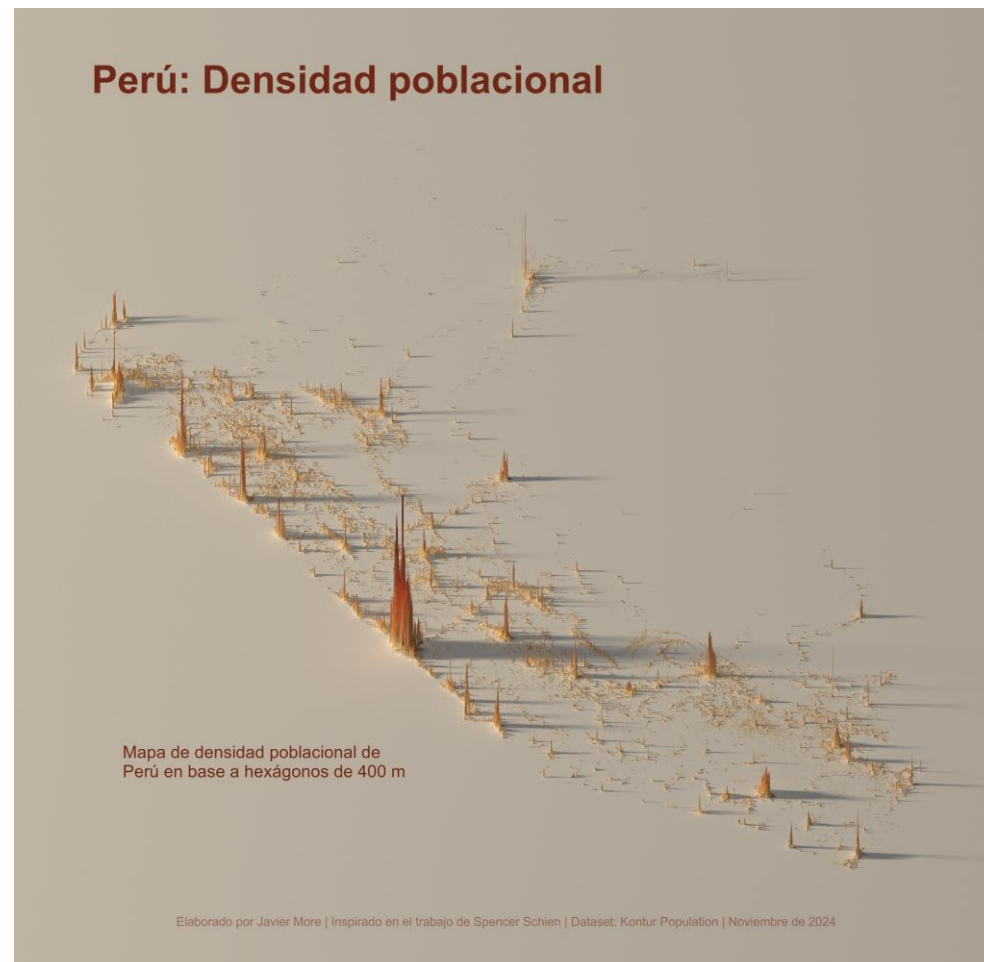


## Perú: Temperatura y precipitación

Temperatura y precipitación promedio (1981-2010)



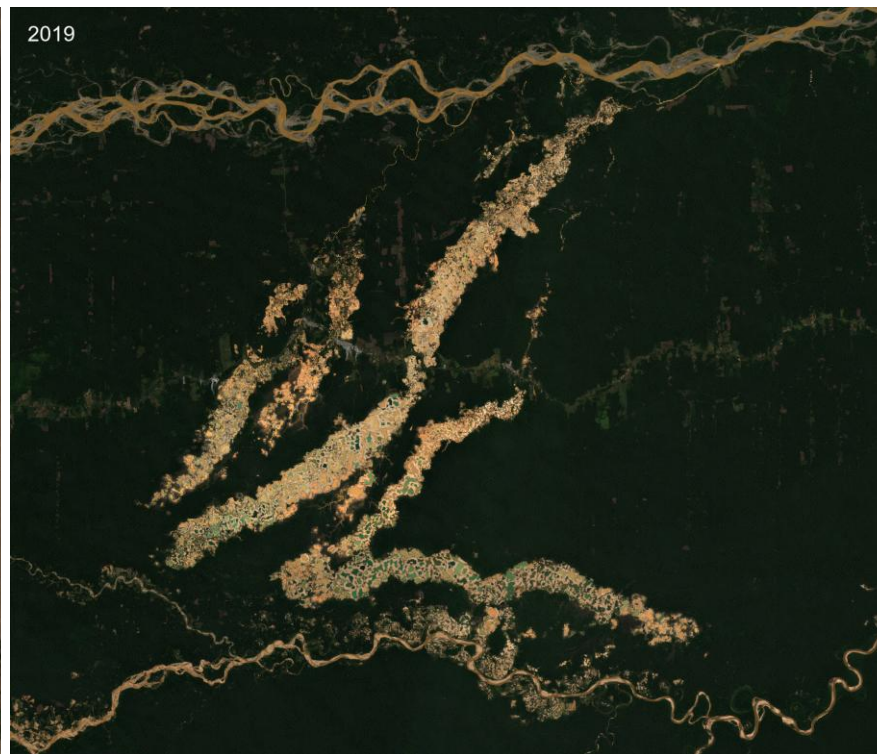
## Perú: Densidad poblacional



# Mapas con QGIS + Plugin Timelapse + Google Earth Engine



**Puerto de Chancay**

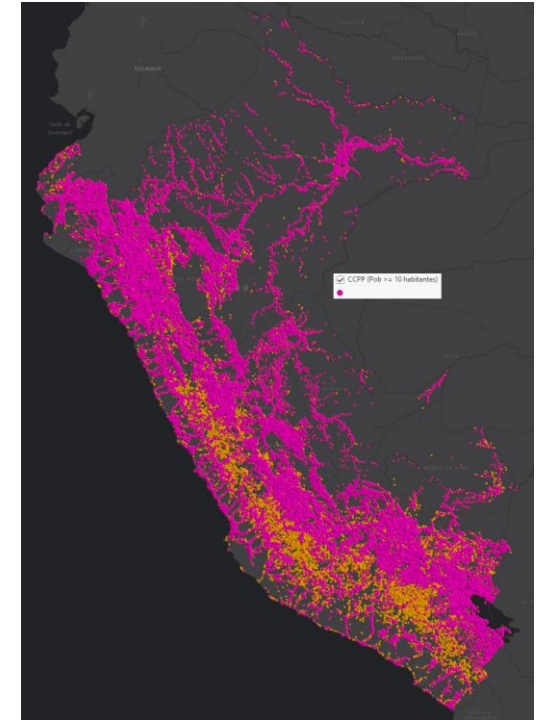
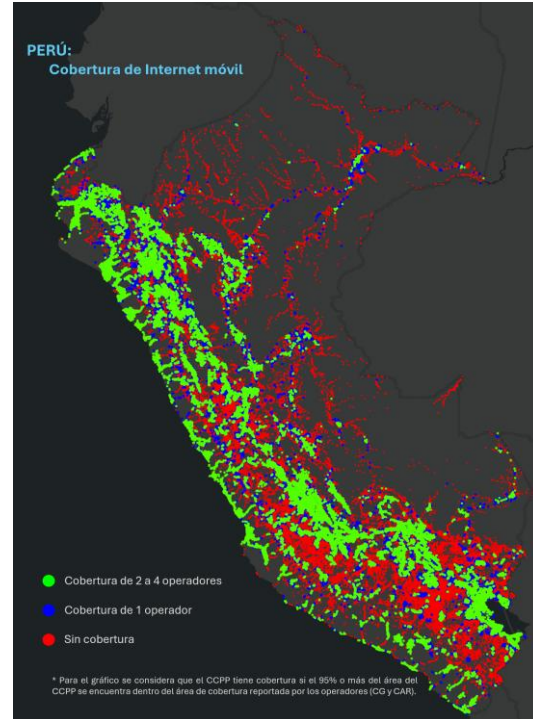
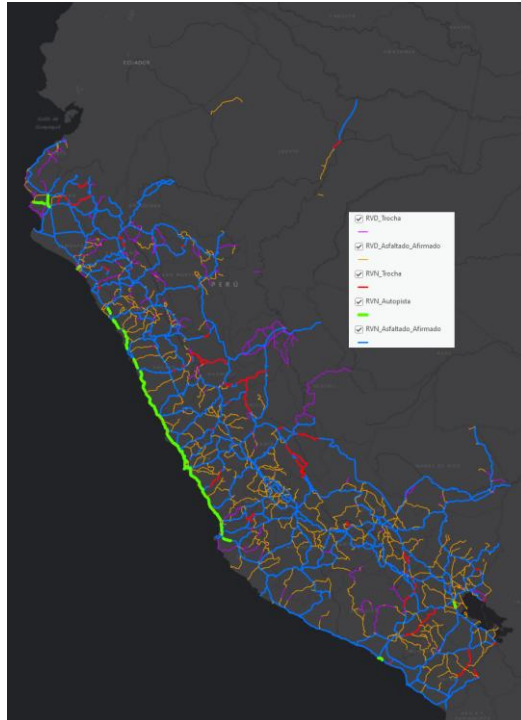


**La Pampa – Madre de Dios**

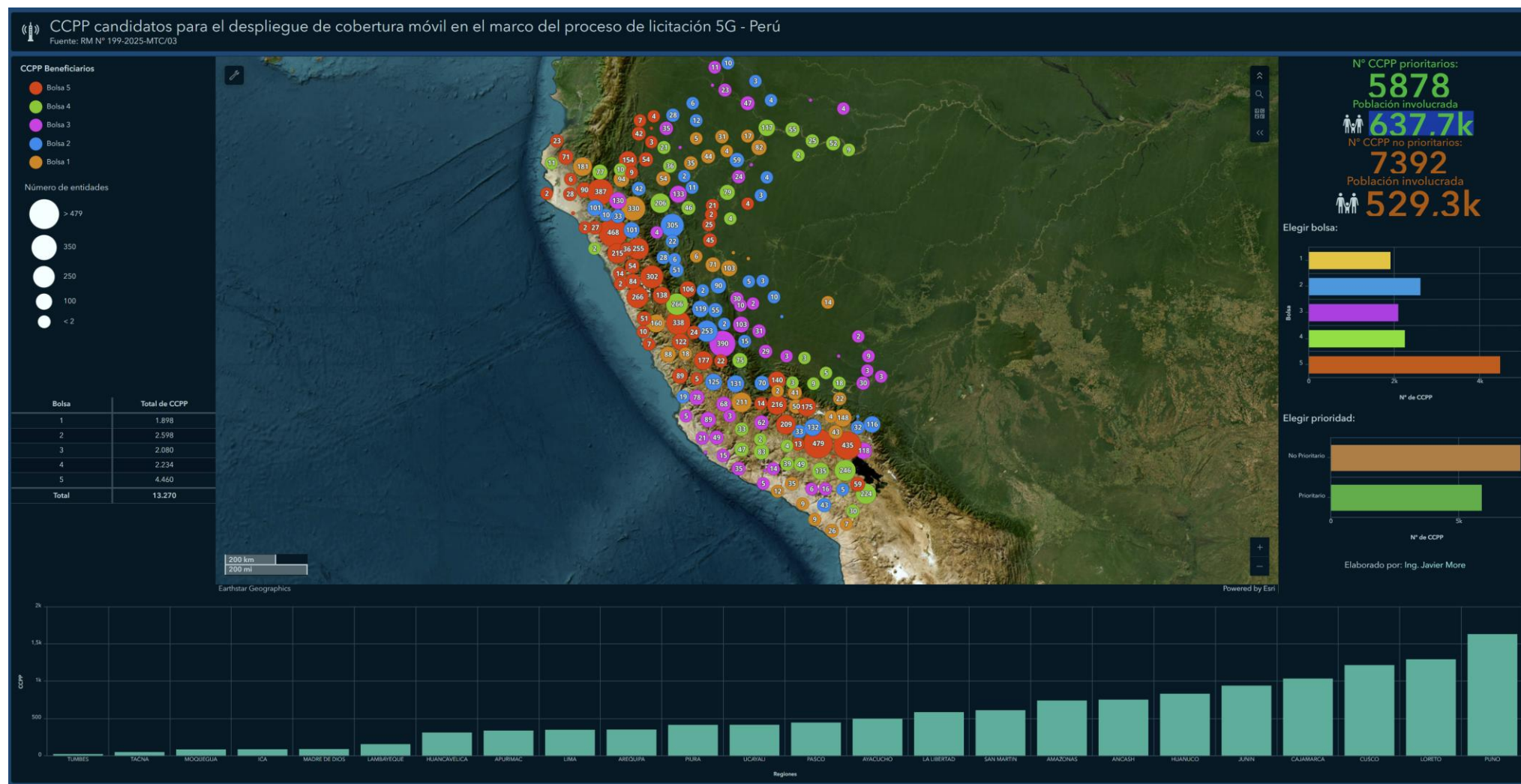


**23 de Octubre – Piura**

# Mapas con ArcGIS

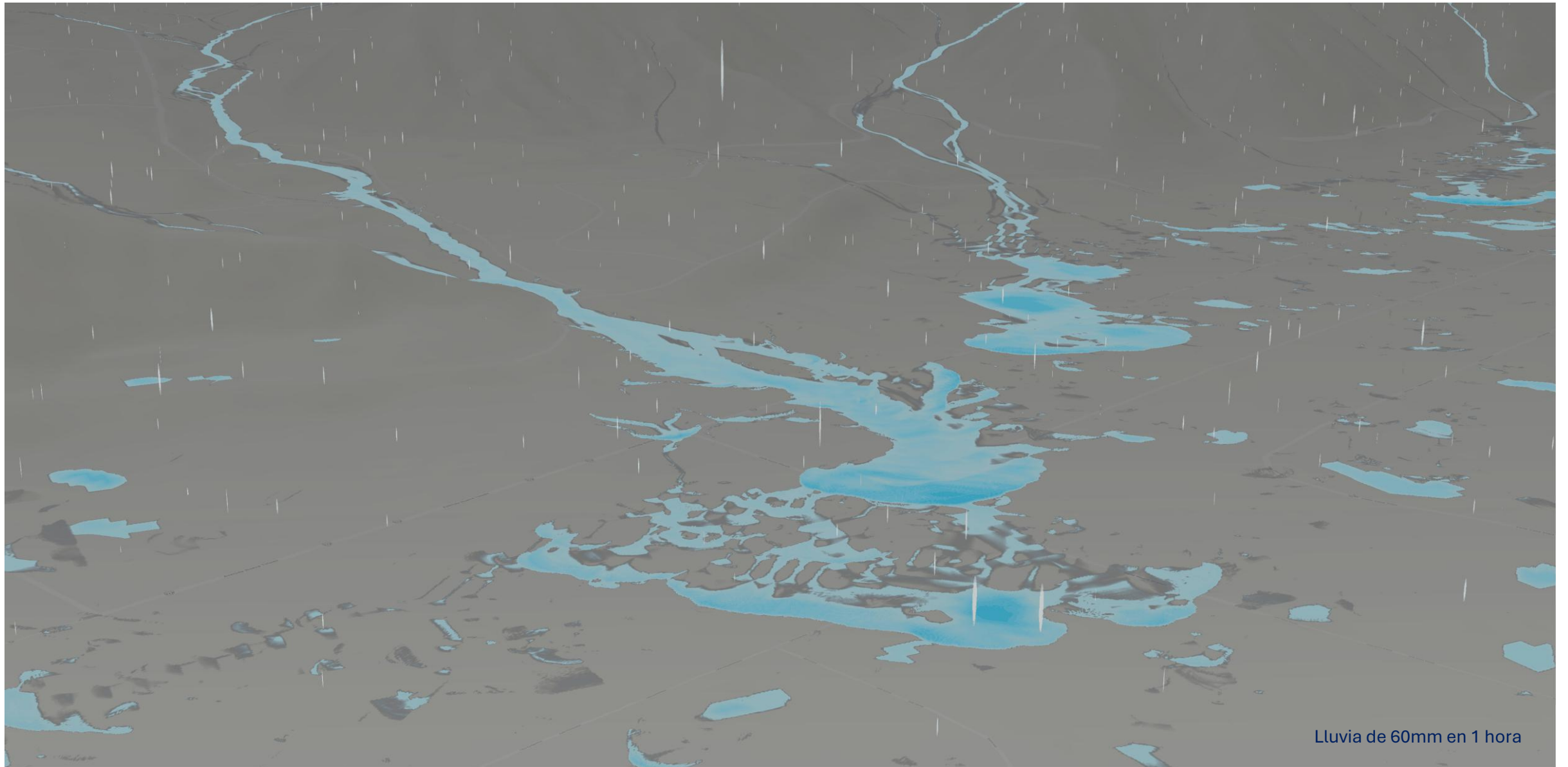


# Dashboard con ArcGIS Online



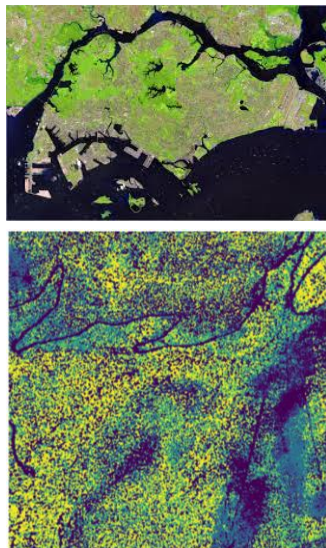
Disponible en: <https://javiermore.maps.arcgis.com/apps/dashboards/d09268f060c14f1eb91b69c0957053f2>

# Simulación de inundaciones con ArcGIS PRO

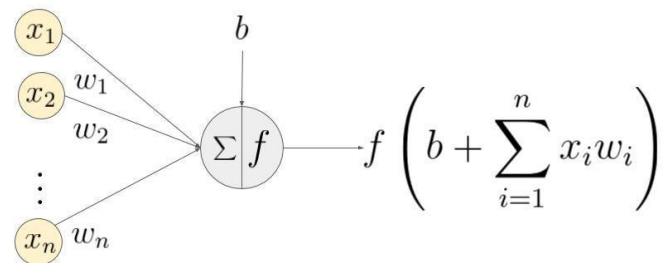


# Estamos en la era de la Inteligencia Artificial

## Datos



## Algoritmos



## Hardware

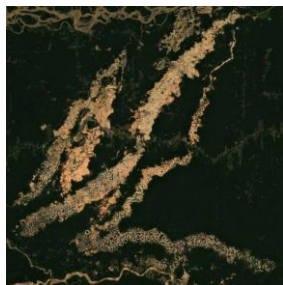


# Librerías para trabajar GeoAI



# GeoAI Workflow

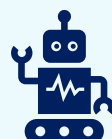
Sentinel 1/2  
Landsat 8/9  
Copernicus DEM  
SRTM  
ASTER GDEM  
ALOS PALSAR



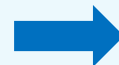
Dataset



Preprocesamiento



Entrenamiento del  
modelo de DL



Evaluación y  
selección de mejor  
modelo

**Training phase**

**Inference phase**

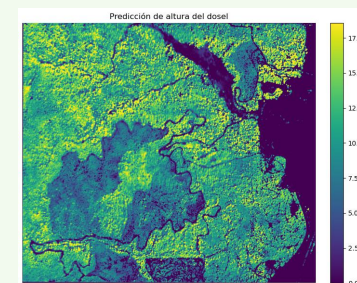
Nuevos  
datos



Modelo  
preentrenado



Data de salida



# Detección de altura de vegetación – Modelo **HighResCanopyHeight** (Meta)



Elaboración: Ing. Javier More con información de:

- **Notebook:** [https://opengeoai.org/examples/canopy\\_height/](https://opengeoai.org/examples/canopy_height/)
- **Github:** <https://github.com/opengeos/qgis-timelapse-plugin>
- **Paper:** <https://www.sciencedirect.com/science/article/pii/S003442572300439X>

# Detección de paneles solares con modelo preentrenado



```
#Carga del modelo preentrenado
detector = geoi.SolarPanelDetector()

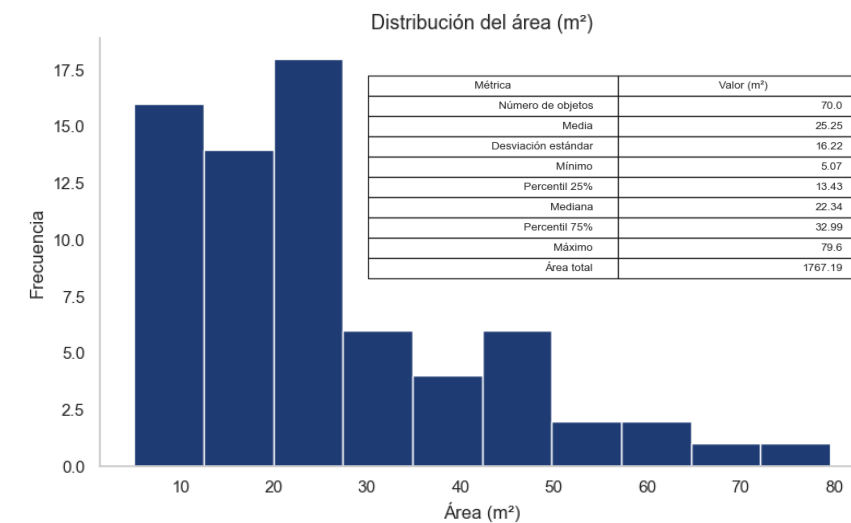
#Imagen de salida
output_path = "solar_panel_masks.tif"

#Entrenamiento
masks_path = detector.generate_masks(
    raster_path,
    output_path=output_path,
    confidence_threshold=0.4,
    mask_threshold=0.5,
    min_object_area=100,
    overlap=0.25,
    chip_size=(400, 400),
    batch_size=4,
    verbose=False,
)
✓ 17.5s
```



Ejemplo adaptado de:

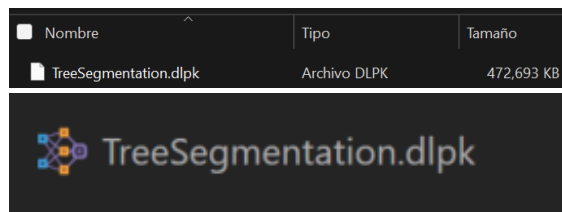
- **Notebook:** [https://opengeoai.org/examples/solar\\_panel\\_detection](https://opengeoai.org/examples/solar_panel_detection)



# Modelo de Deep Learning para detectar árboles en imágenes de alta resolución



**Data de entrada**



Elapsed Time: 12 Minutes 1 Second

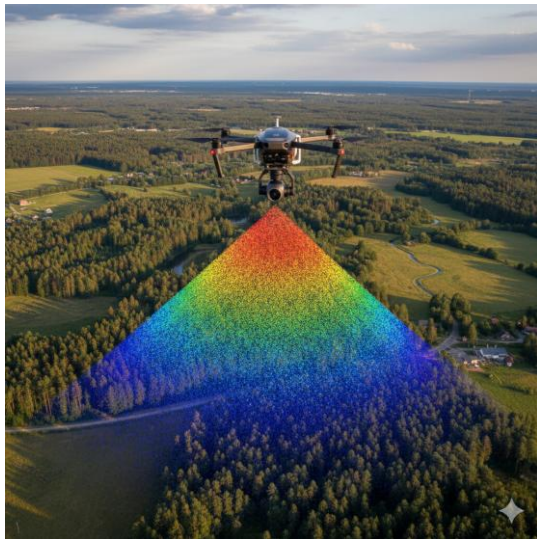


**Data de salida**

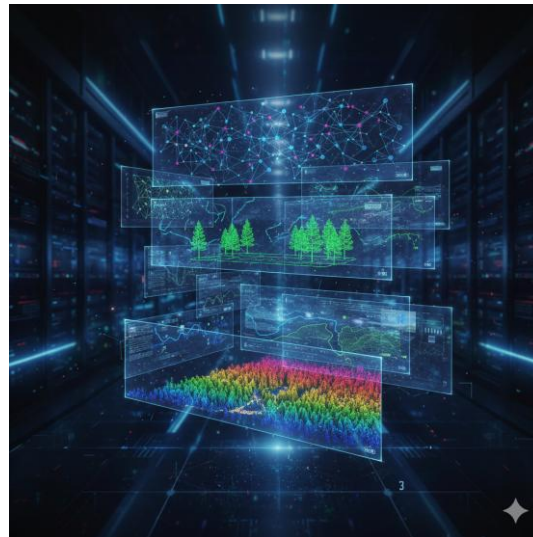
Modelo disponible en: <https://doc.arcgis.com/en/pretrained-models/latest/imagery/introduction-to-tree-segmentation.htm>

# GeoAI: El presente y futuro del GIS

## Geospatial Data



## Deep Learning



## High Performance Computing



Integración con LLM

GeoAI no es una herramienta. Es un nuevo paradigma.



**Javier More Sanchez** ✓

Ing. Electrónico | Mag. Telecomunicaciones | Mag. Regulación SSPP |  
Ms. Gestión de Infraestructuras

Peru · [Contact info](#)



Universidad de Lima



Universidad Nacional Mayor  
de San Marcos

**Gracias**