



EXAMEN UNIDAD II

APLICACIÓN PARA MAPEAR INUNDACIONES DESDE PYTHON, USANDO SNAPPY

Para esta actividad, desarrollaremos una interfaz gráfica de una aplicación, a partir del código de ejemplo provisto en el Campus, para la delimitación de áreas inundadas a partir de imágenes Sentinel 1.

La aplicación debe ser capaz de realizar, al menos, los siguientes pasos:

1. **Seleccionar la imagen a utilizar:** Debe desplegar un panel de navegación para que el usuario pueda seleccionar la ubicación de la imagen. Deberá filtrar únicamente los formatos compatibles con las imágenes Sentinel-1 (.zip). Para esto le será útil explorar la opción “filetypes” del método [filedialog.askopenfilename\(\)](#).
2. **Seleccionar un shapefile como máscara:** Debe desplegar un panel de navegación para que el usuario pueda seleccionar la ubicación de la capa shapefile a utilizar. Deberá filtrar únicamente los formatos compatibles para archivos shapefiles (.shp). Para esto le será útil explorar la opción “filetypes” del método [filedialog.askopenfilename\(\)](#).
3. **Pre-procesar la imagen:** Deberá mostrarle un botón al usuario para que proceda a realizar el pre-procesamiento de la imagen. Este proceso incluirá: Corrección orbital, recorte de la imagen, calibración, aplicación de filtro speckle, Corrección del terreno. Una vez finalizado este proceso, deberá mostrarle, en la misma ventana, la imagen resultante. Para ello le será útil explorar la función “plotBand” incluida en el código, y adaptándola con el método [FigureCanvasTkAgg de matplotlib](#).
4. **Definir umbral para la máscara de agua:** Deberá solicitarse al usuario que ingrese el umbral para la separación del agua, y aplicar la máscara mediante un botón. Este botón deberá aplicar únicamente la máscara, a partir de la imagen ya pre-procesada. Después de finalizado, deberá reemplazar la imagen calibrada, por la imagen de la máscara. Para ello deberá utilizar la misma función “plotBand” debiendo eliminar el canvas creado con el método `FigureCanvasTkAgg`, para volver a mostrar la nueva imagen en el mismo lugar. Para esto le puede ser útil explorar el método `.get_tk_widget().destroy()`.
5. **Crear la imagen en formato GeoTIFF:** Por último, deberá mostrar otro botón para que el usuario cree la imagen final. La imagen deberá crearse en el mismo directorio de la imagen original.



PARA ENTREGAR.

1. El enlace del repositorio en Github con el(los) códigos utilizados para la actividad. **Deberá crear el repositorio antes de comenzar a trabajar**, para que se reflejen todos los cambios realizados a los archivos, haciendo los “commits” necesarios. Le puede ser útil ir construyendo cada función y probándola, para estar seguro de que hace lo que se espera.