

MANUAL DE USUARIO DEL USO DEL PROTOTIPO O6S-CALIBNET

Última actualización: 17 de diciembre del 2025

Versión del manual: 1.0

Compatibilidad: Sistema O6S-CalibNet v1.0+



Mg. Yury Oscar Tello Canchapoma
VºBº Responsable Técnico

DICIEMBRE – 2025

CONTENIDO

1.	Introducción	3
2.	¿Qué hace el sistema?	4
3.	Requisitos básicos para el usuario	4
3.1.	Equipo recomendado	4
3.2.	Software necesario	4
4.	Instalación rápida (paso a paso)	5
4.1.	Paso 1: Crear el entorno del sistema	5
4.2.	Paso 2: Activar el entorno	5
4.3.	Paso 3: Verificar que el sistema esté listo	5
4.4.	Paso 4: Configurar Carpetas:	5
5.	Organización de carpetas (qué debe preparar el usuario)	5
6.	Cómo ejecutar el sistema (uso básico)	6
6.1.	Colocar la imagen satelital	6
6.2.	Ejecutar el procesamiento principal (Fases 1 y 2)	6
6.3.	Resultado esperado	6
7.	Personalización según su uso	6
8.	Productos que genera el sistema	7
9.	Interpretación de Resultados	7
9.1.	Archivos Generados	7
9.2.	Calidad de los Resultados	8
10.	Configuración sencilla (opcional)	8
11.	Buenas prácticas para el usuario	8
12.	¿Para qué sirve este sistema en la práctica?	8
13.	Solución de Problemas y Soporte	9
14.	Solución de Problemas Comunes	9
14.1.	Error: "No se encuentra el entorno py6s-env"	9
14.2.	Error: "Imagen no encontrada"	9
14.3.	Procesamiento muy lento	9
14.4.	Resultados con errores o valores extraños	9
14.5.	No se conecta a Google Earth Engine	9
15.	Reportar Problemas	10
16.	1Enlaces Útiles	10
17.	Checklist de Uso Rápido	10

1. Introducción

El sistema O6S-CalibNet es una herramienta diseñada para procesar imágenes satelitales PeruSat-1 y generar productos listos para análisis (ARD). Está optimizado para condiciones amazónicas y combina correcciones físicas con técnicas de aprendizaje profundo para obtener imágenes radiométricamente consistentes y de alta calidad.

Está pensado para usuarios que necesiten corregir imágenes atmosféricamente y armonizarlas con datos Sentinel-2, sin necesidad de profundizar en el código.

Público objetivo:

- ✓ Técnicos en teledetección
- ✓ Investigadores
- ✓ Operadores de sistemas de monitoreo ambiental
- ✓ Usuarios con conocimientos básicos de Python y línea de comandos

Este sistema está orientado a apoyar aplicaciones como:

- ✓ Monitoreo agrícola sostenible
- ✓ Detección de cambios en la cobertura del suelo
- ✓ Análisis multitemporal con inteligencia artificial

El usuario no necesita modificar código, ya que el funcionamiento se controla mediante archivos de configuración y comandos simples desde la línea de comandos.

El sistema O6S-CalibNet procesa imágenes satelitales en 4 fases principales:

- ✓ Reconstrucción de datos atmosféricos (AOD) usando CatBoost.
- ✓ Corrección atmosférica física con el modelo O6S (basado en 6S).
- ✓ Armonización radiométrica con la red neuronal CalibNet.
- ✓ Validación y control de calidad.

2. ¿Qué hace el sistema?

El sistema ejecuta cuatro pasos automáticos:

- ✓ Reconstrucción de información atmosférica (rellena vacíos de datos).
- ✓ Corrección atmosférica física (elimina efectos de la atmósfera).
- ✓ Armonización radiométrica inteligente (ajusta diferencias entre sensores).
- ✓ Validación automática de calidad (verifica precisión y consistencia).

El resultado final es una imagen satelital confiable, comparable en el tiempo y lista para análisis.

3. Requisitos básicos para el usuario

3.1. Equipo recomendado

Para que el sistema funcione correctamente, su equipo debe cumplir con lo siguiente:

Componente	Requisito Mínimo	Recomendado (Óptimo)
Procesador	Intel i5 (4 núcleos)	Intel i7 / Ryzen 7 (8 núcleos)
Memoria RAM	8 GB	32 GB
Disco Duro	20 GB libres	50 GB SSD
Tarjeta de Video	No integrada	NVIDIA GPU (4GB+ VRAM)

3.2. Software necesario

- ✓ Sistema Operativo: Windows 10/11 o Linux (Ubuntu 20.04+).
- ✓ Gestor de entorno: Anaconda o Miniconda.
- ✓ Acceso a Datos: Una cuenta activa en Google Earth Engine (GEE) para obtener datos climáticos.
- ✓ Anaconda o Miniconda instalado.
- ✓ Acceso a Anaconda Prompt (terminal).

El entorno ya está definido; no se instalan librerías manualmente.

4. Instalación rápida (paso a paso)

4.1. Paso 1: Crear el entorno del sistema

Abra su terminal, en *Anaconda Prompt*, ubicado en la carpeta del proyecto:

```
mamba env create -f environment.yml -n py6s-env
```

4.2. Paso 2: Activar el entorno

```
conda activate py6s-env
```

4.3. Paso 3: Verificar que el sistema esté listo

```
python
from Py6S import SixS
s = SixS()
s.run()
exit() # Si no aparece ningún error, el sistema está correctamente instalado.
```

4.4. Paso 4: Configurar Carpetas:

Debe crear manualmente las carpetas `data/raw`, `data/processed` y `secrets/` en el directorio del proyecto.

5. Organización de carpetas (qué debe preparar el usuario)

El usuario solo debe asegurarse de tener esta estructura básica:

```
data/
├── raw/      ← aquí van las imágenes PeruSat-1
├── processed/ ← aquí se guardan los resultados
├── secrets/   # Crear manualmente
└── proyecto-gee-automatizacion-*.json
```

Las carpetas `data/raw`, `data/processed` y `data/secrets` deben crearse manualmente si no existen.

6. Cómo ejecutar el sistema (uso básico)

6.1. Colocar la imagen satelital

Copie la carpeta completa de la imagen PeruSat-1 (formato DIMAP) dentro de:

`data/raw/`

Ejemplo:

`data/raw/CO_2509101658301/VOL_PER1_ORT_001_001586/IMG_PER1_ORT_MS_001586/`

6.2. Ejecutar el procesamiento principal (Fases 1 y 2)

En *Anaconda Prompt*:

```
python main.py -i data/raw/RUTA_DE_LA_IMAGEN --runner sequential
```

proceso puede tardar varios minutos.

No cierre la ventana hasta que finalice.

6.3. Resultado esperado

Al finalizar, el sistema generará automáticamente archivos en:

`data/processed/`

Estos archivos corresponden a imágenes corregidas atmosféricamente y listas para el siguiente paso.

7. Personalización según su uso

Usted puede cambiar cómo se comporta el programa editando el archivo

`configs/satellites/producto-perusat1.yml`.

- ✓ **Uso Agrícola:** Active el enmascarador-nubes para limpiar las imágenes y calcular índices de vegetación.
- ✓ **Uso en Investigación:** Active el corrector-topografico-scs si su zona de estudio tiene muchas montañas o sombras.
- ✓ **Uso Operativo (Rápido):** Utilice la opción `max_procesos: 4` y `uso_cache: true` para procesar múltiples imágenes en menor tiempo.

8. Productos que genera el sistema

Al finalizar el proceso completo, el usuario obtendrá:

- ✓ Imagen de Reflectancia Superficial Intermedia (GeoTIFF)
- ✓ Métricas de calidad radiométrica (archivo JSON)
- ✓ Modelo entrenado (interno)

Estos resultados permiten:

- ✓ Comparar imágenes en el tiempo
- ✓ Alimentar modelos de IA / CNN
- ✓ Cumplir estándares de calidad ARD

¿Qué archivos se generan?

Después del procesamiento encontrarás:

- ✓ **Archivo principal:** nombre_imagen_O6S.tif - Tu imagen corregida
- ✓ **Archivos auxiliares:**
 - *_cloud_mask.tif - Máscara de nubes
 - *_shadow_mask.tif - Máscara de sombras
 - *_metadata.json - Información del procesamiento

¿Dónde están los resultados?

Todos los archivos procesados se guardan en: data/processed/

9. Interpretación de Resultados

9.1. Archivos Generados

Después del procesamiento, encontrarás en data/processed/:

- ✓ *_O6S.tif → Imagen corregida atmosféricamente
- ✓ *_log.txt → Registro del procesamiento
- ✓ *_metadata.json → Metadatos técnicos

9.2. Calidad de los Resultados

- ✓ Revisa el archivo de log para advertencias o errores
- ✓ Abre la imagen .tif en QGIS o ArcGIS para inspección visual
- ✓ Verifica que los valores estén en un rango razonable (ej: 0–10000)

10. Configuración sencilla (opcional)

Si el usuario desea cambiar el comportamiento del sistema (por ejemplo, activar o desactivar pasos), NO debe editar código.

Solo debe modificar el archivo:

configs/satellites/producto-perusat1.yml

Ejemplo simple:

pipeline_receta:

- ✓ Calibrador-lineal
- ✓ Corrector-atmosferico-6s
- ✓ Ffinalizador-uint16
- ✓ Active el modo detallado si es necesario:
--log-level DEBUG

11. Buenas prácticas para el usuario

- ✓ Usar siempre el entorno py6s-env
- ✓ Mantener ordenadas las carpetas raw y processed
- ✓ Revisar los mensajes en pantalla durante la ejecución
- ✓ Conservar los archivos de salida como respaldo

12. ¿Para qué sirve este sistema en la práctica?

Este prototipo puede utilizarse para:

- ✓ Monitoreo agrícola sostenible
- ✓ Detección de deforestación
- ✓ Preparación de datos para CNN
- ✓ Soporte a regulaciones ambientales
- ✓ Investigación científica y operativa

13. Solución de Problemas y Soporte

Si el programa falla, asegúrese de:

- ✓ Revisar el archivo de registro en logs/pipeline.log.
- ✓ Verificar que tiene conexión a internet para las funciones de Google Earth Engine.
- ✓ Contacto: Si necesita soporte técnico, adjunte siempre su archivo de configuración y el log en modo DEBUG.

14. Solución de Problemas Comunes

14.1. Error: "No se encuentra el entorno py6s-env"

- ✓ Solución: Activar el entorno con conda activate py6s-env

14.2. Error: "Imagen no encontrada"

- ✓ Verifica que la ruta sea correcta y use / en lugar de \
- ✓ Asegúrate de que la carpeta data/raw/ exista

14.3. Procesamiento muy lento

- ✓ Usa --runner parallel -n 8 para paralelizar
- ✓ Verifica que tengas suficiente RAM

14.4. Resultados con errores o valores extraños

- ✓ Revisa los logs en logs/pipeline.log
- ✓ Ejecuta con --log-level DEBUG para más detalle

14.5. No se conecta a Google Earth Engine

- ✓ Configura las credenciales en secrets/
- ✓ O cambia a fuentes locales en la configuración YAML

15. Reportar Problemas

Incluye siempre:

- ✓ Versión del entorno (conda list)
- ✓ Comando exacto que ejecutaste
- ✓ Archivo de log completo
- ✓ Captura de pantalla del error (si aplica)

16. Enlaces Útiles

- ✓ Py6S: <https://py6s.readthedocs.io/>
- ✓ Google Earth Engine: <https://earthengine.google.com/>

17. Checklist de Uso Rápido

- ✓ Instalado Miniconda y creado el entorno
- ✓ Carpeta `data/raw/` creada con imágenes dentro
- ✓ Entorno activado (`conda activate py6s-env`)
- ✓ Ejecutado `python main.py -i ruta/imagen` sin errores
- ✓ Resultados verificados en `data/processed/`
- ✓ Logs revisados para advertencias