

Google Earth Engine

1. ¿Qué es Google Earth Engine?

Google Earth Engine (GEE) es una plataforma basada en la nube para el análisis y procesamiento de grandes volúmenes de datos geoespaciales. Su principal ventaja radica en la capacidad de acceder y analizar colecciones masivas de imágenes satelitales, permitiendo realizar estudios ambientales, monitoreo de cambios en la superficie terrestre, análisis de desastres naturales, entre otros.

GEE proporciona una interfaz de programación basada en JavaScript a través de su **Code Editor**, así como una API en **Python**, facilitando el acceso a datos geoespaciales desde cualquier parte del mundo sin necesidad de infraestructura computacional avanzada.

2. Características principales

Google Earth Engine destaca por las siguientes características:

- **Acceso a datos geoespaciales:** dispone de una amplia colección de imágenes satelitales y datos geoespaciales, incluyendo sensores como **Landsat**, **MODIS**, **Sentinel-2**, entre otros.
- **Computación en la nube:** los cálculos se realizan en servidores de Google, eliminando la necesidad de procesar datos localmente.
- **Interfaz interactiva:** permite la visualización de datos en mapas de forma rápida e intuitiva.
- **APIs en JavaScript y Python:** facilita la integración con otros entornos de desarrollo y herramientas como **QGIS** y **Jupyter Notebook**.

- **Soporte para big data:** procesa grandes volúmenes de datos de manera eficiente, utilizando métodos de reducción y filtrado espacial y temporal.
- **Descarga de datos:** permite exportar imágenes y resultados en diferentes formatos para su posterior análisis.

3. Registro en Google Earth Engine

Para utilizar Google Earth Engine es necesario registrarse con una cuenta de Google:

1. Acceder a la página oficial de **Google Earth Engine** (<https://earthengine.google.com/>).
2. Hacer clic en **Sign Up** (Registrarse) y completar el formulario con información personal y un propósito de uso.
3. Aceptar los términos y condiciones.
4. Esperar la confirmación por correo electrónico.
5. Una vez aprobado, acceder al **Code Editor** en <https://code.earthengine.google.com/>.

4. Elementos de Google Earth Engine

GEE cuenta con diversos elementos que facilitan la manipulación y análisis de datos geoespaciales:

- **Code Editor:** entorno de desarrollo donde se pueden escribir y ejecutar scripts.
- **Catálogo de datos:** biblioteca con múltiples colecciones de imágenes satelitales y datos vectoriales.
- **APIs (JavaScript y Python):** conjunto de funciones para la manipulación de datos.
- **Almacenamiento en la nube:** Permite cargar y manejar datos propios.

- **Sistema de permisos:** control de acceso a los datos y colaboración en proyectos.

5. New Script o Nuevo Guion

El **New Script** o **Nuevo Guion** en Google Earth Engine permite a los usuarios escribir y ejecutar código en JavaScript para analizar imágenes satelitales y otros datos geoespaciales.

Pasos para crear un script:

1. Ingresar al **Code Editor**.
2. Hacer clic en **File → New Script**.
3. Asignar un nombre y guardarlo.
4. Escribir el código y ejecutarlo con el botón **Run**.

Ejemplo de script para visualizar la cobertura terrestre de España con datos MODIS:

```
// Cargar los límites de España
var spain = ee.FeatureCollection(«FA0/GAUL/2015/level0»)
.filter(ee.Filter.eq('ADM0_NAME', 'Spain'));

// Cargar imagen MODIS
var landcover = ee.Image('MODIS/006/MCD12Q1/2020_01_01')
.select('LC_Type1');

// Visualizar en el mapa
Map.addLayer(landcover, {min: 1, max: 17, palette: ['05450a',
'086a10', '54a708']}, 'Cobertura MODIS');
Map.centerObject(spain, 6);
```

6. Ventana del «Inspector», «Tasks»

y «Console»

Google Earth Engine proporciona herramientas clave dentro del **Code Editor**:

- **Inspector**: permite examinar valores de píxeles al hacer clic en el mapa.
- **Tasks**: gestiona las tareas de exportación de datos e imágenes.
- **Console**: muestra información de depuración y mensajes de salida de los scripts.

Ejemplo de uso del **Inspector**:

```
Map.onClick(function(coords) {  
  var point = ee.Geometry.Point(coords.lon, coords.lat);  
  var value = landcover.sample(point,  
30).first().get('LC_Type1');  
  value.evaluate(console.log);  
});
```

7. Descarga de datos desde Google Earth Engine

GEE permite descargar datos en formatos como **GeoTIFF** y **CSV**.

Ejemplo de exportación de una imagen:

```
Export.image.toDrive({  
  image: landcover,  
  description: 'MODIS_Landcover_Spain',  
  scale: 500,  
  region: spain.geometry(),  
  fileFormat: 'GeoTIFF'  
});
```

8. Gestión de errores

En la programación con GEE, pueden surgir errores debido a formatos de datos incorrectos o funciones mal implementadas. Algunas estrategias para gestionar errores incluyen:

- **Uso de try...catch en Python.**
- **Depuración con print() en JavaScript.**
- **Revisión de la consola para mensajes de error.**
- **Validación de geometrías antes de ejecutar funciones.**

Ejemplo de manejo de errores en JavaScript:

```
try {  
  var resultado = landcover.reduceRegion({  
    reducer: ee.Reducer.mean(),  
    geometry: spain.geometry(),  
    scale: 500  
  });  
  print(resultado);  
} catch (error) {  
  print('Error en la ejecución:', error.message);  
}
```

9. Visualización de datos en QGIS

Para integrar datos de GEE en **QGIS**, se pueden seguir estos pasos:

1. **Usar la extensión GEE Plugin en QGIS.**
2. **Descargar imágenes desde GEE y cargarlas en QGIS.**
3. **Conectar con la API de GEE en Python.**

Ejemplo de código en Python para obtener datos desde GEE en QGIS:

```
import ee
import geemap

# Inicializar Earth Engine
ee.Initialize()

# Cargar imagen de cobertura terrestre
landcover = ee.Image('MODIS/006/MCD12Q1/2020_01_01').select('LC_Type1')

# Mostrar en QGIS
Map = geemap.Map()
Map.addLayer(landcover, {}, 'Cobertura MODIS')
Map
```