

Clase 02: Fundamentos de Geocomputación

Historia, herramientas y ecosistema de desarrollo

Prof. Francisco Parra O.

Geólogo, PhD en Informática

20 de agosto de 2025

USACH - Ingeniería Civil en Informática




Agenda de hoy

 **Duración: 80 minutos**

Conceptos clave:

- Definición de Geocomputación
- Tipos de datos espaciales
- Aplicaciones en Chile
- Proyecto semestral

Recordatorio:

-  Instalar software antes del lab
-  Pensar en grupos de proyecto
-  Completar encuesta diagnóstica

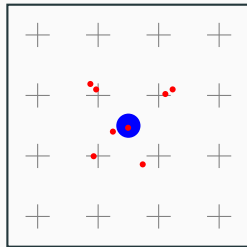
Hoy profundizaremos en los fundamentos y herramientas

Historia y evolución de la Geocomputación

Los precursores: Cartografía y computación

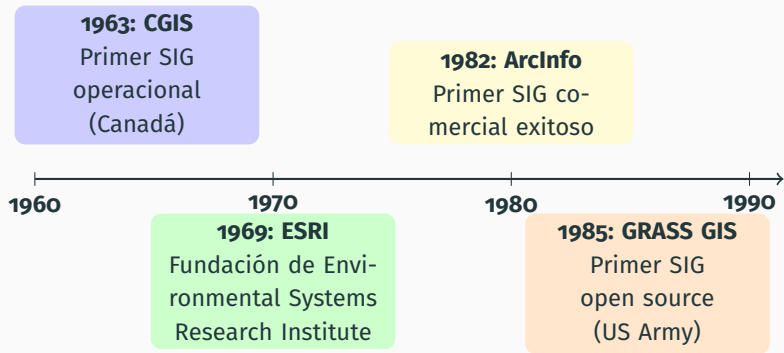
Antes de los SIG (pre-1960):

- Mapas en papel y overlay manual
- Primeros intentos de cuantificación espacial
- John Snow (1854): Mapa del cólera en Londres
- Von Thünen (1826): Modelo de uso del suelo



Mapa de John Snow (1854)

La era de los SIG (1960-1990)



Características de esta era:

- Digitalización de mapas analógicos
- Mainframes y minicomputadoras
- Alto costo y especialización
- Enfoque en gestión de datos más que análisis

El nacimiento de la Geocomputación (1990-2000)

Conferencia inaugural de Geocomputación (1996)

Universidad de Leeds, Reino Unido - Primera conferencia internacional que acuñó el término

¿Qué diferencia la Geocomputación del SIG tradicional?

Cambio de paradigma:

- De gestión a análisis
- De datos a conocimiento
- De mapas a modelos
- De estático a dinámico

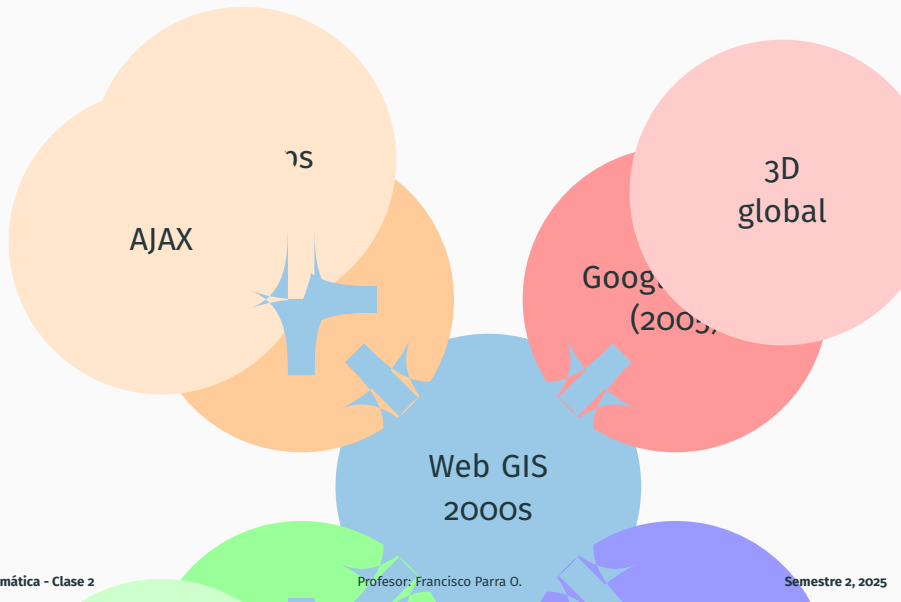
Nuevas técnicas:

- Autómatas celulares
- Redes neuronales espaciales
- Algoritmos genéticos
- Simulación basada en agentes

"Geocomputation is about using the various different types of geodata and






about developing relevant geo-tools within an 'intelligent' IT

La revolución Web GIS (2000-2010)

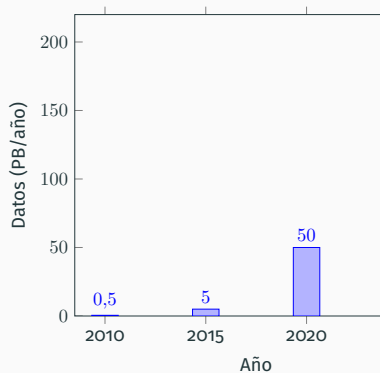


Era actual: Big Data y AI Geoespacial (2010-presente)

Características actuales:

-  **Big Data:** Petabytes de datos satelitales
-  **Cloud Computing:** Google Earth Engine, AWS
-  **AI/ML:** Deep Learning para imágenes
-  **Ubicuidad:** GPS en cada smartphone
-  **Real-time:** Análisis en tiempo real

Ejemplo Chile: Plataforma IDE Chile integra 50+ servicios geoespaciales

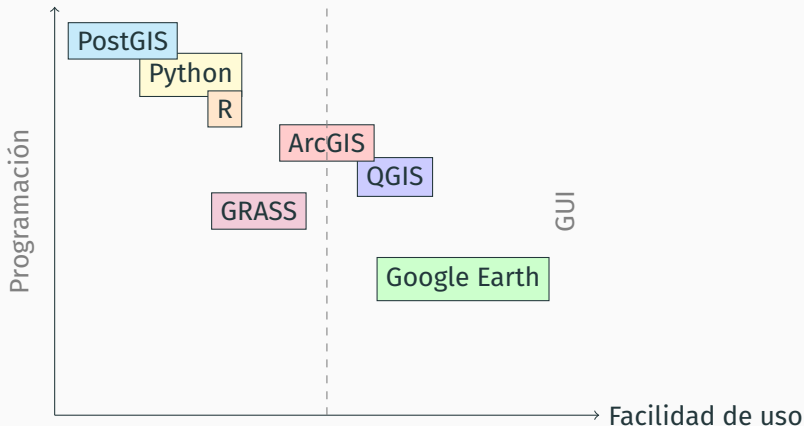


Crecimiento exponencial de datos geoespaciales

Software para análisis geoespacial

Panorama del software geoespacial

Capacidad analítica



Software Desktop GIS

Open Source:

- **QGIS** (Quantum GIS)
 - Más popular
 - Interfaz amigable
 - Plugins extensivos
- **GRASS GIS**
 - Análisis avanzado
 - Desde 1985
- **SAGA GIS**
 - Geomorfología
 - Análisis de terreno

Comercial:

- **ArcGIS Pro** (ESRI)
 - Estándar industria
 - Integración completa
 - Alto costo
- **MapInfo**
 - Business intelligence
- **Global Mapper**
 - LiDAR processing

Para este curso:

Usaremos QGIS como complemento visual (10 % del tiempo)

QGIS vs ArcGIS Pro: Comparación

Característica	QGIS	ArcGIS Pro
Costo	Gratis	\$700/año (estudiante)
Sistema Operativo	Win/Mac/Linux	Solo Windows
Curva aprendizaje	Moderada	Empinada
Documentación	Comunitaria	Profesional
Soporte Python	✓	✓
Análisis 3D	Básico	Avanzado
Procesamiento	PyQGIS	ArcPy
Cloud Integration	Limitada	ArcGIS Online

💡 *QGIS es perfectamente capaz para el 95 % de tareas GIS*

Plataformas Cloud para Geocomputación

Google Earth Engine

- Petabytes de imágenes satelitales
- Procesamiento en la nube
- JavaScript/Python API
- Gratuito para académicos

AWS

- S3 para almacenamiento
- EC2 para procesamiento
- SageMaker para ML

Microsoft Azure

- Azure Maps
- Planetary Computer
- AI for Earth

Mapbox

- Mapas personalizados
- APIs de geocoding
- Navegación

 Cloud computing es esencial para Big Data geoespacial

Bases de Datos Espaciales

¿Por qué bases de datos espaciales?

- Consultas espaciales SQL
- Índices espaciales para performance
- Integridad referencial
- Multi-usuario concurrente

PostGIS (PostgreSQL)

- Open source líder
- Completo soporte OGC
- Funciones espaciales avanzadas

Otras opciones:

- Oracle Spatial
- MySQL Spatial
- SQLite/SpatiaLite
- MongoDB (NoSQL)

```
SELECT nombre, ST_Area(geom)/10000 as hectareas  
FROM parcelas  
WHERE ST_Within(geom, (SELECT geom FROM comunas WHERE  
nombre='Santiago'))
```

Herramientas de línea de comandos

GDAL/OGR: La navaja suiza del GIS

GDAL (Raster)

- `gdalinfo`: Información
- `gdal_translate`: Conversión
- `gdalwarp`: Reproyección
- `gdal_calc.py`: Álgebra

OGR (Vector)

- `ogrinfo`: Información
- `ogr2ogr`: Conversión
- `ogrmerge`: Fusión
- `ogr_layer_algebra`: Operaciones


```
# Convertir shapefile a GeoJSON
ogr2ogr -f "GeoJSON" output.json input.shp

# Reproyectar raster a UTM 19S
gdalwarp -t_srs EPSG:32719 input.tif output.tif
```


Ecosistema Python y R para geodatos

¿Por qué programar para GIS?

Ventajas de programar:

-  Reproducibilidad
-  Automatización
-  Escalabilidad
-  Versionado
-  Compartir código

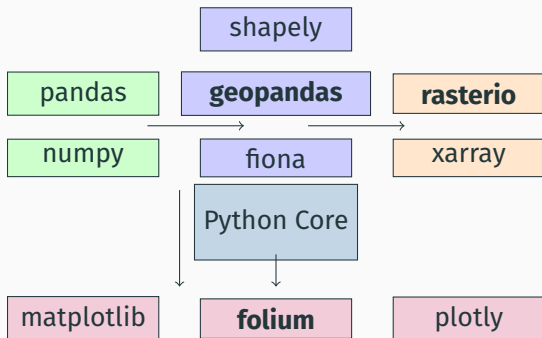
Proceso Manual
100 archivos
8 horas

Script Python
100 archivos
10 minutos

Regla de oro:

Si lo haces más de 3 veces → automatízalo con código

Ecosistema Python para Geoinformática



Bibliotecas clave:

- **geopandas:**
DataFrames espaciales
- **rasterio:**
Lectura/escritura

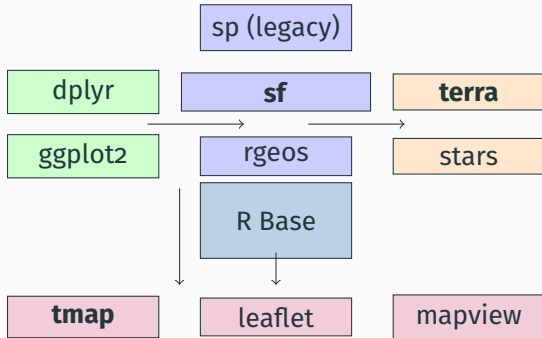
- raster**
- **shapely:**
Geometrías
- **folium:** Mapas

- web**
- **contextily:**
Mapas base
- **osmnx:** Redes OSM

Ejemplo: Análisis espacial con Python

```
1 import geopandas as gpd
2 import matplotlib.pyplot as plt
3 from shapely.geometry import Point
4
5 # Leer datos de comunas de Santiago
6 comunas = gpd.read_file('santiago_comunas.shp')
7
8 # Crear puntos de inter s (hospitales)
9 hospitales = gpd.GeoDataFrame(
10     {'nombre': ['Hospital 1', 'Hospital 2'],
11     'geometry': [Point(-70.65, -33.45),
12                 Point(-70.60, -33.42)]},
13     crs='EPSG:4326')
14
15 # Buffer de 2km alrededor de hospitales
16 areas_servicio = hospitales.buffer(0.02)
17
18 # Encontrar comunas servidas
19 comunas_servidas = comunas[comunas.intersects(
20     areas_servicio.unary_union)]
21
22 print("Comunas con cobertura: ", len(comunas_servidas))
```

Ecosistema R para Geoinformática



Paquetes clave:

- **sf**: Simple Features
- **terra**: Datos raster
- **tmap**: Mapas temáticos
- **leaflet**: Mapas interactivos
- **rayshader**: 3D
- **gstat**: Geoestadística

Ejemplo: Análisis espacial con R

```
1 library(sf)
2 library(tmap)
3 library(dplyr)
4
5 # Leer datos de comunas
6 comunas <- st_read("santiago_comunas.shp")
7
8 # Crear puntos de hospitales
9 hospitales <- data.frame(
10   nombre = c("Hospital 1", "Hospital 2"),
11   lon = c(-70.65, -70.60),
12   lat = c(-33.45, -33.42)
13 ) %>%
14   st_as_sf(coords = c("lon", "lat"), crs = 4326)
15
16 # Buffer de 2km
17 areas_servicio <- st_buffer(hospitales, dist = 2000)
18
19 # Intersección con comunas
20 comunas_servidas <- comunas[areas_servicio, ]
21
22 # Visualizar
```

Python vs R para Geoinformática

Aspecto	Python	R
Sintaxis	Más general	Más estadística
Curva aprendizaje	Más suave	Más empinada
Velocidad	✓ Más rápido	Más lento
Visualización estática	Buena	✓ Excelente
Machine Learning	✓ scikit-learn	Bueno
Deep Learning	✓ TensorFlow	Limitado
Estadística espacial	Buena	✓ Excelente
Integración web	✓ Django/Flask	Shiny
Comunidad GIS	Grande	Grande

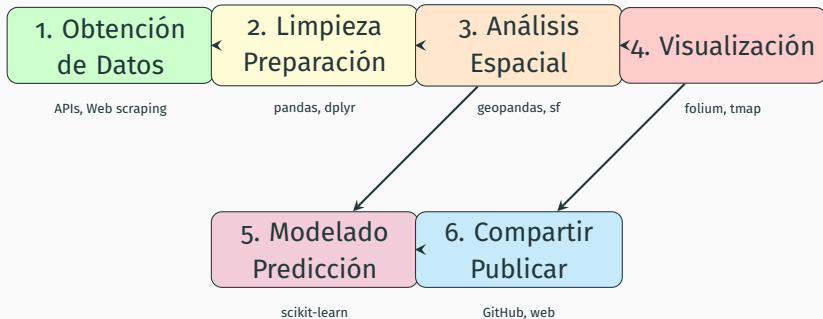
Recomendación del curso:

Python para producción y escalabilidad

R para exploración y visualización estadística

Primeros ejemplos prácticos

Flujo de trabajo típico en Geocomputación



Caso práctico: Análisis COVID-19 en RM

Objetivo: Identificar zonas de alto riesgo en Santiago

Datos necesarios:

- Casos por comuna (MINSAL)
- Población (INE)
- Geometrías comunas
- Centros de salud
- Movilidad (opcional)

Pasos del análisis:

1. Calcular tasa incidencia
2. Detectar clusters (Moran's I)
3. Analizar accesibilidad salud
4. Crear mapa de riesgo
5. Dashboard interactivo

Resultado: Mapa de calor identificando Las Condes, Vitacura y Providencia como clusters iniciales (marzo 2020)

Demo: Configuración del ambiente

1. Verificar instalación Python:

```
1 python --version # Debe ser 3.8+
2 pip --version
3
```

2. Crear ambiente virtual:

```
1 python -m venv geo_env
2 source geo_env/bin/activate # Linux/Mac
3 # o
4 geo_env\Scripts\activate # Windows
5
```

3. Instalar bibliotecas geoespaciales:

```
1 pip install geopandas folium matplotlib
2 pip install rasterio contextily
3 pip install jupyter notebook
4
```



Esto lo haremos juntos en el laboratorio




Tu primer script geoespacial

```
1 # archivo: mi_primer_mapa.py
2 import geopandas as gpd
3 import matplotlib.pyplot as plt
4
5 # 1. Descargar datos de ejemplo
6 world = gpd.read_file(gpd.datasets.get_path('naturalearth_lowres'))
7
8 # 2. Filtrar Am rica del Sur
9 sudamerica = world[world['continent'] == 'South America']
10
11 # 3. Calcular centroides
12 sudamerica['centroid'] = sudamerica.geometry.centroid
13
14 # 4. Crear figura con subplots
15 fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(15, 8))
16
17 # 5. Mapa 1: Poblaci n
18 sudamerica.plot(column='pop_est',
19                 ax=ax1,
20                 legend=True,
21                 cmap='YlOrRd',
22                 edgecolor='black')
23 ax1.set_title('Poblaci n de Sudam rica')
24
25 # 6. Mapa 2: PIB per c pita
26 sudamerica.plot(column='gdp_md_est',
27                 ax=ax2,
28                 legend=True,
29                 cmap='Greens',
30                 edgecolor='black')
31 ax2.set_title('PIB de pa ses sudamericanos')
32
33 plt.tight_layout()
34 plt.savefig('sudamerica_analisis.png', dpi=300)
35 plt.show()
```

Preparación para el laboratorio

Laboratorio 1: Configuración del ambiente

Objetivos del laboratorio de hoy:

1.  **Instalación completa**
 - Python + Anaconda/Miniconda
 - R + RStudio
 - QGIS (opcional)
 - Git
2.  **Verificación**
 - Importar bibliotecas sin errores
 - Ejecutar script de prueba
3.  **Primer mapa**
 - Cargar datos de Chile
 - Visualización básica
 - Exportar resultado

Importante:

Traer computador con permisos de administrador para instalar software

Recursos para profundizar

Libros online gratuitos:

- Geocomputation with Python
- Geocomputation with R
- Geographic Data Science





Documentación oficial:

- GeoPandas docs
- sf package docs
- QGIS docs

Cursos online:

- Coursera: GIS Specialization
- edX: Spatial Data Science
- YouTube: GeoDelta Labs

Comunidades:

-  Stack Overflow GIS
-  r/gis
-  #gischat
-  awesome-gis

Proyecto semestral: Empezar a pensar

Brainstorming de ideas para proyecto:

Comercial:

- Optimización de rutas delivery
- Análisis inmobiliario Santiago
- Ubicación óptima de sucursales
- Segmentación de clientes

Científico:

- Contaminación del aire RM
- Expansión urbana Santiago
- Riesgo de incendios V Región
- Acceso a áreas verdes

Semana 4: Formación de grupos (1-3 personas)

Empiencen a conversar con compañeros sobre intereses comunes

Resumen: Conceptos clave de hoy

Historia:

- 1960s: Primeros SIG
- 1996: Nace Geocomputación
- 2005: Revolución Web GIS
- Hoy: Big Data + AI

Software:

- Desktop: QGIS vs ArcGIS
- Cloud: Google Earth Engine
- BD: PostGIS
- CLI: GDAL/OGR




Programación:

- Python: geopandas, rasterio
- R: sf, terra, tmap
- Reproducibilidad es clave
- Automatización ahorra tiempo

Para recordar:

- Open source es viable
- Python líder en la industria
- R excelente para estadística
- Práctica hace al maestro

Para la próxima clase

1.  **URGENTE:** Instalar software antes del laboratorio
 - Python (Anaconda) + bibliotecas
 - R + RStudio + paquetes
 - QGIS (opcional pero recomendado)
2.  **Lectura:** Capítulo 1 de Geocomputation with Python
 - <https://py.geocompx.org/01-introduction>
3.  **Reflexionar:**
 - ¿Qué problema espacial te gustaría resolver?
 - ¿Qué datos necesitarías?

 Dudas: francisco.parra.o@usach.cl

¡Nos vemos en el laboratorio!

¿Preguntas?

A continuación:

Laboratorio 1 - Configuración del ambiente

Mismo salón, 10 minutos de break



Pueden ir por un café mientras preparamos el laboratorio