

# 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
-  Pensar en grupos
-  Encuesta diagnóstica

*Hoy profundizaremos en los fundamentos y herramientas*

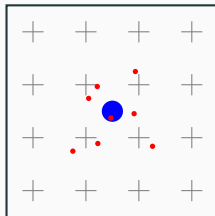
# **Historia y evolución de la Geocomputación**

---

# Los precursores: Cartografía y computación

## Antes de los SIG (pre-1960):

- Mapas en papel
- Cuantificación espacial
- John Snow (1854): Cólera
- Von Thünen (1826)



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

# El nacimiento de la Geocomputación (1990-2000)

Conferencia inaugural (1996)

Universidad de Leeds - Primera conferencia internacional

## ¿Qué diferencia la Geocomputación del SIG tradicional?

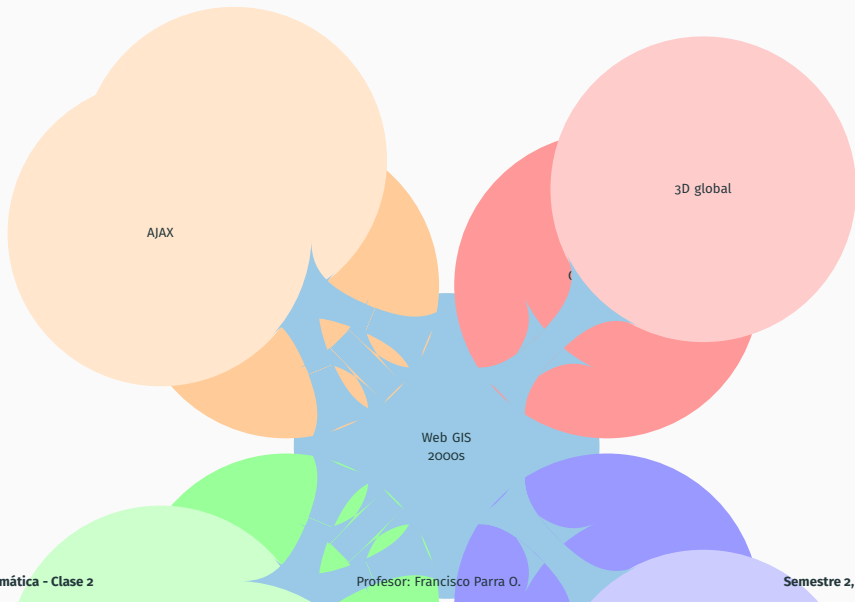
### Cambio de paradigma:

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

### Nuevas técnicas:

- Autómatas celulares
- Redes neuronales
- Algoritmos genéticos
- Simulación de agentes






# La revolución Web GIS (2000-2010)

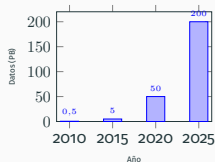




# Era actual: Big Data y AI Geoespacial (2010-presente)

## Características actuales:

-  **Big Data:** Petabytes
-  **Cloud:** GEE, AWS
-  **AI/ML:** Deep Learning
-  **Ubicuidad:** GPS
-  **Real-time**

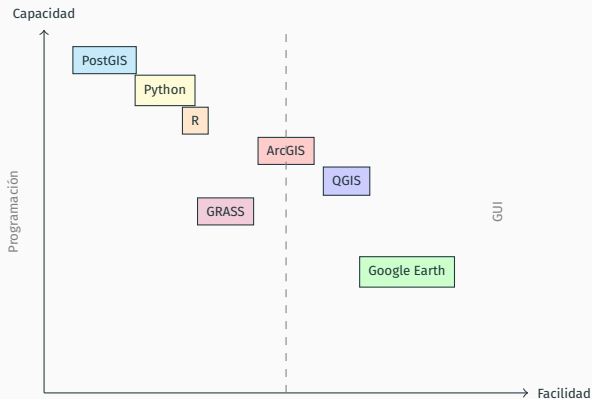


**Ejemplo Chile:** IDE Chile integra 50+ servicios geoespaciales

# **Software para análisis geoespacial**

---

# Panorama del software geoespacial



# Software Desktop GIS

## Open Source:

- **QGIS**
  - Más popular
  - Interfaz amigable
- **GRASS GIS**
  - Análisis avanzado
- **SAGA GIS**
  - Geomorfología

## Comercial:

- **ArcGIS Pro**
  - Estándar industria
  - Alto costo
- **MapInfo**
  - Business intelligence
- **Global Mapper**
  - LiDAR processing

# QGIS vs ArcGIS Pro: Comparación

Característica	QGIS	ArcGIS
Costo	Gratis	\$700/año
Sistema Operativo	Todos	Windows
Curva aprendizaje	Moderada	Empinada
Documentación	Comunitaria	Profesional
Python	✓	✓
Análisis 3D	Básico	Avanzado
Cloud	Limitada	ArcGIS Online

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

# Plataformas Cloud para Geocomputación

## Google Earth Engine

- Petabytes de imágenes
- Procesamiento en nube
- JavaScript/Python API

## AWS

- S3 almacenamiento
- EC2 procesamiento
- SageMaker ML

## Microsoft Azure

- Azure Maps
- Planetary Computer
- AI for Earth

## Mapbox

- Mapas personalizados
- APIs geocoding
- Navegación

# 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
- Soporte OGC completo
- Funciones 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, zona)
```

# Herramientas de línea de comandos

## GDAL/OGR: La navaja suiza del GIS

### GDAL (Raster)

- `gdalinfo`: Info
- `gdal_translate`: Conversión
- `gdalwarp`: Reproyección

### OGR (Vector)

- `ogrinfo`: Info
- `ogr2ogr`: Conversión
- `ogrmerge`: Fusión

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



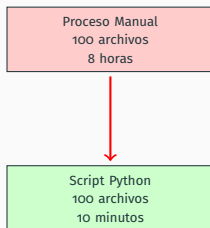
# **Ecosistema Python y R para geodatos**

---

# ¿Por qué programar para GIS?

## Ventajas de programar:

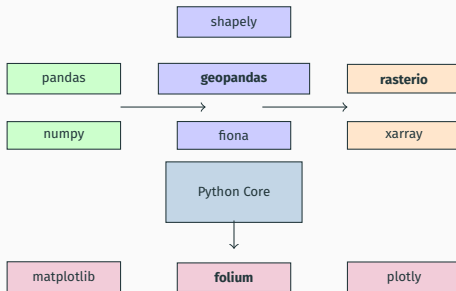
-  Reproducibilidad
-  Automatización
-  Escalabilidad
-  Versionado
-  Compartir



## Regla de oro:

Si lo haces más de 3 veces → automatízalo

# Ecosistema Python para Geoinformática



## Bibliotecas clave:

- geopandas
- rasterio
- shapely
- folium
- contextily
- osmnx

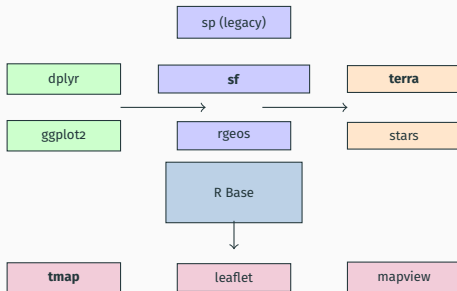
# Ejemplo Python: Configuración

```
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 interes (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
```

# Ejemplo Python: Análisis

```
1 # Buffer de 2km alrededor de hospitales
2 areas_servicio = hospitales.buffer(0.02)
3
4 # Encontrar comunas servidas
5 comunas_servidas = comunas[
6     comunas.intersects(areas_servicio.unary_union)
7 ]
8
9 # Visualizar resultado
10 ax = comunas.plot(color='lightgray', edgecolor='black')
11 areas_servicio.plot(ax=ax, color='blue', alpha=0.3)
12 hospitales.plot(ax=ax, color='red', markersize=50)
13
14 print(f"Comunas con cobertura: {len(comunas_servidas)}")
15
```

# Ecosistema R para Geoinformática



## Paquetes clave:

- `sf`
- `terra`
- `tmap`
- `leaflet`
- `rayshader`
- `gstat`

# Ejemplo R: Configuración

```
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
```

# Ejemplo R: Análisis y visualización

```
1 # Buffer de 2km
2 areas_servicio <- st_buffer(hospitales, dist = 2000)
3
4 # Intersección con comunas
5 comunas_servidas <- comunas[areas_servicio, ]
6
7 # Visualizar con tmap
8 tm_shape(comunas) +
9   tm_polygons(col = "gray90") +
10  tm_shape(areas_servicio) +
11    tm_borders("red", lwd = 2) +
12  tm_shape(hospitales) +
13    tm_dots(size = 0.5, col = "blue")
14
```



# Python vs R para Geoinformática

Aspecto	Python	R
Sintaxis	General	Estadística
Velocidad	✓ Rápido	Más lento
Visualización	Buena	✓ Excelente
Machine Learning	✓ sklearn	Bueno
Deep Learning	✓ TensorFlow	Limitado
Estadística espacial	Buena	✓ Excelente
Integración web	✓ Django	Shiny

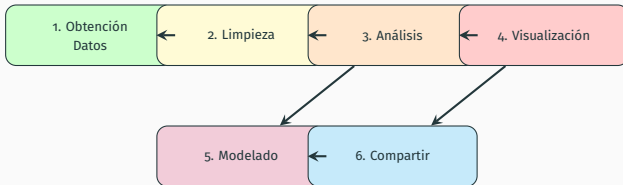
## Recomendación:

**Python** para producción, **R** para exploración

## **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
- Población (INE)
- Geometrías comunas
- Centros de salud

**Pasos del análisis:**

1. Calcular tasa incidencia
2. Detectar clusters
3. Analizar accesibilidad
4. Crear mapa de riesgo

**Resultado:** 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 geo_env\Scripts\activate    # Windows
4
```

## 3. Instalar bibliotecas:

```
1 pip install geopandas folium matplotlib
2 pip install jupyter notebook
3
```

# Tu primer script: Importar y filtrar

```
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(
7     gpd.datasets.get_path('naturalearth_lowres')
8 )
9
10 # 2. Filtrar América del Sur
11 sudamerica = world[
12     world['continent'] == 'South America'
13 ]
14
15 # 3. Calcular centroides
16 sudamerica['centroid'] = sudamerica.geometry.centroid
17
```

# Tu primer script: Visualizar

```
1 # 4. Crear figura con subplots
2 fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(12, 6))
3
4 # 5. Mapa 1: Poblaci n
5 sudamerica.plot(column='pop_est', ax=ax1,
6                 legend=True, cmap='YlOrRd',
7                 edgecolor='black')
8 ax1.set_title('Poblaci n de Sudam rica')
9
10 # 6. Mapa 2: PIB
11 sudamerica.plot(column='gdp_md_est', ax=ax2,
12                 legend=True, cmap='Greens',
13                 edgecolor='black')
14 ax2.set_title('PIB de pa ses sudamericanos')
15
16 plt.tight_layout()
17 plt.savefig('sudamerica.png', dpi=150)
18 plt.show()
19
```



# **Preparación para el laboratorio**

---



# Laboratorio 1: Configuración del ambiente

## Objetivos del laboratorio de hoy:

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

Importante:

Traer computador con permisos de administrador

# Recursos para profundizar

## Libros online gratuitos:

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

## Documentación:

- GeoPandas docs
- sf package docs
- QGIS docs

## Cursos online:

- Coursera: GIS
- edX: Spatial Data
- YouTube: GeoDelta

## Comunidades:

- Stack Overflow GIS
- r/gis
- #gischat

# Proyecto semestral: Ideas

## Comercial:

- Optimización rutas
- Análisis inmobiliario
- Ubicación sucursales

## Científico:

- Contaminación aire
- Expansión urbana
- Riesgo incendios

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

# Resumen: Conceptos clave

## Historia:

- 1963: Primer SIG
- 1996: Geocomputación
- 2005: Web GIS
- Hoy: Big Data + AI

## Software:

- QGIS vs ArcGIS
- Google Earth Engine
- PostGIS




## Programación:


- Python: geopandas
- R: sf, terra
- Reproducibilidad
- Automatización

## Recordar:

- Open source viable
- Python en industria
- R para estadística

# Para la próxima clase

1.  **URGENTE:** Instalar software
  - Python (Anaconda)
  - R + RStudio
  - QGIS (opcional)
2.  **Lectura:** Cap. 1 Geocomputation with Python
3.  **Reflexionar:**
  - ¿Qué problema espacial resolver?
  - ¿Qué datos necesitarías?

 francisco.parra.o@usach.cl

¡Nos vemos en el laboratorio!

¿Preguntas?

**A continuación:**

Laboratorio 1 - Configuración

10 minutos de break



*Pueden ir por un café*