

Manual del Profesor

Clase 02: Fundamentos de Geocomputación

Historia, herramientas y ecosistema de desarrollo

Prof. Francisco Parra O.
Geólogo, PhD en Informática
`francisco.parra.o@usach.cl`

Semestre 2, 2025
Duración: 80 minutos

Índice

1. Introducción	3
1.1. Objetivos de Aprendizaje	3
2. Preparación Previa (15 minutos antes de clase)	3
2.1. Checklist de Materiales	3
2.2. Configuración del Ambiente	4
3. Desarrollo de la Clase	5
3.1. Apertura y Bienvenida [0:00-0:02]	5
3.2. Repaso de Clase Anterior [0:03-0:06]	5
4. Sección 1: Historia y Evolución [20 minutos totales]	6
4.1. Los Precursores de la Geocomputación [0:06-0:09]	6
4.1.1. Historia de John Snow (1854)	6
4.1.2. Conexión con el Presente	6
4.2. La Era de los SIG (1960-1990) [0:09-0:13]	6
4.2.1. CGIS - El Primer SIG	6
4.2.2. Nacimiento de ESRI	7
4.3. El Nacimiento de la Geocomputación [0:13-0:17]	7
4.3.1. Cambio de Paradigma (1996)	7
4.3.2. Diferencia Fundamental	7
4.4. Era Web GIS y Big Data [0:17-0:26]	8
4.4.1. La Revolución de Google Maps (2005)	8
4.4.2. Caso Chile: Terremoto 27F	8
4.4.3. Big Data Geoespacial Actual	8
5. Sección 2: Software para Análisis Geoespacial [20 minutos]	9
5.1. Panorama del Software [0:26-0:29]	9
5.2. QGIS vs ArcGIS [0:29-0:36]	9
5.2.1. Historia de QGIS	9
5.2.2. Comparación Objetiva	9
5.3. Plataformas Cloud [0:36-0:40]	10
5.3.1. Google Earth Engine	10
5.3.2. Otras Plataformas	10
5.4. Bases de Datos Espaciales [0:40-0:46]	10
5.4.1. ¿Por qué PostGIS?	10
6. Sección 3: Ecosistema Python y R [25 minutos]	11
6.1. ¿Por qué Programar para GIS? [0:46-0:49]	11
6.1.1. El Argumento de Eficiencia	11
6.2. Ecosistema Python [0:49-0:58]	11
6.2.1. Bibliotecas Principales	11
6.2.2. Ejemplo Completo Python	12
6.3. Ecosistema R [0:58-1:07]	12
6.3.1. Filosofía de R	12
6.3.2. Ejemplo R	12
6.4. Python vs R: La Verdad [1:07-1:11]	12

7. Sección 4: Ejemplos Prácticos [15 minutos]	14
7.1. Flujo de Trabajo Típico [1:11-1:14]	14
7.2. Caso COVID-19 en Santiago [1:14-1:19]	14
7.3. Configuración del Ambiente [1:19-1:22]	14
7.4. Primer Script Completo [1:22-1:26]	15
8. Cierre y Preparación para Laboratorio [10 minutos]	16
8.1. Objetivos del Laboratorio [1:26-1:29]	16
8.2. Recursos para Profundizar [1:29-1:32]	16
8.3. Ideas para Proyecto [1:32-1:34]	16
8.4. Tareas y Cierre [1:34-1:40]	17
9. Anexos	18
9.1. Preguntas Frecuentes	18
9.2. Actividades Interactivas	18
9.3. Troubleshooting Común	19
9.4. Checklist Post-Clase	19
9.5. Métricas de Éxito	19

1 Introducción

Este manual proporciona una guía detallada para dictar la Clase 02 del curso de Geoinformática. Está diseñado para ser completamente autocontenido, permitiendo que cualquier profesor pueda conducir la clase exitosamente.

1.1 Objetivos de Aprendizaje

Al finalizar esta clase de 80 minutos, los estudiantes serán capaces de:

1. **Comprender** la evolución histórica desde los SIG tradicionales hasta la Geocomputación moderna
2. **Identificar** las principales herramientas de software geoespacial y sus casos de uso
3. **Comparar** las fortalezas y debilidades de Python vs R para análisis espacial
4. **Ejecutar** un primer script geoespacial básico en Python
5. **Planificar** su ambiente de trabajo para el laboratorio

2 Preparación Previa (15 minutos antes de clase)

2.1 Checklist de Materiales

Hardware y Software:

- ☐ Computador con Python (Anaconda) y R (RStudio) instalados
- ☐ Presentación PDF cargada en modo presentación
- ☐ Conexión a internet estable (tener hotspot móvil como backup)
- ☐ Proyector/pantalla funcionando correctamente
- ☐ Adaptadores de video necesarios

Materiales de Clase:

- ☐ Pizarra y al menos 3 plumones de colores diferentes
- ☐ Mouse presentador o puntero láser
- ☐ Botella de agua para hidratación
- ☐ Lista de asistencia (si aplica)

Archivos Digitales:

- ☐ santiago_comunas.shp - Shapefile de comunas
- ☐ hospitales.csv - Datos de ubicación
- ☐ demo_script.py - Script Python funcional
- ☐ demo_script.R - Script R funcional
- ☐ Links de recursos en documento compartido

2.2 Configuración del Ambiente

```
1 # Activar ambiente Python
2 cd ~/demos_clase02
3 conda activate geo # o source geo_env/bin/activate
4
5 # Verificar instalaciones
6 python --version
7 python -c "import geopandas; print('OK')"
8
9 # Abrir Jupyter
10 jupyter notebook
```

Listing 1: Terminal 1: Preparar Python

```
1 # En RStudio, pre-cargar bibliotecas
2 library(sf)
3 library(tmap)
4 library(dplyr)
5
6 # Verificar datos
7 file.exists("santiago_comunas.shp")
```

Listing 2: Terminal 2: Preparar R

3 Desarrollo de la Clase

3.1 Apertura y Bienvenida [0:00-0:02]

DECIR

Buenos días a todos. Bienvenidos a nuestra segunda clase de Geoinformática. Hoy profundizaremos en los fundamentos de la Geocomputación, su historia fascinante, y las herramientas que usaremos durante todo el semestre.

HACER

- Hacer contacto visual con varios estudiantes
- Verificar que todos pueden ver la presentación
- Ajustar iluminación si es necesario
- Confirmar funcionamiento del audio

NOTA

El tono debe ser entusiasta pero profesional. Es jueves primera hora, los estudiantes pueden estar cansados.

3.2 Repaso de Clase Anterior [0:03-0:06]

DECIR

Antes de comenzar, hagamos un breve repaso. En nuestra primera clase definimos qué es la Geocomputación. ¿Alguien recuerda la diferencia entre datos vectoriales y raster?

HACER

- Esperar 30 segundos para respuestas voluntarias
- Si no hay respuestas, señalar a un estudiante específicamente
- Agradecer cada participación: "Exacto, muy bien"
- Complementar las respuestas si es necesario

Respuesta esperada: Vectorial = puntos, líneas, polígonos con coordenadas precisas. Raster = grilla de celdas/píxeles con valores.

ALERTA

Si menos del 30 % instaló software, dedicar 5 minutos extra en el laboratorio para instalación.

4 Sección 1: Historia y Evolución [20 minutos totales]

4.1 Los Precursores de la Geocomputación [0:06-0:09]

4.1.1. Historia de John Snow (1854)

DECIR

La Geocomputación no comienza con computadores. Comienza en 1854 con un médico llamado John Snow en Londres, durante una epidemia de cólera.

HACER

- Dibujar en la pizarra un mapa simple:
 - Rectángulos para manzanas de casas
 - X para marcar casos de cólera
 - Círculo para la bomba de agua
- Usar colores diferentes para enfatizar el patrón

Elementos clave para mencionar:

- Teoría predominante: "miasmas." aire contaminado
- Método de Snow: mapeo sistemático de casos
- Descubrimiento: clustering alrededor de bomba de Broad Street
- Solución: remover manija de la bomba
- Resultado: caída dramática de casos

NOTA

Esta historia conecta emocionalmente a los estudiantes con el poder del análisis espacial.

4.1.2. Conexión con el Presente

DECIR

¿Les suena familiar? Es exactamente lo que hicimos con COVID-19 en 2020, pero ahora con millones de datos procesados en tiempo real. La pregunta fundamental no ha cambiado: ¿dónde están los casos y qué patrón revelan?

4.2 La Era de los SIG (1960-1990) [0:09-0:13]

4.2.1. CGIS - El Primer SIG

DECIR

1963 marca el nacimiento del primer SIG operacional: CGIS - Canada Geographic Information System.

Contexto importante:

- Canadá: 2do país más grande del mundo (10 millones km²)
- Necesidad: inventario de recursos naturales
- Imposible mapear manualmente millones de hectáreas
- Costo: \$15 millones CAD (1960) = \$150 millones actuales

Capacidad revolucionaria de CGIS:

DECIR

CGIS podía superponer capas de información - algo revolucionario. Imaginen: una capa de suelos, otra de vegetación, otra de hidrología. La pregunta "muéstrame áreas con suelo tipo A, bosque de coníferas, cerca de un río" tomaba horas en mainframes del tamaño de esta sala.

4.2.2. Nacimiento de ESRI

DECIR

1969: Jack y Laura Dangermond fundan ESRI en su garage en California. Hoy es una empresa de 2 billones de dólares. ArcGIS sigue siendo el estándar de la industria.

4.3 El Nacimiento de la Geocomputación [0:13-0:17]

4.3.1. Cambio de Paradigma (1996)

DECIR

En los 90s, los académicos sentían que los SIG se habían estancado. Eran buenos para mapear y almacenar, pero ¿para entender procesos? ¿Para predecir? ¿Para simular?

La cita fundamental de Stan Openshaw:

"Geocomputation is about using the various different types of geodata and about developing relevant geo-tools within an 'intelligent' IT framework"

4.3.2. Diferencia Fundamental

HACER

Escribir en la pizarra una tabla comparativa:

SIG Tradicional	Geocomputación
¿Dónde está el hospital más cercano?	¿Dónde construir el próximo hospital?
¿Cuántos incendios hubo?	¿Dónde será el próximo incendio?
¿Cuál es la ruta más corta?	¿Cómo evolucionará el tráfico?
Responde preguntas	Hace predicciones

Nuevas técnicas introducidas:

- Autómatas celulares (física)
- Redes neuronales (IA)

- Algoritmos genéticos (biología)
- Simulación basada en agentes

4.4 Era Web GIS y Big Data [0:17-0:26]

4.4.1. La Revolución de Google Maps (2005)

DECIR

El 8 de febrero de 2005 cambió todo. Google lanza Google Maps. De repente, cualquier persona podía hacer análisis espacial sin saberlo.

Contraste dramático:

- ANTES: Software \$5000, semanas de training, datos caros
- DESPUÉS: Navegador web, gratis, intuitivo, datos incluidos

4.4.2. Caso Chile: Terremoto 27F

DECIR

El terremoto del 27F en 2010 fue uno de los primeros desastres mapeados colaborativamente. Voluntarios mundiales usaron OpenStreetMap para mapear zonas afectadas en tiempo real. El gobierno chileno usó estos mapas para coordinar ayuda.

4.4.3. Big Data Geoespacial Actual

Crecimiento exponencial de datos:

- 2010: 0.5 petabytes/año
- 2025: 200 petabytes/año (400x más)
- Contexto: 1 petabyte = 31 años si 1GB = 1 segundo

Fuentes de datos actuales:

1. Sentinel-2: fotografía toda la Tierra cada 5 días (1.5 TB/día)
2. 5 billones de smartphones con GPS
3. IoT: Santiago tiene 500+ sensores de calidad del aire
4. Un vuelo LiDAR: 100 GB/hora

5 Sección 2: Software para Análisis Geoespacial [20 minutos]

5.1 Panorama del Software [0:26-0:29]

HACER

Mostrar y explicar el diagrama de dispersión en la diapositiva

Explicación del diagrama:

- Eje X: Facilidad de uso
- Eje Y: Capacidad analítica

Analogía útil:

DECIR

Es como herramientas de un mecánico:

- Google Earth = Destornillador básico
- QGIS = Caja de herramientas completa
- Python/R = Taller profesional
- PostGIS = Fábrica de herramientas

5.2 QGIS vs ArcGIS [0:29-0:36]

5.2.1. Historia de QGIS

DECIR

2002: Gary Sherman, frustrado con el costo de ArcGIS, crea QGIS. Lo libera gratis. Hoy tiene millones de usuarios. Un desarrollador cambió la industria.

5.2.2. Comparación Objetiva

Característica	QGIS	ArcGIS Pro
Costo	Gratis	\$700-3000/año
Sistema Operativo	Win/Mac/Linux	Solo Windows
Curva aprendizaje	Moderada	Empinada
Documentación	Comunitaria	Profesional
Python	Sí	Sí
3D	Básico	Avanzado

NOTA

Ser honesto: ArcGIS es más pulido, pero QGIS hace 95 % de las tareas igual de bien.

5.3 Plataformas Cloud [0:36-0:40]

5.3.1. Google Earth Engine

DECIR

Google Earth Engine procesa 40 años de imágenes satelitales en segundos. Es procesamiento distribuido masivo.

ALERTA

Si hay internet estable, considerar demo de 1 minuto en code.earthengine.google.com

5.3.2. Otras Plataformas

- **Microsoft Planetary Computer:** Enfoque en datos abiertos
- **AWS:** Poder computacional crudo
- **Mapbox:** APIs de mapas y navegación

Problema del cloud: Vendor lock-in y costos ocultos a largo plazo.

5.4 Bases de Datos Espaciales [0:40-0:46]

5.4.1. ¿Por qué PostGIS?

DECIR

Imaginen Uber: “Encuentra el conductor más cercano”. Con 10,000 conductores activos, calcular distancia a cada uno tomaría segundos. Inaceptable. PostGIS usa índices R-tree: encuentra el más cercano en milisegundos.

Ejemplo de query espacial:

```
1 SELECT nombre, ST_Area(geom)/10000 as hectareas
2 FROM parcelas
3 WHERE ST_Within(geom,
4   (SELECT geom FROM comunas WHERE nombre='Santiago')
5 );
```

6 Sección 3: Ecosistema Python y R [25 minutos]

6.1 ¿Por qué Programar para GIS? [0:46-0:49]

6.1.1. El Argumento de Eficiencia

DECIR

Caso real: Cliente necesita análisis mensual de 100 comunas. Manual: 2 días. Con Python: 10 minutos. Después de 3 meses, el script ahorró 35 días de trabajo.

HACER

Escribir en la pizarra la regla de oro:

Si lo haces más de 3 veces → AUTOMATÍZALO

Ventajas de programar:

1. Reproducibilidad
2. Automatización
3. Escalabilidad
4. Versionado (Git)
5. Compartir código

6.2 Ecosistema Python [0:49-0:58]

6.2.1. Bibliotecas Principales

GeoPandas - El corazón:

DECIR

GeoPandas es pandas + geografía. Si saben pandas, ya saben 70 % de GeoPandas.

```
1 # Pandas normal
2 df[df['poblacion'] > 100000]
3
4 # GeoPandas - igual pero con superpoderes
5 gdf[gdf['poblacion'] > 100000].plot() # Mapa!
```

Otras bibliotecas clave:

- **Shapely:** Geometrías puras
- **Rasterio:** Datos raster
- **Folium:** Mapas web (3 líneas = mapa interactivo)
- **Contextily:** Mapas base

6.2.2. Ejemplo Completo Python

```

1 import geopandas as gpd
2 from shapely.geometry import Point
3
4 # Leer comunas
5 comunas = gpd.read_file('santiago_comunas.shp')
6
7 # Crear hospitales
8 hospitales = gpd.GeoDataFrame(
9     {'nombre': ['Hospital 1', 'Hospital 2'],
10      'geometry': [Point(-70.65, -33.45),
11                  Point(-70.60, -33.42)]},
12     crs='EPSG:4326')
13
14 # Buffer 2km y análisis
15 areas = hospitales.buffer(0.02)
16 servidas = comunas[comunas.intersects(areas.unary_union)]
17
18 print(f"Comunas con cobertura: {len(servidas)}")

```

6.3 Ecosistema R [0:58-1:07]

6.3.1. Filosofía de R

DECIR

R fue creado por estadísticos, para estadísticos. Trata todo como datos para analizar. Python trata todo como objetos para procesar.

Paquetes principales:

- **sf**: Simple Features - revolucionó R espacial
- **terra**: Raster, 10x más rápido que el antiguo 'raster'
- **tmap**: Mapas de calidad publicación
- **leaflet**: Mapas interactivos

6.3.2. Ejemplo R

```

1 library(sf)
2 library(tmap)
3
4 # Leer datos
5 comunas <- st_read("santiago_comunas.shp")
6
7 # Pipe operator para flujo de trabajo
8 comunas %>%
9   filter(poblacion > 100000) %>%
10   tm_shape() +
11   tm_polygons("poblacion", palette = "Blues")

```

6.4 Python vs R: La Verdad [1:07-1:11]

DECIR

La pregunta del millón: ¿Python o R? Mi respuesta: DEPENDE.

Usa Python cuando:

- Integración con aplicaciones web
- Deep Learning (TensorFlow, PyTorch)
- Pipelines de producción
- Trabajas con ingenieros

Usa R cuando:

- Análisis estadístico espacial complejo
- Visualización para publicación académica
- Investigación
- Trabajas con estadísticos

NOTA

Secreto profesional: Los expertos usan ambos. Python para producción, R para exploración.

7 Sección 4: Ejemplos Prácticos [15 minutos]

7.1 Flujo de Trabajo Típico [1:11-1:14]

HACER

Señalar el diagrama de flujo en la diapositiva

Los 6 pasos del flujo:

1. **Obtención de datos:** APIs, web scraping, archivos
2. **Limpieza:** 80 % del tiempo se va aquí
3. **Análisis espacial:** La parte divertida
4. **Visualización:** Un mapa vale más que mil tablas
5. **Modelado:** Predicción y optimización
6. **Compartir:** GitHub, web apps, reportes

7.2 Caso COVID-19 en Santiago [1:14-1:19]

DECIR

Marzo 2020. Primer caso COVID en Chile. Necesitamos entender la propagación. Este es un caso real donde la geocomputación salvó vidas.

Pasos del análisis:

1. Join espacial de datos MINSAL con comunas
2. Normalización: casos por 100,000 habitantes
3. Autocorrelación espacial: Moran's I = 0.7
4. Identificación de clusters: Las Condes, Vitacura, Providencia
5. Predicción con modelo SIR espacial

Resultado: 8 de 10 comunas predichas correctamente para siguientes brotes.

7.3 Configuración del Ambiente [1:19-1:22]

HACER

Demostración en vivo de configuración

```

1 # Verificar Python
2 python --version # Debe ser 3.8+
3
4 # Crear ambiente virtual
5 python -m venv geo_env
6 source geo_env/bin/activate # Linux/Mac
7 # o
8 geo_env\Scripts\activate # Windows
9
10 # Instalar bibliotecas
11 pip install geopandas folium matplotlib jupyter

```

ALERTA

Windows puede dar problemas con Fiona. Solución: usar Anaconda

7.4 Primer Script Completo [1:22-1:26]**DECIR**

Este script es su plantilla base para todo el semestre. Guárdenlo.

```
1 import geopandas as gpd
2 import matplotlib.pyplot as plt
3
4 # Cargar datos del mundo
5 world = gpd.read_file(
6     gpd.datasets.get_path('naturalearth_lowres'))
7
8 # Filtrar Sudamérica
9 sudamerica = world[world['continent'] == 'South America']
10
11 # Crear visualización
12 fig, ax = plt.subplots(figsize=(10, 8))
13 sudamerica.plot(column='pop_est',
14                 ax=ax,
15                 legend=True,
16                 cmap='YlOrRd')
17 ax.set_title('Población de Sudamérica')
18 plt.show()
```


8 Cierre y Preparación para Laboratorio [10 minutos]

8.1 Objetivos del Laboratorio [1:26-1:29]

DECIR

En 10 minutos comenzamos el laboratorio. Será hands-on, práctico, posiblemente frustrante. Es normal.

Qué haremos:

1. Instalación completa de software
2. Verificación con script de prueba
3. Generar primer mapa de Chile
4. Configurar Git para entregas

Problemas anticipados:

- Windows: Permisos de administrador
- Mac: Xcode Command Line Tools (2GB)
- Linux: Generalmente sin problemas

8.2 Recursos para Profundizar [1:29-1:32]

Libros esenciales (gratuitos):

- Geocomputation with Python - py.geocompx.org
- Geocomputation with R - r.geocompx.org
- Geographic Data Science - geographicdata.science/book

Comunidades:

- Stack Overflow tag 'gis'
- r/gis en Reddit
- GitHub: awesome-gis

8.3 Ideas para Proyecto [1:32-1:34]

HACER

Actividad rápida: 30 segundos cada estudiante con su compañero para compartir una idea de proyecto

Ideas comerciales:

- Optimización de rutas delivery
- Análisis inmobiliario
- Segmentación de clientes

Ideas científicas:

- Contaminación del aire
- Segregación urbana
- Riesgo de incendios

8.4 Tareas y Cierre [1:34-1:40]**Tareas (no negociables):**

1. URGENTE: Instalar software ANTES del lab
2. Leer Capítulo 1 de Geocomputation with Python (30 min)
3. Pensar en un problema espacial de interés

DECIR

¿Preguntas finales?

HACER

Responder 2-3 preguntas máximo

DECIR

En 4 meses estarán haciendo análisis que hoy parecen magia. Confíen en el proceso. 10 minutos de break. Nos vemos aquí para el laboratorio.

9 Anexos

9.1 Preguntas Frecuentes

1. **¿Python o R primero?**
R: Si vienen de programación, Python. Si no programan, pueden empezar con cualquiera.
2. **¿Necesito GPU?**
R: No para el curso. GPU ayuda solo para deep learning avanzado.
3. **¿Por qué no ArcGIS?**
R: Costo, filosofía de enseñanza, y empleabilidad. El que programa GIS aprende ArcGIS rápido.
4. **¿Datos de Chile?**
R: IDE Chile tiene mucho. INE tiene shapefiles buenos. Les compartiré repositorio curado.
5. **¿Proyecto sobre mi comuna?**
R: ¡Absolutamente! Incentivo proyectos locales con contexto conocido.
6. **¿ChatGPT para código?**
R: Sí, pero para aprender, no copiar. Deben entender cada línea.
7. **¿Empleabilidad?**
R: Alta demanda. Junior: 1.2-1.5M CLP. Con experiencia: 2-3M CLP.
8. **¿PostGIS es necesario?**
R: Opcional. Muy poderoso pero requiere SQL. Puedo dar taller extra si hay interés.
9. **¿Imágenes satelitales?**
R: Sí, unidad 2. Sentinel-2 (10m, gratis) y Landsat (30m, histórico).
10. **¿Cuánto programación previa?**
R: Básico: variables, loops, funciones. Si pueden hacer FizzBuzz, están listos.

9.2 Actividades Interactivas

Actividad 1: Encuentra el Patrón

Tiempo: 2 minutos
Cuándo: Al mostrar mapa de John Snow
Qué hacer: Mostrar puntos, preguntar por patrón
Resultado esperado: Identifican clustering

Actividad 2: QGIS vs ArcGIS

Tiempo: 3 minutos
Cuándo: Sección de software
Qué hacer: Parejas debaten pros/contras
Resultado esperado: Comprenden trade-offs

Actividad 3: Idea de Proyecto**Tiempo:** 1 minuto**Cuándo:** Antes del cierre**Qué hacer:** Compartir idea con compañero**Resultado esperado:** Ideas fluyendo**9.3 Troubleshooting Común****Problema:** ImportError en Python**Solución:**

```

1 # Windows con Anaconda
2 conda install -c conda-forge geopandas
3
4 # Mac/Linux
5 pip install --upgrade pip
6 pip install geopandas

```

Problema: Internet caído**Solución:** Tener demos pre-grabados, screenshots, código en USB**Problema:** Proyector falla**Solución:** Compartir pantalla por Teams/Zoom, usar pizarra más**9.4 Checklist Post-Clase**

- ☐ Subir slides al repositorio
- ☐ Compartir links de recursos
- ☐ Preparar datos para laboratorio
- ☐ Responder emails (24h máximo)
- ☐ Anotar qué mejorar
- ☐ Backup de demos utilizados

9.5 Métricas de Éxito

Indicador	Meta
Software instalado al final del lab	¡90 %
Preguntas durante clase	¡5
Participación en actividades	¡80 %
Satisfacción (si hay encuesta)	¡4/5

¡Éxito en tu clase!

Para consultas: francisco.parra.o@usach.cl