

PF-3311 Temas especiales de ingeniería y sistemas de información: Ciencia de datos geoespaciales

Profesor: Manuel Vargas Del Valle

Modalidad: Alto virtual

Horario: M 17:00 a 20:50. Aula: Por definir. Créditos: 4

Horario de atención al estudiantado: A convenir

Correo electrónico institucional: manuel.vargas_d@ucr.ac.cr

II ciclo lectivo 2025

PROGRAMA DEL CURSO

1. DESCRIPCIÓN

Este curso trata sobre el manejo, visualización y análisis de datos geoespaciales mediante herramientas computacionales como sistemas de información geográfica y lenguajes de programación. Se introducen conceptos básicos de cartografía y se presenta una visión general de cómo metodologías y técnicas de ciencia de datos y aprendizaje automatizado pueden ser aplicadas al componente geoespacial de diversos problemas. Estas técnicas y metodologías se implementan principalmente mediante el lenguaje de programación Python.

El enfoque del curso es teórico-práctico, con lecciones teóricas combinadas con sesiones prácticas de programación en las cuales los estudiantes aplican en diversos escenarios de procesamiento de datos los conocimientos y habilidades aprendidos.

Los contenidos del curso y los recursos relacionados se comparten en el sitio web <https://pf3311-cienciadatosgeoespaciales.github.io/2025-ii/>, así como en la plataforma Mediación Virtual de la Universidad de Costa Rica.

2. OBJETIVOS

Al finalizar el curso, el estudiantado será capaz de:

- Leer mapas y comprender conceptos cartográficos básicos como sistemas de coordenadas, escala y resolución, entre otros.



- Utilizar sistemas de información geográfica para importar, visualizar y analizar datos geoespaciales.
- Importar, manipular y analizar datos geoespaciales mediante el lenguaje de programación Python.
- Desarrollar aplicaciones interactivas para el procesamiento, visualización y análisis de datos geoespaciales.
- Aplicar metodologías y técnicas de ciencia de datos y aprendizaje automatizado en el componente geoespacial de problemáticas relacionadas con población, clima, biodiversidad y transporte, entre otras áreas.

3. CONTENIDO DEL CURSO

SEMANA	CONTENIDO
I Introducción a la ciencia de datos geoespaciales	
1	<ul style="list-style-type: none">• El componente espacial de los datos:<ul style="list-style-type: none">◦ Observaciones y variables espaciales.◦ El proceso de ciencia de datos espaciales.• Fundamentos de cartografía y geodesia:<ul style="list-style-type: none">◦ Mapas.◦ Sistemas de coordenadas.• Modelos de datos espaciales:<ul style="list-style-type: none">◦ Vectorial.◦ Raster.• Sistemas de información geográfica – el sistema QGIS.
2	<ul style="list-style-type: none">• Fuentes de datos:<ul style="list-style-type: none">◦ Infraestructuras de datos espaciales (ej. SNIT).◦ Imágenes de satélite (ej. Landsat, Sentinel, MODIS).• Formatos de archivos, bases de datos y geoservicios:<ul style="list-style-type: none">◦ Formatos tradicionales: SHP, GPKG, GeoTIFF, PostgreSQL/PostGIS, SpatiaLite, WMS, WFS.◦ Formatos nuevos: GeoArrow, GeoParquet, DuckDB.• Geoprocesamiento:<ul style="list-style-type: none">◦ Operaciones con datos vectoriales.◦ Operaciones con datos raster.
3	<ul style="list-style-type: none">• Programas utilitarios de línea de comandos – la biblioteca GDAL.
4	<ul style="list-style-type: none">• Herramientas para investigación reproducible y desarrollo colaborativo:<ul style="list-style-type: none">◦ Markdown.◦ Git.

II El lenguaje de programación Python

5	<ul style="list-style-type: none"> • Introducción al lenguaje. • Herramientas de desarrollo: <ul style="list-style-type: none"> ◦ Ambientes de desarrollo locales - VS Code. ◦ Ambientes de desarrollo en la nube - Google Colab. • Principios de diseño y convenciones: <ul style="list-style-type: none"> ◦ El Zen de Python. ◦ Guías de estilo. • Sintaxis básica: <ul style="list-style-type: none"> ◦ Comentarios. ◦ Bloques. • Variables y tipos de datos. • Estructuras de control: <ul style="list-style-type: none"> ◦ Condicionales. ◦ Ciclos. • Funciones y modularidad. • Manejo de errores y excepciones.
6	<ul style="list-style-type: none"> • Programación orientada a objetos. • Manejo de archivos. • Interfaces de programación de aplicaciones (API).

III Análisis estadístico y visualización de datos mediante Python

7	<ul style="list-style-type: none"> • Análisis estadístico: <ul style="list-style-type: none"> ◦ Álgebra lineal - la biblioteca Numpy. ◦ Manipulación de datos - la biblioteca Pandas: <ul style="list-style-type: none"> ▪ Importación. ▪ Filtrado. ▪ Ordenamiento. ▪ Agrupación.
8	<ul style="list-style-type: none"> • Visualización de datos: <ul style="list-style-type: none"> ◦ Gráfica estadística - la biblioteca Matplotlib: <ul style="list-style-type: none"> ▪ Gráficos de dispersión. ▪ Gráficos de líneas. ▪ Histogramas. ▪ Gráficos de barras. ▪ Otros tipos de gráficos. ◦ Gráfica estadística interactiva - la biblioteca Plotly: <ul style="list-style-type: none"> ▪ Gráficos estadísticos.



IV Procesamiento de datos geoespaciales mediante Python	
9	<ul style="list-style-type: none">• Manipulación y análisis de datos vectoriales - la biblioteca Geopandas:<ul style="list-style-type: none">◦ Tipos de datos.◦ Operaciones.
10	<ul style="list-style-type: none">• Ejemplos de manipulación y análisis de datos vectoriales.
11	<ul style="list-style-type: none">• Manipulación y análisis de datos raster - la biblioteca Rasterio:<ul style="list-style-type: none">◦ Tipos de datos.◦ Operaciones.
12	<ul style="list-style-type: none">• Ejemplos de manipulación y análisis de datos raster.
13	<ul style="list-style-type: none">• Visualización de datos geoespaciales:<ul style="list-style-type: none">◦ Mapas interactivos - las bibliotecas Folium y Leafmap.◦ Mapeo de conjuntos de datos grandes - la biblioteca Lonboard.
V Visualización y análisis avanzados	
14	<ul style="list-style-type: none">• Desarrollo de aplicaciones web de ciencia de datos y aprendizaje automatizado - el marco de trabajo Streamlit.
15	<ul style="list-style-type: none">• Desarrollo de aplicaciones web de ciencia de datos y aprendizaje automatizado - el marco de trabajo Streamlit (continuación).
16	<ul style="list-style-type: none">• Publicación de aplicaciones web de ciencia de datos y aprendizaje automatizado - la plataforma Streamlit Cloud.

4. METODOLOGÍA

El curso se desarrolla mediante clases teórico-prácticas. Los conceptos teóricos son explicados por el profesor del curso durante las lecciones y también a través de lecturas previamente asignadas. Las sesiones prácticas se destinan a la realización de diferentes ejercicios de programación por parte de los estudiantes.

Los contenidos de las lecciones están disponibles en la plataforma Mediación Virtual y también en el sitio web del curso <https://pf3311-cienciadatosgeoespaciales.github.io/2025-ii/>, en el que hay enlaces a la bibliografía y a otros recursos de aprendizaje como tutoriales y videos.

Se recomienda a los estudiantes probar las diferentes herramientas y conceptos fuera del tiempo de clase y aprovechar las lecciones y las horas de consulta para aclarar dudas.



5. EVALUACIÓN

La evaluación incluye los siguientes componentes:

- **Exámenes cortos (30%).** Tienen como propósito principal evaluar las lecturas y los conceptos teóricos cubiertos en clase.
- **Tareas programadas (70%).** Consisten en ejercicios de programación que deben ser resueltos por los estudiantes fuera del tiempo de clase.

6. BIBLIOGRAFÍA

Abba, I. V. (2021). *Git and GitHub Tutorial - Version Control for Beginners*. FreeCodeCamp.Org. <https://www.freecodecamp.org/news/git-and-github-for-beginners/>

Çetinkaya-Rundel, Mine, & Hardin, Johanna (2021). *Introduction to Modern Statistics* (1st ed.). OpenIntro, Inc. <https://openintro-ims.netlify.app/>

Development Seed. (s.f.). *lonboard*. Recuperado 1 de agosto de 2024, de <https://developmentseed.org/lonboard/>

Downey, Allen B. (2024). *Think Python: How to Think Like a Computer Scientist* (3rd ed.). O'Reilly Media. <https://greenteapress.com/wp/think-python-3rd-edition/>

Geopandas contributors. (s.f.). *geopandas: Geographic pandas extensions*. Recuperado 1 de enero de 2022, de <http://geopandas.org>

Gillies, S. (s.f.). *rasterio: Fast and direct raster I/O for use with Numpy and SciPy*. Recuperado 1 de enero de 2022, de <https://github.com/mapbox/rasterio>

Kaggle. (s.f.-a). *Learn Pandas*. Recuperado 1 de agosto de 2024, de <https://www.kaggle.com/learn/pandas>

Kaggle. (s.f.-b). *Learn Data Visualization*. Recuperado 1 de agosto de 2024, de <https://www.kaggle.com/learn/data-visualization>



Kaggle. (s.f.-c). *Learn GeospatialAnalysis*. Recuperado 1 de agosto de 2024, de <https://www.kaggle.com/learn/geospatial-analysis>

Markdown Tutorial. (s.f.). Recuperado 19 de marzo de 2022, de <https://www.markdowntutorial.com/>

P, C. (s.f.). *plotly: An open-source, interactive data visualization library for Python*. Recuperado 1 de agosto de 2024, de <https://plotly.com/python/>

Rey, S. J., Arribas-Bel, D., & Wolf, L. J. (2020). *Geographic Data Science with Python*. <https://geographicdata.science/book/>

Richards, T. (2021). *Getting started with Streamlit for data science: Create and deploy Streamlit web applications from scratch in Python*. Packt Publishing. <https://www.packtpub.com/product/getting-started-with-streamlit-for-data-science/9781800565500>

Severance, D. C. R. (2016). *Python for Everybody: Exploring Data in Python 3* (S. Blumenberg & E. Hauser, Eds.). CreateSpace Independent Publishing Platform. <https://www.py4e.com/html3/>

The Pandas Development Team. (s.f.). *pandas: Powerful data structures for data analysis, time series, and statistics*. Recuperado 1 de enero de 2022, de <https://pandas.pydata.org>

Story, R. (s.f.). *folium: Make beautiful maps with Leaflet.js & Python*. Recuperado 1 de enero de 2022, de <https://github.com/python-visualization/folium>

Wickham, Hadley; Çetinkaya-Rundel, Mirne; & Grolemund, Garret (2023). *R for Data Science: Import, Tidy, Transform, Visualize, and Model Data* (2nd ed.). O'Reilly Media. <https://r4ds.hadley.nz/>



PF-3312 Laboratorio de Temas especiales de ingeniería y sistemas de información: Ciencia de datos geoespaciales

Profesor: Manuel Vargas Del Valle

Modalidad: Alto virtual

Horario: A convenir. Aula: Por definir. Créditos: 2

Horario de atención al estudiantado: A convenir

Correo electrónico institucional: manuel.vargas_d@ucr.ac.cr

II ciclo lectivo 2025

PROGRAMA DEL CURSO

1. DESCRIPCIÓN

Este laboratorio es complementario del curso “Temas especiales de ingeniería y sistemas de información: Ciencia de datos geoespaciales” y le permite al estudiantado poner en un nivel mayor de práctica los conocimientos y habilidades adquiridos en ese curso.

2. OBJETIVOS

Objetivo general

Diseñar, implementar, desplegar y documentar una aplicación web geoespacial interactiva que incorpore la ingesta, el análisis y la visualización de datos vectoriales y raster, así como la publicación en la nube.

Objetivos específicos

- Formular un caso de uso geoespacial (problema, datos, entregables, etc.) y un plan de acción.
- Desarrollar un flujo de trabajo reproducible en Python que incluya adquisición de datos, limpieza, geoprocésamiento y generación de mapas interactivos.
- Construir la interfaz empleando componentes visuales avanzados y buenas prácticas de experiencia de usuario.
- Automatizar la integración y el despliegue continuo con control de versiones en GitHub.
- Documentar y comunicar los resultados.



3. METODOLOGÍA

Este curso proporciona un espacio para que el estudiantado explore y ejecute las actividades propias del proceso de desarrollo de una aplicación web geoespacial. El curso se enfocará en tres pilares fundamentales: el correcto planteamiento de una propuesta, el desarrollo de la aplicación y la presentación de resultados. Conforme se cubren los contenidos del curso teórico, los estudiantes deben avanzar en el desarrollo de la aplicación.

Cada grupo de trabajo debe presentar una propuesta que demuestre la viabilidad del proyecto y sirva como guía para todo el ciclo de desarrollo. En ella deben quedar claros el problema geoespacial por resolver, las necesidades de los usuarios, las fuentes y el formato de los datos, la arquitectura de la solución y un cronograma realista con hitos medibles.

El desarrollo incremental se evalúa a lo largo de todo el proceso mediante la revisión de *commits*, ramas y *pull requests* en GitHub (u otra plataforma similar para compartir código), y la ejecución automática de pruebas en cada integración.

La aplicación final desplegada debe estar accesible públicamente, funcionar sin errores y reproducir todo el flujo de datos desde la ingesta hasta la visualización interactiva. Se revisará la estabilidad del despliegue en la nube, el rendimiento frente a diferentes tamaños de datos, la usabilidad de la interfaz, la correcta gestión de dependencias y secretos, y la trazabilidad que permita reconstruir el entorno con instrucciones mínimas.

Por último, la presentación de resultados combina una demostración en vivo de la aplicación y una breve exposición de las lecciones aprendidas. Durante la demo se espera que los equipos expliquen el caso de uso, justifiquen las decisiones técnicas, muestren los principales flujos de la interfaz e ilustren cómo los usuarios finales pueden beneficiarse de la herramienta.

4. EVALUACIÓN

La evaluación incluye los siguientes componentes:



- **Propuesta (20%).** Alcance, datos, criterios de éxito, diagrama de arquitectura y cronograma.
- **Desarrollo incremental (30%).** Commits documentados, pruebas automatizadas y cumplimiento de hitos especificados en el cronograma de la propuesta.
- **Aplicación final desplegada (30%).** Aplicación operativa en una plataforma en la nube (ej. Streamlit Cloud).
- **Presentación de resultados (20%).** Exposición oral.