

# Programación en TIG (GIT Programming)

## Máster en TIG

**Curso Académico 2024/2025**

**1º curso– 1º cuatrimestre**

## GUÍA DOCENTE

Nombre de la asignatura:	<b>Programación en TIG</b>
Código:	<b>202169</b>
Titulación en la que se imparte:	<b>Máster en Tecnologías de la Información Geográfica</b>
Departamento y Área de Conocimiento:	<b>Automática. Arquitectura y Tecnología de Computadores</b>
Carácter:	<b>Obligatoria</b>
Créditos ECTS:	<b>6</b>
Curso y cuatrimestre:	<b>1º Curso-1º Cuatrimestre</b>
Profesorado:	Julia María Clemente Párraga (julia.clemente@uah.es)
Horario de Tutoría:	A fijar en función del horario de clase
Idioma en el que se imparte:	Español/ English Friendly

### 1.a PRESENTACIÓN

La asignatura de Programación en Tecnologías de Información Geográfica (TIG) pretende introducir al alumno en las características, conceptos y componentes esenciales de la programación informática. La meta es que el alumno adquiera habilidades básicas para diseñar e implementar soluciones software utilizando como soporte práctico los elementos y estructuras de datos proporcionados por un lenguaje de programación de alto nivel y amplia difusión como es Python y, a la vez, aprenda a utilizar los entornos de programación. Con estas herramientas se aborda la resolución de problemas territoriales en aplicaciones de *scripting* que permitan automatizar tareas de geoprocésamiento. Para lograr este fin, el alumno realizará ejercicios y proyectos de programación de tareas TIG adquiriendo experiencia y dominio en la programación con Python para software de Sistemas de Información Geográfica (SIG) actualmente usados (ArcGIS, QGIS, etc.), así como en el uso de librerías o bibliotecas tanto estándares como externas para propósitos más específicos.

### 1.b PRESENTATION (en inglés)

Programming in Geographic Information Technologies (GIT) aims to introduce the student to the characteristics, concepts and essential components of computer programming. The goal is for the student to acquire basic skills to design and implement software solutions using as a practical support the elements and data

structures provided by a high-level and widely used programming language such as Python and, at the same time, to learn how to use programming environments. These tools are used to solve territorial problems in scripting applications that allow the automation of geoprocessing tasks. To achieve this goal, the student will perform exercises and projects of programming TIG tasks while acquiring experience and mastery in programming with Python for Geographic Information Systems (GIS) software currently used (ArcGIS, QGIS, etc.), as well as in the use of libraries or libraries both standard and external for more specific purposes.

## 2. COMPETENCIAS y RESULTADOS DE APRENDIZAJE

- Competencias básicas y generales:

CG1 - Comprender los problemas territoriales que pueden ser estudiados con las Tecnologías de la Información Geográfica.

CG2 - Aplicar correctamente las funciones de análisis y representación de la información geográfica para solucionar problemas territoriales de distinta naturaleza.

CG3 - Combinar conocimientos y destrezas propios de las TIG para avanzar soluciones a problemas territoriales aún no resueltos.

CG4 - Evaluar y comunicar adecuadamente las soluciones basadas en las TIG a los problemas territoriales.

CB6 – Poseer y comprender conocimientos que aporten una base u oportunidad de ser originales en el desarrollo y/o aplicación de ideas, a menudo en un contexto de investigación.

CB7 - Que los estudiantes sepan aplicar los conocimientos adquiridos y su capacidad de resolución de problemas en entornos nuevos o poco conocidos dentro de contextos más amplios (o multidisciplinares) relacionados con su área de estudio.

CB8 - Que los estudiantes sean capaces de integrar conocimientos y enfrentarse a la complejidad de formular juicios a partir de una información que, siendo incompleta o limitada, incluya reflexiones sobre las responsabilidades sociales y éticas vinculadas a la aplicación de sus conocimientos y juicios.

CB9 – Que los estudiantes sepan comunicar sus conclusiones y los conocimientos y razones últimas que las sustentan a públicos especializados y no especializados de un modo claro y sin ambigüedades.

CB10 - Que los estudiantes posean las habilidades de aprendizaje que les permitan continuar estudiando de un modo que habrá de ser en gran medida autodirigido o autónomo.

- Competencias específicas:

CE16 - Aplicar los principales conceptos, tipos de datos y variables, tipos de operadores, componentes y estructuras de programación en la escritura de macros y *scripts* operativos en diferentes programas del ámbito de las TIG.

CE17 - Interpretar el código de lenguajes de programación de uso habitual en el ámbito de las TIG.

CE19 - Desarrollar y documentar procedimientos de captación, análisis o publicación de información territorial de las TIG mediante lenguajes de programación y procedimientos de validación adecuados.

### **Resultados del aprendizaje:**

1. Analizar los componentes y requerimientos de un procedimiento de análisis en TIG susceptible de automatización o mejora.
  - a. Comprender los requisitos funcionales, no funcionales, de datos, de Interfaz de usuario, de integración de los diferentes componentes, rendimiento, etc., de los componentes.
  - b. Documentar apropiadamente los diferentes requerimientos asociados a los componentes de un procedimiento.
2. Aplicar los principales conceptos y estructuras de programación en la escritura de macros y scripts operativos en diferentes programas del ámbito de las TIG.
  - a. Conocer los conceptos y estructuras fundamentales aplicables a todos los lenguajes de programación y los específicos del lenguaje a aplicar.
  - b. Comprender cómo organizar y estructurar el código de diferentes formas: funciones, módulos o paquetes, definición de objetos, etc. para su fácil reutilización y mantenimiento.
  - c. Aplicar la sintaxis y semántica de lenguajes de uso común en el ámbito TIG a problemas afines.
3. Entender el código de lenguajes de programación de uso habitual en el ámbito de las TIG.
  - a. Interpretar el código de terceros.
  - b. Explorar las diferentes bibliotecas idóneas para resolver un determinado problema en el ámbito TIG o para mejorar un procedimiento ya existente.
  - c. Comprender la diferente documentación de una determinada biblioteca o módulo del lenguaje de programación a utilizar.
4. Crear o adaptar nuevos procedimientos capaces de enriquecer la potencialidad actual de captación, análisis o difusión de información territorial de las TIG

utilizando los lenguajes de programación y procedimientos de validación adecuados.

- a. Conocer entornos de desarrollo y entornos de aprendizaje de desarrollo integrados (IDEs e IDLEs, respectivamente) para facilitar la creación de programas/scripts en el ámbito de las TIG.
  - b. Aplicar las diferentes herramientas de validación adecuadas para evaluar correctamente una aplicación.
5. Documentar, presentar y justificar adecuadamente las líneas de código y los módulos escritos de manera que puedan ser reutilizados por otros usuarios.
- a. Conocer las buenas prácticas de programación generales y específicas del lenguaje de programación a usar para el desarrollo de scripts en el ámbito GIS.
  - b. Representar un algoritmo independientemente del lenguaje de programación a utilizar.
  - c. Diseñar los componentes de una aplicación.

### 3. CONTENIDOS

Bloques de contenido	Total de clases, créditos u horas
<b>Tema 1. Introducción al entorno de trabajo</b>	4 horas
<b>Tema 2. Introducción a Python</b> 2.1. ¿Qué es Python? ¿Por qué Python? 2.2. Características 2.3. Conceptos básicos	2 horas
<b>Tema 3. Flujo de control</b> 3.1. Condiciones y bucles 3.2. Funciones 3.3. Diseño de algoritmos y buenas prácticas de codificación	4 horas

<p><b>Tema 4. Estructuras de datos</b></p> <p>4.1. Introducción.</p> <p>4.2. Estructuras de datos y Tipos Abstractos de datos (TADs).</p> <p>4.3. Secuencias: Listas, tuplas, cadenas, rangos, etc.</p> <p>4.4. Conjuntos</p> <p>4.5. Diccionarios</p> <p>4.6. Técnicas de <i>looping</i></p>	<p>4 horas</p>
<p><b>Tema 5. Entrada/Salida</b></p> <p>5.1. Introducción</p> <p>5.2. Formateo</p> <p>5.3. Manejo de archivos</p> <p>5.4. Serialización en Python</p>	<p>4 horas</p>
<p><b>Tema 6. Módulos y paquetes</b></p> <p>6.1. Introducción</p> <p>6.2. Módulos</p> <p>6.3. Paquetes</p> <p>6.4. Biblioteca estándar de Python</p> <p>6.5. Paquetes de ciencia de datos: <i>numpy</i>, <i>pandas</i>, <i>matplotlib</i> y <i>seaborn</i></p>	<p>6 horas</p>
<p><b>Tema 7. Programación Orientada a Objetos en Python</b></p> <p>7.1. Paradigmas de programación</p> <p>7.2. Conceptos básicos de Programación Orientado a Objetos (POO)</p> <p>7.3. POO en Python</p>	<p>4 horas</p>
<p><b>Tema 8. Errores y manejo de excepciones</b></p> <p>7.4. Tipos de errores</p> <p>7.5. Qué es una excepción</p> <p>7.6. Manejo de excepciones</p> <p>7.7. Excepciones en Python</p>	<p>2 horas</p>

<b>Tema 9. Exploración de datos GIS</b>  7.8. Introducción. 7.9. Paquete ArcPy 7.10. Descripción y listado de datos GIS 7.11. Manejo de cursores	6 horas
<b>Tema 10. Lectura, escritura y manejo de datos geoespaciales</b>  7.12. OGR: generalidades 7.13. Cómo trabajar con datos vectoriales 7.14. Leer y escribir datos vectoriales 7.15. Manejar geometrías y proyecciones 7.16. Filtros 7.17. GDAL: generalidades 7.18. Cómo trabajar con datos ráster 7.19. Obtener información del ráster 7.20. Leer y escribir datos raster 7.21. Análisis geoespacial con <i>geopandas</i>	9 horas
<b>Pruebas de evaluación</b>	3 horas

#### 4. METODOLOGÍAS DE ENSEÑANZA-APRENDIZAJE- ACTIVIDADES FORMATIVAS

##### 4.1. Distribución de créditos (especificar en horas)

Actividades formativas	Horas
Presencialidad o interactividad síncrona (clases teóricas y clases prácticas y evaluación formativa)	48
Trabajo autónomo del estudiante	102
<b>Total horas</b>	<b>150</b>

## 4.2. Metodologías, materiales y recursos didácticos

Metodologías	Materiales y recursos didácticos
Expositiva y aula invertida	Materiales didácticos elaborados por el/la profesor@: presentaciones de clase, <i>notebooks</i> para entrenar en el aprendizaje del lenguaje de programación y potenciar la capacidad algorítmica del alumno así como otros a resolver por el alumno tras el estudio de materiales ( <i>flipped classroom</i> ).
Uso de herramientas software	Uso de plataformas como Anaconda y su herramienta Jupyter Notebook, IDEs como PyCharm, GitHub, numerosos módulos y paquetes en lenguaje python de amplia y diversa utilidad en el ámbito TIG, para facilitar diferentes tipos de aprendizaje: interactivo, individualizado, colaborativo, etc.
Basada en problemas	Elaboración de prácticas enfocadas a que el estudiante, mediante la ejercitación de rutinas, aplicación de algoritmos, procedimientos de transformación de la información disponible y la interpretación de resultados, desarrolle las competencias de la asignatura proponiendo soluciones adecuadas a problemas territoriales y aplicando las habilidades y conocimientos adquiridos.
Basada en proyectos	Se propone a los estudiantes seleccionar un proyecto de entre un conjunto de ellos en el ámbito TIG. El alumno debe plantear el objetivo claramente, diseñar, codificar en Python la solución al problema, así como su validación. Adicionalmente, la presentación del proyecto y discusión final frente al resto de estudiantes contribuye a fomentar el trabajo en equipo, habilidades de comunicación, análisis crítico, pero a la vez también autonomía.
Uso de aula virtual	Se utiliza un espacio virtual para la asignatura en la plataforma e-learning Blackboard donde están accesibles a los estudiantes todos los materiales didácticos utilizados en las sesiones presenciales, incluyendo las actividades y proyectos propuestos al



estudiante. En este último caso, se usa como recurso una wiki donde el alumno puede seleccionar una de las múltiples propuestas planteadas como proyecto.

Adicionalmente, se utiliza como material didáctico soporte de todas las metodologías aplicadas las siguientes:

- Referencias bibliográficas. Pueden consultarse en Bibliotecas (Edificio Politécnico, CRAI, Facultad de Medicina y Ciencias de la Salud, etc).
- Otros recursos web. Sendos enlaces y vídeos de terceros como apoyo en el aprendizaje.

## 5. EVALUACIÓN: Procedimientos, criterios de evaluación y de calificación

### Evaluación continua:

Todo el proceso de evaluación estará inspirado en la evaluación continua del estudiante, de tal forma que se garantice la adquisición tanto de los contenidos como de las competencias de la asignatura. La evaluación se adecua a los establecido en la normativa de evaluación de los aprendizajes de la UAH<sup>1</sup>

- **Convocatoria ordinaria**

Los criterios de evaluación deben atender al grado de adquisición de las competencias por parte del estudiante. Para ello, se definen los siguientes criterios:

**CE1:** El alumnado es capaz de conocer los fundamentos, componentes básicos y técnicas para elaborar programas informáticos.

**CE2:** El alumnado es capaz de interpretar el código de lenguajes de programación de uso habitual en el ámbito de las TIG.

**CE3:** El alumnado muestra capacidad e iniciativa a la hora de desarrollar *scripts* o aplicaciones utilizando los elementos y estructuras de datos de un lenguaje de programación de amplia difusión para resolver tareas comunes en TIG.

---

<sup>1</sup> Normativa de evaluación de los aprendizajes de la UAH (30 septiembre de 2021).

<https://www.uah.es/export/sites/uah/es/conoce-la-uah/organizacion-y-gobierno/.galleries/Galeria-Secretaria-General/Normativa-Evaluacion-Aprendizajes.pdf>

**CE4:** El alumnado es capaz de cumplir con las tareas encomendadas, realiza sus propios aportes con rigor y justifica las soluciones a problemas planteados.

**CE5:** El alumnado muestra claridad expositiva y cuidado formal en la exposición de los contenidos relativos a la materia estudiada.

**CE6:** El alumnado muestra interés por los contenidos y la materia estudiada participando en los ejercicios relacionados con métodos *flipped classroom*.

**CE7:** El alumnado es capaz de profundizar en el análisis, atendiendo a matices y discriminación de las partes.

**CE8:** El alumnado demuestra utilizar métodos de comprobación y validación de la corrección y coherencia de los programas informáticos realizados.

Los instrumentos de evaluación a emplear son los siguientes:

- Pruebas de Evaluación Parcial (**PEP**):
  1. Pruebas individuales de resolución de problemas prácticos con el lenguaje de programación (25% de la nota final).
  2. Pruebas individuales de resolución de problemas prácticos con el lenguaje de programación y las herramientas informáticas utilizadas para resolver y automatizar tareas TIG (35% de la nota final).
- Trabajo Final de la Asignatura (**TFA**): El trabajo de la asignatura consiste en el desarrollo grupal de un prototipo software aplicado a tareas TIG utilizando el lenguaje y herramientas de programación manejadas durante el periodo docente. Se presentará un informe final y se realizará una presentación y defensa del trabajo en clase. Para su evaluación se utilizará como herramienta una rúbrica que permita además la autoevaluación y coevaluación de los alumnos. Las guías para el desarrollo de este trabajo así como la descripción de la rúbrica utilizada se proporcionarán a través de la página web de la asignatura (35% de la nota final).
- Participación y actitud (**PAC**):
  - Participación en clase u *online* (a través de la plataforma, virtual, etc.).
  - Notebooks de seguimiento de aula invertida.
  - Actitud proactiva en la resolución de problemas.  
(5% de la nota final)

La relación entre criterios, instrumentos y calificación se muestra a continuación:

Criterio de evaluación	Instrumentos de evaluación	%
CE1-CE4, CE6-CE8	PEP1-PEP2. Pruebas de resolución de problemas prácticos sobre <b>el lenguaje de programación Python (Temas 2-8)</b>	25%
	PEP3-PEP4. Pruebas de resolución de problemas prácticos sobre <b>Programación en Python para GIS (Temas 9-10)</b>	35%

Criterio de evaluación	Instrumentos de evaluación	%
CE1- CE8	TFA. Con uso de Rúbrica.	35%
CE6	PAC	5%

- **Convocatoria extraordinaria**

Los criterios de evaluación y calificación para la convocatoria extraordinaria serán exactamente los mismos que para la convocatoria ordinaria. El alumnado podrá entregar/realizar en esta convocatoria aquellas actividades de evaluación que considere oportunas; se mantendrá la calificación de la convocatoria ordinaria para el resto de las actividades no entregadas/realizadas.

### Evaluación final:

El estudiantado podrá acogerse a la evaluación final, sin perjuicio de que sus causas tengan que ser valoradas en cada caso concreto, la realización de prácticas presenciales, las obligaciones laborales, las obligaciones familiares, los motivos de salud y la discapacidad. El hecho de seguir los estudios a tiempo parcial no otorga por sí mismo el derecho a optar por la evaluación final.

Los estudiantes de Máster Universitario, para acogerse a la evaluación final, tendrán que solicitarlo por escrito al director del Máster en las dos primeras semanas de impartición de la asignatura, explicando las razones que le impiden seguir el sistema de evaluación continua. En el caso de aquellos estudiantes que por razones justificadas no tengan formalizada su matrícula en la fecha de inicio del curso o del periodo de impartición de la asignatura, el plazo indicado comenzará a computar desde su incorporación a la titulación. El director de Máster deberá valorar las circunstancias alegadas por el estudiante y tomar una decisión motivada. Transcurridos 15 días hábiles sin que el estudiante haya recibido respuesta expresa por escrito a su solicitud, se entenderá que ha sido estimada.

Los criterios de evaluación, instrumentos y calificación para la evaluación final serán los mismos que para la evaluación continua, en convocatoria ordinaria o extraordinaria, con la excepción de PAC, en la que se excluirá de su calificación la actividad relacionada con el seguimiento de aula invertida (*notebooks*), y TFA, que se realizará de forma individual.

La relación entre criterios, instrumentos y calificación se muestra a continuación:

Criterio de evaluación	Instrumentos de evaluación	%
CE1-CE4, CE6-CE8	PEP1-PEP2. Pruebas de resolución de problemas prácticos sobre <b>el lenguaje de programación Python (Temas 2-8)</b>	25%

Criterio de evaluación	Instrumentos de evaluación	%
	PE3-PEP4. Pruebas de resolución de problemas prácticos sobre <b>Programación en Python para GIS (Temas 9-10)</b>	35%
CE1- CE8	TFA1. Con uso de Rúbrica.	35%
CE6	PAC1	5%

Es requisito para la superación de la asignatura, tanto en la convocatoria ordinaria como extraordinaria, aprobar las prácticas (es decir, obtener 3/6 puntos en el total de calificación de las prácticas) y aprobar el trabajo final de la asignatura (es decir, obtener 1,75/3,5 puntos en esta actividad).

Como criterio general, aquellos alumnos que no entreguen todas las prácticas o no realicen el trabajo final se considerarán *No Presentados*.

## 6. BIBLIOGRAFÍA

### Bibliografía Básica

- Python Tutorial Release 3.12.2. Guido van Rossum and the Python development team, 2024.
- Learning Python: Powerful Object-Oriented Programming, 5th Edition. Mark Lutz. Ed. O'Reilly, 2013.
- Learning Geospatial Analysis with Python, 3th Edition. Joel Lawhead. Packt Publishing, 2019.
- Programming ArcGIS with Python Cookbook, Second Edition. Eric Pimpler. Packt Publishing, 2015.
- Advanced Python Scripting for ArcGIS Pro. Paul A. Zandbergen. Esri Press, 2020.
- Python For ArcGIS Pro. Python for ArcGIS Pro: Automate cartography and data analysis using ArcPy, ArcGIS API for Python, Notebooks, and pandas. Packt Publishing, 2022.
- Geoprocessing with python. Manning, 2016.
- Python for Geospatial Analysis: Theory, Tools, and Practice for Location Intelligence. O'Reilly, 2022.
- PyQGIS developer cookbook (QGIS 3.34).  
[https://docs.qgis.org/3.34/en/docs/pyqgis\\_developer\\_cookbook/index.html](https://docs.qgis.org/3.34/en/docs/pyqgis_developer_cookbook/index.html)

### Bibliografía Complementaria

- Python For ArcGIS, 1<sup>st</sup> ed. Laura Tateosian. Springer, 2015.
- Think Python. How to Think Like a Computer Scientist. Allen B. Green Tea Press, 2015.
- Dive into Python 3. Mark Pilgrim. Ed. Apress, 2009.
- Python Scripting for Computational Science. Hans Petter Langtangen, Springer Science & Business Media, 2008.
- Python 3. Object Oriented Programming, 3th Edition. Dusty Phillips. Packt Publishing, 2018.
- Programming for ArcGIS Pro, Second Edition. Packt Publishing, 2021.
- Python Scripting for ArcGIS Pro. Paul A. Zandbergen. Esri Press, 2020.

- [Mastering Geospatial Analysis with Python](#). Paul Crickard, Eric van Rees, Silas toms. Packt Publishing, 2018.
- [The PyQGIS Programmer's Guide. Extending QGIS 3 with Python 3](#). Gary Sherman. Locate Press, 2018.