



Universidad
de Alcalá

GUÍA DOCENTE

ASIGNATURA

PROGRAMACIÓN EN TIG

**Máster Universitario en
Tecnologías de la Información Geográfica**

Universidad de Alcalá

Curso Académico 2018/19

GUÍA DOCENTE

Nombre de la asignatura:	PROGRAMACIÓN EN TIG
Código:	202169
Titulación en la que se imparte:	Master Profesional en Tecnologías de la Información Geográfica
Departamento y Área de Conocimiento:	Automática. Arquitectura y Tecnología de Computadores
Carácter:	OBLIGATORIA
Créditos ECTS:	6
Curso y cuatrimestre:	Único. Primer cuatrimestre.
Profesorado:	Julia María Clemente Párraga (julia.clemente@uah.es)
Horario de Tutoría:	Lunes y Jueves de 17:30 a 19:30
Idioma en el que se imparte:	Español

1. PRESENTACIÓN

La asignatura de Programación en Tecnologías de Información Geográfica (TIG) tiene como uno de sus objetivos fundamentales 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 como es Python y aprenda a utilizar los entornos de programación.

Estrechamente relacionado con el objetivo previamente mencionado, otro de los objetivos principales es que los estudiantes aprendan a utilizar un lenguaje de amplia difusión como Python para resolver 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 a la vez 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 estándar como externas para propósitos más específicos.

2. COMPETENCIAS

Competencias genéricas:

CG-1. Comprender los problemas territoriales que pueden ser estudiados con las Tecnologías de la Información Geográfica.

CG-2. Aplicar correctamente las funciones de análisis y representación de la información geográfica para solucionar problemas territoriales de distinta naturaleza.

CG-3. Combinar conocimientos y destrezas propios de las TIG para avanzar soluciones a problemas territoriales aún no resueltos.

CG-4. Evaluar y comunicar adecuadamente las soluciones basadas en las TIG a los problemas territoriales.

Competencias específicas:

CE-16. 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.

CE-17. Interpretar el código de lenguajes de programación de uso habitual en el ámbito de las TIG.

CE-19. Desarrollar y documentar procedimientos de captación, análisis o publicación de información territorial de las TIG mediante lenguajes de programación y procedimientos de validación adecuados.

3. CONTENIDOS

Bloques de contenido (se pueden especificar los temas si se considera necesario)	Total de clases, créditos u horas
Bloque I: Introducción a la programación <ul style="list-style-type: none"> • Definición de Computador y Sistema Operativo. • Codificación de la información. • Lenguajes de programación. • Programas y algoritmos 	<ul style="list-style-type: none"> • 2 horas
Bloque II: El lenguaje de programación Python <ul style="list-style-type: none"> • Introducción a Python. • Conceptos básicos <ul style="list-style-type: none"> ○ Tipos de datos y operadores ○ Variables y asignaciones. • Flujo de control <ul style="list-style-type: none"> ○ Tipos de sentencias: condicionales e iterativas. ○ Funciones. ○ Implementación de algoritmos. Estilo de codificación. • Estructuras de datos. <ul style="list-style-type: none"> ○ Listas ○ Tuplas y secuencias ○ Conjuntos ○ Diccionarios. • Módulos y paquetes • Biblioteca Estándar de Python • Entrada/Salida <ul style="list-style-type: none"> ○ Formateo ○ Manejo de archivos ○ Serialización en Python (<i>pickles</i>) • Excepciones • Programación Orientada a objetos en Python <ul style="list-style-type: none"> ○ Paradigma orientado a objetos: Clases, objetos, propiedades y métodos. ○ Herencia, encapsulamiento y Polimorfismo. 	<ul style="list-style-type: none"> • 26 horas

Bloque III: Programación en Python para GIS

- Desarrollo de aplicaciones geoespaciales con Python.
 - Análisis y visualización de datos geoespaciales.
 - Tipos de formatos de datos geoespaciales.
 - Lectura y Escritura de datos geoespaciales con Python.
 - Librerías Python para el desarrollo geoespacial: GDAL y OGR.
 - Generación de mapas usando Python y Mapnik.

- 20 horas.

Cronograma (Optativo)

Semana / Sesión	Contenido
01	<ul style="list-style-type: none"> • Bloque 1: Teoría (2h) • Bloque 2: Teoría (1h) + Práctica (1h)
02^a-07^a	<ul style="list-style-type: none"> • Bloque 2: Teoría (9h) + Práctica (15h)
08^a-12^a	<ul style="list-style-type: none"> • Bloque 3: Teoría (8h) + Práctica (12h)

4. METODOLOGÍAS DE ENSEÑANZA-APRENDIZAJE.-ACTIVIDADES FORMATIVAS

4.1. Distribución de créditos (especificar en horas)

<p>Número de horas presenciales: 48</p>	<p>1. Lección magistral. Presentación en el aula de los conceptos propios de lenguajes de programación y su aplicación a TIG, haciendo uso de metodología expositiva y participativa, mediante la cual se suministra a los alumnos información esencial y organizada procedente de diversas fuentes con unos objetivos específicos predefinidos, con la utilización de medios audiovisuales y otros recursos didácticos.</p> <p>Su propósito es la transmisión de los contenidos a enseñar, motivando al alumnado a la reflexión, facilitándole el descubrimiento de las relaciones entre diversos conceptos y formarle una mentalidad crítica.</p>
<p>Número de horas del trabajo propio del estudiante: 102</p>	<p>2. Resolución de casos prácticos. Este método de enseñanza se utiliza como complemento de las clase de teoría. Se basa en el desarrollo de soluciones, ejercitación de rutinas y aplicación de algoritmos propios para la realización de proyectos de desarrollo software en el ámbito que se estudia; además de la aplicación de procedimientos de transformación de la información disponible y la interpretación de los resultados. La intención principal es la de aplicar lo ya aprendido para favorecer la comprensión, tanto de la importancia como del contenido de un nuevo tema, afianzar conocimientos y estrategias y su aplicación en las situaciones prácticas que se planteen.</p> <p>Evaluación de soluciones a los problemas planteados y pruebas funcionales mediante el desarrollo de proyectos.</p> <p>1. Actividades de autoestudio. Se trata de actividades que el alumno realizará de forma individual, bajo la supervisión o no del</p>

	<p>profesor, mediante tutorías presenciales o virtuales; de modo que el estudiante pueda avanzar en la adquisición de conocimientos y procedimientos de la materia.</p> <p>El propósito principal es desarrollar la capacidad de autorregular su aprendizaje planificándolo, diseñándolo y adecuándolo a sus condiciones particulares.</p> <p>2. Actividades de trabajo en equipo. Se trata de actividades, guiadas y no guiadas por el profesor, en las que un grupo de alumnos resuelve una determinada tarea o proyecto.</p> <p>El propósito principal es favorecer en los estudiantes la generación e intercambio de ideas, la identificación y análisis de diferentes puntos de vista, la generalización o transferencia de conocimiento y la valoración crítica del mismo.</p>
<p>Total horas: 150 horas</p>	<p>Se utiliza un modelo de formación semipresencial, con el soporte de la página web de la asignatura.</p>

4.2. Estrategias metodológicas, materiales y recursos didácticos

<p>Estrategias metodológicas</p>	<ol style="list-style-type: none"> 1. Exposición. Consiste en la transmisión de conocimientos, ofreciendo un enfoque crítico de la materia, que lleve a los alumnos a reflexionar y descubrir las relaciones entre los diversos conceptos para formar una mentalidad crítica en la forma de afrontar los problemas y aplicar una metodología, implicando al alumno en el proceso de enseñanza. 2. Resolución de problemas. El estudiante desarrolla competencias proponiendo soluciones adecuadas y aplicando las habilidades y conocimientos adquiridos; mediante la ejercitación de rutinas, aplicación de algoritmos, procedimientos de transformación de la información disponible y la interpretación de resultados. 3. Basada en proyectos. Consiste en la
----------------------------------	--

	<p>realización de un proyecto para la resolución de un problema mediante el análisis, diseño y realización de un subsistema.</p>
<p>Materiales y recursos didácticos</p>	<ul style="list-style-type: none">- Referencias bibliográficas. Para cada actividad se proporcionará una serie de referencias bibliográficas que pueden consultarse en la biblioteca de la Escuela o en Internet. - Recursos web. En la web de la asignatura se encontrarán los recursos didácticos para el seguimiento de las sesiones presenciales, así como las actividades y proyectos a realizar por el estudiante de forma autónoma o en equipos de trabajo. - Otros medios disponibles:<ul style="list-style-type: none">• Aula de Informática, “Máster”, Edificio de Ciencias Ambientales, Universidad de Alcalá.• Aula Virtual.• Web oficial del máster.• Bibliotecas (CRAI y Facultad de Medicina y Ciencias de la Salud)

5. EVALUACIÓN: Procedimientos, criterios de evaluación y de calificación¹

5.1. Procedimientos

La evaluación puede realizarse de forma continua o mediante una evaluación final, existiendo para cada caso dos convocatorias por matrícula: ordinaria y extraordinaria.

- **Evaluación continua**

La evaluación continua valora el desarrollo de las competencias durante todo el proceso de aprendizaje de la asignatura (NRPEA, Art. 3) mediante una serie de pruebas de carácter formativo distribuidas a lo largo del curso, que permiten al estudiante abordar la asignatura de forma progresiva.

Garantiza la retroalimentación temprana en el proceso de aprendizaje del alumno y permite a los profesores, coordinadores y demás elementos del Sistema de Garantía de Calidad hacer un seguimiento global, con la posibilidad de actuar en caso de que lo aconsejen indicadores o situaciones determinadas.

Es requisito para la superación de la asignatura en la evaluación continua la superación de las Prácticas y del Trabajo de la Asignatura, puesto que ambos conforman la parte práctica según el artículo 6.4 de la normativa.

Como criterio general, aquellos alumnos en convocatoria ordinaria que no se presenten a la evaluación de todas las prácticas se considerarán No Presentados.

- **Evaluación final**

Se solicitará a la Dirección del Máster.

5.2. Criterios de evaluación

¹ Es importante señalar los procedimientos de evaluación: por ejemplo evaluación continua, final, autoevaluación, co-evaluación. Instrumentos y evidencias: trabajos, actividades. Criterios o indicadores que se van a valorar en relación a las competencias: dominio de conocimientos conceptuales, aplicación, transferencia conocimientos. Para el sistema de calificación hay que recordar la **Normativa del Consejo de Gobierno del 16 de Julio de 2009**: la calificación de la evaluación continua representará, **al menos, el 60%**. Se puede elevar este % en la guía.

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 alumno ha adquirido conocimientos sobre los fundamentos, componentes básicos y técnicas para elaborar programas informáticos.

CE2: El alumno es capaz de Interpretar el código de lenguajes de programación de uso habitual en el ámbito de las TIG.

CE3: El alumno 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.

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

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

CE6: El alumno muestra interés por los contenidos y la materia estudiada.

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

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

5.3. Instrumentos de evaluación

Sistema de evaluación continua

El rendimiento de los alumnos será valorado por su trabajo, conocimientos y destrezas adquiridas y la mejora de su proceso de aprendizaje. Los instrumentos a emplear son:

- Pruebas de Evaluación Parcial (**PEP**):
 1. Pruebas de resolución de problemas prácticos con el lenguaje de programación (25% de la nota final).
 2. Pruebas 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 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 en clase (**PAC**):
 - Preguntas y participación en las discusiones en clase.
 - Asistencia a tutorías relacionadas con la asignatura.
 - Actitud proactiva en la resolución de problemas.(5% de la nota final)

Sistema de evaluación final

A aquellos alumnos a los que se les haya concedido la evaluación mediante examen final, se les evaluará mediante un único examen global acerca de todos los contenidos de la asignatura. La **evaluación final** considerará los siguientes instrumentos de evaluación:

1. **PEP**: Pruebas de Evaluación Parcial realizadas durante el periodo docente. Estas pruebas, individuales y en grupo, son de tipo resolución de problemas prácticos con el lenguaje de programación y las herramientas informáticas utilizadas para resolver y automatizar tareas TIG (véase su descripción en **Sistema de evaluación continua**). (25% de la nota final).
2. **TFA1**: El trabajo de la asignatura consistirá en la defensa del desarrollo de un prototipo software aplicado a tareas TIG utilizando el lenguaje y herramientas de programación manejadas durante el periodo docente. (35% de la nota final).
3. **PAC1**: Participación y actitud a través de medios principalmente *online* (e-mail, tutorías virtuales a través de videoconferencias, etc.). (5% de la nota final).

Si el alumno no supera la asignatura en la convocatoria ordinaria, en evaluación continua o final, tendrá la posibilidad de presentarse a un examen extraordinario con los mismos instrumentos de evaluación descritos para **evaluación final**.

5.4. Criterios de calificación

Esta sección muestra la relación entre criterios, instrumentos y calificación tanto en convocatoria ordinaria como en convocatoria extraordinaria.

Convocatoria Ordinaria, Evaluación continua

Criterio de evaluación	Instrumentos de evaluación	%
CE1-CE4, CE6-CE8	PEP1-PEP3. Pruebas de resolución de problemas prácticos sobre el lenguaje de programación Python (Bloques I-II)	25%
	PEP4-PEP7. Pruebas de resolución de problemas prácticos sobre Programación en Python para GIS (Bloque III)	35%
CE1- CE8	TFA. Con uso de Rúbrica.	35%
CE6	PAC	5%

Convocatoria Ordinaria, Evaluación final y Convocatoria Extraordinaria

Criterio de evaluación	Instrumentos de evaluación	%
CE1-CE4, CE6-CE8	PEP1-PEP3. Pruebas de resolución de problemas prácticos sobre el lenguaje de programación Python (Bloques I-II)	25%
	PEP4-PEP7. Pruebas de resolución de problemas prácticos sobre Programación en Python para GIS (Bloque III)	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, la superación de las prácticas y del trabajo final de la asignatura.

El sistema de calificación se ajustará al RD 1125/2003 por el cual se regula el sistema de créditos ECTS.

Como criterio general, aquellos alumnos en convocatoria ordinaria que no se presenten a la evaluación de todas las prácticas se considerarán *No Presentados*.

6. BIBLIOGRAFÍA

Bibliografía Básica

- Python Tutorial Release 3.2.3. Guido van Rossum, Jr. Fred L. Drake. Python Software Foundation, 2012.
- A Byte of Python. Swaroop C H. Creative Commons Attribution-ShareAlike 3.0, 2008.
- Learning Python: Powerful Object-Oriented Programming, 5th Edition. Mark Lutz. Ed. O'Reilly, 2013.
- Learning Geospatial Analysis with Python. Joel Lawhead. Packt Publishing, 2013.
- Programming ArcGIS 10.1 with Python Cookbook. Eric Pimpler. Packt Publishing.
- QGIS Python Programming Cookbook. Joel Lawhead. Packt Publishing, 2015.
- The PyQGIS Programmer's Guide. Extending QGIS 2.X with Python. Gary Sherman. Locate Press, 2014.

Bibliografía Complementaria

- How to Think Like a Computer Scientist: Learning with Python. Allen Downey Jeffrey Elkner Chris Meyers. Green Tea Press, 2008.
- Dive into Python. Mark Pilgrim. Ed. Prentice Hall, 2004
- Python Scripting for Computational Science. Hans Petter Langtangen, Springer Science & Business Media, 2008.
- Python 3. Object Oriented Programming. Dusty Phillips. Packt Publishing, 2010.
- Python Scripting for ArcGIS. Paul A. Zandbergen. Esri Press, 2013.
- GIS Tutorial for Python Scripting, David W. Allen. Esri Press, 2014.
- PyQGIS Developer Cookbook. Release 2.2. QGIS Project, 2014.