



UNIVERSIDAD
POLITÉCNICA
DE MADRID

PROCESO DE
COORDINACIÓN DE LAS
ENSEÑANZAS PR/CL/001



E.T.S. de Arquitectura

ANX-PR/CL/001-01

GUÍA DE APRENDIZAJE

ASIGNATURA

33000750 - Métodos Informáticos En Arquitectura

PLAN DE ESTUDIOS

03AM - Master Universitario En Arquitectura

CURSO ACADÉMICO Y SEMESTRE

2025/26 - Segundo semestre

Índice

Guía de Aprendizaje

1. Datos descriptivos.....	1
2. Profesorado.....	1
3. Competencias y resultados de aprendizaje.....	2
4. Descripción de la asignatura y temario.....	3
5. Cronograma.....	5
6. Actividades y criterios de evaluación.....	9
7. Recursos didácticos.....	12
8. Otra información.....	13

1. Datos descriptivos

1.1. Datos de la asignatura

Nombre de la asignatura	33000750 - Métodos Informáticos en Arquitectura
No de créditos	4 ECTS
Carácter	Optativa
Curso	Primer curso
Semestre	Primer semestre Segundo semestre
Período de impartición	Febrero-Junio
Idioma de impartición	Castellano
Titulación	03AM - Master Universitario en Arquitectura
Centro responsable de la titulación	03 - E.T.S. De Arquitectura
Curso académico	2025-26

2. Profesorado

2.1. Profesorado implicado en la docencia

Nombre	Despacho	Correo electrónico	Horario de tutorías *
Juan Francisco Padial Molina (Coordinador/a)		jf.padial@upm.es	- -

* Las horas de tutoría son orientativas y pueden sufrir modificaciones. Se deberá confirmar los horarios de tutorías con el profesorado.

3. Competencias y resultados de aprendizaje

3.1. Competencias

CT3 - Capacidad para adoptar soluciones creativas que satisfagan adecuadamente las diferentes necesidades planteadas.

CT4 - Capacidad para trabajar de forma efectiva como individuo, organizando y planificando su propio trabajo, de forma independiente o como miembro de un equipo.

CT5 - Capacidad para gestionar la información, identificando las fuentes necesarias, los principales tipos de documentos técnicos y científicos, de una manera adecuada y eficiente

RD10 - 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

RD6 - 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

RD7 - 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

RD9 - 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

3.2. Resultados del aprendizaje

RA107 - Lenguajes de programación. Lenguajes interpretados, programación orientada a objetos.

RA194 - Capacidad para analizar modelos matemáticos que aparecen en la Arquitectura y el Medio Ambiente.

RA109 - Modelización. Cálculo simbólico. Cálculo numérico. Elementos de geometría computacional

RA108 - Diseño paramétrico. Aplicaciones. Optimización. Programación. Python, C#

4. Descripción de la asignatura y temario

4.1. Descripción de la asignatura

En esta asignatura se introduce la programación con Python (lenguaje interpretado) como complemento a las posibilidades que ofrece el software de diseño paramétrico en arquitectura. Python es un lenguaje de programación que está presente en distintas aplicaciones informáticas de uso habitual en arquitectura o bien permiten la comunicación entre distintos tipos de software, como por ejemplo EnergyPlus, TRNSYS, ArcGIS, Rhinoceros o Revit, por ejemplo.

El curso se orienta al diseño paramétrico desde desde Grasshopper (plugin de Rhinoceros) mediante el clúster de Python. La mayor parte del curso está orientada introducirse en la programación con Python con aplicaciones en estadística, tratamiento de imágenes, cálculo numérico, como medio de aprendizaje mediante la suit Anaconda.

Una vez introducida la programación en Python, se trabaja con Rhinoceros y su plugin de Grasshopper incorporando las técnicas de programación introducidas en la primera etapa del curso, con el fin de aplicarlo a problemas de diseño paramétrico.

4.2. Temario de la asignatura

1. Programación y diseño paramétrico

- 1.1. Introducción a software de diseño paramétrico y lenguajes de programación.
- 1.2. Rhinoceros, Grasshopper y Python
- 1.3. Primeros modelos de diseño paramétrico.
- 1.4. La necesidad de la recursividad en el diseño paramétrico
- 1.5. Programación con Grasshopper. Algunas limitaciones
- 1.6. Posibilidades de integrar un lenguaje de programación en software de diseño paramétrico. El caso de Rhinoceros: C#, VisualBasic, Python en Grasshopper

2. Programación en Python

- 2.1. Introducción. Instalación. Módulos
- 2.2. Conceptos básicos: instancias, operaciones, diagramas de flujo, funciones
- 2.3. Módulos en Python: sobre el sistema, math, random, numpy, pyplot
- 2.4. Lectura y escritura de archivos.
- 2.5. Algunos módulos relacionados con Microsoft Office y archivos PDF
- 2.6. Archivos de imágenes. Tratamiento de imágenes.

3. Rhinoceros y Grasshopper

- 3.1. Proyectos
- 3.2. Python en Grasshopper
 - 3.2.1. Interface de entrada y salida
 - 3.2.2. Módulos de Rhinoceros y Grasshopper en Python
- 3.3. Proyectos

5. Cronograma

5.1. Cronograma de la asignatura *

Sem	Actividad tipo 1	Actividad tipo 2	Tele-enseñanza	Actividades de evaluación
1	<p>Introducción de los contenidos de la sesión Duración: 00:30 LM: Actividad del tipo Lección Magistral</p>	<p>Ejemplos, problemas y aplicaciones. Programación. Duración: 01:00 PR: Actividad del tipo Clase de Problemas</p> <p>Resolución de problemas. Algoritmos. Programación. Proyectos Duración: 01:30 PL: Actividad del tipo Prácticas de Laboratorio</p> <p>Resolución de problemas. Algoritmos. Programación. Proyectos Duración: 01:00 AC: Actividad del tipo Acciones Cooperativas</p>		
2	<p>Introducción de los contenidos de la sesión Duración: 00:30 LM: Actividad del tipo Lección Magistral</p>	<p>Ejemplos, problemas y aplicaciones. Programación. Duración: 01:00 PR: Actividad del tipo Clase de Problemas</p> <p>Resolución de problemas. Algoritmos. Programación. Proyectos Duración: 01:30 PL: Actividad del tipo Prácticas de Laboratorio</p> <p>Resolución de problemas. Algoritmos. Programación. Proyectos Duración: 01:00 AC: Actividad del tipo Acciones Cooperativas</p> <p>Participación en clase, entrega de programas y proyectos, colaboración en el trabajo en grupo Duración: 00:00 OT: Otras actividades formativas / Evaluación</p>		<p>Participación en clase, entrega de programas y proyectos, colaboración en el trabajo en grupo PG: Técnica del tipo Presentación en Grupo Evaluación Progresiva Presencial Duración: 00:00</p>
3	<p>Introducción de los contenidos de la sesión Duración: 00:30 LM: Actividad del tipo Lección Magistral</p>	<p>Ejemplos, problemas y aplicaciones. Programación. Duración: 01:00 PR: Actividad del tipo Clase de Problemas</p> <p>Resolución de problemas. Algoritmos. Programación. Proyectos Duración: 01:30 PL: Actividad del tipo Prácticas de Laboratorio</p> <p>Resolución de problemas. Algoritmos.</p>		

		<p>Programación. Proyectos Duración: 01:00 AC: Actividad del tipo Acciones Cooperativas</p>	
4	<p>Introducción de los contenidos de la sesión Duración: 00:30 LM: Actividad del tipo Lección Magistral</p>	<p>Ejemplos, problemas y aplicaciones. Programación. Duración: 01:00 PR: Actividad del tipo Clase de Problemas</p> <p>Resolución de problemas. Algoritmos. Programación. Proyectos Duración: 01:30 PL: Actividad del tipo Prácticas de Laboratorio</p> <p>Resolución de problemas. Algoritmos. Programación. Proyectos Duración: 01:00 AC: Actividad del tipo Acciones Cooperativas</p> <p>Participación en clase, entrega de programas y proyectos, colaboración en el trabajo en grupo Duración: 00:00 OT: Otras actividades formativas / Evaluación</p>	<p>Participación en clase, entrega de programas y proyectos, colaboración en el trabajo en grupo PG: Técnica del tipo Presentación en Grupo Evaluación Progresiva Presencial Duración: 00:00</p>
5	<p>Introducción de los contenidos de la sesión Duración: 00:30 LM: Actividad del tipo Lección Magistral</p>	<p>Ejemplos, problemas y aplicaciones. Programación. Duración: 01:00 PR: Actividad del tipo Clase de Problemas</p> <p>Resolución de problemas. Algoritmos. Programación. Proyectos Duración: 01:30 PL: Actividad del tipo Prácticas de Laboratorio</p> <p>Resolución de problemas. Algoritmos. Programación. Proyectos Duración: 01:00 AC: Actividad del tipo Acciones Cooperativas</p>	
6	<p>Introducción de los contenidos de la sesión Duración: 00:30 LM: Actividad del tipo Lección Magistral</p>	<p>Ejemplos, problemas y aplicaciones. Programación. Duración: 01:00 PR: Actividad del tipo Clase de Problemas</p> <p>Resolución de problemas. Algoritmos. Programación. Proyectos Duración: 01:30 PL: Actividad del tipo Prácticas de Laboratorio</p> <p>Resolución de problemas. Algoritmos. Programación. Proyectos Duración: 01:00 AC: Actividad del tipo Acciones Cooperativas</p> <p>Participación en clase, entrega de programas y proyectos, colaboración en el trabajo en grupo</p>	<p>Participación en clase, entrega de programas y proyectos, colaboración en el trabajo en grupo PG: Técnica del tipo Presentación en Grupo Evaluación Progresiva Presencial Duración: 00:00</p>

		Duración: 00:00 OT: Otras actividades formativas / Evaluación		
7		Resolución de problemas. Algoritmos. Programación. Proyectos Duración: 02:00 PL: Actividad del tipo Prácticas de Laboratorio Resolución de problemas. Algoritmos. Programación. Proyectos Duración: 02:00 AC: Actividad del tipo Acciones Cooperativas		
8		Resolución de problemas. Algoritmos. Programación. Proyectos Duración: 02:00 PL: Actividad del tipo Prácticas de Laboratorio Resolución de problemas. Algoritmos. Programación. Proyectos Duración: 02:00 AC: Actividad del tipo Acciones Cooperativas Participación en clase, entrega de programas y proyectos, colaboración en el trabajo en grupo Duración: 00:00 OT: Otras actividades formativas / Evaluación		Participación en clase, entrega de programas y proyectos, colaboración en el trabajo en grupo PG: Técnica del tipo Presentación en Grupo Evaluación Progresiva Presencial Duración: 00:00
9		Resolución de problemas. Algoritmos. Programación. Proyectos Duración: 02:00 PL: Actividad del tipo Prácticas de Laboratorio Resolución de problemas. Algoritmos. Programación. Proyectos Duración: 02:00 AC: Actividad del tipo Acciones Cooperativas		
10		Entrega prácticas y proyectos. Defensa pública (10 minutos/proyecto). Se evalúa la exposición y el trabajo entregado Duración: 02:00 OT: Otras actividades formativas / Evaluación		Entrega prácticas y proyectos. Defensa pública (10 minutos/proyecto). Se evalúa la exposición y el trabajo entregado PG: Técnica del tipo Presentación en Grupo Evaluación Progresiva Presencial Duración: 02:00
11				
12				
13				
14				

15				
16				
17		<p>Entrega y defensa de la solución a distintas propuestas sobre programación con Python (suit Anaconda) y diseño paramétrico con Grasshopper integrando el clúster de Python</p> <p>Duración: 10:00 OT: Otras actividades formativas / Evaluación</p>		<p>Entrega y defensa de la solución a distintas propuestas sobre programación con Python (suit Anaconda) y diseño paramétrico con Grasshopper integrando el clúster de Python</p> <p>TI: Técnica del tipo Trabajo Individual Evaluación Global No presencial Duración: 10:00</p>

Para el cálculo de los valores totales, se estima que por cada crédito ECTS el alumno dedicará dependiendo del plan de estudios, entre 26 y 27 horas de trabajo presencial y no presencial.

6. Actividades y criterios de evaluación

6.1. Actividades de evaluación de la asignatura

6.1.1. Evaluación (progresiva)

Sem.	Descripción	Modalidad	Tipo	Duración	Peso en la nota	Nota mínima	Competencias evaluadas
2	Participación en clase, entrega de programas y proyectos, colaboración en el trabajo en grupo	PG: Técnica del tipo Presentación en Grupo	Presencial	00:00	5%	5 / 10	CT3 CT5 RD7 CT4 RD10 RD6 RD9
4	Participación en clase, entrega de programas y proyectos, colaboración en el trabajo en grupo	PG: Técnica del tipo Presentación en Grupo	Presencial	00:00	5%	5 / 10	CT3 CT5 RD7 CT4 RD10 RD6 RD9
6	Participación en clase, entrega de programas y proyectos, colaboración en el trabajo en grupo	PG: Técnica del tipo Presentación en Grupo	Presencial	00:00	5%	5 / 10	CT3 CT5 RD7 CT4 RD10 RD6 RD9
8	Participación en clase, entrega de programas y proyectos, colaboración en el trabajo en grupo	PG: Técnica del tipo Presentación en Grupo	Presencial	00:00	5%	5 / 10	CT3 CT5 RD7 CT4 RD10 RD6 RD9
10	Entrega prácticas y proyectos. Defensa pública (10 minutos/proyecto). Se evalúa la exposición y el trabajo entregado	PG: Técnica del tipo Presentación en Grupo	Presencial	02:00	80%	5 / 10	CT3 CT5 RD7 CT4 RD10 RD6 RD9

6.1.2. Prueba evaluación global

Sem	Descripción	Modalidad	Tipo	Duración	Peso en la nota	Nota mínima	Competencias evaluadas
17	Entrega y defensa de la solución a distintas propuestas sobre programación con Python (suit Anaconda) y diseño paramétrico con Grasshopper integrando el clúster de Python	TI: Técnica del tipo Trabajo Individual	No Presencial	10:00	100%	5 / 10	CT3 CT5 RD7 CT4 RD10 RD6 RD9

6.1.3. Evaluación convocatoria extraordinaria

Descripción	Modalidad	Tipo	Duración	Peso en la nota	Nota mínima	Competencias evaluadas
Prueba global. Consiste en la entrega de todas las prácticas que han sido objeto de evaluación durante el curso, más un ejercicio presencial en el día y horario establecido en el calendario de evaluación.	EP: Técnica del tipo Examen de Prácticas	Presencial	03:00	100%	5 / 10	CT3 CT5 RD7 CT4 RD10 RD6 RD9

6.2. Criterios de evaluación

La asignatura puede ser superada mediante el proceso de evaluación continua o mediante solo examen final.

La asignatura puede ser superada mediante el proceso de evaluación progresiva (de la semana 1 a la semana 10) o mediante examen global correspondiente a la convocatoria ordinaria y cronológicamente a la semana 17 cuando

no se ha superado la asignatura mediante la evaluación progresiva. Finalmente se dispone de la evaluación extraordinaria en caso de no superar la asignatura mediante el proceso de evaluación progresiva ni evaluación global en su convocatoria ordinaria.

Criterios de evaluación:

- Se valora la adquisición de conocimientos relacionados con los contenidos de la asignatura
- Se valora la utilización y manejo de software adecuado a la materia
- Se valora el análisis y la crítica
- Se valora la toma de decisiones respaldada por criterios científicos
- Se valora la presentación y defensa de proyectos de la asignatura
- Se valora la participación activa en el aula y en grupo

7. Recursos didácticos

7.1. Recursos didácticos de la asignatura

Nombre	Tipo	Observaciones
Software	Equipamiento	La programación y los proyectos se realizarán mediante software específico: Rhinoceros, Grasshopper y Python
Diversos recursos WEB, Moodle, documentos, videos, etc.	Recursos web	Se indicarán en tiempo real durante el desarrollo de las clases, y se propondrán para la realización de trabajos personales o en grupo.
Diversos recursos WEB, documentos, videos, etc.	Recursos web	https://developer.rhino3d.com/guides/rhinopython/your-first-python-script-in-grasshopper/
Hans Petter Langtangen, A Primer on Scientific Programming with Python	Bibliografía	
Laura Tateosian. Python For ArcGIS	Bibliografía	Bibliografía complementaria
Kent D. Lee. Python Programming Fundamentals	Bibliografía	
Robert Woodbury. Elements of Parametric Design	Bibliografía	
H. Pottmann, J. Wallner, Computational Line Geometry	Bibliografía	Bibliografía complementaria
Mark Summerfield. Programación. Python 3	Bibliografía	

8. Otra información

8.1. Otra información sobre la asignatura

Aplicación del diseño paramétrico y la programación en Python para el diseño, el análisis, la optimización y mejora del diseño y sus implicaciones en los procesos de fabricación industrial, eficiencia energética en edificios y sus procesos de ejecución, minimización de costes.

- ODS9: Industria, innovación e infraestructura
- ODS11: Ciudades y comunidades sostenibles