



FICHA TÉCNICA DE LA ASIGNATURA

Datos de la asignatura	
Nombre completo	Geometría Computacional
Código	DMA-IMAT-311
Título	N/A [GiMAT 26]
Impartido en	Grado en Ingeniería Matemática e Inteligencia Artificial [Tercer Curso]
Créditos	4,5 ECTS
Carácter	Obligatoria (Grado)
Departamento / Área	Departamento de Matemática Aplicada
Responsable	Santiago Canales Cano

Datos del profesorado	
Profesor	
Nombre	Santiago Canales Cano
Departamento / Área	Departamento de Matemática Aplicada
Despacho	Alberto Aguilera 25 [D-208]
Correo electrónico	scanales@icai.comillas.edu
Profesor	
Nombre	Miguel Ángel Berbel López
Departamento / Área	Departamento de Matemática Aplicada
Despacho	Alberto Aguilera 25 [D-204]
Correo electrónico	maberbel@icai.comillas.edu

DATOS ESPECÍFICOS DE LA ASIGNATURA

Contextualización de la asignatura
Aportación al perfil profesional de la titulación
Una característica fundamental de las tecnologías que han aparecido en el presente siglo XXI es que están constituidas por la aportación conjunta de muchas ramas de la ciencia, pero en todas ellas aparece el uso crucial del ordenador, no ya como destinatario de las aplicaciones desarrolladas, sino como herramienta en la investigación y en la producción. Las matemáticas, de una u otra forma, participan en todos estos avances tecnológicos, a veces de forma significativa, pero poco visible, a veces con un protagonismo crucial. Éste es el caso de la <i>Geometría Computacional</i>, que participa de una forma sustancial en diversas áreas de investigación aplicada como por ejemplo la Informática gráfica, el Diseño y fabricación asistidos por ordenador (CAD/CAM), la Caracterización y reconocimiento automático de formas, el Diseño de circuitos integrados gigantes, la Visión artificial, los Sistemas de información geográfica y la Robótica entre otros. Así, el modelado geométrico trata la representación de formas, de cara a visualizarlas, analizarlas, diseñarlas y estudiar su comportamiento. Por tanto, la aportación de la asignatura <i>Geometría Computacional</i> en el perfil profesional de la titulación se puede describir en dos puntos fundamentales:



- Por una parte incrementa las herramientas geométricas que la geometría clásica establece dentro de un conocimiento matemático profundo.
- Por otra parte, dota al alumno de técnicas geométricas computacionales modernas, donde el manejo de datos y algoritmos juega un papel fundamental.

Así, al finalizar el curso el alumno:

- Estará familiarizado con los fundamentos de la *Geometría Computacional*, sus herramientas, técnicas y algoritmos.
- Será capaz de utilizar y diseñar algoritmos de análisis de datos geométricos y de aplicar dichos algoritmos a problemas reales.

Prerrequisitos

Se presuponen conocimientos matemáticos de los estudios de ESO y Bachillerato, en particular:

1. Resolución de ecuaciones e inecuaciones.
2. Resolución de sistemas de ecuaciones.
3. Análisis y representación matemática de elementos geométricos, tanto en el plano como en el espacio.

Se presuponen, así mismo, conocimientos matemáticos adquiridos en tres asignaturas de la titulación cursadas previamente:

1. Fundamentos de productos escalares y vectoriales y de espacios vectoriales euclídeos de la asignatura *Álgebra y Geometría*.
2. Conocimientos de programación en lenguaje Python de la asignatura *Programación*.
3. Fundamentos de estructuras de datos y algoritmos de la asignatura *Algoritmos y estructuras de datos*.

Resultados del proceso de formación y aprendizaje

Conocimientos o contenidos

CON04	Capacidad para comprender y dominar los conceptos básicos de matemática discreta, lógica, algorítmica y complejidad computacional.
--------------	--

Habilidades o destrezas

HAB01	Capacidad para la resolución de los problemas matemáticos generales que puedan plantearse en la ingeniería.
HAB02	Capacidad de razonamiento abstracto y sentido crítico, así como de cálculo, modelado, simulación, optimización y predicción, para dar respuesta a los problemas planteados por la ciencia, la tecnología y la sociedad en general.
HAB04	Capacidad para la resolución de los problemas matemáticos que puedan plantearse en la ingeniería, aplicando con aptitud los conocimientos sobre: álgebra lineal y multilineal, geometría, cálculo diferencial e integral, ecuaciones diferenciales, métodos numéricos, estadística y optimización.
HAB05	Capacidad para saber aplicar las técnicas matemáticas más adecuadas en la resolución de los diferentes problemas, técnicos y tecnológicos, planteados en el ámbito de la ingeniería y la inteligencia artificial. Aptitud para conocer el rango de aplicabilidad y limitaciones en la resolución de problemas de las diferentes herramientas matemáticas.
HAB06	Capacidad para utilizar con habilidad y soltura software matemático, así como para implementar algoritmos y desarrollar programas informáticos que permitan resolver los problemas matemáticos planteados en el ámbito de la ingeniería y de la inteligencia artificial.



HAB07	Aptitud para aplicar las técnicas propias de la matemática discreta y la geometría computacional a la resolución de problemas de optimización discreta, al modelado de la interacción entre las componentes de un sistema, al estudio de bases de datos y diagramas de jerarquía, a problemas de codificación y decodificación de la información, al modelado de redes de computadores y a la resolución de problemas de localización y otros problemas geométricos.
HAB10	Dominio de las principales estructuras de datos y técnicas algorítmicas, siendo capaz de implementarlas en distintos lenguajes de programación conociendo su complejidad computacional.
HAB22	Capacidad para diseñar y aplicar métodos y algoritmos heurísticos de búsqueda para la toma de decisiones.

BLOQUES TEMÁTICOS Y CONTENIDOS

Contenidos – Bloques Temáticos
Geometría Computacional
Tema 1. Introducción a la Geometría Computacional
1. ¿Qué es la Geometría Computacional? Historia. 2. Terminología y aplicaciones. 3. Complejidad de un problema. 4. Operaciones básicas en Geometría Computacional. ◦ Intersección de segmentos. ◦ Cálculo de diagonales. ◦ Cálculo de tangentes. 5. Algunos métodos de programación utilizados. ◦ Método incremental. ◦ Divide y vencerás.
Tema 2. Cierres Convexos
1. Introducción. Definición de cierre convexo. 2. Algoritmos para calcular la envolvente convexa. ◦ Scan de Graham. ◦ Marcha de Jarvis. ◦ Quickhull. ◦ Algoritmo incremental. ◦ Método divide y vencerás. 3. Cierre convexo de un polígono. ◦ Cierre convexo de un polígono. ◦ Algoritmo de Lee. ◦ Algoritmo de Melkman. 4. Aplicaciones. Diámetro, anchura y pares antipodales. ◦ Diámetro de una nube de puntos. ◦ Diámetro de un polígono convexo. ◦ Otras aplicaciones de la rotación de calibres.
Tema 3. Triangulaciones



1. Introducción. Problema de la galería de arte.
2. Triangulación de un polígono.
 - Triangulación por inserción de diagonales.
 - Grafo dual de una triangulación.
 - Triangulación cortando orejas.
 - Triangulación de polígonos especiales.
 - Triangulación de polígonos descomponiendo en particiones monótonas.
3. Sistemas de información geográfica.
4. Triangulación de nubes de puntos.
 - Definiciones y propiedades.
 - Comparación de triangulaciones.
 - Triangulación de Delaunay.
 - Propiedades.
 - Triangulación por flips.
 - Algoritmo aleatorio incremental.

Tema 4. Localización de puntos

1. Localización de un punto en un polígono.
 - Polígono convexo.
 - Polígono estrellado.
2. Localización en subdivisiones planas.
 - Método de las franjas.
 - Mapas trapezoidales.
 - Método de la cadena.

Tema 5. Diagramas de Voronoi

1. Diagramas de Voronoi.
 - Definición. Elementos del diagrama de Voronoi.
 - Propiedades de los diagramas de Voronoi.
 - La triangulación de Delaunay y el diagrama de Voronoi.
2. Métodos de construcción del Diagrama de Voronoi.
 - Método incremental.
 - Método divide y vencerás.
3. Problemas que resuelven el diagrama de Voronoi o la triangulación de Delaunay.
 - Problemas de proximidad.
 - Interpolación espacial.
4. Aplicaciones de los diagramas de Voronoi.

Tema 6. Visibilidad, planificación de trayectorias y colisiones

1. Definiciones. Problema de la galería de arte.
2. Planificación de trayectorias.
 - Grafo de visibilidad.
3. Detección de colisiones.

Tema 7. Técnicas metaheurísticas

1. Introducción a las técnicas heurísticas.



2. Simulated annealing.
3. Ant systems.
4. Algoritmos genéticos.

METODOLOGÍA DOCENTE

Aspectos metodológicos generales de la asignatura

Metodología Presencial: Actividades

1. Clase magistral expositiva-participativa (14 horas)

El profesor explicará los conceptos fundamentales de cada tema mediante una exposición dialogada en la que, apoyándose en un buen material, presentará de forma clara y organizada los contenidos, a la vez que estimulará la participación de los estudiantes en el proceso de enseñanza-aprendizaje. Se fomentará el diálogo a través de la formulación de preguntas diversas, dirigidas a la comprensión de la información, así como preguntas retadoras, orientadas a identificar los conocimientos previos de los estudiantes. También se utilizará la presentación de ejemplos prácticos, problemas tipo y situaciones cercanas que despierten la motivación de los estudiantes en torno al tema, como por ejemplo noticias de actualidad.

2. Sesiones prácticas con uso de software. Ejercicios prácticos y resolución de problemas (15+6 horas)

En estas sesiones se explicarán, corregirán y analizarán problemas prácticos de cada tema, análogos a los resueltos en las lecciones expositivas y también otros de mayor complejidad, previamente propuestos por el profesor y trabajados por el alumno. En estas clases se favorecerá la participación del alumno y la interacción alumno-profesor y alumno-alumno como vía para fomentar el aprendizaje colaborativo y la capacidad de autoaprendizaje. Además, siguiendo la metodología de clase invertida, se propondrá el aprendizaje de ciertos contenidos de la asignatura fuera del aula, liberando tiempo para facilitar la participación de los estudiantes en el aprendizaje activo a través de preguntas.

Para la realización de dichas prácticas se utilizará el software matemático SageMath (System Algebra for Geometry Experimentation), con el cual los alumnos podrán implementar en lenguaje Python los algoritmos explicados en las clases teóricas, realizando experimentos geométricos gráficos que permitan analizar las propiedades de dichos algoritmos, su complejidad y los problemas computacionales de la vida real que pueden resolver.

3. Tutorías para resolver dudas (5 horas)

En el período lectivo se reservará un porcentaje de tiempo de las clases presenciales para resolver todas las dudas que los alumnos vayan manifestando, tanto de los conceptos teóricos expuestos en las clases magistrales, como en las prácticas realizadas en las sesiones prácticas con ordenador. Dichas sesiones de dudas serán participativas y estarán encaminadas a orientar al alumno en la búsqueda personal de las soluciones a los problemas planteados, sobre todo, en las dudas sobre los problemas planteados en las sesiones de prácticas con ordenador y en la elaboración del proyecto final grupal.



4. Casos prácticos-proyecto (2 horas)

A lo largo del cuatrimestre, los alumnos realizarán en grupos de tres alumnos un proyecto final de asignatura. Dicho proyecto consistirá en el desarrollo de una web interactiva utilizando el software SageMath y cuya realización será ampliamente explicada por el profesor en la clases prácticas. En dicho proyecto los alumnos implementarán alguno de los algoritmos descritos en las clases teóricas y que no hayan sido desarrollados e implementados en las clases prácticas, haciendo especial hincapié en la explicación del algoritmo, su complejidad y sus aplicaciones reales en la industria.

Como apoyo al desarrollo de dicho proyecto, se realizarán 2 sesiones de control de 1 hora a lo largo del cuatrimestre, en las que los alumnos contarán con la asistencia del profesor de la asignatura para dirigir adecuadamente su trabajo, ayudarles a detectar posibles fallos en sus decisiones de diseño en momentos tempranos del desarrollo y, en general, para guiar el proceso de desarrollo del proyecto y resolver aquellas dudas que pudieran surgirle a los alumnos durante el mismo.

5. Actividades de evaluación continua del rendimiento (3 horas)

Se realizará una prueba de evaluación continua, con el objetivo de valorar en su totalidad el proceso de aprendizaje del alumnado, y mejorarlo a medida que transcurre el curso. En concreto, se efectuará un examen intercuatrimestral de 1 hora y 30 min, a realizar hacia la mitad del cuatrimestre y cuya fecha exacta y duración queda fijada por la Dirección de la Escuela.

Asimismo se dedicarán 1 hora y 30 minutos de la última semana del curso a la exposición de los proyectos grupales asignados. Dicha exposición la realizarán todos los miembros del grupo y constituirá una parte de la nota de evaluación de dicho proyecto final.

Metodología No presencial: Actividades

1. Sesiones prácticas con uso de software. Ejercicios prácticos y resolución de problemas (30+20 horas)

Partiendo de las explicaciones teóricas impartidas por el profesor y de las descripciones de los algoritmos proporcionadas, los alumnos trabajarán individualmente las prácticas de la asignatura fuera del aula. Dichas prácticas se compondrán de problemas Tipo A (que se realizarán presencialmente en el aula tras el trabajo previo de los alumnos) y problemas de Tipo B (que forma parte del trabajo personal del alumno). Por tanto el trabajo de las prácticas se realizará por una parte antes de las sesiones prácticas presenciales (para intentar los problemas de Tipo A y según indicará el profesor previamente en clase), y por otra posteriormente, para trabajar y depurar todos aquellos elementos que no hayan podido ser finalizados durante las sesiones presenciales de prácticas y constituirán fundamentalmente los problemas de Tipo B de dichas prácticas. De esta forma, el alumno intentará antes de cada práctica resolver los problemas planteados, utilizando posteriormente las explicaciones del profesor (en las sesiones prácticas presenciales) para resolver todas las dudas surgidas durante este trabajo previo. A continuación se terminará fuera del aula todo el trabajo incompleto de la práctica, para de esta forma afianzar y comprender en profundidad todos los conceptos trabajados.

2. Proyectos. Trabajos y casos prácticos (6+4 horas)

Según lo descrito en el apartado de "proyectos" de la sección de actividades presenciales, los alumnos trabajarán en tríos en un proyecto final de asignatura. Los alumnos recibirán una descripción de los requerimientos de cada proyecto y deberán tomar las decisiones de diseño y arquitectura de software necesarias para construir programas capaces de cumplir con la funcionalidad requerida. Para este fin, necesitarán combinar tanto los conocimientos teóricos y prácticos de la asignatura, como conocimientos fundamentales de programación y diseño de algoritmos, así como ciertos conocimientos transversales de programación y procesamiento de datos de otras asignaturas de segundo curso.

3. Estudio personal (30 horas)

El alumno debe realizar un trabajo autónomo para comprender e interiorizar los fundamentos teóricos de la asignatura. Este trabajo de asimilación se realizará después de las clases magistrales y previamente a las sesiones donde se utilice el modelo de clase Invertida.



RESUMEN HORAS DE TRABAJO DEL ALUMNO

HORAS PRESENCIALES					
AF1. Clases magistrales expositivas y participativas	AF3. Sesiones prácticas con uso de software	AF5. Tutorías para resolución de dudas	AF8. Casos prácticos	AF11. Ejercicios prácticos y resolución de problemas	AF12. Actividades de evaluación continua del rendimiento
14.00	15.00	5.00	2.00	6.00	3.00
HORAS NO PRESENCIALES					
AF3. Sesiones prácticas con uso de software	AF6. Trabajos	AF8. Casos prácticos	AF10. Estudio personal	AF11. Ejercicios prácticos y resolución de problemas	
30.00	6.00	4.00	30.00	20.00	
CRÉDITOS ECTS: 4,5 (135,00 horas)					

EVALUACIÓN Y CRITERIOS DE CALIFICACIÓN

El uso de IA para crear trabajos completos o partes relevantes, sin citar la fuente o la herramienta o sin estar permitido expresamente en la descripción del trabajo, será considerado plagio y regulado conforme al Reglamento General de la Universidad.

Actividades de evaluación	Criterios de evaluación	Peso
---------------------------	-------------------------	------



Exámenes escritos teórico-prácticos: - Examen Intercuatrimstral (IE) (15%) - Examen Final (FE) (35%) Nota: Para aplicar las ponderaciones indicadas en el sistema de evaluación general de la asignatura, será necesario obtener una nota mínima de al menos 4 puntos en el examen final de la asignatura tanto en la parte teórica como práctica.	Prueba intercuatrimestral: A mitad del cuatrimestre se realizará una prueba intercuatrimestral, que abarcará todo el temario del cuatrimestre impartido hasta el momento. Esta prueba tendrá dos partes: una escrita, (IE), sobre los conceptos y algoritmos explicados en clase teóricas presenciales y en las sesiones prácticas con ordenador y otra práctica (IP) en la que se evaluará la resolución de problemas computacionales utilizando el software SageMath empleado en las prácticas con ordenador. Ambas partes se evaluarán sobre 10 puntos. Examen final: Al final del cuatrimestre se realizará un examen final, que abarcará toda la materia de teoría y prácticas del curso. De forma equivalente al examen intercuatrimestral, este examen tendrá dos partes: una escrita, (FE), sobre los conceptos y algoritmos explicados en clase teóricas presenciales y en las sesiones prácticas con ordenador y otra práctica (FP) en la que se evaluará la resolución de problemas computacionales utilizando el software SageMath empleado en las prácticas con ordenador. Ambas partes se evaluarán sobre 10 puntos igual que en la prueba intercuatrimestral.	50 %
--	--	-------------



Exámenes prácticos con ordenador: - Examen Intercuatrimstral (IP) (10%) - Examen Final (FP) (20%) Nota: Para aplicar las ponderaciones indicadas en el sistema de evaluación general de la asignatura, será necesario obtener una nota mínima de al menos 4 puntos en el examen final de la asignatura tanto en la parte teórica como práctica.	Prueba intercuatrimestral: A mitad del cuatrimestre se realizará una prueba intercuatrimestral, que abarcará todo el temario del cuatrimestre impartido hasta el momento. Esta prueba tendrá dos partes: una escrita, (IE), sobre los conceptos y algoritmos explicados en clase teóricas presenciales y en las sesiones prácticas con ordenador y otra práctica (IP) en la que se evaluará la resolución de problemas computacionales utilizando el software SageMath empleado en las prácticas con ordenador. Ambas partes se evaluarán sobre 10 puntos. Examen final: Al final del cuatrimestre se realizará un examen final, que abarcará toda la materia de teoría y prácticas del curso. De forma equivalente al examen intercuatrimestral, este examen tendrá dos partes: una escrita, (FE), sobre los conceptos y algoritmos explicados en clase teóricas presenciales y en las sesiones prácticas con ordenador y otra práctica (FP) en la que se evaluará la resolución de problemas computacionales utilizando el software SageMath empleado en las prácticas con ordenador. Ambas partes se evaluarán sobre 10 puntos igual que en la prueba intercuatrimestral.	30 %
Entrega de proyecto (evaluación de proyecto por grupos): Proyecto final (P) (20%)	A lo largo del cuatrimestre se realizará un proyecto final de asignatura relacionado con las sesiones prácticas realizadas durante el curso. Dicho proyecto final será evaluado sobre 10 puntos.	20 %

Calificaciones

Con el objetivo de evaluar de forma continua el trabajo del alumno a lo largo del cuatrimestre, se realizará una **prueba intercuatrimestral**, a mitad del cuatrimestre, que estará compuesta de una parte escrita (**IE**), (que versará sobre los contenidos expuestos tanto en las clases teóricas de la asignatura como en las sesiones prácticas con ordenador: elementos geométricos, propiedades, algoritmos, etc..) y una parte práctica (**IP**), (relacionada con los ejercicios prácticos realizados con el software SageMath). Al finalizar el curso se realizará un **examen al final**, formado también por una parte escrita (**FE**) y una parte práctica (**FP**), de similares características a las del examen intercuatrimestral. Además, con el objetivo de afianzar los conceptos básicos de la asignatura, los alumnos realizarán por grupos un proyecto final (**P**) de asignatura, en el que se aplicarán los conceptos teóricos, usando las destrezas adquiridas en las sesiones prácticas con ordenador.

1.- Evaluación en la convocatoria ordinaria:

La evaluación de la asignatura se compondrá de cinco notas como se indicó anteriormente: **IE, IP, FE, FP y P**.



- **Nota IE (sobre 10 puntos):** nota obtenida por el alumno en la parte escrita del examen intercuatrimestral.
- **Nota IP (sobre 10 puntos):** nota obtenida por el alumno en la parte práctica del examen intercuatrimestral.
- **Nota FE (sobre 10 puntos):** nota obtenida por el alumno en la parte escrita del examen final.
- **Nota FP (sobre 10 puntos):** nota obtenida por el alumno en la parte práctica del examen final.
- **Nota P (sobre 10 puntos):** nota obtenida por el alumno en el proyecto final de asignatura.

Si las notas **FE** y **FP** son mayores o iguales que **4**, la calificación **N** de la signatura será:

$$N = IE*(0.15) + IP*(0.10) + FE*(0.35) + FP*(0.20) + P*(0.20)$$

siendo el mínimo entre **FE** y **FP**, la nota **N** de la asignatura en otro caso.

La asignatura se aprueba en la convocatoria ordinaria si **N ≥ 5**, suspendiendo la asignatura en esta convocatoria en caso contrario.

BIBLIOGRAFÍA Y RECURSOS

Bibliografía Básica

- SATYAN L. DEVADOSS, JOSEPH O'ROURKE. Discrete and Computational Geometry. Princeton University Press 2011.
- M. BERG, M. VAN KREVELD, M. OVERMARS, O. SCHWARZKOPF. Computational Geometry. Algorithms and Applications. Springer-Verlag 1997.

Bibliografía Complementaria

- JOSEPH O'ROURKE. Computational Geometry in C. Cambridge University Press 1993.
- FRANCO P. PREPARATA, MICHAEL IAN SHAMOS. Computational Geometry and Introduction 1990.