

**Proyecto/Guía Docente de la asignatura**

<b>Asignatura</b>	Matemática Discreta		
<b>Materia</b>	Matemáticas		
<b>Módulo</b>	Formación Básica		
<b>Titulación</b>	Grado en Ingeniería Informática de Servicios y Aplicaciones		
<b>Plan</b>	413 / 5471 / 5472	<b>Código</b>	40800
<b>Periodo de impartición</b>	Semestre 1	<b>Tipo/Carácter</b>	FB
<b>Nivel/Ciclo</b>	Grado	<b>Curso</b>	1
<b>Créditos ECTS</b>	6		
<b>Lengua en que se imparte</b>	Español		
<b>Profesor responsable</b>	José Ignacio Farrán Martín		
<b>Datos de contacto (E-mail, teléfono...)</b>	Escuela de Ingeniería Informática Plaza de la Universidad 1 - 40005 Segovia Teléfono : (+34) 921 11 24 20 Fax : (+34) 921 11 24 01 e-mail : jifarran@eii.uva.es web : <a href="http://wmatem.eis.uva.es/~ignfar">http://wmatem.eis.uva.es/~ignfar</a>		
<b>Horario de tutorías</b>	Ver <a href="http://campusvirtual.uva.es">http://campusvirtual.uva.es</a>		
<b>Departamento</b>	Matemática Aplicada		



## 1. Situación / Sentido de la Asignatura

### 1.1 Contextualización

Esta asignatura trata sobre los conceptos y técnicas relacionados con las estructuras discretas y sus aplicaciones, tanto en las Matemáticas como en diversas áreas de la Informática, desde la programación y las estructuras de datos hasta la seguridad o el diseño de redes.

### 1.2 Relación con otras materias

No es prerequisite de ninguna otra asignatura, pero sus conceptos matemáticos estarán presentes en múltiples asignaturas de la titulación. Por este motivo, se imparte en el primer cuatrimestre del primer curso de la titulación.

### 1.3 Prerrequisitos

Ninguno.

## 2. Competencias

### 2.1 Generales

- G01** : Conocimientos generales básicos.
- G03** : Capacidad de análisis y síntesis.
- G05** : Comunicación oral y escrita en la lengua propia.
- G06** : Conocimiento de una segunda lengua (Inglés).
- G07** : Habilidades básicas en el manejo del ordenador.
- G09** : Resolución de problemas.
- G16** : Capacidad de aplicar los conocimientos en la práctica.
- G18** : Capacidad de aprender.

### 2.2 Específicas

- E01** : Capacidad para la resolución de los problemas matemáticos que puedan plantearse en la ingeniería. Aptitud para aplicar los conocimientos sobre: álgebra lineal; cálculo diferencial e integral; métodos numéricos; algorítmica numérica; estadística y optimización.
- E02** : Comprensión y dominio de los conceptos básicos de matemática discreta, lógica, algorítmica y complejidad computacional, y su aplicación para el tratamiento automático de la información por medio de sistemas computacionales y para la resolución de problemas propios de la ingeniería.
- E03** : Conocimientos básicos sobre el uso y programación de los ordenadores, sistemas operativos, bases de datos y programas informáticos con aplicación en ingeniería.
- E11** : Conocimiento y aplicación de los procedimientos algorítmicos básicos de las tecnologías informáticas para diseñar soluciones a problemas, analizando la idoneidad y complejidad de los algoritmos propuestos.



### 3. Objetivos

- Plantear en lenguaje matemático y resolver problemas relacionados con la matemática discreta y sus aplicaciones.
- Describir algorítmicamente la resolución de problemas de matemática discreta, e implementarla eficientemente mediante software matemático.
- Comprender, discutir y expresar (oralmente y por escrito) conceptos y argumentos de tipo lógico matemático relacionados con la matemática discreta.
- Construcción de los modelos discretos necesarios para la resolución de problemas aplicados a la informática y a la empresa.
- Manejar software matemático en aplicaciones prácticas, con un énfasis especial en la interpretación de resultados y la escritura de informes.
- Comprender la interrelación de la matemática discreta con otras materias de la titulación.

### 4. Tabla de dedicación del estudiante a la asignatura

ACTIVIDADES PRESENCIALES	HORAS	ACTIVIDADES NO PRESENCIALES	HORAS
Clases teórico-prácticas (T/M)	30	Estudio y trabajo autónomo individual	90
Clases prácticas de aula (A)	15	Estudio y trabajo autónomo grupal	
Laboratorios (L)	13		
Prácticas externas, clínicas o de campo			
Seminarios (S)			
Tutorías grupales (TG)			
Evaluación	2		
<b>Total presencial</b>	<b>60</b>	<b>Total no presencial</b>	<b>90</b>



## 5. Bloques temáticos

### Bloque 1: Aritmética

Carga de trabajo en créditos ECTS: 

1,2
-----

#### a. Contextualización y justificación

Este tema sirve de repaso de conceptos ya aprendidos en la Educación Secundaria y de introducción a los métodos de la asignatura, a la vez que se aprenden conceptos y métodos que se aplican en asignaturas posteriores (aritmética del ordenador, seguridad informática, etc), y se aprenden ejemplos básicos de algoritmos como introducción teórica a la programación.

#### b. Objetivos de aprendizaje

- Conocer las propiedades fundamentales de los números enteros.
- Comprender y realizar los algoritmos fundamentales relacionados con los números enteros: cambios de base, factorización, algoritmo de Euclides extendido e inversión modular, resolución de congruencias lineales y teorema chino de los restos, exponenciación modular.

#### c. Contenidos

1. División entera: algoritmo de enumeración.
2. Divisibilidad: números primos y factorización.
3. Algoritmo de Euclides y teorema de Bézout.
4. Introducción a las estructuras algebraicas.
5. Aritmética modular: inverso modular y exponenciación modular.
6. Resolución de congruencias lineales: teorema chino de los restos.

#### d. Métodos docentes

1. Lección magistral: exposición de la teoría (4 horas).
2. Prácticas en aula: resolución de problemas (2 horas).
3. Prácticas con ordenador con **python** (6 horas).
4. Estudio autónomo por parte del alumno, incluyendo realización de problemas, consulta bibliográfica, realización de prácticas y preparación de pruebas de evaluación (mínimo 18 horas).

#### e. Plan de trabajo

Alternar sesiones teóricas con prácticas y clases de problemas.

#### f. Evaluación

Realización de un examen de carácter teórico-práctico y una práctica de ordenador.



### **g. Bibliografía básica**

---

**K. H. Rosen:** *Matemática Discreta y sus Aplicaciones*, McGraw-Hill (2010).

**M. Lutz:** *Python Pocket Reference*, O'Reilly (2014).

### **h. Bibliografía complementaria**

---

**R. P. Grimaldi:** *Matemática Discreta y Combinatoria*, Addison Wesley (2003).

**W. K. Grassmann, J. P. Tremblay:** *Matemática Discreta y Lógica*, Prentice-Hall (2000).

**R. Johnsonbaugh:** *Matemáticas Discretas*, Prentice-Hall (2005).

**J. B. Fraleigh:** *A First Course in Abstract Algebra*, Pearson (2002).

**M. Lutz:** *Learning Python*, O'Reilly (2014).

### **i. Recursos necesarios**

---

Aula con pizarra y ordenador con proyector, aula informática, biblioteca, sala de estudio, y despacho o seminario para tutorías.





## Bloque 2: Lógica matemática

Carga de trabajo en créditos ECTS: 1,2

### a. Contextualización y justificación

La lógica es un lenguaje básico tanto en las matemáticas como en el desarrollo de programas y sistemas informáticos. Se hará especial hincapié en la lógica necesaria para escribir programas y algoritmos.

### b. Objetivos de aprendizaje

- Conocer los fundamentos básicos de la lógica proposicional y la lógica de predicados.
- Saber hacer demostraciones por inducción, así como definiciones y programas recursivos.
- Saber detectar errores típicos en demostraciones matemáticas.
- Aplicar los conceptos de lógica para realizar y analizar programas y algoritmos sencillos.

### c. Contenidos

1. Lógica de proposiciones: equivalencia e implicación lógicas.
2. Lógica de predicados: variables y cuantificadores.
3. Demostración lógica: inducción matemática.
4. Definiciones recurrentes y programación recursiva.

### d. Métodos docentes

1. Lección magistral: exposición de la teoría (6 horas).
2. Prácticas en aula: resolución de problemas (3 horas).
3. Prácticas con ordenador con **python** (3 horas).
4. Estudio autónomo por parte del alumno, incluyendo realización de problemas, consulta bibliográfica, realización de prácticas y preparación de pruebas de evaluación (mínimo 18 horas).

### e. Plan de trabajo

- Alternar sesiones teóricas con prácticas y clases de problemas.

### f. Evaluación

Realización de un examen de carácter teórico-práctico y una práctica de ordenador.

### g. Bibliografía básica

**K. H. Rosen:** *Matemática Discreta y sus Aplicaciones*, McGraw-Hill (2010).

**M. Lutz:** *Python Pocket Reference*, O'Reilly (2014).

### h. Bibliografía complementaria

**R. P. Grimaldi:** *Matemática Discreta y Combinatoria*, Addison Wesley (2003).

**W. K. Grassmann, J. P. Tremblay:** *Matemática Discreta y Lógica*, Prentice-Hall (2000).



**R. Johnsonbaugh:** *Matemáticas Discretas*, Prentice-Hall (2005).

**M. Lutz:** *Learning Python*, O'Reilly (2014).

### **i. Recursos necesarios**

---

Aula con pizarra y ordenador con proyector, aula de informática, biblioteca, sala de estudio, y despacho o seminario para tutorías.





### Bloque 3: Teoría de conjuntos

Carga de trabajo en créditos ECTS:

#### a. Contextualización y justificación

La teoría de conjuntos es, asimismo, un lenguaje fundamental para formalizar tanto conceptos matemáticos como estructuras de datos en programación.

#### b. Objetivos de aprendizaje

- Manejar mediante ejemplos prácticos la terminología de conjuntos y funciones.
- Saber realizar las operaciones asociadas con conjuntos y funciones.
- Saber relacionar ejemplos reales de programación con modelos en términos de conjuntos y funciones, e interpretar las operaciones asociadas y la terminología estudiada dentro de dicho contexto.
- Conocer la terminología de las relaciones, junto con sus propiedades y operaciones fundamentales.

#### c. Contenidos

1. Conjuntos: definiciones básicas, conjunto potencia y operaciones Booleanas.
2. Producto cartesiano: funciones y sus propiedades.
3. Composición de funciones: función inversa.
4. Relaciones binarias: propiedades y operaciones fundamentales.
5. Relaciones de equivalencia: particiones y conjunto cociente.
6. Relaciones de orden: diagramas de Hasse, ordenación topológica y retículos.

#### d. Métodos docentes

1. Lección magistral: exposición de la teoría (9 horas).
2. Prácticas en aula: resolución de problemas (5 horas).
3. Prácticas con ordenador con **python** (2 horas).
4. Estudio autónomo por parte del alumno, incluyendo realización de problemas, consulta bibliográfica, realización de prácticas y preparación de pruebas de evaluación (mínimo 24 horas).

#### e. Plan de trabajo

- Alternar sesiones teóricas con prácticas y clases de problemas.

#### f. Evaluación

Realización de un examen de carácter teórico-práctico y una práctica de ordenador.

#### g. Bibliografía básica

**K. H. Rosen:** *Matemática Discreta y sus Aplicaciones*, McGraw-Hill (2010).

**M. Lutz:** *Python Pocket Reference*, O'Reilly (2014).

#### h. Bibliografía complementaria

**R. P. Grimaldi:** *Matemática Discreta y Combinatoria*, Addison Wesley (2003).





**W. K. Grassmann, J. P. Tremblay:** *Matemática Discreta y Lógica*, Prentice-Hall (2000).

**R. Johnsonbaugh:** *Matemáticas Discretas*, Prentice-Hall (2005).

**M. Lutz:** *Learning Python*, O'Reilly (2014).

### **i. Recursos necesarios**

---

Aula con pizarra y ordenador con proyector, aula de informática, biblioteca, sala de estudio, y despacho o seminario para tutorías.





## Bloque 4: Combinatoria

Carga de trabajo en créditos ECTS:

### a. Contextualización y justificación

Este tema proporciona las herramientas básicas para contar y enumerar situaciones que se dan en casos prácticos, incluyendo el conteo de operaciones que se realizan en un algoritmo (complejidad computacional).

### b. Objetivos de aprendizaje

- Saber plantear un problema combinatorio en función de los problemas elementales (permutaciones, combinaciones, etc) y los principios básicos de la Combinatoria.
- Utilizar los números combinatorios para contar elementos de un conjunto finito.
- Saber resolver los tipos básicos de relaciones de recurrencia.
- Conocer las aplicaciones básicas de la combinatoria en el campo de la computación, como es el caso de estimar la complejidad de un algoritmo, tanto iterativo como recursivo.

### c. Contenidos

1. Principios elementales de la combinatoria: principios de la suma y del producto, principios del palomar y de las casillas.
2. Números combinatorios: variaciones, permutaciones y combinaciones, con y sin repetición.
3. Principio de inclusión-exclusión: generalizaciones.

### d. Métodos docentes

1. Lección magistral: exposición de la teoría (4 horas).
2. Prácticas en aula: resolución de problemas (3 horas).
3. Prácticas con ordenador con **python** (1 hora).
4. Estudio autónomo por parte del alumno, incluyendo realización de problemas, consulta bibliográfica, realización de prácticas y preparación de pruebas de evaluación (mínimo 18 horas).

### e. Plan de trabajo

- Alternar sesiones teóricas con prácticas y clases de problemas.

### f. Evaluación

Realización de un examen de carácter teórico-práctico.

### g. Bibliografía básica

**K. H. Rosen:** *Matemática Discreta y sus Aplicaciones*, McGraw-Hill (2010).

**M. Lutz:** *Python Pocket Reference*, O'Reilly (2014).

### h. Bibliografía complementaria



**R. P. Grimaldi:** *Matemática Discreta y Combinatoria*, Addison Wesley (2003).

**W. K. Grassmann, J. P. Tremblay:** *Matemática Discreta y Lógica*, Prentice-Hall (2000).

**R. Johnsonbaugh:** *Matemáticas Discretas*, Prentice-Hall (2005).

**M. Lutz:** *Learning Python*, O'Reilly (2014).

### **i. Recursos necesarios**

---

Aula con pizarra y ordenador con proyector, biblioteca, sala de estudio, y despacho o seminario para tutorías.





## Bloque 5: Grafos

Carga de trabajo en créditos ECTS:

### a. Contextualización y justificación

Los grafos y árboles son estructuras de datos muy útiles en Informática para modelar diversos procesos y sistemas (redes, flujo de datos de un programa, fases de un proyecto, etc).

### b. Objetivos de aprendizaje

- Aprender a través de ejemplos reales la terminología de grafos, junto con sus propiedades fundamentales y ejemplos característicos.
- Conocer ejemplos del campo de la computación en donde se utilizan los grafos y los árboles como modelo y técnica de resolución de problemas.
- Relacionar los grafos y los árboles con estructuras de datos, algoritmos y problemas de combinatoria.
- Conocer de forma introductoria las diferentes técnicas transversales para árboles y grafos, junto con sus aplicaciones prácticas.

### c. Contenidos

1. Generalidades sobre la teoría de grafos: modelos con grafos y aplicaciones, representación de grafos, grados de vértices, grafos dirigidos y no dirigidos.
2. Conexión en grafos: recorridos de aristas y vértices.
3. Grafos planos, coloreado de coloreados y grafos ponderados.
4. Generalidades sobre árboles.
5. Árboles dirigidos y árboles con raíz: árboles binarios.
6. Árboles de expansión y técnicas transversales.

### d. Métodos docentes

1. Lección magistral: exposición de la teoría (7 horas).
2. Prácticas en aula: resolución de problemas (2 horas).
3. Prácticas con ordenador con **python** (1 hora).
4. Evaluación (2 horas).
5. Estudio autónomo por parte del alumno, incluyendo realización de problemas, consulta bibliográfica, realización de prácticas y preparación de pruebas de evaluación (mínimo 18 horas).

### e. Plan de trabajo

- Alternar sesiones teóricas con prácticas y clases de problemas.

### f. Evaluación

Realización de un examen de carácter teórico-práctico.

### g. Bibliografía básica



**K. H. Rosen:** *Matemática Discreta y sus Aplicaciones*, McGraw-Hill (2010).

**M. Lutz:** *Python Pocket Reference*, O'Reilly (2014).

#### **h. Bibliografía complementaria**

---

**R. P. Grimaldi:** *Matemática Discreta y Combinatoria*, Addison Wesley (2003).

**W. K. Grassmann, J. P. Tremblay:** *Matemática Discreta y Lógica*, Prentice-Hall (2000).

**R. Johnsonbaugh:** *Matemáticas Discretas*, Prentice-Hall (2005).

**M. Lutz:** *Learning Python*, O'Reilly (2014).

#### **i. Recursos necesarios**

---

Aula con pizarra y ordenador con proyector, biblioteca, sala de estudio, y despacho o seminario para tutorías.



**6. Temporalización (por bloques temáticos)**

BLOQUE TEMÁTICO	CARGA ECTS	PERIODO PREVISTO DE DESARROLLO
1. Aritmética	1,2	3 semanas
2. Lógica Matemática	1,2	3 semanas
3. Teoría de conjuntos	1,6	4 semanas
4. Combinatoria	0,8	2 semanas
5. Teoría de Grafos	1,2	3 semanas

**7. Tabla resumen de los instrumentos, procedimientos y sistemas de evaluación/calificación**

INSTRUMENTO/PROCEDIMIENTO	PESO EN LA NOTA FINAL	OBSERVACIONES
Realización de examen final escrito de carácter teórico-práctico.	80%	La puntuación obtenida debe ser como mínimo de 4 para hacer la media ponderada con la prueba práctica de ordenador y aprobar la asignatura.
Realización de una prueba práctica de ordenador en Python.	20%	En esta prueba no se exige una nota mínima para hacer la media ponderada y aprobar la asignatura en la evaluación continua.

**CRITERIOS DE CALIFICACIÓN**

- **Convocatoria ordinaria:**
  - Se realizarán las dos pruebas anteriormente descritas.
- **Convocatoria extraordinaria:**
  - Se conservará la nota de la prueba práctica de ordenador, y solo se repetirá el examen final escrito, con los mismos pesos y las mismas observaciones que en la convocatoria ordinaria.

**8. Consideraciones finales**