



1. GUÍA DOCENTE DE AUTÓMATAS Y LENGUAJES

La presente guía docente corresponde a la asignatura Autómatas y Lenguajes (AyL), aprobada para el curso lectivo 2016-2017 en Junta de Centro y publicada en su versión definitiva en la página web de la Escuela Politécnica Superior. La guía docente de AyL aprobada y publicada antes del periodo de matrícula tiene el carácter de contrato con el estudiante.





2. ASIGNATURA

AUTÓMATAS Y LENGUAJES (AyL)

1.1. Código

17838 de la titulación de Grado en Ingeniería Informática

1.2. Materia

Autómatas y Lenguajes

1.3. Tipo

Formación obligatoria

1.4. Nivel

Grado

1.5. Curso

3º Ingeniería Informática, 4º plan conjunto Informática/Matemáticas

1.6. Semestre

1º

1.7. Número de créditos

6 créditos ECTS

1.8. Requisitos previos

Para un buen aprovechamiento del curso, es recomendable haber aprobado las asignaturas *Programación I*, *Programación II* y *Proyecto de programación* de la materia *Programación* del módulo *Programación y Estructura de datos*. De no ser así, el curso



puede seguirse pero requiriendo tal vez un esfuerzo extra lo que puede incidir en el rendimiento del estudiante en otras asignaturas en las que esté matriculado. Su contenido también tiene relación, aunque más marginal, con la materia *Análisis de Algoritmos*, del módulo *Programación y estructuras de datos*, que se imparte como asignatura con el mismo nombre; con *Seminario Taller de software*, del módulo *Seminarios Taller de Informática*; y finalmente con la materia *Estructuras Discretas y Lógica*, del módulo *Fundamentos teóricos de la informática y aplicaciones*.

La asignatura *Autómatas y Lenguajes* se imparte en el primer semestre del tercer curso en el Grado de Informática. Conformar, junto con la asignatura *Proyecto de Autómatas y Lenguajes*, la materia de *Autómatas y Lenguajes (materia 2)* del módulo *Fundamentos teóricos de la informática y aplicaciones*. Ambas asignaturas se imparten en el mismo semestre. Por tanto, es imprescindible el buen aprovechamiento en cada una de ellas para superar con éxito ambas. Esta asignatura también se imparte en el primer semestre del cuarto curso de la titulación conjunta en Informática/Matemáticas

1.9. Requisitos mínimos de asistencia a las sesiones presenciales

Se plantean dos itinerarios, uno con asistencia obligatoria a clase y otro sin ella, los estudiantes deberán optar por uno u otro desde el principio del curso y cumplir con los distintos requisitos de evaluación que conlleva cada uno de los modelos, publicados en la presente guía docente (ver apartado 4).

ITINERARIO CON ASISTENCIA OBLIGATORIA A CLASE

Lo llamaremos a partir de ahora *evaluación presencial*. La asistencia es obligatoria al menos en un 85%.

ITINERARIO SIN ASISTENCIA OBLIGATORIA A CLASE

Lo llamaremos a partir de ahora *evaluación no presencial*. La asistencia es muy recomendable aunque no obligatoria.

1.10. Datos del equipo docente

Dr. Alfonso Ortega de la Puente (coordinador)

Departamento de Ingeniería Informática

Escuela Politécnica Superior

Despacho: B-340

Teléfono: +34 91 497 2279

Correo electrónico: alfonso.ortega@uam.es

Horario de atención al alumnado: Por petición



Dr. Luis Fernando Lago Fernández

Departamento de Ingeniería Informática
 Escuela Politécnica Superior
 Despacho: B-307
 Teléfono: +34 91 497 2211
 Correo electrónico: luis.lago@uam.es
 Horario de atención al alumnado: Por petición

1.11. Objetivos del curso

AyL es una asignatura de carácter mixto teórico-práctico. Constituye una toma de contacto con los dispositivos abstractos de cómputo de diferente potencia expresiva y ofrece simultáneamente su justificación y posibilidades de uso desde el punto de vista de la ingeniería informática en el ámbito del diseño y proceso de los lenguajes de programación, comenzando por los imperativos. Proporciona también al alumno una toma de contacto con las herramientas informáticas a las que puede acceder en la actualidad para facilitar las tareas con las que se va a encontrar en el trabajo en esta área.

Las **competencias** específicas de Computación que se pretenden adquirir con esta asignatura son las siguientes:

CC1. Capacidad para tener un conocimiento profundo de los principios fundamentales y modelos de la computación y saberlos aplicar para interpretar, seleccionar, valorar, modelar, y crear nuevos conceptos, teorías, usos y desarrollos tecnológicos relacionados con la informática.

CC2. Capacidad para conocer los fundamentos teóricos de los lenguajes de programación y las técnicas de procesamiento léxico, sintáctico y semántico asociadas, y saber aplicarlas para la creación, diseño y procesamiento de lenguajes.

Los objetivos que se pretenden alcanzar con esta asignatura son:

OBJETIVOS GENERALES	
G1	Conocer los tipos de lenguajes regulares e independientes del contexto, y sus equivalencias con los autómatas y otros tipos de representaciones.
G2	Conocer las gramáticas de atributos como dispositivo capaz de generar cualquier tipo de lenguaje
G3	Desarrollar procesadores de lenguaje para los que son objeto de la asignatura
G4	Conocer los límites expresivos de cada modelo de cómputo y estar familiarizado con las técnicas formales que permiten determinarlos



OBJETIVOS ESPECÍFICOS POR TEMA	
TEMA 1.- Preliminares de los modelos de cómputo y familias de lenguajes	
1.	Identificar y conocer los objetos matemáticos relacionados con los modelos de cómputo y familias de lenguajes
2.	Conocer la estructura de los conjuntos que forman, las operaciones que se definen en ellos y sus propiedades
TEMA 2.- Autómatas finitos deterministas y no deterministas, expresiones regulares	
3.	Conocer los autómatas finitos (y las diferencias entre deterministas y no deterministas) y las expresiones regulares así como las técnicas básicas para su diseño incluyendo herramientas actuales si procede
4.	Saber analizar de manera informal los lenguajes aceptados por los autómatas finitos y los representados por las expresiones regulares
5.	Saber diseñar autómatas finitos y expresiones regulares que, de manera informal, acepten o representen un lenguaje dado, especialmente aquellos relacionados con las construcciones de los lenguajes de programación
TEMA 3.- Gramáticas Independientes del contexto y autómatas a pila	
6.	Conocer las gramáticas independientes del contexto y los autómatas a pila y sus técnicas básicas de diseño incluyendo herramientas informáticas actuales si procede
7.	Saber analizar de manera informal el lenguaje generado por una gramática independiente del contexto y los reconocidos por los autómatas a pila
8.	Saber diseñar gramáticas independientes del contexto y autómatas a pila que, de manera informal, generen o acepten un lenguaje dado, especialmente aquellos relacionados con las construcciones de los lenguajes de programación
TEMA 4.- Gramáticas de atributos	
9.	Conocer las gramáticas de atributos y las técnicas básicas para su diseño incluyendo herramientas actuales si procede
10.	Saber analizar el proceso de propagación de atributos a partir de gramáticas de atributos dadas sobre los árboles de derivación de las cadenas que generan
11.	Saber diseñar gramáticas de atributos que, de manera informal, generen un lenguaje dado, especialmente aquellos relacionados con las construcciones de los lenguajes de programación
TEMA 5.- Análisis morfológico	
12.	A partir de un lenguaje, y de las características del proceso que se quiera realizar sobre él, saber identificar el subconjunto de ese lenguaje del que se encargará el analizador morfológico
13.	Conocer las técnicas más habituales de diseño de analizadores morfológicos
14.	Utilizar herramientas informáticas actuales para facilitar los objetivos anteriores
TEMA 6.- Análisis sintáctico	



15.	Conocer las diferencias fundamentales entre las técnicas de análisis sintáctico ascendentes y descendentes
16.	A partir de las gramáticas independientes del contexto, saber diseñar analizadores sintácticos ascendentes según las técnicas SLR(1), LR(1) y LALR(1), lo que incluye el diseño de diagramas de estado de análisis LR(0), conjuntos primero y siguiente y símbolos de adelanto
17.	A partir de las gramáticas independientes del contexto, saber diseñar analizadores sintácticos descendentes según la técnica LL(1) a partir de tablas de análisis
18.	Estar familiarizados con la técnica de análisis LL(k)
19.	Conocer el concepto de conflicto y las condiciones por las cuales un lenguaje no es SLR(1), LR(1), LALR(1), LL(1) o LL(k)
20.	Utilizar herramientas informáticas actuales para facilitar los objetivos anteriores
TEMA 7.- Análisis semántico	
21.	Conocer las funciones del análisis semántico en el contexto de los procesadores de lenguaje
22.	Saber construir procesadores de lenguaje (tanto de una pasada como de más) a partir de analizadores sintácticos (tanto ascendentes como descendentes) y las gramáticas de atributos de los lenguajes objeto de estudio
23.	Utilizar herramientas informáticas actuales para facilitar los objetivos anteriores
TEMA 8.- Conjuntos regulares	
24.	Saber minimizar autómatas finitos
25.	Saber obtener autómatas finitos deterministas equivalentes a no deterministas
26.	Estar familiarizado con los algoritmos que permiten obtener expresiones regulares equivalentes a autómatas finitos y viceversa
27.	Estar familiarizado con el lema de bombeo como herramienta para demostrar que un lenguaje predeterminado no es regular
TEMA 9.- Conjuntos independientes del contexto	
28.	Estar familiarizado con los algoritmos que permiten obtener gramáticas independientes del contexto equivalentes a autómatas a pila y viceversa
29.	Estar familiarizado con el lema de anidamiento doble como herramienta para demostrar que un lenguaje predeterminado no es independiente del contexto



1.12. Contenidos del programa

Programa Sintético

Unidad 1: modelos de cómputo y familias de lenguajes.

Tema 1: Preliminares

Tema 2: Autómatas finitos deterministas y no deterministas, expresiones regulares

Tema 3: Gramáticas Independientes del contexto y autómatas a pila

Tema 4: Gramáticas de atributos

Unidad 2: procesadores de lenguaje

Tema 5: Análisis morfológico

Tema 6: Análisis sintáctico

Tema 7: Análisis semántico

Unidad 3: equivalencias y caracterización

Tema 8: Conjuntos regulares

Tema 9: Conjuntos independientes del contexto

Programa Detallado

1. Unidad 1: modelos de cómputo y familias de lenguajes.

1.1. Tema 1: Preliminares

1.2. Tema 2: Autómatas finitos deterministas y no deterministas, expresiones regulares:

- Introducción a los autómatas finitos deterministas y no deterministas



- Introducción a las expresiones regulares
- 1.3. Tema 3: Gramáticas Independientes del contexto y autómatas a pila:
 - Introducción a las gramáticas independientes del contexto
 - Introducción a los autómatas a pila
- 1.4. Tema 4: Gramáticas de atributos
 - Introducción a las gramáticas de atributos
 - Programación de gramáticas de atributos
- 2. **Unidad 2: procesadores de lenguaje**
 - 2.1. Tema 5: Análisis morfológico
 - Diseño de analizadores morfológicos
 - Herramientas informáticas de ayuda al diseño de analizadores morfológicos
 - 2.2. Tema 6: Análisis sintáctico:
 - Introducción al análisis sintáctico ascendente y descendente
 - Análisis sintáctico ascendente SLR(1), LR(1) y LALR(1)
 - Análisis sintáctico descendente LL(1) y LL(k)
 - Herramientas informáticas de ayuda al diseño de analizadores sintácticos ascendentes y descendentes
 - 2.3. Tema 7: Análisis semántico
 - Introducción al análisis semántico
 - Herramientas informáticas de ayuda al diseño de compiladores completos
- 3. **Unidad 3: equivalencias y caracterización**
 - 3.1. Tema 8: Conjuntos regulares
 - Equivalencia y minimización de autómatas finitos y expresiones regulares
 - Lema de bombeo
 - 3.2. Tema 9: Conjuntos independientes del contexto
 - Equivalencias entre autómatas a pila y gramáticas independientes del contexto
 - Lema de anidamiento doble

1.13. Referencias de consulta

Unidades 1 y 3:

1. Alfonseca, M., de la Cruz, M., Ortega, A., Pulido, E. Compiladores e intérpretes: teoría y práctica, Pearson, 2006.
2. Alfonseca, M., Alfonseca, E., Roberto Moriyon, R. Teoría de Autómatas y Lenguajes Formales, McGraw Hill, 2007. ISBN: 978-84-481-5637-4.
3. Hopcroft, J.E., Motwani, R. and Ullman, J.D.: Introducción a la Teoría de Autómatas, Lenguajes y Computación, Ed. Pearson Education, 2002.



4. Alfonseca, M., Sancho, J., Martínez Orga, M.A., Teoría de Lenguajes, Gramáticas y Autómatas. Universidad y Cultura, Madrid, 1987.
5. Linz, P., An Introduction to Formal Languages and Automata. Jones and Bartlett Publishers, 2001.

Unidades 1 y 2:

1. Alfonseca, M., de la Cruz, M., Ortega, A., Pulido, E. Compiladores e intérpretes: teoría y práctica, Pearson, 2006.
2. Aho, A.V., Sethi, R., Ullman, J.D., Compilers: Principles, Techniques and Tools. Addison-Wesley Publishing Company, Reading, MA, 1986. Traducción español, Compiladores: principios, técnicas y herramientas, Pearson Education, 1990-98.
3. Grune, D., Modern Compiler Design, Wiley, 2000.
4. Marcotty, M., Ledgard, H.F., Bochmann, G.V., A Sampler of Formal Definitions. Computing Surveys, 8:2, pp. 191-276, June 1976.

Nota: no se recomienda a los estudiantes comprar ningún libro hasta no haber comparado su contenido con el programa y revisado previamente en la biblioteca.

2. Métodos docentes

La metodología utilizada en el desarrollo de la actividad docente incluye los siguientes tipos de actividades:

*Clases de teoría:

Actividad del profesor

Planteamiento de dinámicas y actividades para capacitar al alumno en el conocimiento de los diferentes aspectos teóricos y la solución de situaciones prácticas (por ejemplo clases expositivas y realización de problemas y ejercicios teórico-prácticos). Se basará en el uso de los recursos didácticos accesibles principalmente desde el aula de teoría.

Actividad del estudiante:

Actividad presencial: Participación activa en las dinámicas planteadas y superación de los objetivos determinados para las mismas.

Actividad no presencial: estudio del material y desarrollo de las tareas necesarias para completar las actividades

*Prácticas:

Actividad del profesor:



Al principio de las sesiones programadas para cada práctica, el profesor explicará las herramientas que se utilizarán en cada ejercicio así como los objetivos de las sesiones asignadas a cada práctica. Contestará las dudas de los alumnos durante la realización de los ejercicios y supervisará las actividades mediante las que se evaluará cada práctica y el examen final de prácticas.

Actividad del estudiante:

Actividad presencial: Realización en el laboratorio los ejercicios y actividades propuestos por su profesor.

Actividad no presencial: Avanzar y anticipar el trabajo de las prácticas durante las semanas que incluyan las sesiones programadas para ellas si él mismo o su profesor lo consideran oportuno.

3. Tiempo de trabajo del estudiante

		Nº de horas	Porcentaje
Presencial	Clases teóricas	44 h (29.3%)	81 h (54%)
	Clases prácticas	26 h (17.3%)	
	Realización y entrega del material evaluable	11h = 5 (3 control+ 2 fin práct.) + 6 h (finales) (7.3%)	
No presencial	Estudio semanal regulado	23 h (2x9+1x5) (15.3%)	69 h (46%)
	Realización de actividades prácticas	14 h (1x14) 1h por sesión (9.3%)	
	Preparación y exámenes finales (conv. ord.)	13 h (8.7%)	
	Preparación y exámenes finales (conv. extra.)	19 h (12.7%)	
Carga total de horas de trabajo: 25 horas x 6 ECTS		150 h	





Asignatura: Autómatas y Lenguajes
Código: 17838
Centro: Escuela Politécnica Superior
Titulación: Grado en Ingeniería Informática
Nivel: Grado
Tipo: Obligatoria
Nº de créditos: 6



4. Métodos de evaluación y porcentaje en la calificación final

Nota final de la asignatura

La asignatura tiene dos partes

- Bloque de teoría
- Bloque de prácticas

Cada parte recibirá una calificación entre 0 y 10 según el cálculo que se especifica a continuación:

75% nota bloque teoría + 25% nota bloque prácticas

Para aprobar la asignatura es obligatorio obtener una nota mayor o igual a 5 puntos, tanto en la parte de teoría como en las prácticas. En caso contrario, la nota final en actas será

$0,25 * \min(5, \text{nota bloque prácticas}) + 0,75 * \min(5, \text{nota bloque teoría})$

La nota de teoría y de prácticas se conserva sólo para la convocatoria extraordinaria del mismo curso académico.

ATENCIÓN: Ante cualquier copia descubierta que se haya realizado a lo largo del curso, tanto en cualquiera de las actividades de teoría desarrolladas, como en cualquiera de los apartados de las prácticas, se aplicará el tratamiento previsto por el departamento y la universidad.

Bloque de teoría

PARA LA EVALUACIÓN PRESENCIAL

Las diferentes partes que serán tenidas en cuenta para la evaluación del bloque de teoría son las siguientes:

1) Participación en las actividades de clase:

- Los alumnos deben participar adecuadamente en las actividades planteadas en cada sesión por el profesor; que podrán incluir pruebas escritas, entrega de diferente material, etc..
- Aquellos alumnos que a juicio de su profesor no hayan alcanzado un mínimo en este aspecto podrán pasar a evaluación no presencial.



- El profesorado evaluará todas esas actividades y las calificará adecuadamente obteniendo una nota de cada una de las siguientes partes cuyo peso en la calificación final se especifica a continuación. Esa calificación tendrá carácter liberatorio pero podrá ser mejorada por el alumno en el examen final en las condiciones que se especifican más adelante. Por defecto, al margen de que se puedan realizar más actividades, se considera la realización de los exámenes parciales liberatorios que se describen a continuación aunque estos exámenes podrán ser sustituidos, si el profesorado lo estima oportuno, por otras actividades o dinámicas que permitan evaluar las competencias del estudiante en la materia.

2) Exámenes parciales liberatorios que agrupan las siguientes partes, con el peso de cada parte mostrado entre paréntesis:

- Modelos regulares (12.5%)
- Modelos independientes del contexto (12.5%)
- Gramáticas de atributos/análisis semántico (25%)
- Análisis sintáctico ascendente (16.25%)
- Análisis sintáctico descendente (8.75%)
- Caracterización y equivalencia de modelos (25%)

3) Examen final que cubrirá las mismas partes anteriores con la misma ponderación. Los estudiantes de evaluación continua no tienen la obligación de realizar el examen final, si bien pueden hacerlo para mejorar la nota obtenida en los controles liberatorios. La nota con la que se evaluará cada una de las partes calificables será la mayor entre las obtenidas en los controles y el examen final.

PARA LA EVALUACIÓN NO PRESENCIAL

Sólo se tendrá en cuenta el examen final.

Cada parte del examen final será calificada entre 0 y 10. El cálculo final de la nota se obtendrá mediante la aplicación de la misma ponderación que en EVALUACIÓN PRESENCIAL

Bloque de prácticas.

PARA LA EVALUACIÓN PRESENCIAL

Como resultado de la realización de la evaluación de las actividades asociadas con cada práctica, cada alumno tendrá una calificación correspondiente a la misma.



La nota final del laboratorio se obtendrá realizando la media aritmética de dichas notas. Los que no superen el 5 en la media se considerarán suspensos en la evaluación presencial y deberán asistir al examen final.

El último día de clase se celebrará el examen final de las prácticas. Sólo los alumnos de evaluación no presencial y aquellos que hayan suspendido las prácticas presenciales tienen la obligación de presentarse a él.

PARA LA EVALUACIÓN NO PRESENCIAL

Sólo se tendrá en cuenta el examen final que será calificado entre 0 y 10 y dará directamente la nota de prácticas del alumno.

Sobre la convocatoria extraordinaria

En la parte de teoría la nota de la convocatoria extraordinaria se obtendrá en todos los casos mediante un examen final.

En la parte de prácticas los profesores especificarán un trabajo que el alumno debe aportar. La calificación de este trabajo será la nota de la convocatoria extraordinaria. Los detalles concretos de estos trabajos serán especificados tras la finalización de la convocatoria ordinaria pues dependen del número de alumnos que puedan acudir a la convocatoria extraordinaria.

El promedio entre la nota de teoría y prácticas de la convocatoria extraordinaria será el mismo que el utilizado en la ordinaria.

Sobre calificación “no evaluado” en evaluación presencial

En evaluación presencial, el número mínimo de calificaciones parciales que el estudiante ha de tener para recibir una calificación numérica, es dos tercios del número máximo de calificaciones parciales posibles. Por debajo de este mínimo, el estudiante recibirá la calificación "No evaluado" que **implica que la convocatoria es consumida**.

Este número de calificaciones por debajo del cual se considerará como no evaluado será el siguiente

- Teoría, 4 (consideramos las calificaciones de los 4 controles y el final, un total de 5 posibles)





Asignatura: Autómatas y Lenguajes
Código: 17838
Centro: Escuela Politécnica Superior
Titulación: Grado en Ingeniería Informática
Nivel: Grado
Tipo: Obligatoria
Nº de créditos: 6

- Prácticas, 2/3 del número de prácticas propuestas.



5. Cronograma

Semana	Contenido	Horas presenciales	Horas no presenciales
1	- Presentación y motivación de la asignatura, descripción del programa, normativa y los métodos de evaluación, descripción del material accesible en la plataforma Moodle. - Unidad 1: modelos de cómputo y familias de lenguajes. <ul style="list-style-type: none"> • Tema 1: preliminares • Tema 2: Gramáticas Independientes del contexto y autómatas a pila: presentación 	3	2
		2	1
2	- Unidad 1: modelos de cómputo y lenguajes. <ul style="list-style-type: none"> • Tema 2: Gramáticas Independientes del contexto y autómatas a pila: problemas • Tema 3: Autómatas finitos deterministas y no deterministas, expresiones regulares: <ul style="list-style-type: none"> o Presentación o Problemas - Práctica	3	2
		2	1
3	- Unidad 1: modelos de cómputo y lenguajes. <ul style="list-style-type: none"> • Tema 3: Autómatas finitos deterministas y no deterministas, expresiones regulares: <ul style="list-style-type: none"> o Presentación o Problemas - Práctica	3	2
		2	1
4	- Unidad 1: modelos de cómputo y lenguajes. <ul style="list-style-type: none"> • Tema 4: Gramáticas de atributos <ul style="list-style-type: none"> o Presentación del modelo o Programación de gramáticas de atributos o Problemas - Práctica	3	2
		2	1
5	- Unidad 1: modelos de cómputo y lenguajes. <ul style="list-style-type: none"> • Tema 4: Gramáticas de atributos <ul style="list-style-type: none"> o Problemas - Unidad 2: procesadores de lenguaje	3	



	<ul style="list-style-type: none"> • Tema 5: Análisis morfológico <ul style="list-style-type: none"> o Presentación y problemas - Práctica		2 (total)
		2	
6	- Unidad 2: procesadores de lenguaje <ul style="list-style-type: none"> • Tema 6: Análisis sintáctico (ascendente): <ul style="list-style-type: none"> o Introducción al análisis sintáctico - Práctica	3	2
		2	1
7	- Unidad 2: procesadores de lenguaje <ul style="list-style-type: none"> • Tema 6: Análisis sintáctico ascendente: <ul style="list-style-type: none"> o SLR(1): presentación y problemas o LR(1): presentación - Práctica	3	2
		2	1
8	- Unidad 2: procesadores de lenguaje <ul style="list-style-type: none"> • Tema 6: Análisis sintáctico ascendente: <ul style="list-style-type: none"> o LR(1) - LALR(1) - Práctica	3	
		2	2 (total)
9	- Unidad 2: procesadores de lenguaje <ul style="list-style-type: none"> • Tema 6: Análisis sintáctico ascendente: <ul style="list-style-type: none"> o Problemas • Tema 7: Análisis sintáctico descendente: <ul style="list-style-type: none"> o LL(1) - LL(k) - Práctica	3	2
		2	1
10	- Unidad 2: procesadores de lenguaje <ul style="list-style-type: none"> • Tema 7: Análisis sintáctico descendente: <ul style="list-style-type: none"> o LL(k) o Problemas - Unidad 2: procesadores de lenguaje <ul style="list-style-type: none"> • Tema 8: Análisis semántico: - Práctica	3	
		2	2 (total)
11	- Unidad 2: procesadores de lenguaje <ul style="list-style-type: none"> • Tema 8: Análisis semántico: <ul style="list-style-type: none"> o Problemas - Práctica	3	
		2	2 (total)
12	- Unidad 3: equivalencias y caracterización <ul style="list-style-type: none"> • Tema 9: Conjuntos regulares <ul style="list-style-type: none"> o Minimización de autómatas finitos o Equivalencias (autómatas finitos deterministas y no deterministas y expresiones regulares) - Práctica	3	2
		2	1



13	<ul style="list-style-type: none"> - Unidad 3: equivalencias y caracterización <ul style="list-style-type: none"> • Tema 9: Conjuntos regulares <ul style="list-style-type: none"> o Lema de bombeo - Práctica 	3	
		2	2 (total)
14	<ul style="list-style-type: none"> - Unidad 3: equivalencias y caracterización <ul style="list-style-type: none"> • Tema 10: Conjuntos independientes del contexto <ul style="list-style-type: none"> o Equivalencias o Lema de anidamiento doble - Examen de prácticas 	3	2
		2	1
15-16	- Examen final de teoría (ordinaria)	3	13
17-19	- Examen final (extraordinaria)	3	19

