UNIVERSIDAD DE CASTILLA - LA MANCHA

GUÍA DOCENTE

1. DATOS GENERALES

Asignatura: PROGRAMACIÓN DECLARATIVA
Código: 42343
Tipología: OPTATIVA
Créditos ECTS: 6
Grado: 406 - GRADO EN INGENIERÍA INFORMÁTICA (AB)_20
Curso académico: 2022-23
Centro: 604 - E.S. DE INGENIERIA INFORMATICA ALBACETE
Grupo(s): 15
Curso: 3
Duración: C2
Lengua principal de impartición: Español
Segunda lengua:
Uso docente de otras lenguas:
English Friendly: N
Página web:
Bilingüe: N

Profesor: **JUAN ANTONIO GUERRERO ABENZA** - Grupo(s): **15**

Edificio/Despacho	Departamento	Teléfono	Correo electrónico	Horario de tutoría
Infante D. Juan Manuel/1A4	SISTEMAS INFORMÁTICOS	926053299	juan.guerrero@uclm.es	Estará disponible al inicio del curso en http://www.esiiab.uclm.es

Profesor: **GINES MORENO VALVERDE** - Grupo(s): **15**

Edificio/Despacho	Departamento	Teléfono	Correo electrónico	Horario de tutoría
Infante D. Juan Manuel/1.C.9	SISTEMAS INFORMÁTICOS	2471	gines.moreno@uclm.es	Estará disponible al inicio del curso en http://www.esiiab.uclm.es

2. REQUISITOS PREVIOS

Esta asignatura se imparte en el tercer curso de la titulación, lo que permite asumir que los alumnos que la vayan a cursar, ya tendrán el suficiente entrenamiento en otras disciplinas afines que les habrán aportado conocimientos, capacidades y niveles de abstracción suficientes para asimilar convenientemente los contenidos que se imparten en la misma.

En lo que respecta a conocimientos previos, se espera una formación media en:

- Formalismos Matemáticos. Aquí se pone especial énfasis en los conocimientos referentes a lógica matemática y, en menor medida, álgebra, ya que suponen la base formal de los lenguajes lógicos y funcionales que se estudian clásicamente dentro del paradigma declarativo.

- Lenguajes y Técnicas de Programación. El alumno debe estar suficientemente familiarizado con algunos lenguajes de programación de estilo imperativo u orientado objeto, así como con técnicas de programación básicas, que en general no tengan por qué ser totalmente dependientes de un lenguaje de programación concreto para su implementación final. Esto permitirá contrastar de forma más satisfactoria el nuevo paradigma declarativo con otros estilos de programación, resaltando el amplio repertorio de recursos expresivos que aportan a la hora de codificar la solución a un problema minimizando el tamaño de las aplicaciones.

- Algorítmica y Estructuras de Datos. En relación con el apartado anterior, se espera que el alumno conozca las estructuras de datos y los esquemas algorítmicos básicos a la hora de representar y manipular la información de forma conveniente, para que posteriormente pueda simplificar tanto las tareas de implementación y mantenimiento, como el cálculo de coste computacional haciendo uso de tecnología declarativa.

- Teoría de Autómatas y Computación. Puesto que la programación declarativa viene aportando lenguajes de programación con altos niveles de abstracción y sólida cimentación matemática, se espera que el alumno conozca y sepa aplicar los fundamentos de la teoría de lenguajes formales, especificaciones algebraicas y métodos formales de la ingeniería del software.

Más específicamente se espera que el alumno posea las competencias y los conocimientos adquiridos en las asignaturas de Lógica, Fundamentos de Programación I y II, Metodología de la Programación y Estructura de Datos.

3. JUSTIFICACIÓN EN EL PLAN DE ESTUDIOS, RELACIÓN CON OTRAS ASIGNATURAS Y CON LA PROFESIÓN

La programación declarativa se basa en la idea de utilizar un cierto tipo de lógica como lenguaje de programación. Esto incluye tanto la programación lógica (o relacional, que usa un lenguaje clausal para programar y el principio de resolución como mecanismo de ejecución) como la funcional (que usa el lenguaje de las funciones matemáticas y la reducción de expresiones como mecanismo computacional).

La programación declarativa ha encontrado una gran variedad de aplicaciones. Sin ánimo de ser exhaustivos, podemos enumerar algunas de éstas:

Procesamiento del lenguaje natural. Representación del conocimiento. Química y biología molecular. Desarrollo de Sistemas de Producción y Sistemas Expertos. Resolución de Problemas. Metaprogramación. Prototipado de aplicaciones. Bases de Datos Deductivas. Servidores y buceadores de información inteligentes. Web Semántica. Diseño de sistemas VLSI. Herramientas de soporte al desarrollo del software.

Lo que da idea de la relevancia de esta materia y justifica su inclusión en un plan de estudios de ingeniería informática. Más generalmente, la programación declarativa se ha aplicado en todos los campos de la computación simbólica (y por esto también los lenguajes declarativos se denominan a veces, lenguajes de computación simbólica, en contraposición a los lenguajes más tradicionales orientados a la computación numérica), la inteligencia artificial y la informática teórica (e.g., teoría de tipos).

La asignatura de Programación Declarativa se integra en la materia de Tecnología Específica de Computación del plan de estudios y sirve de fundamento a, o se relaciona directamente con, las siguientes asignaturas:

Sistemas basados en el Conocimiento
Diseño de Sistemas Interactivos

4. COMPETENCIAS DE LA TITULACIÓN QUE LA ASIGNATURA CONTRIBUYE A ALCANZAR

Competencias propias de la asignatura

Código	Descripción
CM01	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.
INS01	Capacidad de análisis, síntesis y evaluación.
INS04	Capacidad de resolución de problemas aplicando técnicas de ingeniería.
UCLM03	Correcta comunicación oral y escrita.

5. OBJETIVOS O RESULTADOS DE APRENDIZAJE ESPERADOS

Resultados de aprendizaje propios de la asignatura

Descripción

Conocer y usar los distintos métodos, técnicas y herramientas para el desarrollo de aplicaciones mediante un lenguaje de programación declarativa.

Comprender los fundamentos teóricos que sustentan el paradigma de programación declarativa.

6. TEMARIO

Tema 1: Estilos de programación. Programación declarativa.

Tema 2: Programación lógica.

Tema 3: El lenguaje Prolog.

Tema 4: Técnicas avanzadas de programación en Prolog.

Tema 5: Programación funcional.

Tema 6: Programación funcional avanzada.

Tema 7: Programación simbólica. Lisp.

7. ACTIVIDADES O BLOQUES DE ACTIVIDAD Y METODOLOGÍA

Actividad formativa	Metodología	Competencias relacionadas (para títulos anteriores a RD 822/2021)	ECTS	Horas	Ev	Ob	Descripción
Enseñanza presencial (Teoría) [PRESENCIAL]	Método expositivo/Lección magistral	CM01	0.44	11	S	N	Clase (grupos grandes)
Resolución de problemas o casos [PRESENCIAL]	Resolución de ejercicios y problemas	CM01	0.44	11	S	N	Clase (grupos grandes)
Resolución de problemas o casos [PRESENCIAL]	Prácticas	INS04	1.2	30	S	N	Laboratorio (grupos pequeños)
Pruebas de progreso [PRESENCIAL]	Pruebas de evaluación	INS01	0.24	6	S	N	Controles periódicos en clase. Recuperable en examen ordinario/extraordinario de forma oral/escrita.
Elaboración de memorias de Prácticas [AUTÓNOMA]	Aprendizaje basado en problemas (ABP)	UCLM03	1.2	30	S	N	Tareas entregables (Moodle, etc.) o para corregir en clase o tutorías. Recuperable en examen ordinario/extraordinario de forma oral/escrita.
Estudio o preparación de pruebas [AUTÓNOMA]	Autoaprendizaje	INS04	2.4	60	S	N	Repaso y ampliación de temas. Resolución de ejercicios. Terminar prácticas. Preparación de Controles.
Prueba final [PRESENCIAL]		INS01	0.08	2	S	N	Examen de evaluación ordinario/extraordinario. Examen ordinario recuperable en examen extraordinario de forma oral/escrita.
		Total:	**6**	**150**			
		Créditos totales de trabajo presencial: 2.4					**Horas totales de trabajo presencial: 60**
		Créditos totales de trabajo autónomo: 3.6					**Horas totales de trabajo autónomo: 90**

Ev: Actividad formativa evaluable

Ob: Actividad formativa de superación obligatoria (Será imprescindible su superación tanto en evaluación continua como no continua)

8. CRITERIOS DE EVALUACIÓN Y VALORACIONES

Sistema de evaluación	Evaluacion continua	Evaluación no continua*	Descripción
Pruebas de progreso	70.00%	0.00%	Controles sobre conocimientos adquiridos en clase y laboratorio ([ESC]50%, [LAB]20%)
Valoración de la participación con aprovechamiento en clase	20.00%	0.00%	Participación: Exposición de temas, trabajos y resolución de problemas en clase y laboratorio ([INF]5%, [PRES]15%)
Elaboración de memorias de prácticas	10.00%	10.00%	Trabajos entregables (Moodle, etc.) ([INF]10%), cuya valoración podrá ser de forma presencial, de forma oral o escrita.
Prueba final	0.00%	90.00%	Prueba final sobre el temario completo ([ESC]90%)
Total:	**100.00%**	**100.00%**	

* *En **Evaluación no continua** se deben definir los porcentajes de evaluación según lo dispuesto en el art. 4 del Reglamento de Evaluación del Estudiante de la UCLM, que establece que debe facilitarse a los estudiantes que no puedan asistir regularmente a las actividades formativas presenciales la superación de la asignatura, teniendo derecho (art. 12.2) a ser calificado globalmente, en 2 convocatorias anuales por asignatura, una ordinaria y otra extraordinaria (evaluándose el 100% de las competencias).*

Criterios de evaluación de la convocatoria ordinaria:

Evaluación continua:

No hay examen final. La nota de la convocatoria ordinaria será el resultado de la evaluación continua (Pruebas de progreso + Memorias de prácticas + Participación). Para aprobar no se exige nota mínima en ninguna parte, pero la suma no podrá ser inferior al 50% de la nota máxima alcanzable.

Por defecto, el estudiante será evaluado por evaluación continua. Si desea cambiarse a evaluación no continua, debe indicarlo a través del siguiente enlace https://www.esiiab.uclm.es/alumnos/evaluacion.php antes de la finalización del periodo lectivo del cuatrimestre, siempre y cuando no haya sido evaluado del 50% de la asignatura, tal como establece el reglamento de evaluación del estudiante.

Evaluación no continua:

Se está en la modalidad de evaluación no continua siempre que no se haya participado durante el periodo de impartición de clases en actividades evaluables que supongan en su conjunto al menos el 50% de la evaluación total de la asignatura.

Por defecto, el estudiante será evaluado por evaluación continua. Si desea cambiarse a evaluación no continua, debe indicarlo a través del siguiente enlace https://www.esiiab.uclm.es/alumnos/evaluacion.php antes de la finalización del periodo lectivo del cuatrimestre, siempre y cuando no haya sido evaluado del 50% de la asignatura, tal como establece el reglamento de evaluación del estudiante.

Hay dos pruebas finales exclusivamente para quien no siga la evaluación continua: una sobre el temario completo de la asignatura (90%), y otra sobre las memorias de prácticas (10%). Estas memorias son las mismas que se ofertan en evaluación continua, aunque su evaluación presencial se hace de forma conjunta de forma oral o escrita.

Para aprobar, la suma de notas de la anteriores pruebas no podrá ser inferior al 50% de la nota máxima alcanzable.

Particularidades de la convocatoria extraordinaria:

Como en las pruebas finales de evaluación no continua.

Particularidades de la convocatoria especial de finalización:

Como en las pruebas finales de evaluación no continua y extraordinaria.

9. SECUENCIA DE TRABAJO, CALENDARIO, HITOS IMPORTANTES E INVERSIÓN TEMPORAL

No asignables a temas

Horas	**Suma horas**
Prueba final [PRESENCIAL][]	2

Comentarios generales sobre la planificación: Esta planificación es ORIENTATIVA, pudiendo variar a lo largo del curso en función de las necesidades docentes, festividades, etc. La asignatura se imparte en tres sesiones semanales de 1,5 horas. La planificación semana a semana de la asignatura podrá encontrarse en la plataforma Campus Virtual (Moodle).

Tema 1 (de 7): Estilos de programación. Programación declarativa.

Actividades formativas	**Horas**
Enseñanza presencial (Teoría) [PRESENCIAL][Método expositivo/Lección magistral]	1
Resolución de problemas o casos [PRESENCIAL][Resolución de ejercicios y problemas]	1
Resolución de problemas o casos [PRESENCIAL][Prácticas]	2
Elaboración de memorias de Prácticas [AUTÓNOMA][Aprendizaje basado en problemas (ABP)]	2
Estudio o preparación de pruebas [AUTÓNOMA][Autoaprendizaje]	1

Tema 2 (de 7): Programación lógica.

Actividades formativas	**Horas**
Enseñanza presencial (Teoría) [PRESENCIAL][Método expositivo/Lección magistral]	4
Resolución de problemas o casos [PRESENCIAL][Resolución de ejercicios y problemas]	4
Resolución de problemas o casos [PRESENCIAL][Prácticas]	8
Elaboración de memorias de Prácticas [AUTÓNOMA][Aprendizaje basado en problemas (ABP)]	8
Estudio o preparación de pruebas [AUTÓNOMA][Autoaprendizaje]	16

Tema 3 (de 7): El lenguaje Prolog.

Actividades formativas	**Horas**
Enseñanza presencial (Teoría) [PRESENCIAL][Método expositivo/Lección magistral]	1
Resolución de problemas o casos [PRESENCIAL][Resolución de ejercicios y problemas]	1
Resolución de problemas o casos [PRESENCIAL][Prácticas]	4
Pruebas de progreso [PRESENCIAL][Pruebas de evaluación]	1.5
Elaboración de memorias de Prácticas [AUTÓNOMA][Aprendizaje basado en problemas (ABP)]	3
Estudio o preparación de pruebas [AUTÓNOMA][Autoaprendizaje]	10

Tema 4 (de 7): Técnicas avanzadas de programación en Prolog.

Actividades formativas	**Horas**
Enseñanza presencial (Teoría) [PRESENCIAL][Método expositivo/Lección magistral]	2
Resolución de problemas o casos [PRESENCIAL][Resolución de ejercicios y problemas]	2
Resolución de problemas o casos [PRESENCIAL][Prácticas]	6
Pruebas de progreso [PRESENCIAL][Pruebas de evaluación]	1.5
Elaboración de memorias de Prácticas [AUTÓNOMA][Aprendizaje basado en problemas (ABP)]	4
Estudio o preparación de pruebas [AUTÓNOMA][Autoaprendizaje]	12

Tema 5 (de 7): Programación funcional.

Actividades formativas	**Horas**
Enseñanza presencial (Teoría) [PRESENCIAL][Método expositivo/Lección magistral]	2
Resolución de problemas o casos [PRESENCIAL][Resolución de ejercicios y problemas]	2
Resolución de problemas o casos [PRESENCIAL][Prácticas]	4

Elaboración de memorias de Prácticas [AUTÓNOMA][Aprendizaje basado en problemas (ABP)]	4
Estudio o preparación de pruebas [AUTÓNOMA][Autoaprendizaje]	8

Tema 6 (de 7): Programación funcional avanzada.

Actividades formativas	**Horas**
Resolución de problemas o casos [PRESENCIAL][Prácticas]	2
Pruebas de progreso [PRESENCIAL][Pruebas de evaluación]	1.5
Elaboración de memorias de Prácticas [AUTÓNOMA][Aprendizaje basado en problemas (ABP)]	5
Estudio o preparación de pruebas [AUTÓNOMA][Autoaprendizaje]	3

Tema 7 (de 7): Programación simbólica. Lisp.

Actividades formativas	**Horas**
Enseñanza presencial (Teoría) [PRESENCIAL][Método expositivo/Lección magistral]	1
Resolución de problemas o casos [PRESENCIAL][Resolución de ejercicios y problemas]	1
Resolución de problemas o casos [PRESENCIAL][Prácticas]	4
Pruebas de progreso [PRESENCIAL][Pruebas de evaluación]	1.5
Elaboración de memorias de Prácticas [AUTÓNOMA][Aprendizaje basado en problemas (ABP)]	4
Estudio o preparación de pruebas [AUTÓNOMA][Autoaprendizaje]	10

Actividad global

Actividades formativas	**Suma horas**
Enseñanza presencial (Teoría) [PRESENCIAL][Método expositivo/Lección magistral]	11
Resolución de problemas o casos [PRESENCIAL][Resolución de ejercicios y problemas]	11
Resolución de problemas o casos [PRESENCIAL][Prácticas]	30
Pruebas de progreso [PRESENCIAL][Pruebas de evaluación]	6
Elaboración de memorias de Prácticas [AUTÓNOMA][Aprendizaje basado en problemas (ABP)]	30
Estudio o preparación de pruebas [AUTÓNOMA][Autoaprendizaje]	60
Prueba final [PRESENCIAL][]	2
Total horas:	150

10. BIBLIOGRAFÍA, RECURSOS

Autor/es	Título/Enlace Web	Editorial	Población	ISBN	Año	Descripción
Apt, Krysztof	From logic programming to prolog	Prentice-Hall		0-13-230-368-X	1999	
Bird, Richard	Introducción a la programación funcional con Haskell	Prentice Hall		84-8322-176-4	2000	
Clocksin, W. F.	Programming in PROLOG	Springer		3-540-00678-8	2003	
Covington, Michael A.	Prolog programming in depth	Prentice Hall		0-13-138645-x	1997	
Deransart, Pierre	Prolog : the standard : reference manual	Springer		3-540-59304-7	1996	
Doets, Kees	From logic to logic programming	The MIT Press		0-262-04142-1	1994	
Field, Anthony J.	Functional programming	Addison-Wesley		0-201-19249-7	1989	
Gregory, Steve	Parallel logic programming in PARLOG : the language and its	Addison-Wesley		0-201-19241-1	1987	
Hudak, Paul	The haskell school of expression : learning functional prog	Cambridge University Press		0-521-64338-4	2000	
Julián Iranzo, Pascual	Programación lógica : teoría y práctica	Pearson Prentice Hall		978-84-8322-368-0	2007	
Lalement, Rene	Computation as logic	Masson Prentice Hall		0-13-770009-1	1993	
Lloyd, J.W.	Foundations of logic programming	Springer-Verlag		3-540-18199-7	1993	
Maier, David (1953-)	Computing with Logic : logic programming with Prolog	The Benjamin-Cummins Publishing Company		0-8053-6681-4	1988	
PEYTON JONES, Simon L.	The implementation of functional programming languages	Prentice-Hall		0-13-453333-X	1987	
THOMPSON, Simon	Type theory and functional programming	Wokingham, England [etc.] Addison-Wesley, [1991]		0-201-41667-0		
Thompson, Simon	Haskell the craft of functional programming	Addison-Wesley		0-201-34275-8	1999	
Touretzky, David S.	Common LISP : a gentle introduction to symbolic computation	The benjamin cummings		0-8053-0492-4	1990	
	Razonando con Haskell : un curso sobre programación funciona	Thomson-Paraninfo		84-9732-277-0	2004	
	Sicstus Prolog User's Manual	Swedish Institute of Computer Science			2008	