

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA

INGENIERO EN INFORMÁTICA

ASIGNATURAS DE PRIMER CICLO

TERCER CURSO

PRIMER CUATRIMESTRE

312 - PROGRAMACIÓN DECLARATIVA		
Departamento: LENGUAJES Y CIENCIAS DE LA COMPUTACION	Créditos: 9	Obligatoria
OBJETIVOS		
<p>1. Introducir los principios y técnicas de la programación lógica y funcional, y presentar las diferencias del estilo declarativo con el estilo de programación imperativa.</p> <p>2. Tener una primera toma de contacto con los lenguajes declarativos más representativos (Prolog y Haskell) resolviendo problemas típicos con el uso de tales lenguajes.</p>		
CONTENIDO		
<p>BLOQUE TEMATICO: Nombre Bloque Temático</p> <p>1. Introducción a la Programación Declarativa Primera Parte: Programación Lógica</p> <p>2. Principios de la programación lógica</p> <p style="padding-left: 20px;">2.1 Elementos de la lógica de cláusulas de Horn. Programa lógico</p> <p style="padding-left: 20px;">2.2 Semántica declarativa de un programa lógico</p> <p style="padding-left: 20px;">2.3 Semántica operacional de un programa lógico</p> <p style="padding-left: 20px;">2.4 Resolución. Unificación. Árboles de búsqueda</p> <p>3. El lenguaje lógico Prolog</p> <p style="padding-left: 20px;">3.1 Aspectos básicos del lenguaje</p> <p style="padding-left: 20px;">3.2 Programación elemental con bases de datos</p> <p style="padding-left: 20px;">3.3 Programación recursiva. Modos de uso de un predicado</p> <p style="padding-left: 20px;">3.4 Aplicación: Aritmética Natural</p> <p>4. Predicados extralógicos y módulos</p> <p style="padding-left: 20px;">4.1 Predicados aritméticos predefinidos</p> <p style="padding-left: 20px;">4.2 Predicados de entrada/salida</p> <p style="padding-left: 20px;">4.3 Sistema de módulos de Prolog</p> <p>5. Programación lógica con estructuras</p> <p style="padding-left: 20px;">5.1 Listas</p> <p style="padding-left: 20px;">5.2 Árboles</p> <p style="padding-left: 20px;">5.3 Grafos y otras estructuras</p> <p>6. Control en Prolog</p> <p style="padding-left: 20px;">6.1 Corte</p> <p style="padding-left: 20px;">6.2 Aplicaciones del corte</p> <p style="padding-left: 20px;">6.3 Unificación como mecanismo de control</p> <p>7. Metaprogramación en Prolog</p> <p style="padding-left: 20px;">7.1 Predicados metalógicos</p> <p style="padding-left: 20px;">7.2 Manipulación de estructuras</p> <p>8. Técnicas de programación lógica</p> <p style="padding-left: 20px;">8.1 Programación no determinista</p> <p style="padding-left: 20px;">8.2 Técnicas de búsqueda</p> <p style="padding-left: 20px;">8.3 Programación de segundo orden</p> <p style="padding-left: 20px;">8.4 Programación con información parcial</p> <p>Segunda Parte: Programación Funcional.</p> <p>9. Principios de la programación funcional</p> <p style="padding-left: 20px;">9.1 Introducción al lambda-cálculo</p> <p style="padding-left: 20px;">9.2 Semántica operacional de un programa funcional</p> <p style="padding-left: 20px;">9.3 Programación en un lenguaje funcional</p> <p>10. Elementos del lenguaje funcional Haskell</p> <p style="padding-left: 20px;">10.1 Aspectos básicos del lenguaje</p> <p style="padding-left: 20px;">10.2 El sistema de clases de Haskell</p> <p style="padding-left: 20px;">10.3 El sistema de módulos de Haskell</p> <p style="padding-left: 20px;">10.4 Entrada y salida</p> <p>11. Programación funcional con listas</p> <p style="padding-left: 20px;">11.1 Listas</p> <p style="padding-left: 20px;">11.2 Iteradores sobre listas</p> <p style="padding-left: 20px;">11.3 Plegado de listas</p>		

- 11.4 Problemas combinatorios
- 12. Evaluación perezosa, redes de procesos y listas infinitas
 - 12.1 Evaluación perezosa
 - 12.2 Listas parciales
 - 12.3 Redes de procesos
- 13. Programación con Árboles y Grafos. Aplicaciones.
 - 13.1. Árboles
 - 13.2 Iteradores sobre árboles
 - 13.3 Plegado de árboles
 - 13.4 Grafos
 - 13.5 Aplicaciones
- 14. Aspectos teóricos y herramientas de la programación funcional
 - 14.1 Sistema de tipos e inferencia de tipos
 - 14.2 Aspectos sobre la corrección de programas
 - 14.3 Transformación de programas

BIBLIOGRAFÍA

BIRD, R. *Introducción a la Programación Funcional con Haskell* Prentice-Hall 1999
 RUIZ, B.C., GUTIERREZ, F., GUERRERO, P., GALLARDO, J.E. *Razonando con Haskell. Un Curso sobre Programación Funcional* Ed. Thomson 2004
 STERLING, L., SHAPIRO, E *The Art of Prolog* The MIT Press 1994
 CLOCK SIN, W., MELLISH, C. *Programming in Prolog* Springer-Verlag 1994

METODOLOGÍA DOCENTE

Los dos estilos declarativos podrán ser impartidos simultáneamente, sin perjuicio del correcto desarrollo de los contenidos. Las clases de teoría/práctica generales (o de pizarra) serán en un aula y hará uso cuando sea necesario de técnicas estándares (cañón, retroproyector, pizarra, ejecución simultánea de código a través de portátil/cañón, ¿) El material para las exposiciones (ejercicios, códigos, transparencias, ¿) estará disponible con suficiente antelación en la página web de la asignatura. Las exposiciones realizadas por los alumnos también serán en el aula y harán uso de estas técnicas. También se hará ocasionalmente recogidas de apuntes y encuestas tras alguna exposición por parte de un profesor o grupo de alumnos, e incluso pruebas cortas (controles) con una periodicidad mínima de un mes. Las restantes actividades presenciales se celebrarán en los laboratorios. Al menos dos laboratorios serán necesarios. Entre estas actividades figuran:

- realización tutorizada de ejercicios con ordenador
- entrega de problemas/prácticas en forma personal/tutorizada: Cada alumno, además del soporte correspondiente (papel, fichero con programas, etc.), deberá antes "defender" su trabajo en pocos minutos ante el profesor. Posteriormente (un día o dos después), entregará en tutoría estándar, por correo electrónico, o por internet, los ejercicios que hayan sido modificados tras la entrevista. Los alumnos también deben acostumbrarse a corregir/comentar las soluciones de sus propios compañeros. A tal fin se seleccionaran cada dos semanas algunas.
- realización de ejercicios teórico/prácticos personales con la asistencia y revisión de los profesores.

EVALUACION

La evaluación puede ser continua o por un examen final. La continua consta de la entrega de ejercicios y la realización de los exámenes parciales programados. La final consta únicamente de un examen final. La elección entre un esquema u otro la toma libremente el alumno. Para la realización de los parciales, dependiendo de las disponibilidades de laboratorios, podría utilizarse un ordenador como elemento auxiliar o ayuda. Los exámenes tendrán una componente teórica y otra práctica, en el que se pueda medir la capacitación del alumno para resolver problemas utilizando cada uno de los estilos de programación declarativa. Ambas partes de la asignatura (Programación Funcional y Programación Lógica) deben evaluarse por el mismo esquema. Los exámenes parciales no serán eliminatorios, pero sí acumulativos: la materia de cada parcial será la tratada desde el inicio del curso hasta la clase anterior al examen, si bien se enfatizarán los contenidos de los temas correspondientes al parcial. El número de exámenes parciales es dos, coincidiendo el segundo con el examen ordinario de la asignatura.

TÉCNICAS DOCENTES

Sesiones académicas teóricas:	SI
Sesiones académicas prácticas:	SI
Exposición y debate:	SI
Tutorías especializadas:	SI
Visitas y excursiones:	NO
Controles de lectura obligatorias:	SI
Otros:	
Desarrollo y Justificación:	

NÚMERO DE HORAS DE TRABAJO DEL ALUMNO

Primer Semestre

Actividad	Nº de horas
Clases Teóricas:	30
Clases Prácticas:	20
Exposiciones y Seminarios:	20
Tutorías Especializadas (presenciales o virtuales):	
A) Colectivas:	4
B) Individuales:	4
Realización de Actividades Académicas Dirigidas:	
A) Con presencia del profesor:	4

B) Sin presencia del profesor:	20
Otro Trabajo Personal Autónomo:	
A) Horas de estudio	60
B) Preparación de Trabajo Personal:...	20
C) ?	
Realización de Exámenes:	
A) Examen escrito:	8
B) Exámenes orales (control del Trabajo Personal):	
COMPETENCIAS TRANSVERSALES/GENÉRICAS	
<p>Capacidad de análisis y síntesis: Alto. Capacidad de organización y planificación: Medio. Comunicación oral y escrita en lengua nativa: Medio. Resolución de problemas: Alto. Toma de decisiones: Medio. Trabajo en equipo: Bajo. Trabajo en un equipo de carácter interdisciplinar: Bajo. Aprendizaje autónomo: Medio. Adaptación a nuevas situaciones: Medio. Capacidad de aplicar los conocimientos en la práctica: Alto. Habilidad para trabajar de forma autónoma: Alto. Creatividad: Medio.</p>	
COMPETENCIAS ESPECÍFICAS	
Cognitivas(Saber)	
<p>Entre las competencias cognitivas que el alumno adquirirá podemos destacar: el aprendizaje de un lenguaje de programación funcional (Haskell) y de un lenguaje para la programación lógica (Prolog), el uso de mecanismos propios de estos lenguajes para resolver problemas de programación (variables lógicas, funciones de orden superior,..), aptitud para seleccionar estructuras de datos adecuadas a los problemas a resolver.</p>	
Procedimentales/Instrumentales (Saber hacer)	
<p>Planteamiento de soluciones algorítmicas a problemas concretos: Alto. Participación en la implementación de programas informáticos: Alto. Diseño e implementación de algoritmos: Alto. Identificación y localización de errores: Alto. Capacidad de poner la teoría en conexión con la práctica: Medio. Diseño de experimentos y estrategias: Alto. Utilización de herramientas: Medio. Capacidad para ajustarse a unas especificaciones funcionales establecidas:Alto.</p>	
Actitudinales(Ser)	
<p>Capacidad de abstracción: Alto. Razonamiento lógico e identificación de relaciones entre los elementos de un problema: Alto. Capacidad de presentación de soluciones informáticas: Medio.</p>	