

TEORÍA DE LOS LENGUAJES DE PROGRAMACIÓN

Curso 2011/2012

(Código: 71012024)

1. PRESENTACIÓN DE LA ASIGNATURA

Esta asignatura presenta los contenidos relacionados con la teoría y la práctica de los lenguajes de programación (LP). Los objetivos generales de la asignatura son:

- El estudio formal de los Lenguajes de Programación (LP) y su especificación, para que el estudiante tenga una visión general que le permita entender el papel de la sintaxis y la semántica en los LP.
- Conocer los mecanismos de los LP y sus construcciones, para distinguir entre el control de secuencia y el control de datos en un programa, los rasgos que caracterizan a los elementos de datos en un lenguaje, su sistema de tipos o su ausencia.
- El estudio de los paradigmas de los lenguajes de programación existentes, enmarcar los LP en paradigmas y destacar las características de los LP que los hacen adecuados para la función que se les otorga.

Su estudio está orientado a:

- El programador profesional, que necesita saber las diferencias entre los LP para poder elegir el más adecuado para el proyecto que quiera realizar o el problema a resolver o, si ya ha tomado esta decisión y trabaja con uno concreto, para mejorarlo con técnicas disponibles en otros LP.
- El diseñador de un nuevo LP que necesita saber tanto los fundamentos de los diferentes paradigmas de los LP como el panorama completo de los existentes, para considerar con todo rigor el nicho que va a ocupar su nuevo LP.
- El implementador de un LP que desarrolla su compilador o intérprete sobre una máquina particular o un tipo de máquina y pueda aprovechar lo que ha aprendido en este mismo proceso para futuros LP.

En general el programador, el diseñador y el implementador de un LP deben conocer el área de los LP, su evolución histórica, sus fundamentos teóricos, los distintos usos, etc para poder apreciar sus características diferentes, entender la influencia que ejercen las arquitecturas y aplicaciones de los ordenadores sobre el diseño de los LP y evitar futuros defectos de diseño de programas aprendiendo las lecciones del pasado.

La asignatura incluye una parte práctica con dos sesiones presenciales con el tutor, de asistencia obligatoria, que evaluará su práctica concreta.

2. CONTEXTUALIZACIÓN EN EL PLAN DE ESTUDIOS

Contextualización:

En el plan formativo del grado, esta asignatura se encuadra en la materia "Lenguajes de Programación", altamente relacionada con las materias "Fundamentos de Programación", y "Metodología de la Programación y Algoritmia".

- "Fundamentos de Programación" con dos asignaturas:
 - Fundamentos de Programación (primero, primer semestre o 1S)
 - Programación Orientada a Objetos (primero 2S)
- "Metodología de la Programación y Algoritmia", compuesta por las asignaturas:
 - Estrategias de Programación y Estructura de Datos (primero, 2S)



- Programación y Estructuras de Datos Avanzadas (segundo, 1S)
 - Complejidad y Computabilidad (cuarto, 1S)
- “Lenguajes de Programación”:
 - Autómatas, Gramáticas y Lenguajes (primero, 2S)
 - Teoría de los Lenguajes de Programación (segundo, 2S)
 - Procesadores del Lenguaje I (tercero, 1S)
 - Procesadores del Lenguaje II (tercero, 2S)

En la tabla oficial de reconocimiento de créditos de planes de estudios en la UNED, se puede observar que para esta asignatura solo se reconocen los créditos de la asignatura “Lenguajes de Programación” de la Ingeniería Técnica en Informática de Sistemas de la UNED. Esto se debe a que la asignatura planteada cubre casi un 80% de los aspectos teóricos de la de ITI Sistemas, aunque menos en los aspectos prácticos.

Competencias genéricas:

Las competencias genéricas que propone la UNED se articulan en cuatro áreas competenciales, entre las que se encuentran las dos relacionadas con esta asignatura G.2 y G.5:

- Gestión autónoma y autorregulada del trabajo,
 - (G.2) Competencias cognitivas superiores: selección y manejo adecuado de conocimientos, recursos y estrategias cognitivas de nivel superior apropiados para el afrontamiento y resolución de diversos tipos de tareas/problemas con distinto nivel de complejidad y novedad: Análisis y Síntesis. Aplicación de los conocimientos a la práctica Resolución de problemas en entornos nuevos o poco conocidos. Pensamiento creativo. Razonamiento crítico. Toma de decisiones.
- Gestión de los procesos de comunicación e información.
 - (G.5) Competencias en el uso de las herramientas y recursos de la Sociedad del Conocimiento: Manejo de las TIC. Competencia en la búsqueda de información relevante. Competencia en la gestión y organización de la información. Competencia en la recolección de datos, el manejo de bases de datos y su presentación
- Trabajo en equipo
- Compromiso ético.

Competencias específicas:

Las competencias específicas a alcanzar durante el transcurso de los estudios de la titulación se han clasificado en tres apartados, entre las que se encuentran las relacionadas con esta asignatura (FB.4, FB.5.2):

- Bloque de Formación Básica
 - (FB.5) Conocimientos básicos sobre el uso y programación de los ordenadores, sistemas operativos, bases de datos y programas informáticos con aplicación en ingeniería.
 - (FB.4) Comprensión de la estructura, funcionamiento e interconexión de los sistemas informáticos, así como de los fundamentos de su programación.
- Bloque Común a la Rama de Informática
 - (BC.1) Diseñar, desarrollar, seleccionar y evaluar, aplicaciones y sistemas informáticos, asegurando su fiabilidad, seguridad y calidad, conforme a los principios éticos y a la legislación y normativa vigente.
 - (BC.5) Aplicar los conocimientos sobre los procedimientos algorítmicos básicos de las tecnologías informáticas para diseñar soluciones a problemas, analizando la idoneidad y complejidad de los algoritmos propuestos
 - (BC.6) Conocer, diseñar y utilizar de forma eficiente los tipos y estructuras de datos más adecuados a la resolución de un problema.
 - (BC.7) Programar aplicaciones de forma robusta, segura y eficiente, eligiendo el paradigma y los lenguajes de programación más adecuados.
- Bloque de Tecnologías Específicas: Computación
 - (BTEc.1) Tener un conocimiento profundo de los principios fundamentales de la computación y saberlos aplicar para interpretar, seleccionar, valorar, modelar, y crear nuevos conceptos, teorías, usos y desarrollos tecnológicos relacionados con la informática.
 - (BTEc.2) 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.
- Bloque de Tecnologías Específicas: Competencias de otras áreas tecnológicas, Tecnologías de la



Información

(BTEti.2) Capacidad para seleccionar, diseñar, desplegar, integrar, evaluar, explotar y mantener las tecnologías de hardware, software y redes, dentro de los parámetros de coste y calidad adecuados.

3.REQUISITOS PREVIOS REQUERIDOS PARA CURSAR LA ASIGNATURA

No hay ningún prerrequisito diferente de los generales, ya que esta asignatura puede ser cursada aisladamente, aunque el estudiante se beneficiaría si hubiera cursado previamente las dos asignaturas de primero de Fundamentos de Programación y la de Metodología de la programación y Algoritmia del primer semestre de segundo.

4.RESULTADOS DE APRENDIZAJE

El objetivo general de la materia es proporcionar al estudiante formación relacionada con los lenguajes de programación (LP) en aspectos como:

- La habilidad para desarrollar algoritmos eficaces: muchos LP incluyen características que, cuando se usan de forma apropiada, benefician al programador pero si se usan incorrectamente, pueden o desperdiciar tiempo de cómputo o conducir al programador a errores lógicos.
- El uso correcto de un LP: conociendo cómo se implementan las características del LP que se use, se mejora la habilidad para escribir programas.
- El conocimiento de construcciones útiles de programación: el LP sirve a la vez como una ayuda y como una restricción para el pensamiento. Un LP permite estructurar la manera en que uno concibe un programa, y ayuda a expresarse en formas (construcciones de programación) difíciles de expresar solo con palabras.
- La elección de un LP según el problema a resolver o proyecto a realizar: cuando se presenta la situación, el conocimiento teórico de los diferentes paradigmas de programación y de los LP permite la elección del LP idóneo para un proyecto particular.
- El aprendizaje de nuevos LP: el conocimiento profundo de las construcciones y otros elementos de un LP, junto con sus técnicas de ejecución de programas, permite al programador aprender nuevos LP con más facilidad.
- El diseño de un nuevo LP: todo programa tiene una interfaz de usuario que es, de hecho, una forma de LP, ya que la interfaz de usuario se compone de los comandos y formatos de datos que permiten comunicarse con el programa. El diseñador de la interfaz de usuario para un proyecto complejo debe ocuparse de muchas de las cuestiones que están presentes en el diseño de un LP para usos generales. Por lo tanto es doble el resultado del aprendizaje: diseño de programas y métodos de implementación de LP.

Los resultados de aprendizaje y sus competencias en esta asignatura según el plan de estudios del Grado en Ingeniería Informática de la UNED, y que se cubren convenientemente con el planteamiento anterior son:

- Conocer y comprender los constructores propios de los lenguajes de programación y sus diferentes paradigmas, así como las técnicas para su implementación (Competencias: G2, FB3, BC6, BC7, BTEc1, BTEc2)
- Saber elegir el lenguaje más apropiado para el desarrollo de una aplicación. Conocer las alternativas y sus características (Competencias: G2,G5, FB3, BC1, BC7, BTEti2, BTEc1, BTEc2)
- Utilizar los constructores de distintos lenguajes en la implementación de programas (Competencias: G5, FB3, FB4.2, BC5, BC7, BTEc1, BTEc2)

5.CONTENIDOS DE LA ASIGNATURA

A continuación, se presenta la estructura de los contenidos de la asignatura: cuatro unidades didácticas que incluyen diferentes temas. Dentro de las actividades de aprendizaje de cada unidad didáctica, se especifican las lecturas más adecuadas para cada uno de los temas. Se presentarán y se facilitará, en su caso, el acceso a diferentes ejemplos y herramientas accesibles desde el aula virtual.



UNIDAD DIDÁCTICA 1: Principios de diseño de los lenguajes de programación

Esta unidad didáctica es una introducción al diseño de los lenguajes y los paradigmas existentes. Está compuesta por los siguientes temas:

Tema 1: Paradigmas de computación

Tema 2: Principios de diseño de los lenguajes

- 2.1 Historia y criterios de diseño
- 2.2 Eficiencia
- 2.3 Regularidad
- 2.4 Principios adicionales sobre diseño de los lenguajes (apartado 3.4)

UNIDAD DIDÁCTICA 2: Programación Declarativa

El objetivo global de esta unidad didáctica es el conocimiento de los lenguajes de programación declarativa, en sus dos modalidades, la declarativa y la funcional. La práctica obligatoria de la asignatura se corresponde con esta unidad didáctica.

Esta unidad didáctica está formada por los siguientes temas:

Tema 3: Programación Funcional

- 3.1 Programas como funciones
- 3.2 Evaluación perezosa
- 3.3 El lenguaje Haskell
- 3.4 Funciones recursivas

Tema 4: Programación Lógica

- 4.1 Lógica y programas lógicos
- 4.2 Resolución y unificación
- 4.3 El lenguaje Prolog
- 4.4 Problemas que se presentan con la programación lógica

UNIDAD DIDÁCTICA 3: Principios básicos del diseño de lenguajes

El objetivo global de esta unidad didáctica es el conocimiento teórico y práctico de los principios relacionados con la sintaxis y semántica de los lenguajes de programación no declarativos. Además se revisarán conceptos relacionados con los tipos de datos de un lenguaje de programación.

Esta unidad didáctica está formada por los siguientes temas:

Tema 5: Sintaxis de los lenguajes de programación

- 5.1 Estructura léxica de los lenguajes de programación
- 5.2 Gramáticas libres de contexto y BNF
- 5.3 Árboles de análisis sintáctico y árboles de sintaxis abstracta
- 5.4 Ambigüedad, asociatividad y precedencia
- 5.5 EBNF y diagramas sintácticos

Tema 6: Semántica básica

- 6.1 Atributos ligaduras y funciones semánticas
- 6.2 Declaraciones, bloques y alcance
- 6.3 La tabla de símbolos
- 6.4 Asignación, tiempo de vida y el entorno
- 6.5 Variables y constantes

Tema 7: Tipos de datos

- 7.1 Tipos de datos e información de tipos
- 7.2 Tipos simples
- 7.3 Constructores de tipos
- 7.4 Nomenclatura de tipos en lenguajes de ejemplo
- 7.5 Equivalencia de tipos
- 7.6 Conversión de tipos



UNIDAD DIDÁCTICA 4: Abstracción y estructura del control y la organización de entornos de ejecución

El objetivo global de esta unidad didáctica es el conocimiento teórico y práctico de los principios relacionados con las estructuras de control y los entornos de ejecución de los lenguajes de programación. En el tema 8 se estudian las expresiones, cuya evaluación permitirá que el programa pueda variar el flujo de ejecución mediante las estructuras de control. En el tema 9 se tratan los procedimientos, que son bloques de código cuya ejecución se pospone y que cuentan con una interfaz claramente especificada.

Esta unidad didáctica está formada por los siguientes temas:

Tema 8: Control I: Expresiones y enunciados

- 8.1 Expresiones
- 8.2 Enunciados y guardias condicionales
- 8.3 Ciclos y variaciones sobre WHILE
- 8.4 Manejo de excepciones

Tema 9: Control II: Procedimientos y ambientes

- 9.1 Definición y activación de los procedimientos
- 9.2 Semántica de los procedimientos
- 9.3 Mecanismos de paso de parámetros
- 9.4 Ambientes, activación y asignación de los procedimientos

6.EQUIPO DOCENTE

- [ANA Mª GARCIA SERRANO](#)
- [FERNANDO LOPEZ OSTENERO](#)

7.METODOLOGÍA Y ACTIVIDADES DE APRENDIZAJE

El plan de trabajo de acuerdo con el contenido de la asignatura se organiza como sigue:

Tema	Horas de teoría	Horas de práctica o de actividades complementarias
1	4	0
2	4	0
3	14	25
4	14	10
5	10	5
6	12	5
7	12	5
8	10	5
9	10	5

La asignatura no tiene clases presenciales. Los contenidos teóricos se impartirán a distancia, de acuerdo con las normas y estructuras de soporte telemático de la enseñanza en la UNED.

Junto a las actividades y enlaces con fuentes de información externas, existe material didáctico propio preparado por el equipo docente con ejercicios que organizan las actividades prácticas del alumno.

8.EVALUACIÓN

La evaluación se realizará a partir de los ejercicios propuestos a lo largo del curso, así como de un trabajo práctico final individual que organizará y ampliará el trabajo realizado en los ejercicios, según el calendario que se publicará en el aula virtual al comienzo del periodo lectivo de la asignatura.

La calificación consistirá en:



- un 60% del examen presencial
- 30% la práctica obligatoria
- 10% ejercicios opcionales teórico-prácticos realizados

Para aprobar la asignatura se ha de obtener al menos un 4 sobre 10 en el examen y aprobar la práctica obligatoria. Con respecto a la práctica:

- Deberá entregarse al tutor con anterioridad al examen para que éste la evalúe.
- Si la práctica está aprobada para la convocatoria ordinaria, también se considera aprobada para la convocatoria extraordinaria.
- Si en una convocatoria no se tienen las dos notas (práctica y examen), la calificación de dicha convocatoria será suspenso.
- No se guardarán notas para el siguiente curso.

Para optar a matrícula en esta asignatura, al menos hay que realizar todos los ejercicios opcionales.

9. BIBLIOGRAFÍA BÁSICA

ISBN(13): 9789706862846
 Título: LENGUAJES DE PROGRAMACIÓN (2ª)
 Autor/es: Louden, Kenneth C. ;
 Editorial: THOMSON PARANINFO, S.A.

Buscarlo en Editorial UNED

Buscarlo en librería virtual UNED

Buscarlo en bibliotecas UNED

Buscarlo en la Biblioteca de Educación

Comentarios y anexos:

El libro base cubre el temario de la asignatura, aunque no en el mismo orden. Para realizar el estudio se remite al alumno a la segunda parte de la guía de estudios donde podrá encontrar referencias a los capítulos del libro para cada punto del temario de la asignatura.

10. BIBLIOGRAFÍA COMPLEMENTARIA

ISBN(13): 9780072381115
 Título: PROGRAMMING LANGUAGES: PRINCIPLES AND PARADIGMS (1st. ed.)
 Autor/es: Tucker, Allen B., ;
 Editorial: : MCGRAW-HILL

Buscarlo en librería virtual UNED

Buscarlo en bibliotecas UNED

Buscarlo en la Biblioteca de Educación

Buscarlo en Catálogo del Patrimonio Bibliográfico



ISBN(13): 9789701700464
Título: LENGUAJES DE PROGRAMACIÓN : (3ª ed.)
Autor/es: Zelkowitz, Marvin V. ;
Editorial: PRENTICE-HALL HISPANOAMERICANA

Buscarlo en librería virtual UNED

Buscarlo en bibliotecas UNED

Buscarlo en la Biblioteca de Educación

Buscarlo en Catálogo del Patrimonio Bibliográfico

Comentarios y anexos:

- APPLEBY, D. y VANDEKOPPLE, J. J.: *Lenguajes de programación: paradigma y práctica*. McGraw-Hill (2.a edición), 1997.
- Sebesta, R. 2002. *Concepts of Programming Languages*. Quinta edición. Addison-Wesley.
- Tucker, A. y R. Noonan. 2002. *Programming Languages Principles and Paradigms*. McGraw-Hill.
- Wilson, L.B. y R.G. Clark. 2001. *Comparative Programming Languages*. Tercera edición. Addison-Wesley.

11.RECURSOS DE APOYO

La plataforma de e-Learning *Alf* proporcionará el aula virtual y el adecuado interfaz de interacción entre el alumno y sus profesores. *Alf* permite impartir y recibir formación, gestionar y compartir documentos, crear y participar en comunidades temáticas, así como realizar proyectos online. Se usarán las herramientas necesarias para que, tanto el equipo docente como el alumnado, encuentren la manera de compaginar el trabajo individual con el aprendizaje cooperativo.

Para las actividades complementarias se utilizarán recursos audiovisuales de la UNED (televisión, visualización etc), de acuerdo a la planificación que el equipo docente publicará anualmente.

12.TUTORIZACIÓN

La tutorización de los alumnos se llevará a cabo por los profesores de la asignatura, a través de la plataforma de e-Learning *Alf*, por teléfono y por correo electrónico (agarcia@lsi.uned.es, flopez@lsi.uned.es, 91 398 7993 y 7793 respectivamente).

El horario de guardia del equipo docente será los jueves del segundo cuatrimestre de 16:00 a 20:00.

13.Práctica Obligatoria

Esta asignatura tiene una práctica de carácter obligatorio, existiendo dos sesiones presenciales de asistencia obligatoria organizadas por los Centros Asociados.

La práctica, que se realizará en Haskell o Hugs, estará dividida en dos tareas principales:

- La primera de ellas, de carácter más práctico, consistirá en resolver un problema mediante un programa realizado en un lenguaje de programación funcional, que se estudia en los primeros temas de la asignatura.
- Además de esta tarea de programación, el alumno deberá identificar los conceptos teóricos de la asignatura dentro del código obtenido y relacionarlos con sus equivalentes en otros lenguajes de programación representativos de los diferentes paradigmas de programación que se ven en la asignatura.

Puntos importantes que se deben destacar son:

- El alumno debe dirigirse a su tutor o Centro Asociado con suficiente antelación para conocer el



calendario de las sesiones presenciales obligatorias de prácticas.

- Habrá un cierto número de sesiones de control y orientación sobre las prácticas, de asistencia obligatoria, y que se celebrarán en los Centros Asociados, bajo la supervisión y guía de los Tutores. Los Tutores propondrán una fecha límite de entrega de la práctica.
- Para poder aprobar la asignatura, será requisito imprescindible entregar en tiempo y forma la práctica resuelta y superar los mínimos requeridos para aprobar dicho trabajo.
- El anterior punto implica que todos los alumnos deberán haber entregado la documentación antes de la fecha establecida por su tutor, ya sea en junio o en septiembre.
- Los Tutores corregirán las prácticas, establecerán días de revisión e informarán al alumno de las mismas a través del Centro Asociado.
- No se guardan calificaciones de prácticas de un curso para otro.

