

UNIVERSIDAD NACIONAL DE ASUNCIÓN
FACULTAD POLITÉCNICA
INGENIERIA EN INFORMÁTICA
PLAN 2008
PROGRAMA DE ESTUDIO

I. IDENTIFICACIÓN

- | | |
|------------------------------------|-------------------------------|
| 1. Asignatura | : Estructura de los Lenguajes |
| 2. Semestre | : Sexto |
| 3. Horas semanales | : 7 horas |
| 3.1. Clases teóricas | : 3.5 horas |
| 3.2. Clases prácticas | : 3.5 horas |
| 4. Total real de horas disponibles | : 112 horas |
| 4.1. Clases teóricas | : 56 horas |
| 4.2. Clases prácticas | : 56 horas |

II. JUSTIFICACIÓN

A través de los años se han desarrollado un gran número de lenguajes de programación que intentan satisfacer las necesidades de los programadores en diferentes dominios de aplicación. Estos lenguajes pueden ser vistos como herramientas que permiten no solo desarrollar programas que puedan ser leídos y modificados con facilidad, sino también como herramientas para expresar ideas. La capacidad expresiva de las ideas está limitada por el poder expresivo del lenguaje en términos de estructuras de control, estructuras de datos y las abstracciones que maneja el lenguaje. Tener conocimiento de las estructuras de un gran número de lenguajes de programación permite al desarrollador superar las limitaciones que el lenguaje le ofrece al poder simular unas características en otros lenguajes que no las soportan. Igualmente, un conocimiento de las características de los diferentes lenguajes permite realizar una mejor selección del lenguaje a utilizar para realizar una tarea determinada.

Los lenguajes de programación se encuentran en una evolución constante, por lo que la capacidad de un aprendizaje continuo es esencial para el desarrollador. El conocimiento de las estructuras de los lenguajes permite mejorar la habilidad de aprender nuevos lenguajes. Además, permite la incorporación de nuevos lenguajes y paradigmas al permitir un entendimiento sobre los conceptos en los cuales estos se basan. Igualmente, un conocimiento sobre las características de las implementaciones de los lenguajes permite mejorar el uso de los mismos, utilizando dichas características de manera correcta. Por último, a medida que se conocen más características de los lenguajes es posible mejorar la habilidad para el diseño de nuevos lenguajes.

III. OBJETIVOS GENERALES

- Enunciar, describir y aplicar criterios y características para evaluar lenguajes de programación y elegir el lenguaje más adecuado en cada situación.
- Describir e interpretar el proceso histórico del desarrollo de los lenguajes de programación.
- Enunciar y describir características y propiedades de los distintos tipos de lenguajes de programación.
- Enunciar y describir las estructuras fundamentales de los lenguajes de programación.
- Describir componentes léxicos y sintácticos de los lenguajes de programación.
- Adquirir experiencia en el uso de al menos un lenguaje de programación de cada uno de los tipos mostrados.
- Facilitar la comprensión de los temas relacionados con la implementación de los lenguajes de programación.

IV. OBJETIVOS ESPECÍFICOS

A. Conocimientos

1. Describir las características que pueden servir para evaluar un lenguaje de programación.
2. Clasificar y diferenciar los diferentes lenguajes de programación.
3. Enunciar y comprender las herramientas básicas utilizadas para describir y aceptar los componentes léxicos y sintácticos de los lenguajes: expresiones regulares, gramáticas, autómatas.
4. Enunciar y comprender los criterios de diseño que se utilizan para el desarrollo de las diferentes estructuras de los lenguajes de programación.

B. Habilidades

1. Describir componentes léxicos utilizando expresiones regulares.
2. Realizar autómatas finitos deterministas y no-deterministas.
3. Emplear criterios adecuados para evaluar lenguajes de programación.
4. Desarrollar programas que realicen análisis léxico y sintáctico de un lenguaje utilizando una herramienta como lex y yacc.
5. Desarrollar programas en lenguajes que representen diferentes tipos de lenguajes vistos.



C. Competencias

1. Capacidad de aplicar los conocimientos en la práctica.
2. Disposición para el trabajo en equipo.
3. Capacidad de abstracción, análisis y síntesis y presentaciones orales.
4. Habilidades para buscar, procesar y analizar información procedente de fuentes diversas.
5. Capacidad para identificar, plantear y resolver problemas
6. Capacidad de comunicación oral y escrita.
7. Capacidad de desarrollar programas en diferentes lenguajes de programación

V. PRE - REQUISITO

1. Lenguajes de programación III

VI. CONTENIDO

A. 6.1. Unidades programáticas

1. Introducción a Lenguajes de Programación
2. Fundamentos de Lenguajes de Programación
3. Tipos Abstractos de Datos y Programación Orientada a Objetos
4. Programación Funcional
5. Programación Lógica

B. 6.2. Desarrollo de las unidades programáticas

1. Introducción a Lenguajes de Programación
 - 1.1. Dominios de programación.
 - 1.2. Paradigmas de Programación.
 - 1.3. Criterios de evaluación.
 - 1.4. Métodos de implementación.
 - 1.5. Ambientes de programación.
 - 1.6. Evolución histórica.
 - 1.7. Estandarización.
2. Fundamentos de Lenguajes de Programación
 - 2.1. Descripción de sintaxis y semántica.
 - 2.2. Expresiones regulares, Automatas, Gramáticas libre de contexto.
 - 2.3. Variables y constantes. Nombres y ligado. Tipos de datos.
 - 2.4. Expresiones y asignación.
 - 2.5. Estructuras de control.
 - 2.6. Subprogramas.
 - 2.7. Semántica de paso de parámetros Prueba de tipos.
 - 2.8. Funciones como parámetro.
 - 2.9. Funciones genéricas.
 - 2.10. Compilación independiente.
 - 2.11. Aspectos de implementación de subprogramas.
3. Tipos Abstractos de Datos y Programación Orientada a Objetos
 - 3.1. Modularidad y reutilización de software.
 - 3.2. Tipos abstractos de datos.
 - 3.3. Encapsulación y clases.
 - 3.4. Tipos de datos abstractos parametrizados.
 - 3.5. Herencia.
 - 3.6. Herencia múltiple.
 - 3.7. Polimorfismo y enlace dinámico.
 - 3.8. Objetos y Clases. Herencia.
 - 3.9. Interfaces Manejo de excepciones.
 - 3.10. Paquetes y otras facilidades del lenguaje.
4. Programación Funcional
 - 4.1. Programas como funciones.
 - 4.2. Evaluación retardada.
 - 4.3. Funciones recursivas.
 - 4.4. Cálculo Lambda.
 - 4.5. Gestión dinámica de la memoria.
5. Programación Lógica
 - 5.1. Lógica de primer orden y cláusulas de Horn.



5.2. Resolución e inferencia.

VII. ESTRATEGIAS METODOLÓGICAS

1. La asignatura está concebida sobre las prácticas de laboratorio que los estudiantes irán desarrollando durante las clases, apoyados sobre la base teórica presentada en clases magistrales.
2. Las clases teóricas se desarrollan en clases magistrales y trabajos grupales, dirigidos por el docente. Además los estudiantes participarán activamente de las clases al realizar lecturas previas de un tema determinado, indicadas a través del sitio virtual de la Facultad.
3. Los estudiantes realizarán los trabajos de laboratorios realizados en grupos o individuales y serán supervisados por los docentes.
4. Presentación y defensa de memorias de prácticas de laboratorio y de artículos científicos relacionados con el área en cuestión.
5. Enseñanza basada en trabajo y evaluación continua, que incluyen el aprendizaje basado en problemas y el trabajo en grupo.
6. En la plataforma virtual de la Facultad se realizarán: foros de discusión, tareas individuales y grupales, video con tutoriales, talleres, entregas de memorias, etc..

A. Medios Auxiliares

1. Pizarras acrílicas.
2. Marcadores.
3. Borrador de pizarra acrílica.
4. Computadoras. Sistemas operativo Linux, librerías OpenMP, MPI.
5. Proyector multimedia.
6. Parlantes para multimedia.
7. Plataforma virtual "EDUCA".
8. Entornos de desarrollo y manuales de uso de los mismos.
9. Sala de laboratorio equipada para las prácticas.
10. Computadoras en red.
11. Acceso a internet.

VIII. EVALUACIÓN

Los exámenes parciales requeridos por los reglamentos de la Facultad. Cada examen parcial constará de una parte teórica y otra práctica.

La calificación final será establecida de acuerdo a la escala vigente en la Facultad.

Trabajos grupales: para los cuales se dividirá el curso en grupos de hasta tres personas.

Trabajo prácticos desarrollados en distintos lenguajes de programación correspondientes a cada uno de los paradigmas estudiados.

Investigación en un tópico sobre Estructura de los lenguajes de programación y presentación de un documento de investigación. Cada grupo investigará un tópico relevante al contenido del curso. Cuando finalice la investigación el grupo entregará un artículo impreso de no menos de ocho páginas. El artículo debe presentar el tema de investigación a un nivel entendible por un estudiante del curso. Además cada grupo realizará una presentación a la clase describiendo lo encontrado, en esta presentación, a cada miembro del grupo se le requerirá participar en la presentación de una manera igualitaria. Esta presentación tendrá una duración de 40 minutos. Previa a la presentación al curso se presentará previamente el contenido de la misma al Docente de la asignatura.

IX. BIBLIOGRAFÍA**A. Básica**

- ☐ R. Sebesta, Concepts of Programming Languages Addison Wesley. XI Edición, 2016
- ☐ K. Louden. Construcción de Compiladores, Thompson. 2004.

B. Complementaria

- ☐ John C. Mitchell, Concepts in Programming Languages, Stanford University, Cambridge University Press, 2002
- ☐ Aho, Alfred V., Ravi Sethi, and Jeffrey D. Ullman. Compilers: Principles, Techniques, and Tools. Addison-Wesley, Reading, MA, USA, 1986.
- ☐ Bruce Tate, Seven Languages in Seven Weeks: A Pragmatic Guide to Learning Programming Languages, Pragmatic Bookshelf, 2010.
- ☐ Aho, Alfred V.; Sethi, Ravi y Ullman, Jeffrey D. Compiladores: Principios, técnicas y prácticas. Addison Wesley. 2da Edición, 1990.
- ☒ Compiladores: principios, técnicas y herramientas, 2da Edición, Alfred V. Aho, Monica S. Lam, Ravi Sethi & Jeffrey D. Ullman, 2007.



