

UNIVERSIDAD NACIONAL DE ASUNCIÓN
FACULTAD POLITÉCNICA
LICENCIATURA EN CIENCIAS INFORMÁTICAS
ÉNFASIS EN PROGRAMACIÓN DE COMPUTADORAS
PLAN 2009
PROGRAMA DE ESTUDIOS

I. - IDENTIFICACIÓN

1. Asignatura	: Estructura de los Lenguajes
2. Código	: 5.2.A
3. Horas cátedra semanal	: 5 horas.
4. Total de horas	: 80 horas

II. - JUSTIFICACIÓN

Este curso presenta al alumno los principios y problemas relacionados con el diseño y la implementación de lenguajes de programación de alto nivel, es un estudio de importantes conceptos que se encuentran en los actuales lenguajes de programación, posibilitando también al alumno tomar contacto con paradigmas de programación alternativos.

Mediante los conceptos aprendidos en la materia, se pretende que el alumno acreciente su vocabulario con construcciones útiles sobre programación y utilice mejor un determinado lenguaje; así como también, que sea capaz de desarrollar algoritmos eficaces que ayuden a utilizar eficientemente los recursos computacionales; que pueda seleccionar con criterios racionales el lenguaje más apropiado para un problema determinado y pueda aprender con mayor facilidad nuevos lenguajes de programación.

III. - OBJETIVOS GENERALES

- Acrecentar el vocabulario técnico con construcciones útiles sobre los lenguajes de programación
- Utilizar de mejor manera los lenguajes de programación
- Seleccionar los lenguajes de programación adecuados según el contexto de cada necesidad
- Comprender los fundamentos y características de diseño de los lenguajes de programación más representativos de cada paradigma

IV. OBJETIVOS ESPECIFICOS

- **Conocimientos**
 - Describir la evolución histórica de los lenguajes de programación más representativos
 - Comprender el proceso de traducción que sigue un lenguaje de alto nivel hasta su ejecución en la máquina
 - Describir las características esenciales de cada paradigma de programación
 - Enumerar las principales cuestiones de diseño a considerar en el momento de diseñar un nuevo lenguaje de programación o en el momento de aprender un nuevo lenguaje de programación
- **Habilidades**
 - Evaluar críticamente características y construcciones de los lenguajes de programación.
 - Evaluar nuevos lenguajes de programación aplicando principios generales de diseño de lenguajes y realizando analogías con lenguajes históricamente importantes.
- **Competencias**
 - Distinguir las diferencias entre distintos paradigmas de programación.
 - Comprender las motivaciones de las decisiones de diseño de algunos lenguajes y las implicancias de éstas.
 - Potenciar la capacidad de expresar ideas y apreciar las características valiosas de los lenguajes para fomentar su uso.

V. PRE – REQUISITO

1. Paradigmas de la Programación

VI. CONTENIDO

6.1. Unidades programáticas

- Introducción
- Evolución de los principales lenguajes de programación.
- Estructuras sintácticas.
- Identificadores. Enlaces, chequeo de tipos y ámbito
- Tipo de datos



- Expresiones y sentencias de asignación
- Estructuras de control
- Subprogramas.
- Lenguajes orientados a objetos
- Lenguajes funcionales
- Lenguajes lógicos
- Lenguajes con soporte para la programación concurrente

6.2. Desarrollo de las unidades programáticas

- Introducción
 - Razones para estudiar los conceptos de lenguajes de programación
 - Dominios de programación
 - Criterios de evaluación de lenguajes
 - Influencias en el diseño de lenguajes
 - Categorías de lenguajes y decisiones de diseño
 - Métodos de implementación y entornos de programación
- Evolución de los principales lenguajes de programación.
 - Repaso de evolución histórica de lenguajes
 - Plankalkul, Pseudocódigos y Fortran
 - Lisp, Algol, Cobol, Basic, PL/1
 - APL, Snobol, Simula, Prolog, Ada, Smalltalk
 - C, C++, Java, C#, Perl, Python, y más contemporáneos
- Estructuras sintácticas.
 - Descripción de sintaxis
 - Métodos formales
 - Semántica
 - Análisis Léxico
 - Análisis Sintáctico
 - Parseo, Parseo descendente recursivo, Parseobottom-up
 - Análisis Semántico
- Identificadores. Enlaces, chequeo de tipos y ámbito
 - Nombres, variables y enlaces
 - Chequeo y compatibilidad de tipos. Lenguajes fuertemente tipados
 - Alcance y tiempo de vida
 - Entornos de referencia
 - Constantes nombradas
- Tipo de datos
 - Tipos primitivos
 - Cadenas de caracteres
 - Tipos ordinales
 - Arreglos
 - Arreglos Asociativos
 - Registros, Uniones, Conjuntos y Punteros
- Expresiones y sentencias de asignación
 - Expresiones aritméticas
 - Sobrecarga de operadores
 - Conversiones de tipos
 - Expresiones relacionales y booleanas
 - Evaluación en corto-circuito
 - Sentencias de asignación y asignaciones en modo mixto
- Estructuras de control
 - Sentencias compuestas
 - Sentencias de selección
 - Sentencias iterativas
 - Saltos incondicionales
 - Comandos con guardas
- Subprogramas.
 - Definiciones, fundamentos y cuestiones de diseño
 - Entornos de referencia local
 - Métodos de pasos de parámetros
 - Subprogramas sobrecargados y genéricos
 - Cuestiones de diseño
- Lenguajes orientados a objetos



- Abstracción, Encapsulamiento, Herencia y Polimorfismo
- Cuestiones de diseño de tipos de datos abstractos
- Tipos de datos abstractos parametrizados
- Programación orientada a objetos. Conceptos, características y cuestiones de diseño
- Soporte para la programación orientada a objetos en Java
- Manejo de excepciones
- Lenguajes funcionales
 - Funciones matemáticas
 - Fundamentos de los lenguajes de programación funcionales
 - Conceptos, características y cuestiones de diseño
 - Soporte para la programación funcional con Scheme
 - Aplicaciones de los lenguajes funcionales
- Lenguajes lógicos
 - Introducción al cálculo de predicados y demostración de teoremas
 - Orígenes de Prolog
 - Conceptos, características y cuestiones de diseño
 - Soporte para la programación lógica con Prolog
 - Aplicaciones de los lenguajes lógicos
- Lenguajes con soporte para la programación concurrente
 - Principales motivaciones del paralelismo
 - Aplicaciones posibles
 - Paralelismo implícito
 - Concurrencia física y lógica
 - Plataformas de cómputo

VII. ESTRATEGIAS METODOLÓGICAS

- Desarrollo de clases teóricas y prácticas
- Desarrollo de clases Taller en el aula
- Prácticas de Laboratorio

VIII. MEDIOS AUXILIARES

- Pizarra.
- Proyector de transparencias.
- Materiales impresos.
- Equipos Multimedia.
- Equipos de Laboratorio.

IX. EVALUACIÓN

- Exámenes Parciales y Finales establecidos en el Reglamento, los cuales estarán orientados a evaluar la capacidad de crítica de los alumnos hacia las diferentes construcciones en los lenguajes de programación así como la habilidad de usar estas construcciones para un mejor aprovechamiento de los recursos computacionales.
- Trabajos Prácticos
- Tareas en Laboratorios

X. BIBLIOGRAFÍA

- Sebesta, Robert W. Concepts of Programming Languages / Robert W. Sebesta.
- Loudon, Kenneth C. Lenguajes de programación. principios y prácticas / Kenneth C. Loudon.
- Watt, Davis A. Programming Language Design Concepts / David A. Watt
- Tucker, A. Lenguajes de Programación: principios y paradigmas / A. Tucker, R. Noonan
- Bal, Henri E. Programming Language Essentials. Henri E. Bal, DicjGrune.
- Pratt, Terrence W. Lenguajes de programación: diseño e implementación / Terrence W. Pratt, Marvin V. Zelkowitz
- Libros de referencia de lenguajes de programación: C, Java, Haskell, Fortran, Basic, Pascal, Cobol, C#, C++, Ada, Prolog.



