

LENGUAJES DE PROGRAMACIÓN

CLAVE:		SECTOR:	OPTATIVO
SEMESTRE:	6 - 8	ÁREA:	INFORMÁTICA
CRÉDITOS:	10	SERIACIÓN:	
		ASIGNATURA PRECEDENTE INDICATIVA: Materias del sector básico del Área de Informática	
		ASIGNATURA SUBSECUENTE INDICATIVA: Ninguna	
HORAS POR CLASE		TEÓRICA:	1
		PRÁCTICAS:	2
CLASES POR SEMANA		TEÓRICA:	4
		PRÁCTICAS:	1
HORAS POR SEMESTRE		TEÓRICA:	64
		PRÁCTICAS:	32

Objetivos generales: Al finalizar el curso el alumno:

- Conocerá y aplicará los principios fundamentales de algunos lenguajes de programación.

Tema 1. Historia y evolución de los lenguajes de programación **6 horas**

teóricas

3 horas prácticas

Revisará de manera breve e histórica los principales desarrollos en los lenguajes de programación.

- 1.1 Primeros lenguajes: Algol, Fortran, Cobol.
- 1.2 La evolución de los lenguajes orientados a procedimientos (las cadenas de desarrollo de Algol, PL/I, Pascal, Euclid, Modula y Ada).
- 1.3 Paradigmas y lenguajes no orientados a procedimientos
 - Funcionales (LISP).
 - Lógicos (Progol).
 - Orientados a objetos (Smalltalk).
 - Paralelos (Occam).

Tema 2. Organización de la computadora a nivel de ensamblador **10 horas**

teóricas

5 horas prácticas

Comparará distintos juegos de instrucciones y sus correspondientes modos de direccionamiento, haciendo énfasis en la interrelación de conjuntos de instrucciones, operaciones de carga y ejecución y la arquitectura subyacente, además de otros tópicos de importancia.

- 2.1 Organización básica: Von Neumann, diagramas de bloque, rutas para datos, rutas de control, unidades funcionales (i.e. unidad aritmético-lógica, memoria, registros), ciclo de instrucción.
- 2.2 Conjuntos y tipos de instrucciones.
- 2.3 Lenguaje ensamblador y de máquina.
- 2.4 Modos de direccionamiento (i.e. directo, indirecto, de desplazamiento, de registro, indexamiento).
- 2.5 Unidad de control; carga y ejecución de la instrucción; carga del operando.
- 2.6 Interrupciones de entrada/salida.
- 2.7 Instrumentación por alambrado.
- 2.8 Instrumentación por microprogramación; formatos y codificación.

**Tema 3. Máquinas virtuales
teóricas**

10 horas

5 horas prácticas

Analizará las diferencias entre las máquinas reales y las virtuales, y algunas características importantes de estas últimas.

- 3.1 Máquinas virtuales para los lenguajes de programación.
- 3.2 Jerarquía de máquinas virtuales presentadas al usuario a través del programa: el traductor, el sistema operativo, etcétera.
- 3.3 Consecuencias para la traducción de los distintos momentos en los que se hace el ligado.

**Tema 4. Control de secuencia
teóricas**

15 horas

7 horas prácticas

Estudiará el flujo de control en los lenguajes de programación para la evaluación de expresiones y la ejecución de enunciados.

- 4.1 Expresiones, orden de evaluación y efectos laterales.
- 4.2 Enunciados simples y compuestos.
- 4.3 Subprogramas y corrutinas como una abstracción de expresiones y enunciados.

Tema 5. Control de los datos, maneras de compartirlos y chequeos de tipos

23 horas teóricas

12 horas prácticas

Conocerá algunos métodos para compartir datos, así como la restricción del acceso a datos en los lenguajes de programación.

- 5.1 Mecanismos para compartir y restringir el acceso a datos (estructura de bloques, COMMON, ADT's y alias).
- 5.2 Rangos estáticos vs. dinámicos. Extensión y visibilidad.
- 5.3 Mecanismos para el paso de parámetros
 - Por referencia.
 - Por valor.
 - Por nombre.
 - Por resultado
 - Otros.
- 5.4 Variedad en las disciplinas para el chequeo de tipos y sus mecánicas: estática vs dinámica vs sin tipo, explícita vs implícita, polimorfismo vs sobrecarga.

Bibliografía básica:

- Friedman, D. P. *et al. Essentials of Programming Languages*. (s. l.) MIT Press. 1992.
- Sethi, R. *Programming Languages, Concepts and Constructs*. (s. l.) Addison Wesley Publishing Company. 1989.

Bibliografía complementaria:

- Scragg, G. W. *Computer Organization, A Top-Down Approach*. (s. l.) McGraw-Hill Publishing Company, Inc. 1992.

- Budd, T. *An Introduction To Object-Oriented Programming*. (s. l.) Addison Wesley Publishing Company. 1991.
- Field, A. J. and P. G. Harrison. *Functional Programming*. (s. l.) Addison Wesley Publishing Company. 1989.
- Friedman, L. W. *Comparative Programming Languages, Generalizing The Programming Function*. (s. l.) Prentice Hall. 1991.
- Kogge, P. M. *The Architecture of Symbolic Computers*. (s. l.) McGraw-Hill Inc. 1991.
- Tucker, A. B. *Lenguajes de Programación*. 2^{da} edición. España. McGraw-Hill. 1987.

Sugerencias didácticas:

Se recomiendan tareas regulares en las cuales el alumno aplique el material visto en clase y esté obligado a revisar diversas fuentes bibliográficas para que amplíe sus conocimientos con diferentes enfoques. Asimismo se sugieren prácticas de cómputo para la experimentación con los algoritmos vistos en clase y el análisis de casos prácticos.

Forma de evaluación:

Se recomiendan de 3 a 4 exámenes parciales y un examen final, así como la realización de tareas sobre los temas vistos en clase para reforzar los conocimientos teóricos adquiridos.

Perfil profesiográfico:

El profesor que imparta el curso deberá ser egresado de las carreras de Ciencias de la Computación, Actuaría, Matemáticas o alguna afín, de preferencia tener un postgrado, y deberá tener experiencia docente en el área o en las aplicaciones los lenguajes de programación.