Nombre de la materia:	Laboratorio de Paradigmas de Programación
Clave:	IA7300-L
No. de horas/semana:	3
Total de horas:	48
No. de créditos:	6
Prerrequisitos:	* (*)

Objetivo general: Se pretende que el alumno conozca algunos de los lenguajes de programación contemporáneos y las diferencias entre ellos, para que sea capaz de seleccionar el lenguaje y/o paradigma que adecuado para cada situación. Introduciendo al estudiante a algunos de los lenguajes de relacionados con los paradigmas de programación actuales, empezando por el lenguaje ensamblador, lenguajes imperativos (C y Shell), un lenguaje orientado a objetos (Java/c++), un lenguaje funcional (LISP), un lenguaje declarativo (PROLOG) finalizando con una introducción a Wolfram Mathematica.

Contribución a los atributos de egreso y su nivel de aportación

- **AE1.** Aplicar los conocimientos de ingeniería adquiridos durante sus estudios para elaborar proyectos de ingeniería que resuelvan problemas específicos. **(Medio)**
- **AE2.** Identificar, formular y resolver problemas de ingeniería mediante un pensamiento crítico y asertivo, basados en los principios de ciencias básicas e ingeniería. **(Avanzado)**
- **AE3.** Presentar y defender su trabajo en diversos foros, tanto académicos como profesionales. **(Inicial)**
- **AE4.** Intercambiar su conocimiento y puntos de vista con profesionales del área e integrarse en equipos de trabajo multidisciplinarios. **(Avanzado)**

Programa sintético

1. Programación Imperativa o procedural y scripting ..... 16 hrs.
2. Examen parcial 1 ..... 2 hrs.
3. Programación Orientada a Objetos ..... 10 hrs.
4. Examen parcial 2 ..... 2 hrs.
5. Programación Declarativa ..... 12 hrs.
6. Wolfram Mathematica ..... 4 hrs.
7. Examen parcial 3 ..... 2 hrs.

Total: 48 hrs.

Programa desarrollado

1. Programación Imperativa o procedural y scripting ..... 16 hrs.
   - 1.1 El lenguaje de programación Shell
     - 1.1.1 Introducción.
     - 1.1.2 El shell, tipos de shell, procesos, como ejecutar un script.
     - 1.1.3 El shell como intérprete de ordenes

1.1.4 Entrada y salida estándar, tuberías, otros operadores.

1.1.5 El shell como lenguaje de programación

1.1.6 Variables de entorno, asignación de variables, acentos graves, comillas simples y dobles, paso de parámetros

1.1.7 Estructuras de control.

1.1.8 Funciones.

1.2 El lenguaje de programación C

1.2.1 Introducción a C (principios, compilación).

1.2.2 Características, estructura, Variables, constantes, Operadores (aritméticos, comparación, lógicos), entrada/salida.

1.2.3 Estructuras de control.

1.2.4 Apuntadores

1.2.5 Definición, funciones, arreglos.

1.2.6 Paso de parámetros por valor y por referencia.

1.2.7 Macros.

1.2.8 goto.

1.3 El lenguaje ensamblador

1.3.1 Estructura del procesador pentium, tipos de registros y modos de direccionamiento.

1.3.2 Conocimientos básicos de sintaxis AT& T

1.3.3 Estructura de un programa y funciones de la interrupción 0x80H.

1.3.4 Llamado de código ensamblador desde un programa en C.

1.3.5 ¿Que es y para que sirve el inline assembly?

2. Examen parcial 1 ..... 2 hrs.

3. Programación Orientada a Objetos ..... 10 hrs.

3.1 El lenguaje de programación Java y/o c++

3.1.1 Clases y herencia.

3.1.2 Características

3.1.3 Estructura, Variables, constantes, Operadores, flujos,

3.1.4 Estructuras condicionales e iteración

3.1.5 Tipos de dato.

4. Examen parcial 2 ..... 2 hrs.

5. Programación Declarativa ..... 12 hrs.

5.1 Programación funcional y lenguaje de programación LISP.

5.1.1 Objetos básicos (átomos y listas).

5.1.2 Evaluación de los átomos y de las listas.

5.1.3 Definición de funciones (Anónimas y con nombres).

5.1.4 Predicados (Valores lógicos con tipos, de igualdad y operadores lógicos).

5.1.5 Estructuras de control (Condicionales e iteración).

5.2 Programación lógica y el lenguaje de programación prolog

5.2.1 Características de la programación lógica
5.2.2 Ejecutando queries, declaración de predicados, asignación e instanciación
5.2.3 Objetos de tipo lista, Operadores !, . y ;.

6. Wolfram Mathematica ..... 4 hrs.
6.1 Introducción (Conceptos básicos).
6.2 Variables.
6.3 Arreglos y listas
6.4 Estructuras de control.
6.5 Funciones.
6.6 Visualización de datos.
6.6.1 Tablas, árboles y gráficos.

7. Examen parcial 3 ..... 2 hrs.

Bibliografía básica:

- Michael L. Scott. Programming Language Pragmatics Second Edition 2006.
- Robert W. Sebeta. Concepts of programming languages eighth edition 2007.

Bibliografía complementaria:

- P. Wellin, Programming with Mathematica an introduction, First Edition, 2014.
- R.Stones. Beginning Linux Programming. Wrox; Second Edition, 2000.
- B. Neveln. Linux Assembly Languaje Programming. Prentice Hall 2000.
- H.M. Deitel et al. Java How to Program. 5 th Edition. Prentice Hall 2002.
- R.K. Dybvig, J-P Hebert. The Scheme Programming Language. MIT Press. 2003
- H. Abelson, G.J. Sussman and J. Sussman. Structure and Interpretation of Computer Programs. MIT Press. 1996.

Metodologías de enseñanza-aprendizaje:

- Revisión de conceptos, análisis y solución de problemas en clase (X)
- Lectura de material fuera de clase (X)
- Ejercicios fuera de clase (tareas) (X)
- Elaboración de reportes técnicos o proyectos (X)

Metodologías de evaluación:

- Tareas (X)
- Elaboracion de reportes técnicos o proyectos (X)
- Exámenes de academia o departamentales (X)

Revisores:

Programa propuesto por el M.I. José Rafael Rodríguez Ochoa y modificado por el M.C. Adan Garnica Carrillo.

Notas: *Se deja a criterio de la academia la pertinencia del tema "Lenguaje ensamblador" dado que es difícil que el alumno logre un aprendizaje adecuado sobre el tema en tan poco tiempo, sugiriendo destinar el tiempo a la explicación de otro lenguaje de programación como (c++, python, javascript, etc.) por ser más modernos y porque representan mayor interés de los alumnos.*