

## 1.- DATOS DE LA ASIGNATURA

Nombre de la asignatura: <b>Lógica de Programación</b>
Carrera: <b>Ingeniería Mecatrónica</b>
Clave de la asignatura: <b>MTE-0523</b>
Horas teoría-horas práctica-créditos: <b>2-2-6</b>

## 2.- HISTORIA DEL PROGRAMA

<b>Lugar y fecha de elaboración o revisión</b>	<b>Participantes</b>	<b>Observaciones (cambios y justificación)</b>
Instituto Tecnológico de Reynosa, del 6 al 10 de diciembre del 2004.	Representante de las academias de ingeniería Mecatrónica de los Institutos Tecnológicos.	Reunión nacional de evaluación curricular de la carrera de Ingeniería Mecatrónica
Institutos Tecnológicos de Querétaro y Toluca, de enero a abril del 2005.	Academias de Ingenierías Mecatrónica y Sistemas Computacionales	Análisis y enriquecimiento de las propuestas de los programas diseñados en la reunión nacional de evaluación
Instituto Tecnológico de Toluca, del 16 al 20 de mayo del 2005	Comité de consolidación de la carrera de Ingeniería Mecatrónica	Definición de los programas de estudio de la carrera de Ingeniería Mecatrónica

## 3.- UBICACIÓN DE LA ASIGNATURA:

### a). Relación con otras asignaturas del plan de estudio

<b>Anteriores</b>		<b>Posteriores</b>	
<b>Asignaturas</b>	<b>Temas</b>	<b>Asignaturas</b>	<b>Temas</b>
Ninguna		Matemáticas discreta	- Proporciona las bases para la solución de problemas

### b). Aportación de la asignatura al perfil del egresado:

La materia aporta conocimientos y habilidades para las materias de Matemáticas discretas, matemáticas computacionales.

#### 4.- OBJETIVOS(S) GENERAL (ES) DEL CURSO

El estudiante será capaz de plantear metodológicamente la solución de problemas susceptibles de ser computarizados a través del manejo de técnicas estructuradas de diseño y formulación de algoritmos, de manera que estos queden debidamente documentados.

#### 5.- TEMARIO

Unidad	Temas	Subtemas
1	Conceptos básicos	1.1 Introducción 1.1.1 De los problemas a los programas 1.1.2 Breves prácticas de programación 1.2 Definición de lenguaje 1.3 Definición de algoritmo 1.4 Algoritmos cotidianos 1.5 Definición de lenguajes algorítmicos.
2	Metodología para la solución de problemas por medio de computadoras.	2.1 Definición de problema 2.2 Análisis de los datos 2.3 Diseño de la solución 2.4 Prueba y depuración
3	Técnicas para la formulación de algoritmos	3.1 Diagramas de flujo. 3.2 Pseudocódigo. 3.3 Diagramas estructurados (nassi-schneiderman)
4	Principios de Programación	4.1 Tipos de Datos. 4.2 Estructura Básica de un programa. 4.3 Aplicaciones
5	Estructuras Básicas de Control	5.1 Secuenciales • Asignación • Entrada • Salida 5.2 Condicionales • Simple • Múltiple

## 5.- TEMARIO (Continuación)

Unidad	Temas	Subtemas
6	Arreglos	6.1 Vectores 6.2 Matrices
7	Apuntadores	7.1 Concepto de Apuntadores. 7.2 Lista simplemente Ligada
8	Manejo de módulos	8.1 Concepto y características de un módulo 8.2 Clasificación de los módulos <ul style="list-style-type: none"><li>• Funciones</li><li>• Procedimientos</li></ul> 8.3 Operación de módulos y sus parámetros.

## 6.- APRENDIZAJES REQUERIDOS

Ninguno

## 7.- SUGERENCIAS DIDÁCTICAS

- Que el maestro plantee problemas para resolver fuera de clase propiciando que el alumno investigue diferentes métodos de solución y elabore algoritmos para algunos de los métodos.
- El maestro promoverá el trabajo en equipo tanto en clase como fuera de ella, buscando incorporar en cada uno de los equipos por lo menos a un alumno destacado.
- Que el maestro promueva la exposición en clase de los algoritmos diseñados por los alumnos y equipos de alumnos, buscando que se presenten diferentes algoritmos para implementar el mismo método, de tal forma que se propicie la discusión y comparación de los mismos.
- Establecer asesoría no sólo con el maestro, sino con alumnos de semestres avanzados que prestan servicio social.
- Que el maestro organice talleres operativos con el apoyo de alumnos de semestres avanzados (como servicio social) en donde se resuelvan dudas sobre problemas planteados por el maestro.
- Establecer los requerimientos mínimos de documentación que deben contener los trabajos entregados por los alumnos.
- Utilizar un lenguaje de programación básico como C.

## 8.- SUGERENCIAS DE EVALUACIÓN

- Exámenes escritos.
- Entrega de tareas.
- Desempeño del alumno dentro del aula.
- Exámenes resueltos en casa

## 9.- UNIDADES DE APRENDIZAJE

### Unidad 1: Conceptos Básicos

<b>Objetivo Educativo</b>	<b>Actividades de Aprendizaje</b>	<b>Fuentes de Información</b>
El estudiante conocerá la terminología relacionada con los algoritmos, así como la importancia de aplicar técnicas adecuadas de programación.	<ul style="list-style-type: none"><li>• Comprender el significado y concepto de algoritmo.</li><li>• Desarrollar y aplicar el concepto de algoritmo en problemas reales aplicados en su entorno.</li><li>• Realizar una práctica breve de programación para algún algoritmo específico, seleccionado ya sea por el alumno o sugerido por el maestro.</li></ul>	1 2 3

### Unidad 2: Metodología para la Solución de Problemas por Medio de Computadoras.

<b>Objetivo Educativo</b>	<b>Actividades de Aprendizaje</b>	<b>Fuentes de Información</b>
Conocerá la metodología en cada una de sus etapas para proponer la solución de un problema, así como la importancia de éstas como documento del proceso de desarrollo de algoritmos	<ul style="list-style-type: none"><li>• Visualizar algún problema de su ejercicio profesional y proponer por escrito un diseño de la solución (éste deberá incluir todos los pasos de la metodología de solución de problemas).</li></ul>	1 2 3 4

### Unidad 3: Técnicas Para la Formulación de Algoritmos

<b>Objetivo Educativo</b>	<b>Actividades de Aprendizaje</b>	<b>Fuentes de Información</b>
Será capaz de diferenciar los métodos de representación y formulación de algoritmos, así como de conocer las características más importantes de cada técnica.	<ul style="list-style-type: none"><li>• Explicar los métodos de representación y formulación de algoritmos, así como sus ventajas y desventajas.</li><li>• Programar algoritmos de casos específicos usando la secuencia de diagramas de flujo N-S y pseudocódigos.</li></ul>	1 2 3 4 5 6

### Unidad 4: Principios de Programación

<b>Objetivo Educativo</b>	<b>Actividades de Aprendizaje</b>	<b>Fuentes de Información</b>
Desarrollará diferentes programas para la resolución de problemas reales	<ul style="list-style-type: none"><li>• Programar algoritmos de casos específicos usando un lenguaje programación</li></ul>	1, 2 3, 4 5, 6

### Unidad 5: Estructuras Básicas de Control

<b>Objetivo Educativo</b>	<b>Actividades de Aprendizaje</b>	<b>Fuentes de Información</b>
Conocerá las diferentes estructuras de control.	<ul style="list-style-type: none"><li>• Aplicar las estructuras de control a problemas prácticos.</li></ul>	1, 2 3, 4 5, 6

## Unidad 6: Arreglos

<b>Objetivo Educativo</b>	<b>Actividades de Aprendizaje</b>	<b>Fuentes de Información</b>
Será capaz de utilizar los datos de tipo arreglo para plantear la solución de problemas que requieran de esta estructura	<ul style="list-style-type: none"><li>• Explicar los conceptos básicos y tipos de arreglos que existen para diferentes tipos de datos, así como el enfoque usado para su manejo computacional.</li><li>• Elaborar algoritmos y programas para la solución de problemas que involucren arreglos de una o más dimensiones.</li></ul>	1 2 3 4 5 6

## Unidad 7: Apuntadores

<b>Objetivo Educativo</b>	<b>Actividades de Aprendizaje</b>	<b>Fuentes de Información</b>
Conocerá el concepto y programará apuntadores	<ul style="list-style-type: none"><li>• Explicar el concepto, objetivo y manipulación de apuntadores.</li><li>• Elaborar programas para generar una lista simplemente ligada</li></ul>	1 2 3 4 5 6

## Unidad 8: Manejo de Módulos

<b>Objetivo Educativo</b>	<b>Actividades de Aprendizaje</b>	<b>Fuentes de Información</b>
Conocerá el concepto de módulo como unidad contenida de código y las ventajas de su uso en la construcción de programas grandes	<ul style="list-style-type: none"><li>• Explicar el concepto y la necesidad de la existencia de módulos como forma estructurada de diseño y manejo de problemas grandes, así como las formas de clasificación y modularización generalmente aceptadas.</li><li>• Elaborar programas sencillos, pero aplicando el concepto de módulos que hace uso de un "menú" principal.</li></ul>	1 2 3 4

## 10.- FUENTES DE INFORMACIÓN

1. Aho, Hopcroft, Ullman, *Estructura de Datos y Algoritmos*, Ed. Sitesa
2. Luis Joyanes Aguilar, *Fundamentos de Programación, Algoritmos y Estructura de Datos*.
3. Guillermo Levine, *Introducción a la Computación y la Programación Estructurada*, Ed. Mc. Graw Hill
4. Goodman. Hedetmienic, *Introductionto the Design and Analysis of Algorithmis*, Ed. Mc. Graw Hill
5. Dale y Lilly, *Pascal y Estructura de Datos*, Ed, Mc. Graw Hill
6. Francisc Scheid, *Computers and Programing*, Ed. Mc. Graw Hill
7. Tremblay Sorensen, *An Introduction to Data Structures With Applications*, Ed. Mc. Graw Hill