

## 1.- DATOS DE LA ASIGNATURA

Nombre de la asignatura: <b>Fundamentos de programación.</b>
Carrera: <b>Licenciatura en Informática</b>
Clave de la asignatura: <b>IFM - 0415</b>
Horas teoría-horas práctica-créditos <b>3-2-8</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 Puebla del 8 al 12 septiembre 2003.	Representantes de la academia de sistemas y computación de los Institutos Tecnológicos.	Reunión nacional de evaluación curricular de la carrera de Licenciatura en Informática.
Instituto Tecnológico de: Cancún, Cd. Victoria, Iguala, Matamoros, Reynosa 13 septiembre al 28 de noviembre 2003.	Academia de de sistemas y computación.	Análisis y enriquecimiento de las propuestas de los programas diseñados en la reunión nacional de evaluación.
Instituto Tecnológico de Tepic 15 al 19 de marzo 2004.	Comité de consolidación de la carrera de Licenciatura en Informática.	Definición de los programas de estudio de la carrera de Licenciatura en Informática.

### 3.- UBICACIÓN DE LA ASIGNATURA

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

Anteriores		Posteriores	
Asignaturas	Temas	Asignaturas	Temas
Ninguna		Programación I	Fundamentos del lenguaje, subprogramas, punteros, referencias y arreglos, clases y objetos, referencias, Herencia.

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

- Desarrolla habilidades y actitudes que le permiten identificar problemas y oportunidades donde se aplique el tratamiento de información para proponer soluciones por medio de modelos y facilitar con ello la toma de decisiones.
- Desarrolla y administra sistemas de información para aumentar la productividad y competitividad de las organizaciones.
- Conformar, adecua y administra eficientemente las estructuras y bases de datos, que respondan a los requerimientos operativos de información.
- Integra conocimientos para especializarse en el desarrollo de software de aplicación y tecnología de vanguardia en diversas áreas.
- Mantiene una actitud de aprendizaje continuo.

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

- Diseñará metodológicamente la solución de problemas, utilizando técnicas de análisis, diseño y formulación de algoritmos, con un enfoque de programación orientada a objetos.

## 5.- TEMARIO

Unidad	Temas	Subtemas
1	Evolución de los paradigmas de programación.	1.1 Programación lineal. 1.2 Programación estructurada. 1.3 Programación orientada a objetos. 1.4 Relación entre la programación orientada a objetos y la programación estructurada. 1.5 Metodología para la solución de problemas basados en computadora.
2	Fundamentos del paradigma orientado a objetos.	2.1 Introducción a la Programación orientada a objetos. 2.1.1 Orígenes de la POO. 2.1.2 Beneficios de la POO. 2.2 Clases. 2.2.1 Encapsulamiento. 2.2.1.1 Concepto de caja negra. 2.2.1.2 Concepto de interfaz. 2.2.2 Abstracción. 2.2.2.1 Abstracción efectiva. 2.2.2.2 Ocultamiento de información. 2.2.2.3 Atributos. 2.2.2.4 Métodos. 2.2.2.5 Mensajes. 2.2.3 Diseño de una clase. 2.3 Objetos. 2.3.1 Ciclo de vida de un objeto. 2.3.2 Estado y comportamiento de los objetos. 2.4 Herencia y Polimorfismo. 2.4.1 Conceptos básicos. 2.4.2 Relación entre la herencia y el polimorfismo.
3	Construcción del estado de un objeto (atributos).	3.1 Tipos de datos primitivos. 3.1.1 Numéricos. 3.1.2 Caracteres. 3.1.3 Cadenas. 3.1.4 Lógicos. 3.2 Tipos de datos no primitivos. 3.2.1 Arreglos. 3.2.2 Apuntadores. 3.3 Identificadores, variables y constantes.

## 5.- TEMARIO (Continuación)

4	Construcción del comportamiento de un objeto (operaciones).	<ul style="list-style-type: none"><li>4.1 Diseño de métodos (funciones).<ul style="list-style-type: none"><li>4.1.1 Técnicas de formulación de algoritmos.</li><li>4.1.2 Tipos de algoritmos.<ul style="list-style-type: none"><li>4.1.2.1 Algoritmos determinísticos.</li><li>4.1.2.2 Algoritmos aproximados.</li><li>4.1.2.3 Algoritmos heurísticos.</li><li>4.1.2.4 Algoritmos voraces.</li></ul></li><li>4.1.3 Algoritmos probabilísticos.</li><li>4.1.4 Identificación del problema.</li><li>4.1.5 Planteamiento del problema.</li><li>4.1.6 Análisis del problema. (Alternativas de solución).</li><li>4.1.7 Diseño de la solución.</li><li>4.1.8 Pruebas y depuración.</li><li>4.1.9 Documentación.</li></ul></li><li>4.2 Operadores.<ul style="list-style-type: none"><li>4.2.1 Aritméticos.</li><li>4.2.2 Lógicos.</li><li>4.2.3 Jerarquía de operadores.</li></ul></li><li>4.3 Expresiones.</li><li>4.4 Estructuras de control.<ul style="list-style-type: none"><li>4.4.1 Selectivas.</li><li>4.4.2 Iterativas.</li></ul></li></ul>
5	Modelado de aplicaciones utilizando la POO.	<ul style="list-style-type: none"><li>5.1 Identificar y plantear el problema.</li><li>5.2 Principios básicos de modelado de objetos.</li><li>5.3 Introducción a un lenguaje de modelado.</li><li>5.4 Modelar sus clases.</li><li>5.5 Modelar relaciones entre clases<ul style="list-style-type: none"><li>5.5.1 Dependencia.</li><li>5.5.2 Asociación.</li><li>5.5.3 Composición.</li><li>5.5.4 Generalización.</li></ul></li></ul>

## **6.- APRENDIZAJES REQUERIDOS**

- Ninguno..

## **7.- SUGERENCIAS DIDÁCTICAS**

- Investigar antes de la clase y ubicar en el tiempo la evolución de los diferentes paradigmas de programación, llegar a conclusiones mediante una lluvia de ideas.
- Hacer comparaciones entre los diferentes paradigmas de programación.
- Promover durante el curso actividades de exposición que propicie el intercambio de ideas y la discusión dirigida.
- Propiciar la lectura y realización de informes sobre paradigmas de programación actuales y emergentes.
- Promover la investigación de tópicos relacionados con la asignatura.
- Promover el trabajo en equipo para que el estudiante construya su aprendizaje
- Organizar paneles con estudiantes de semestres avanzados para sensibilizar a los estudiantes de nuevo ingreso sobre la importancia de la materia en su estructura reticular y el perfil de su carrera.
- Elaborar en coordinación con los estudiantes un conjunto de problemas actuales asociados al entorno.
- Plantear problemas para que sean resueltos en equipos durante el curso.

## **8.- SUGERENCIAS DE EVALUACIÓN**

- Trabajos de investigación.
- Tareas relacionadas con el modelado de la solución de problemas.
- Exámenes escritos.
- Ejercicios dentro del aula.
- Establecer de común acuerdo con los estudiantes, la ponderación de las diferentes actividades del curso.
- Desempeño académico del estudiante.

## 9.- UNIDADES DE APRENDIZAJE

### UNIDAD 1.- Evolución de los paradigmas de programación.

<b>Objetivo Educativo</b>	<b>Actividades de Aprendizaje</b>	<b>Fuentes de Información</b>
El estudiante comprenderá las características principales de los paradigmas de programación actuales y emergentes.	<p>1.1 Investigar en fuentes diversas de información las características principales de los diferentes paradigmas de programación y elaborar un informe.</p> <p>1.2 Analizar y discutir en el aula la investigación realizada en el punto anterior, donde se resalten las diferencias identificadas.</p> <p>1.3 Comparar las ventajas y desventajas de la programación estructurada y la programación orientada a objetos.</p>	1,2,3,6,8,9,10

### UNIDAD 2.- Fundamentos del paradigma orientado a objetos.

<b>Objetivo Educativo</b>	<b>Actividades de Aprendizaje</b>	<b>Fuentes de Información</b>
Conocerá y comprenderá los mecanismo de la programación orientada a objetos.	<p>2.1 Investigar en empresas que utilicen el paradigma de la programación orientada a objetos, los beneficios que han obtenido de su utilización.</p> <p>2.2 Analizar y discutir en el aula la investigación realizada en el punto anterior, donde se resalten las diferencias identificadas.</p> <p>2.3 Identificar y hacer clasificaciones de objetos cotidianos y discutir en el aula los criterios seguidos para realizar tal clasificación y resaltar el concepto de abstracción.</p> <p>2.4 Investigar en fuentes diversas de información lenguajes de modelado de clases e identificar la representación de una clase.</p> <p>2.5 Seleccionar un objeto cotidiano y representar su ciclo de vida.</p> <p>2.6 Mediante casos cotidianos, identificar el concepto de herencia y polimorfismo.</p>	1,2,3,6,8,9

**UNIDAD 3.-** Construcción del estado de un objeto (atributos).

<b>Objetivo Educativo</b>	<b>Actividades de Aprendizaje</b>	<b>Fuentes de Información</b>
Comprenderá y representará la construcción del estado de un objeto.	3.1 Investigar en fuentes diversas de información los tipos de datos primitivos que soporta un lenguaje de programación actual, elaborar una tabla para comparar este tema en al menos tres lenguajes. 3.2 Representar mediante un lenguaje de modelado, el estado de las clases identificadas en un problema.	1,2,5,8,9

**UNIDAD 4.-** Construcción del comportamiento de un objeto (operaciones).

<b>Objetivo Educativo</b>	<b>Actividades de Aprendizaje</b>	<b>Fuentes de Información</b>
Comprenderá la construcción y representará el comportamiento de un objeto.	4.1 Investigar al menos tres técnicas para la construcción de algoritmos, elaborar un informe. 4.2 Modelar la solución de problemas propuestos en el aula.	3,4,5,6,7,9

**UNIDAD 5.-** Modelado de aplicaciones utilizando la POO.

<b>Objetivo Educativo</b>	<b>Actividades de Aprendizaje</b>	<b>Fuentes de Información</b>
Aprenderá a solucionar problemas utilizando la POO.	5.1 Modelar la solución de problemas.	1,2,5,7,8,9

## 10. FUENTES DE INFORMACIÓN

1. Greg Perry.  
Aprendiendo Principios de Programación en 24 horas,  
Ed. Prentice Hall.
2. Anthony Sintés.  
Aprendiendo Programación Orientada a Objetos en 21 Lecciones Avanzadas,  
Ed. Pearson Educación, México, 2002.
3. Timothy Budd.  
Introducción a la Programación Orientada a Objetos.  
Ed. Addison-Wesley Iberoamericana, Usa 1994.
4. Grady Booch/Addison-Wesley/Diaz de Santos.  
Análisis y Diseño Orientado a Objetos con Aplicaciones.  
Segunda Edición, México 1994.  
Ed. Addison-Wesley.
5. James Martin/James J. Odell.  
Análisis y Diseño Orientado a Objetos.  
Ed. Prentice Hall Hispano-Americana.
6. Andrew C. Stauggard Jr.  
Técnicas Estructuradas y Orientadas a Objetos.  
Segunda Edición 1998.  
Ed. Prentice Hall Hispano-Americana.
7. Michael Blaha/William Premerlani/Fred Eddy/William Lorenzen,  
Programación Orientada A Objetos. Conceptos, Modelado, Diseño y Codificación en C++.  
Ed. Prentice Hall, Madrid España 1996.
8. Luis Joyanes Aguilar.  
Problemas de Metodología de la Programación.  
1ª. Edición 1997.  
Ed. Mc Graw Hill.
9. Meyer Bertrand.  
Construcción de Software Orientado a Objetos.  
Ed. Prentice Hall.

## Referencias en Internet

- [10] [www.jugarijuegos.com./juegos/java](http://www.jugarijuegos.com./juegos/java)  
Desarrollo de estrategias de solución de problemas)
- [11] [www.omg.org/uml](http://www.omg.org/uml)
- [12] [www.uml.org](http://www.uml.org)

## 11. PRÁCTICAS

### Unidad Práctica

- 1 Describir las características y funciones de componentes de objetos utilizados cotidianamente tales como: video casetera, horno de microondas, teléfono público, refrigerador público expendedor de refresco, etc.
- 2 Modelar clases básicas como: empleado, estudiante, cadena de caracteres, vector, etc.