

## 1.- DATOS DE LA ASIGNATURA

Nombre de la asignatura: <b>Programación II</b>
Carrera: <b>Licenciatura en Informática</b>
Clave de la asignatura: <b>IFM - 0429</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: Colima, Comitancillo, Ocotlán 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
Programación I.	Clases y objetos. Subprogramas. arreglos.  Archivos.	Organización de datos.	Métodos de ordenación y búsqueda.  Archivos.

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

Proporciona conocimientos teórico - prácticos para la solución de problemas usando un lenguaje orientado a eventos.

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

El estudiante aplicará los conocimientos de un lenguaje de programación orientado a eventos, para resolver problemas dentro de los procesos administrativos.

## 5.- TEMARIO

Unidad	Temas	Subtemas
1	Fundamentos del lenguaje.	1.1 Programación orientada a eventos. 1.1.1 Objetos. 1.1.2 Eventos. 1.1.3 Métodos. 1.1.4 Propiedades. 1.2 Entorno integrado de desarrollo. 1.3 Ventanas (proyecto, propiedades, ayuda). 1.4 Tipos de proyectos. 1.4.1 Configuración. 1.5 Comentarios. 1.6 Constantes. 1.7 Tipos de datos. 1.8 Variables. 1.9 Operadores. 1.10 Sentencias. 1.11 Matrices. 1.12 Procedimientos y funciones. 1.13 Estructuras de control. 1.14 Controles básicos y sus propiedades.
2	Controles, eventos, métodos y componentes.	2.1 Controles estándar. 2.2 Eventos del formulario. 2.3 Cajas de diálogo. 2.3.1 Modales y no modales. 2.3.2 Predefinidas. 2.4 Componentes estándar.
3	Sistema de archivos.	3.1 Acceso secuencial. 3.2 Acceso directo. 3.3 Manejo de archivos. 3.4 Controlar errores en tiempo de ejecución.
4	Acceso a datos.	4.1 Introducción. 4.2 Control de datos ADO. 4.3 Objetos ADO. 4.4 Herramientas de acceso a datos.
5	Componentes y controles definidos por el usuario.	5.1 Modelo de componentes. 5.2 Crear un componente Active X. 5.3 Crear un Control Active X.

## **6.- APRENDIZAJES REQUERIDOS**

- Conocimiento de los paradigmas de programación.
- Dominio de las metodologías para la solución de problemas mediante la construcción de algoritmos.
- Dominio de las técnicas de diseño de algoritmos.

## **7.- SUGERENCIAS DIDÁCTICAS**

- Propiciar el trabajo en equipo.
- Desarrollar ejemplos de lo simple a lo complejo, buscando que el estudiante, asocie el tema con elementos significativos de su entorno.
- Realizar un taller donde el profesor conduzca paso a paso a sus estudiantes en el desarrollo del ejemplo, teniendo el cuidado que durante la solución se resalten los puntos de importancia que la teoría define, ya sea como concepto o el uso de la sintaxis en el momento de su aplicación.
- Propiciar el desarrollo y la realización de prácticas.
- Elaborar un conjunto de problemas actuales asociados al entorno.
- Solicitar al estudiante el desarrollo de un proyecto de su interés, aplicando los conocimientos adquiridos durante el curso.
- Involucrar al estudiante en las innovaciones que la evolución de la programación genera día con día, propiciando que éste se documente en distintas fuentes de información impresas y electrónicas.
- Seleccionar en acuerdo de academia, el lenguaje de programación a utilizar conforme a las necesidades del entorno, el contenido del programa y los recursos disponibles.

## **8.- SUGERENCIAS DE EVALUACIÓN**

- Aplicar un examen inicial para conocer en nivel de los aprendizajes requeridos.
- Establecer de común acuerdo con los estudiantes, la ponderación de las diferentes actividades del curso.
- Participar en clase y en el taller.
- Exponer temas específicos en aula.
- Elaborar reportes de trabajos de investigación.
- Elaborar reportes de prácticas.
- Aplicar exámenes escritos y prácticos (en computadora) correspondientes a cada unidad.
- Dar seguimiento a los avances y documentación del proyecto propuesto.

## 9.- UNIDADES DE APRENDIZAJE

### UNIDAD 1.- Fundamentos del lenguaje.

<b>Objetivo Educativo</b>	<b>Actividades de Aprendizaje</b>	<b>Fuentes de Información</b>
<p>El estudiante identificará la evolución, aplicaciones y sintaxis del lenguaje de programación, así como sus estructuras básicas.</p> <p>Identificará un problema que será utilizado para elaborar la propuesta de proyecto del curso.</p>	<p>1.1 Desarrollar una investigación en distintas fuentes y elaborar un informe donde identifique las principales etapas de la evolución que ha tenido el lenguaje a utilizar.</p> <p>1.2 Configurar y manipular el entorno de desarrollo del lenguaje, resolviendo problemas que hagan énfasis a cada uno de los elementos de éste.</p> <p>1.3 Desarrollar un programa de aplicación en el cual globalice los puntos comprendidos en la unidad.</p> <p>1.4 Identificar el área de aplicación del lenguaje.</p> <p>1.5 Identificar un problema que pueda ser utilizado como proyecto de curso de acuerdo con el profesor.</p>	<p>1, 2, 5, 6, 8</p>

### UNIDAD 2.- Controles, eventos, métodos y componentes.

<b>Objetivo Educativo</b>	<b>Actividades de Aprendizaje</b>	<b>Fuentes de Información</b>
<p>Identificará, comprenderá y utilizará los controles y componentes estándar definidos en el lenguaje.</p>	<p>2.1 Investigar en fuentes diversas de información los controles estándar y los eventos a los que responden, elaborar un informe comparativo con al menos 2 lenguajes.</p> <p>2.2 Desarrollar aplicaciones que ejemplifiquen el uso de controles y componentes estándar.</p> <p>2.3 Presentar el avance del proyecto.</p>	<p>1, 2, 5, 6, 8</p>

**UNIDAD 3.-** Sistema de archivos.

<b>Objetivo Educativo</b>	<b>Actividades de Aprendizaje</b>	<b>Fuentes de Información</b>
Aprenderá las diferentes formas que el lenguaje ofrece para almacenar datos en disco.	3.1 Investigar en fuentes diversas de información las diferencias existentes entre los archivos secuenciales y directos, elaborar un informe donde reporte los resultados. 3.2 Investigar en fuentes diversas de información las acciones que realiza el sistema operativo cuando se ejecuta una operación de apertura de archivo. 3.3 Elaborar aplicaciones que involucren el almacenamiento de datos en archivos, secuenciales y de acceso directo. 3.4 Incluir en las aplicaciones a desarrollar el uso de procedimientos de captura de errores en el manejo de archivos. 3.5 Presentar el avance del proyecto.	1, 2, 5, 6, 8

**UNIDAD 4.-** Acceso a datos.

<b>Objetivo Educativo</b>	<b>Actividades de Aprendizaje</b>	<b>Fuentes de Información</b>
Aprenderá el uso del acceso a datos mediante ADO.	4.1 Investigar en fuentes diversas de información los diferentes proveedores para el acceso a datos dentro de una aplicación. 4.2 Desarrollar aplicaciones utilizando ADO. 4.3 Presentar el avance del proyecto.	1, 3, 6, 8

**UNIDAD 5.-** Componentes y controles definidos por el usuario.

<b>Objetivo Educativo</b>	<b>Actividades de Aprendizaje</b>	<b>Fuentes de Información</b>
Aprenderá la utilidad de los diferentes componentes para la solución de problemas.	5.1 Investigar en fuentes diversas de información los diferentes modelos de componentes (OLE, COM, DCOM, OLE2, ActiveX). 5.2 Elaborar una guía rápida para el uso de los componentes. 5.3 Diseñar un componente ActiveX que solucione algún problema o simplifique algunas acciones. 5.4 Presentar el avance del proyecto.	6, 8

## 10. FUENTES DE INFORMACIÓN

1. Microsoft Visual Basic .NET. Referencia del lenguaje. Microsoft Corporation. Ed. McGraw Hill, 2002.
2. Microsoft Visual Basic .NET. Edición de aprendizaje. Microsoft Corporation. Ed. McGraw Hill, 2002.
3. Riordan. Aprende ADO.NET ya. Ed. McGraw Hill, 2002.
4. CEBALLOS, F.J., 2002. El lenguaje de programación Visual Basic .NET.
5. Rodríguez. Desarrollo de aplicaciones .NET con Visual C#. Ed. McGraw Hill, 2002.
6. Sharp & Jagger. Aprende Microsoft Visual C# . NET ya. Ed. McGraw Hill, 2002.
7. Appleman. Desarrollo de componentes COM/ActiveX con Visual Basic 6. Ed. Pearson.
8. Fco. Javier Ceballos. Enciclopedia de Visual BASIC 6. Ed. Alfaomega\_ra-ma.
9. Anthony Sintés. Aprendiendo Programación Orientada a Objetos en 21 Lecciones Avanzadas. Ed. Prearson Educación, México, 2002.

## 11. PRÁCTICAS

### Unidad Práctica

- 1 Utilizar el entorno integrado de desarrollo para el diseño de la interfaz.
- 2 Resolución de problemas donde utilice el formulario, conversión de:  
Controles básicos sus propiedades y eventos.  
Pesos y medidas.  
Divisas.  
Temperaturas.  
  
Validación de clave de acceso, el estudiante utilizará la caja de texto para validar una clave de acceso, para lograrlo utilizará las diversas propiedades y eventos del objeto. (variación: acepte solo números).
- 3 Desarrollar aplicaciones en las cuales utilice los controles y componentes tales como: Listas, Menús, Barras de herramientas, barras de estado, contenedores de imágenes. Tomar en cuenta que dichas aplicaciones se enfoquen a problemas cotidianos y que sean del interés del estudiante
- 4 Solución de problemas basados el uso de subprogramas y definidos por el usuario.
- 5 Desarrollar aplicaciones donde utilice el acceso a datos.