

1.- DATOS DE LA ASIGNATURA

Nombre de la asignatura: Introducción a la Programación
Carrera: Ingeniería en Desarrollo Comunitario
Clave de la asignatura: DCH-0521
Horas teoría-horas práctica-créditos: 0-4-4

2. HISTORIA DEL PROGRAMA

LUGAR Y FECHA DE EABORACION Y REVISION	PARTICIPANTES	OBSERVACIONES CAMBIOS Y JUSTIFICACION
Instituto Tecnológico de Oaxaca del 24 al 31 de enero al 4 de febrero de 2005	Representantes de las academias de Ingeniería en desarrollo comunitario de los Institutos Tecnológicos	Reunión Nacional de Evaluación de la carrera de Ingeniería en Desarrollo comunitario
Instituto Tecnológico de la Región Mixe, Instituto Tecnológico de Durango, de enero a abril del 2005	Academia de Ingeniería	Análisis y enriquecimiento de las propuestas de los programas diseñados en la reunión Nacional de Evaluación
Instituto Tecnológico Agropecuario de Oaxaca, del 30 de mayo al 3 de junio del 2005	Comité de consolidación de la carrera de Ingeniería en Desarrollo Comunitario	Definición de los programas de estudio de la carrera de Ingeniería en Desarrollo Comunitario

3.- UBICACIÓN DE LA ASIGNATURA

a). Relación con otras asignaturas del plan de estudio

Anteriores		Posteriores	
Asignaturas	Temas	Asignaturas	Temas
		Cálculo Diferencial e Integral (Matemáticas II)	<ul style="list-style-type: none"> • Matrices y Determinantes • Solución de sistemas de ecuaciones lineales
		Cálculo Integral y Ecuaciones Diferenciales (Matemáticas III)	<ul style="list-style-type: none"> • Solución de ecuaciones diferenciales de orden superior

b). Aportación de la asignatura al perfil del egresado

Esta materia proporciona al alumno herramientas computacionales para su aplicación en la solución de problemas de métodos numéricos, ecuaciones lineales y en las demás asignaturas del plan de estudio en las cuales se requiere del uso de la programación.

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

El alumno desarrollará programas en un lenguaje estructurado de propósito general, que le permitirá modelar situaciones de su entorno real.

5.- Temario

NUMERO	TEMA	SUBTEMA
1	Introducción a la programación	1.1. Introducción a un lenguaje de programación 1.1.1. Definición de lenguaje de programación 1.1.2. Tipos de lenguaje de programación 1.2. Metodología para la solución de problemas 1.3. Identificadores, constantes y variables 1.4. Entidades primitivas 1.4.1. Tipos de datos 1.4.2. Operadores

		1.4.3. Expresiones
2	Algoritmos	<ul style="list-style-type: none"> 2.1. Definición de algoritmos 2.2. Técnicas para la formulación de algoritmos 2.3. Estructuras algorítmicas <ul style="list-style-type: none"> 2.3.1. Secuenciales 2.3.2. Condicionales 2.3.3. Repetitivas
3	Entorno de trabajo	<ul style="list-style-type: none"> 3.1. Ventana principal 3.2. Editor de formas 3.3. Editor de código 3.4. Objetos, propiedades y eventos 3.5. Inspector de objetos <ul style="list-style-type: none"> 3.5.1. Ejecución controlada de un programa 3.5.2. Ventanas Inmediatas, Locals y Watches 3.5.3. Otras posibilidades del Debugger 3.6. Creación de programas ejecutables 3.7. Cómo utilizar el Help
4	Instrucciones fundamentales	<ul style="list-style-type: none"> 4.1. Instrucciones de asignación 4.2. Instrucciones de E/S 4.3. Instrucciones condicionales 4.4. Instrucciones repetitivas
5	Estructuras de datos básicos	<ul style="list-style-type: none"> 5.1. Estructuras lineales que guardan datos de un mismo tipo 5.2. Estructuras lineales que guardan datos de diferentes tipos 5.3. Estructuras para guardar información en almacenamiento secundario
6	Métodos Numéricos	<ul style="list-style-type: none"> 6.1. Solución de ecuaciones lineales <ul style="list-style-type: none"> 6.1.1. Gauss 6.1.2. Gauss Jordan 6.2. Solución de ecuaciones polinomiales <ul style="list-style-type: none"> 6.2.1. Método de interacción simple 6.2.2. Método de Newton Raphson

6.- APRENDIZAJES REQUERIDOS

- Manejo y uso del equipo de cómputo.
- Ecuaciones de primer orden con una, dos y tres incógnitas
- Factorización
- Productos Notables

7.- SUGERENCIAS DIDÁCTICAS

- Mostrar al alumnos programas completos de menor a mayor grado de dificultad y cada una de las instrucciones que los componen
- Enseñar la sintaxis del lenguaje y la finalidad de cada una de ellas.
- Solicitar que corrobore la validez del mismo, ejecutándolo en la computadora.
- Solicitar la elaboración de programas similares, agregándoles algunas variantes.
- Plantearle al alumno problemas para que los resuelva utilizando algoritmos que lo lleven a la solución del mismo, expresado en el lenguaje en cuestión, lo ejecute en la computadora y lo discuta en una sesión grupal.
- El maestro debe promover el trabajo en equipo tanto en clase como fuera de ella, buscando incorporar a cada uno de los equipos por lo menos un alumno destacado
- El maestro definirá los lineamientos mínimos de documentación que deberán contener las tareas relacionadas con el punto anterior.
- Solicitarle al alumno propuestas de problemas a resolver y que sean significativas para él.
- Propiciar que el alumno experimente con nuevos programas encontrados en revistas y libros de la especialidad, que lo lleven a descubrir nuevos conocimientos.
- Propiciar búsqueda y lectura de información de cada tema, aplicación de principios a prácticas de computación, solución de problemas reales, desarrollo de algoritmos y elaboración de programas

8.- SUGERENCIAS DE EVALUACION

- Presentación oral de los trabajos de manera individual identificando la bibliografía y haciendo discusión del tema.
- Reporte de investigación escrito de aplicaciones del tema visto en clase.
- Examen escrito
- Auto evaluación

9.- Unidades de aprendizaje

Unidad 1: Introducción a la programación

Objetivo Educativo	Actividades de Aprendizajes	Fuentes de información
<p>Distinguirá los diferentes tipos de lenguaje de programación. Aprenderá a declarar tipos de datos simples, constantes, etiquetas, y variables</p> <p>Describirá los pasos principales en la solución de problemas</p>	<ul style="list-style-type: none"> • Investigar los diferentes tipos de lenguajes de programación y analizar en forma grupal. • En forma grupal analizar los conceptos del inciso anterior. • Investigar los términos relacionados con el lenguaje, palabras reservadas, identificadores, etc. • En forma grupal analizar los conceptos del inciso anterior. • Estudiar en forma individual y analizar en grupos los diferentes tipos de declaraciones de datos y variables necesarios en un programa • Construir y evaluar expresiones aritméticas. 	<p>1 2 3 4 5</p>

Unidad 2: Algoritmos

Objetivo Educativo	Actividades de Aprendizajes	Fuentes de información
<p>El alumno reconocerá la terminología relacionada con los algoritmos, así como la importancia de aplicar técnicas adecuadas de programación. Conocerá la metodología de los algoritmos en cada una de sus etapas. Conocer las</p>	<ul style="list-style-type: none"> • Investigar la historia, significado y concepto de algoritmo. • Explicar las diferentes técnicas de diseño de algoritmos, así como sus ventajas, desventajas y aplicación a casos específicos. • El alumno deberá visualizar algún problema de la vida cotidiana y propondrá por escrito su diseño de la solución. (éste deberá incluir todos los pasos de la metodología de solución de problemas) • Realizar algoritmos que involucren fórmulas matemáticas, con el fin de crear las expresiones válidas para la computadora utilizando las diferentes estructuras 	<p>6 7 8 9 10 11</p>

diferentes estructuras algorítmicas como componentes básicos de los programas.	<p>algorítmicas.</p> <ul style="list-style-type: none"> • Explicar las estructuras algorítmicas de secuencia y condición, en forma simple o aislada y como parte de problemas más complejos. 	
--	---	--

Unidad 3: Entorno de Trabajo

Objetivo Educativo	Actividades de Aprendizajes	Fuentes de información
El alumno reconocerá los conceptos básicos necesarios dentro de la programación visual.	<ul style="list-style-type: none"> • Investigar los siguientes conceptos: componentes, propiedades y métodos. • En forma grupal analizar los conceptos del inciso anterior. • Desarrollo de programas sencillos donde aplique los conceptos vistos en la unidad 	<p>12</p> <p>13</p> <p>14</p> <p>15</p> <p>16</p> <p>17</p> <p>18</p>

Unidad 4: Instrucciones Fundamentales

Objetivo Educativo	Actividades de Aprendizajes	Fuentes de información
<p>El alumno desarrollará un programa sencillo en el lenguaje bajo estudio.</p> <p>El alumno resolverá problemas relacionados con la selección y de proposiciones cíclicas.</p> <p>Conocerá y aplicará las instrucciones básicas del lenguaje, en la codificación de programas.</p>	<ul style="list-style-type: none"> • Aplicar las proposiciones de E/S del lenguaje estudiado • Construir y documentar un programa, dándole solución a un problema planteado por el maestro. • Distinguir las diferentes proposiciones de selección y cíclicas según el lenguaje de programación en estudio, tales como: <ul style="list-style-type: none"> - If-then, if-then-else, If anidados, If aritmético - Case - Do, While, For, Repeat, etc. • Desarrollar aplicaciones utilizando las estructuras básicas de programación (selectivas y repetitivas) 	<p>12</p> <p>14</p> <p>15</p> <p>16</p> <p>17</p> <p>18</p>

Unidad 5: Estructuras de datos básicos

Objetivo Educativo	Actividades de Aprendizajes	Fuentes de información
El alumno 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">• Investigar la declaración de los diferentes tipos de datos de arreglos.• Explicar los conceptos básicos y tipos de arreglos que existen para diferentes tipos de datos, así como el enfoque usado para su manejo computacional.• Elaborar programas para la solución de problemas que involucren arreglos de una o más.• Diseñar problemas planteados por el maestro, donde se puedan aplicar el concepto de arreglo.	12 13 14 15 16 17 18

Unidad 6: Métodos Numéricos

Objetivo Educativo	Actividades de Aprendizajes	Fuentes de información
El estudiante generalizará los diferentes tipos de error que es posible detectar al emplear un determinado método numérico, con el auxilio de la programación computacional.	<ul style="list-style-type: none">• Emplear la programación para realizar operaciones matriciales: suma, resta y multiplicación, reconociendo su facilidad operativa.• Producir programas computacionales que le permitan aplicar métodos adecuados para calcular la raíz de una ecuación algebraica, de diferente grado.• Construir problemas representativos de situaciones reales, guiados por el maestro.	19

10.- FUENTES DE INFORMACIÓN

1. FRANCIS SCHEID. INTRODUCCIÓN A LA CIENCIA DE LAS COMPUTADORA 2 ED, ED. MC GRAW HILL
2. LUIS JOYANES AGUILAR. FUNDAMENTOS DE PROGRAMACIÓN, ALGORITMOS Y ESTRUCTURA DE DATOS. ED. MC GRAW-HILL
3. LUIS JOYANES AGUILAR. PASCAL 7. INICIACIÓN Y REFERENCIA ED. MCGRAW-HILL
4. LUIS JOYANES AGUILAR. PROGRAMACIÓN EN C. EDITORIAL MCGRAW-HILL
5. METODOLOGÍA DE LA PROGRAMACIÓN, DIAGRAMA DE FLUJOS Y PROGRAMAS OSVALDO CAIRO BATTISTUTTI ED. ALFAOMEGA
6. AHO, HOPCROFT, ULLMAN ESTRUCTURA DE DATOS Y ALGORITMOS ED. SITESA
7. GUILLERMO LEVINE INTRODUCCIÓN A LA COMPUTACIÓN Y LA PROGRAMACIÓN ESTRUCTURADA ED. MC GRAW-HILL
8. GOODMAN, HEDETMENIC INTRODUCTION TO THE DESIGN AND ANALYSIS OF ALGORITHMS ED. MC GRAW-HILL
9. DALE Y LILLY PASCAL Y ESTRUCTURA DE DATOS ED. MC GRAW-HILL
10. FRANCIS SCHEID COMPUTERS AND PROGRAMING ED. MC GRAW-HILL
11. TREMBLAY SORENSEN AN INTRODUCTION TO DATA STRUCTURES WITH APPLICATIONS ED. MC GRAW-HILL
12. FCO. JAVIER CEVALLOS SIERRA ECICLOPEDIA DE MICROSOFT VISUAL BASIC ED. COMPUTEC RAMA
13. FRANCISCO JAVIER CEBALLOS CURSO DE PROGRAMACIÓN VISUAL BASIC 6 ED. ALFAOMEGA
14. MICHAEL HALVORSON APRENDA VISUAL BASIC 6.0 YA ED. MCGRAW-HILL
15. LUIS JOYANES AGUILAR INTRODUCCIÓN A LA PROGRAMACIÓN ORIENTADA A OBJETOS MC GRAW-HILL
16. TIMOTHY BUDD

- PROGRAMACIÓN ORIENTADA A OBJETOS
ED. ADDISON WESLEY
17. BECK ZARATIAN
VISUAL C++6.0
MANUAL DEL PROGRAMADOR
MC GRAW HILL.
18. DEITEL & DEITEL
COMO PROGRAMAR EN C/C++
SEGUNDA EDICIÓN
PRENTICE
19. STEVEN C. CHAPRA / RAYMOND P. CANALE
MÉTODOS NUMÉRICOS PARA INGENIEROS
CON APLICACIONES EN COMPUTADORAS PERSONALES
MC GRAW HILL

PÁGINA ELECTRÓNICA

<http://www.itlp.edu.mx/publica/tutoriales/pascal/pascal.html>
<http://www.itlp.edu.mx/publica/tutoriales/algoritmos/index.htm>
<http://mat21.etsii.upm.es/ayudainf/aprendainf/VisualBasic6/vbasic60.pdf>
<http://www.abcdatos.com/tutoriales/tutorial/l4657.html>