



Universidad Nacional Autónoma de México
Colegio de Ciencias y Humanidades
Área Matemáticas

Programas de
Estudio de
Cibernética y
Computación II



PROGRAMA DE CIBERNÉTICA Y COMPUTACIÓN II

UBICACIÓN DEL CURSO

La asignatura de Cibernética y Computación II tiene una duración de 64 horas en el semestre, de las cuales 48 están orientadas a que el alumno aplique y profundice los aprendizajes adquiridos en el curso de Cibernética y Computación I, en particular, en la metodología para la solución de problemas y en la elaboración de programas, mediante el estudio del lenguaje Pascal; las 16 horas restantes el alumno obtendrá un panorama general de la programación, utilizando el lenguaje Delphi (Kylix).

La unidad uno está organizada para que el alumno profundice en la codificación de algoritmos al lenguaje Pascal. En la unidad dos, elaborará programas para resolver problemas utilizando las estructuras de control. En la unidad tres, construirá programas modulares aplicando procedimientos y funciones. En la unidad cuatro, utilizará las estructuras de datos definidas por el usuario con el fin de generalizar el concepto de variable. En la unidad cinco, conocerá el manejo básico del lenguaje Delphi (Kylix), para ampliar su visión de los lenguajes de programación.

PROPÓSITOS GENERALES

Al finalizar el curso, el alumno:

- ✍ Solucionará problemas utilizando el lenguaje de programación *Turbo Pascal*.
- ✍ Conocerá el manejo básico del lenguaje de programación *Delphi (Kylix)*.
- ✍ Distinguirá entre diferentes lenguajes de programación.

CONTENIDOS TEMÁTICOS

- 1) Lenguaje de programación *pascal*.
- 2) Estructura de control de secuencia.
- 3) Procedimientos y funciones.
- 4) Estructuras de datos definidos por el usuario.
- 5) Introducción a la programación en *delphi (kylix)*.

EVALUACIÓN

La evaluación debe considerar los aprendizajes de los alumnos (conceptuales, procedimentales y actitudinales).

Se recomienda una evaluación permanente, práctica y teórica, que tiene por objeto, informar al alumno el grado de conocimientos alcanzados, permitiendo detectar y corregir errores en su aprendizaje. La evaluación estará basada en exámenes escritos, ejercicios prácticos y trabajos de investigación.

CIBERNÉTICA Y COMPUTACIÓN II

UNIDAD I. LENGUAJE DE PROGRAMACIÓN PASCAL

Propósito:

- ✍ Al finalizar la unidad, el alumno resolverá problemas usando los elementos de pascal, mediante el desarrollo de programas para aplicar y ampliar los conocimientos y habilidades adquiridas en el curso anterior.

TIEMPO: 8 Horas

APRENDIZAJES	ESTRATEGIAS	TEMÁTICA
<p>El alumno:</p> <ul style="list-style-type: none"> ? Describe la estructura del lenguaje de programación. ? Conoce la sintaxis y semántica de las sentencias del lenguaje de programación. ? Resuelve problemas que involucran funciones y procedimientos estándar. ? Resuelve problemas con el proceso de ejecución de un programa. 	<ul style="list-style-type: none"> ? Conocer los aprendizajes, temática, metodología y la forma en que se evaluará. ? Retomar y complementar los algoritmos desarrollados en el curso anterior y codificarlos en el lenguaje. ? Utilizar los diagramas sintácticos de las distintas sentencias del lenguaje. ? Resaltar las etapas de codificación, edición, almacenamiento, compilación y ejecución de un programa. ? Remarcar las características de cada tipo de sentencia: de entrada/salida, referencia a memoria, transferencia de control, asignación de recursos, entre otros. ? Utilizar la ayuda en línea. ? Utilizará el monitoreo de variables, para seguir el cambio de valor. 	<p>Elementos del lenguaje</p> <ul style="list-style-type: none"> ? Estructura del programa. <ul style="list-style-type: none"> ✍ Encabezado. ✍ Zona de declaraciones. ✍ Zona de sentencias. ? Identificadores, constantes, variables, palabras reservadas. ? Tipos de datos primitivos ? Sentencias <ul style="list-style-type: none"> ✍ Lectura, asignación, escritura. ✍ Estructura de la sentencia condicional (<i>If-then-else</i>). ? Operadores, expresiones y orden de evaluación. ? Funciones y procedimientos estándar.

EVALUACIÓN

APRENDIZAJES	Evaluación	Evaluación
	Parcial	Global
✍ Describe la estructura del lenguaje de programación.	10%	1%
✍ Conoce la sintaxis y semántica de las sentencias del lenguaje de programación.	20%	2%
✍ Resuelve problemas que involucran funciones y procedimientos estándar.	20%	2%
✍ Resuelve problemas con el proceso de ejecución de un programa.	50%	5%
TOTAL	100%	10 %

BIBLIOGRAFÍA BÁSICA

Joyanes, Aguilar Luis, *Programación en Turbo Pascal versión 5.5, 6.0 y 7.0*, 1993.

UNIDAD II. ESTRUCTURA DE CONTROL DE SECUENCIA

Propósito:

- ✍ Al finalizar la unidad, el alumno utilizará las sentencias de condición, selección y ciclo, en la construcción de programas para resolver problemas en el lenguaje de programación.

TIEMPO: 10 Horas

APRENDIZAJES	ESTRATEGIAS	TEMÁTICA
<p>El alumno:</p> <ul style="list-style-type: none"> ? Identifica que el orden de ejecución de sentencias es de arriba hacia abajo (<i>Top-Down</i>). ? Utiliza como estructura condicional de selección <i>IF-THEN-ELSE</i> y como un caso particular <i>CASE</i>. ? Emplea las estructuras condicionales de ciclo: <i>WHILE-DO</i>, <i>FOR-DO</i> y <i>REPEAT-UNTIL</i>. ? Explica las diferencias entre las estructuras de control. ? Describe la sintaxis y semántica de las estructuras de control. ? Elabora programas que involucren las estructuras de control. 	<ul style="list-style-type: none"> ? Presentar las sentencias compuestas (sentencias en bloques) como agrupaciones de sentencias sobre las que actuarán las sentencias condicionales. ? Enfatizar la importancia de las expresiones lógicas en el funcionamiento de las sentencias condicionales. ? Enfatizar la conveniencia de utilizar la indentación en los programas como una ayuda visual para la identificación de sentencias compuestas. ? Resaltar la conveniencia del uso de comentarios dentro de los programas para facilitar su comprensión. ? Ejemplificar el uso de las sentencias por medio de programas ya elaborados. ? Utilizar un mismo problema en el que se pueda mostrar cada una de las diferentes estructuras. ? Utilizar los problemas desarrollados en el curso anterior y otros. 	<p>Estructuras de control de secuencia</p> <ul style="list-style-type: none"> ? Incondicionales. <ul style="list-style-type: none"> ✍ Simples. ✍ Compuestas. ✍ La sentencia nula y el uso del terminador de sentencia (“;”) ? Sentencias condicionales: <ul style="list-style-type: none"> ✍ Teorema de la programación estructurada. ? Estructura de Control <ul style="list-style-type: none"> ✍ <i>IF-THEN-ELSE</i>. ✍ <i>WHILE-DO</i>. ✍ <i>CASE</i>. ✍ <i>FOR-DO</i>. ✍ <i>REPEAT-UNTIL</i>.

EVALUACIÓN

APRENDIZAJES	Evaluación	Evaluación
	Parcial	Global
✍ Identifica que el orden de ejecución de sentencias es de arriba hacia abajo (<i>Top-Down</i>).	10%	1.5%
✍ Utiliza como estructura condicional de selección <i>IF-THEN-ELSE</i> y como un caso particular <i>CASE</i> .	10%	1.5%
✍ Emplea las estructuras condicionales de ciclo: <i>WHILE-DO</i> , <i>FOR-DO</i> y <i>REPEAT-UNTIL</i> .	10%	1.5%
✍ Explica las diferencias entre las estructuras de control.	10%	1.5%
✍ Describe la sintaxis y semántica de las estructuras de control.	10%	1.5%
✍ Elabora programas que involucren las estructuras de control.	50%	7.5%
TOTAL	100%	15 %

BIBLIOGRAFÍA BÁSICA

Joyanes, Aguilar Luis, *Programación en Turbo Pascal versión 5.5, 6.0 y 7.0*, 1993.

UNIDAD III. PROCEDIMIENTOS Y FUNCIONES

Propósitos:

- ✍ Al finalizar la unidad, el alumno utilizará los procedimientos y funciones, para elaborar programas de estructura modular, mediante el desarrollo de programas de solución de problemas.

TIEMPO: 10 Horas

APRENDIZAJES	ESTRATEGIAS	TEMÁTICA
<p>El alumno:</p> <ul style="list-style-type: none"> ? Comprende la importancia de dividir un programa en módulos. ? Describe el concepto de programación modular, funciones, procedimientos y parámetros por valor, referencia y variable. ? Identifica en la estructura de programas, la zona para la declaración e invocación de funciones y procedimientos. ? Elabora programas utilizando la programación modular. 	<ul style="list-style-type: none"> ? Realizar una investigación documental sobre la programación modular. ? Propiciar una discusión colectiva sobre la investigación realizada. ? Resaltar la técnica de programación descendente o “divide y vencerás” ? Ejemplificar programas sencillos donde se practiquen los procedimientos y funciones primero sin parámetros y de preferencia que los llamen varias veces. ? Mostrar que el empleo de los procedimientos y funciones proporciona facilidad en el desarrollo de programas. ? Los ejemplos y ejercicios desarrollados para esta unidad deberán realizarse desde la etapa de análisis y diseño, para enfatizar la importancia de construir los programas como colecciones de tareas genéricas interrelacionadas entre sí y no como una lista de instrucciones individuales. 	<p>La modularidad como una herramienta para el manejo de programas</p> <ul style="list-style-type: none"> ? Concepto. ? Importancia. <p>Procedimientos y funciones</p> <ul style="list-style-type: none"> ? Concepto de procedimiento. ? Concepto de función. <ul style="list-style-type: none"> ✍ Valor de regreso. ✍ Tipos permitidos. ? Definición, declaración e invocación de procedimientos y funciones. ? Parámetros por: <ul style="list-style-type: none"> ✍ Valor. ✍ Referencia. ✍ Variable.

<p>? Conoce la utilidad de manejar parámetros globales como medios de comunicación de información entre los módulos y el programa.</p> <p>? Explica las diferencias entre los parámetros por valor, referencia y variable.</p>	<p>? Desarrollar los programas de ejemplo empleando la secuencia:</p> <p>a) Definición de la estructura del cuerpo principal, señalando encabezados de procedimientos y funciones involucradas, así como los parámetros requeridos.</p> <p>b) Declaración de variables globales.</p> <p>c) Construcción de procedimientos y funciones, utilizando, en caso de ser necesario, nuevamente los pasos a) y b) para la construcción de los bloques de instrucciones de los módulos.</p> <p>? Señalar las ventajas y limitaciones de las funciones en relación con los procedimientos (valor de retorno, tipos de datos que pueden regresarse, entre otros).</p> <p>? Enfatizar la importancia de evitar la utilización de variables globales como medios de comunicación, de información a lo largo del programa.</p> <p>? Señalar las ventajas y limitaciones de las funciones en relación con los procedimientos (valor de retorno, tipos de datos que pueden regresarse, entre otros).</p> <p>? Enfatizar la importancia de evitar la utilización de variables globales en procedimientos y funciones debido a que se “oculta” con facilidad la modificación de los contenidos de las variables.</p> <p>? Mostrar el programa del factorial, identificando la parte recursiva y la terminal.</p> <p>? Mostrar con programas la estructura modular.</p>	<p>? Alcance de identificadores.</p> <ul style="list-style-type: none"> ✍ Variables globales y locales. ✍ Nombre de procedimientos y funciones. <p>? Recursividad.</p> <ul style="list-style-type: none"> ✍ Concepto matemático ✍ Implementación en el lenguaje. ✍ Parte recursiva y parte terminal de un procedimiento recursivo.
--	--	---

EVALUACIÓN

APRENDIZAJES	Evaluación	Evaluación
	Parcial	Global
✍ Comprende la importancia de dividir un programa en módulos.	15%	2.25%
✍ Describe el concepto de programación modular, funciones, procedimientos y parámetros por valor, referencia y variable.	15%	2.25%
✍ Identifica en la estructura de programas, la zona para la declaración e invocación de funciones y procedimientos.	15%	2.25%
✍ Elabora programas utilizando la programación modular.	30%	4.5%
✍ Conoce la utilidad de manejar parámetros globales como medios de comunicación de información entre los módulos y el programa.	15%	2.25%
✍ Explica las diferencias entre los parámetros por valor, referencia y variable.	10%	1.5%
TOTAL	100%	15 %

BIBLIOGRAFÍA BÁSICA

Joyanes, Aguilar Luis, *Programación en Turbo Pascal versión 5.5, 6.0 y 7.0*, 1993.

UNIDAD IV. ESTRUCTURAS DE DATOS DEFINIDOS POR EL USUARIO

Propósito:

✍ Al finalizar la unidad, el alumno utilizará las estructuras de datos mediante el desarrollo de programas, para generalizar el concepto de variable.

TIEMPO: 20 Horas

APRENDIZAJES	ESTRATEGIAS	TEMÁTICA
<p>El alumno:</p> <ul style="list-style-type: none"> ? Describe las características de las estructuras de datos de tipo arreglo, enumerado, subrango, cadena, conjunto, registro y archivo. ? Explica la declaración de tipos de datos: arreglo, cadena, registro, archivos y la forma de acceder a los elementos de los mismos. ? Describe las funciones y procedimientos para el manejo de cadenas y archivos. ? Explica las diferencias entre archivos de acceso secuencial y directo. 	<ul style="list-style-type: none"> ? Realizar una investigación documental sobre las estructuras de datos. ? Organizar discusiones colectivas, para abordar los conceptos de la investigación. ? Presentar programas que contengan tipos de datos enumerado, subrango, arreglos, cadenas, registros y archivos para mostrar la declaración y el acceso a los elementos de los mismos. ? Solicitar documentación en línea de la ayuda del lenguaje, para retroalimentar los tipos de datos abordados en los programas presentados. ? Plantear y resolver problemas que involucran la utilización de las estructuras de datos. ? Plantear y resolver problemas donde surge la necesidad de declarar grandes cantidades de variables. 	<p>Estructura de datos</p> <ul style="list-style-type: none"> ? Arreglos. <ul style="list-style-type: none"> ✍ Elementos de un arreglo. ✍ Índices y selección. ✍ Almacenamiento de arreglos. ✍ Dimensión de los arreglos. ✍ Implementación. ? Cadenas de caracteres. <ul style="list-style-type: none"> ✍ Implementación en el lenguaje de alto nivel. ✍ Funciones y procedimientos de manejo de cadenas. ✍ Conversiones a otros tipos de datos. ? Tipos enumerado y subrango. ? Conjuntos. ? Registros. <ul style="list-style-type: none"> ✍ Elementos de un registro.

<p>? Elabora programas que involucren los tipos de datos.</p> <p>? Conoce la diferencia entre estructuras estáticas y estructuras dinámicas.</p>	<p>? Enfatizar que los arreglos son estructuras de elementos del mismo tipo.</p> <p>? Enfatizar que los registros son estructuras de elementos con diferentes tipos.</p> <p>? Señalar el uso de apuntadores al ir al inicio o final de un archivo</p> <p>? Comentar las desventajas de reservar espacio para los arreglos y las ventajas del uso de estructuras dinámicas.</p>	<ul style="list-style-type: none"> ✍ Campos y selección de elementos. ✍ Almacenamiento de registros. ✍ La sentencia With. ✍ Registros variantes. <p>Manejo de archivos</p> <ul style="list-style-type: none"> ? Medios de almacenamiento secundario. ? Procesamiento secuencial. <ul style="list-style-type: none"> ✍ Archivos de tipo texto. ✍ Funciones y procedimientos para el manejo de archivos de tipo texto. ? Procesamiento de tipo directo <ul style="list-style-type: none"> ✍ Archivos de acceso directo. ✍ Funciones y procedimientos para el manejo de archivos de acceso directo. <p>Estructuras dinámicas</p> <ul style="list-style-type: none"> ? Apuntadores.
--	--	---

EVALUACIÓN

APRENDIZAJES	Evaluación	Evaluación
	Parcial	Global
✍ Describe las características de las estructuras de datos de tipo arreglo, enumerado, subrango, cadena, conjunto, registro y archivo.	15%	5.25%
✍ Explica la declaración de tipos de datos: arreglo, cadena, registro, archivos y la forma de acceder a los elementos de los mismos.	15%	5.25%
✍ Describe las funciones y procedimientos para el manejo de cadenas y archivos.	15%	5.25%
✍ Explica las diferencias entre archivos de acceso secuencial y directo.	15%	5.25%
✍ Elabora programas que involucran los tipos de datos.	30%	10.5%
✍ Conoce la diferencia entre estructuras estáticas y estructuras dinámicas.	10%	3.5%
TOTAL	100%	35 %

BIBLIOGRAFÍA BÁSICA

Joyanes, Aguilar Luis, *Programación en Turbo Pascal versión 5.5, 6.0 y 7.0*, 1993.

UNIDAD V. INTRODUCCIÓN A LA PROGRAMACIÓN EN DELPHI (KYLIX)

Propósito:

✍ Al finalizar la unidad, el alumno conocerá el manejo básico del lenguaje Delphi (Kylix), para ampliar la visión de los lenguajes de programación, mediante la exploración y presentación de programas.

TIEMPO: 16 Horas

APRENDIZAJES	ESTRATEGIAS	TEMÁTICA
<p>El alumno:</p> <ul style="list-style-type: none"> ? Describe las características del lenguaje de programación. ? Conoce los conceptos básicos del lenguaje de programación. ? Identifica los elementos del lenguaje de programación. ? Distingue la diferencia entre lenguajes de programación. ? Identifica las ventajas y desventajas entre los lenguajes de programación. 	<ul style="list-style-type: none"> ? Realizar una investigación documental sobre el lenguaje de programación. ? Propiciar una discusión colectiva para obtener las características y conceptos del lenguaje de programación. ? Presentar los elementos del lenguaje de programación. ? Describir los elementos principales del lenguaje de programación. ? Relacionar la estructura del lenguaje Pascal y Delphi (Kylix). ? Resaltar las ventajas y desventajas de desarrollo para la resolución de problemas con herramientas prediseñadas en Delphi (Kylix). ? Mostrar un proyecto de trabajo en Delphi (Kylix). 	<p>Elementos de programación Delphi (Kylix)</p> <ul style="list-style-type: none"> ? Objetos, eventos, diseño de ventanas, proyectos. ? Terminología básica, clases, instancias, herencia, polimorfismo. <p>Ambiente de trabajo.</p> <ul style="list-style-type: none"> ? Pantalla principal. <p>Elementos básicos.</p> <ul style="list-style-type: none"> ? Tablas. ? Formularios. ? Controles. <ul style="list-style-type: none"> ✍ Cuadros de texto. ✍ Etiquetas. ✍ Botones. ✍ Cajas de lista. ✍ Cuadros combinados. ✍ Botones de radio. ✍ Casillas de verificación. ✍ Contenedores. ✍ Cuadros de imagen.

		<ul style="list-style-type: none">? Propiedades<ul style="list-style-type: none">✎ Apariencia.✎ Comportamiento.✎ Posición.✎ Propiedades del control. ? Métodos<ul style="list-style-type: none">✎ Mostrar y ocultar.✎ Actualizar.✎ Maximizar y minimizar.✎ Manipular el enfoque. ? Eventos.<ul style="list-style-type: none">✎ Inicialización y cierre de control.✎ Recepción y pérdida del enfoque.✎ Interacción con ratón.✎ Interacción con teclado. ? Reportes.
--	--	--

EVALUACIÓN

APRENDIZAJES	Evaluación	Evaluación
	Parcial	Global
✍ Describe las características del lenguaje de programación.	20%	5%
✍ Conoce los conceptos básicos del lenguaje de programación.	20%	5%
✍ Identifica los elementos del lenguaje de programación.	20%	5%
✍ Distingue la diferencia entre los lenguajes.	20%	5%
✍ Identifica las ventajas y desventajas entre los lenguajes de programación.	20%	5%
TOTAL	100%	25 %

BIBLIOGRAFÍA BÁSICA

- Goldstein, Larry Joel.- *Turbo Pascal. Introducción a la Programación orientada a objetos*. Editorial Prentice-Hall. México, 1993.
- Joyanes, Aguilar Luis.- *Programación en Turbo Pascal / Borland. Pascal 7*. Editorial MC. Graw-Hill. Madrid, 1998.
- López, R. Leobardo.- *Programación estructurada. Turbo Pascal 7*. Editorial Computec. México, 1993.
- Schneider, Michael G. et al.- *Introducción a la Programación y solución de Problemas con Pascal*. Editorial Limusa Noriega. México, 1990.
- Swedan, Fathi M.- *Turbo Pascal 7. Referencia rápida*. Editorial Ra-Ma, 1994.
- Cantú, Marco. *Delphi (Kylix) 7*. Ediciones Anaya Multimedia. Madrid, 2002.
- Charte Ojeda Francisco, *Guía Práctica para Usuarios de Delphi 7*. Editorial Anaya Multimedia. Madrid, 2002.

BIBLIOGRAFÍA ELECTRÓNICA

<http://www.marcocantu.com>.

<http://www.lapolitecnica.net>.