



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

Programas de  
Estudio de  
Cibernética y  
Computación I y II



# ÍNDICE

PRESENTACIÓN	3
ENFOQUE DE LA MATERIA	4
SECUENCIA DE UNIDADES POR SEMESTRE	6
CIBERNÉTICA Y COMPUTACIÓN I	9
UNIDAD I. LA CIBERNÉTICA	9
UNIDAD II. CIRCUITOS LÓGICOS	12
UNIDAD III. METODOLOGÍA DE SOLUCIÓN DE PROBLEMAS	15
UNIDAD IV. INTRODUCCIÓN A LA PROGRAMACIÓN Y AL LENGUAJE DE PROGRAMACIÓN PASCAL	18
CIBERNÉTICA Y COMPUTACIÓN II	23
UNIDAD I. LENGUAJE DE PROGRAMACIÓN PASCAL	23
UNIDAD II. ESTRUCTURA DE CONTROL DE SECUENCIA	25
UNIDAD III. PROCEDIMIENTOS Y FUNCIONES	27
UNIDAD IV. ESTRUCTURAS DE DATOS DEFINIDOS POR EL USUARIO	30
UNIDAD V. INTRODUCCIÓN A LA PROGRAMACIÓN EN DELPHI (KYLIX)	33
COMISIÓN DE REVISIÓN Y AJUSTE DE LOS PROGRAMAS	36

## **PROGRAMAS DE CIBERNÉTICA Y COMPUTACIÓN I Y II**

### **PRESENTACIÓN**

#### **ORIENTACIÓN GENERAL DE LOS CURSOS**

El Colegio de Ciencias y Humanidades tiene el compromiso de proporcionar a sus alumnos educación, conocimientos y habilidades que contribuyan a desenvolverse en sus actividades profesionales y personales, de tal manera que incidan en la adquisición de la cultura básica, con fundamento en los principios del modelo educativo del Colegio. Es importante que el alumno del Colegio que curse la materia de Cibernética y Computación, adquiera los conocimientos fundamentales que le permitan comprender a la cibernética como una ciencia interdisciplinaria que incluye a la computación, que ha modificado y utilizado la información en todos los campos de la actividad humana, repercutiendo en la sociedad.

El alumno comprenderá la importancia del procesamiento de la información, adquirirá conocimientos y habilidades mediante el desarrollo de estrategias que se puedan aplicar a situaciones problemáticas; comprenderá la vinculación de la matemática con la cibernética en el estudio de sistemas naturales y artificiales; adquirirá una metodología para la solución de problemas, la elaboración de algoritmos y la programación en un lenguaje de alto nivel.

Las asignaturas de Cibernética y Computación pertenecen al Área de Matemáticas, deben cursarse en dos semestres (quinto y sexto), son opcionales u obligatorias de acuerdo al área de elección. Se orientan a la síntesis de lo aprendido durante los primeros cuatro semestres y a su aplicación en el campo de la cibernética y la computación, a fomentar en los alumnos la reflexión de los procesos de aprendizaje y la construcción de conocimientos, haciendo énfasis en la investigación, el desarrollo de habilidades y conocimientos que forman parte de la cultura básica, necesarios para el inicio de estudios superiores.

## ENFOQUE DE LA MATERIA

### ENFOQUE DISCIPLINARIO

La materia debe propiciar en los alumnos una visión general sobre la cibernética y la computación, sus avances, perspectivas y el aprovechamiento de las herramientas computacionales en la solución de problemas.

El enfoque proporcionará al alumno: una visión global de la cibernética, mediante el estudio y análisis de sistemas naturales y artificiales, a través del diseño de modelos de sistemas; mostrando la vinculación de los circuitos lógicos, el álgebra de Boole y los sistemas de numeración en el desarrollo de la cibernética, en particular de la computación; así como, una metodología que le permita poner en práctica el análisis, el razonamiento estructurado en el desarrollo de algoritmos, la codificación en un lenguaje de programación de alto nivel, sin pretender hacer del alumno un programador.

### ENFOQUE DIDÁCTICO

El enfoque debe orientarse para que el alumno logre los aprendizajes indicados en cada una de las unidades, haciendo énfasis en los aspectos:

- ✍ **Histórico.** Con el fin de que obtengan los conocimientos del desarrollo, espacial y temporal.
- ✍ **Teórico.** Para que adquieran los conceptos necesarios para comprender los procesos de manejo de la información y los elementos de la programación.
- ✍ **Práctico.** Deben desarrollar destrezas y habilidades en la solución de problemas, así como en la elaboración y ejecución de programas.
- ✍ **Analítico.** Para evaluar los aspectos que permitan la utilización de la cibernética y la computación, sus límites y perspectivas de desarrollo tanto en la disciplina como en las diversas ramas del conocimiento; así como los procesos de solución, los programas y sus resultados.

Es necesario que durante todo el curso, se consideren los principios del Colegio: aprender a aprender, aprender a hacer y aprender a ser. Se sugiere realizar las estrategias de aprendizajes especificadas en las unidades de los programas.

### **CONTRIBUCIÓN DE LA MATERIA AL PERFIL DEL EGRESADO**

En el proceso educativo, la enseñanza de las matemáticas contribuye a la formación de la personalidad del adolescente mediante el desarrollo de conocimientos, habilidades y destrezas intelectuales, la evolución de sus formas de pensamiento y la adquisición de valores, actitudes y normas. En particular, la materia de Cibernética y Computación propiciará:

La valoración de la dimensión tecnológica de los conocimientos que adquiere y aplicación de los mismos en la solución de problemas.

- ✍ La valoración del conocimiento científico.
- ✍ La aplicación de los conocimientos en distintos ámbitos de su actividad, con actitudes de seguridad en si mismo y de autoestima.
- ✍ La comprensión de las relaciones entre distintos campos del saber, el proceso de evolución histórica de los conocimientos y la relación con la sociedad en la cual se producen.
- ✍ La habilidad de resolver problemas y establecer relaciones con conocimientos adquiridos, planteando métodos de solución y su comprobación a través de procedimientos adecuados.
- ✍ El aprender por si mismo, adquirir habilidades de trabajo intelectual y conocimientos específicos que le permitan aumentar o construir otros y generar estrategias propias para alcanzar aprendizajes cada vez mas independientes y complejos.
- ✍ Fundamentar con racionalidad, responsabilidad y rigor sus conocimientos e ideas.
- ✍ El asimilar en su manera de ser, hacer y pensar, los conocimientos y habilidades que lo lleven a mejorar su propia interpretación del mundo y a adquirir madurez intelectual.
- ✍ Desarrollar un pensamiento lógico, reflexivo, crítico y flexible que se manifiesta en su capacidad para innovar en las diversas esferas de su actividad.

- ✍ Utiliza adecuadamente los algoritmos, de tal forma que resuelve los problemas y expresa sus resultados y conclusiones de manera adecuada.
- ✍ La habilidad para el manejo de estrategias de solución de problemas usando la computadora.
- ✍ El interés por la lectura y comprensión de textos diversos, particularmente científicos y de divulgación.

### **SECUENCIA DE UNIDADES POR SEMESTRE**

<b>QUINTO SEMESTRE</b>	<b>SEXTO SEMESTRE</b>
<b>CIBERNÉTICA Y COMPUTACIÓN I</b>	<b>CIBERNÉTICA Y COMPUTACIÓN II</b>
1. LA CIBERNÉTICA.  <b>(12 horas)</b>	1. LENGUAJE DE PROGRAMACIÓN PASCAL.  <b>(8 horas)</b>
2. CIRCUITOS LÓGICOS.  <b>(12 horas)</b>	2. ESTRUCTURA DE CONTROL DE SECUENCIA.  <b>(10 horas)</b>
3. METODOLOGÍA DE SOLUCIÓN DE PROBLEMAS.  <b>(22 horas)</b>	3. PROCEDIMIENTOS Y FUNCIONES.  <b>(10 horas)</b>
4. INTRODUCCIÓN A LA PROGRAMACIÓN.  <b>(18 horas)</b>	4. ESTRUCTURA DE DATOS DEFINIDOS POR EL USUARIO.  <b>(20 horas)</b>
	5. INTRODUCCIÓN A LA PROGRAMACIÓN EN DELPHI (KYLIX).  <b>(16 horas)</b>

En cada una de las unidades se señalan los propósitos, los aprendizajes esperados, las propuestas de estrategias didácticas, la temática, sugerencias de evaluación y la bibliografía.

## **PROGRAMA DE CIBERNÉTICA Y COMPUTACIÓN I**

### **UBICACIÓN DEL CURSO**

La asignatura de Cibernética y Computación I tiene una duración de 64 horas en el semestre, de las cuales las primeras 24 se dedican al estudio de los conceptos y elementos que dieron surgimiento a la cibernética como una ciencia interdisciplinaria, en la que se incluyen los sistemas de numeración, el álgebra de Boole y los circuitos lógicos. El resto a la metodología de la solución de problemas y al uso básico del lenguaje de programación Pascal.

La unidad uno está organizada para que el alumno obtenga una visión general de la cibernética mediante el estudio y análisis de los sistemas naturales y artificiales. En la unidad dos utilizará algunos elementos del álgebra de Boole y circuitos lógicos. En la unidad tres iniciará el estudio y aplicación de la metodología de solución de problemas. En la unidad cuatro los aspectos básicos del lenguaje de programación Pascal.

### **PROPÓSITOS GENERALES**

Al finalizar el curso, el alumno:

- ✍ Adquirirá una visión integradora de la cibernética y la computación para el estudio de los sistemas naturales y artificiales.
- ✍ Utilizará algunos elementos del álgebra de Boole y circuitos lógicos.
- ✍ Obtendrá una metodología para la resolución de problemas con el apoyo de la computadora y los lenguajes de programación.

## **CONTENIDOS TEMÁTICOS**

- 1) La cibernética.
- 2) Circuitos lógicos.
- 3) Metodología de solución de problemas.
- 4) Introducción a la programación.

## **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 I

### UNIDAD I. LA CIBERNÉTICA

#### Propósito:

- ☞ Al finalizar la unidad el alumno obtendrá una visión general de la cibernética mediante el estudio y análisis de sistemas naturales y artificiales para el diseño de sistemas.

**Tiempo: 12 Horas**

APRENDIZAJES	ESTRATEGIAS	TEMÁTICA
<p>El alumno</p> <ul style="list-style-type: none"> <li>? Describe los antecedentes históricos del origen de la cibernética y personajes que contribuyeron a su desarrollo.</li> <li>? Comprende la cibernética como ciencia Interdisciplinaria.</li> <li>? Compara los sistemas naturales y artificiales con el fin de comprender sus similitudes y diferencias.</li> <li>? Identifica los elementos de los sistemas.</li> <li>? Diseña un sistema.</li> </ul>	<ul style="list-style-type: none"> <li>? Conocer los aprendizajes, temática, metodología y la forma en que se evaluará el curso.</li> <li>? Realizar investigación documental sobre la historia de la cibernética, la aportación de quienes intervinieron en su creación, evolución, conceptos y aplicaciones de la cibernética.</li> <li>? Propiciar una discusión colectiva para obtener conclusiones de los conceptos abordados en la investigación realizada.</li> <li>? Representar con modelos los sistemas naturales y artificiales para conocer y comprender los mecanismos de control y de comunicación. Por ejemplo, al menos dos sistemas (computadora, refrigerador, automóvil, fábrica, hospital, educación, entre otros).</li> </ul>	<p>Origen y objetivos de la cibernética.</p> <p>Sistemas</p> <ul style="list-style-type: none"> <li>? Concepto.</li> <li>? Tipos.</li> <li>? Diseño.</li> <li>? Aplicaciones.</li> <li>? Control y comunicación.               <ul style="list-style-type: none"> <li>☞ Retroalimentación.</li> <li>☞ De información.</li> </ul> </li> </ul> <p>Modelos</p> <ul style="list-style-type: none"> <li>? Naturales y artificiales.</li> <li>? Analógicos y digitales.</li> <li>? Híbridos.</li> <li>? Matemáticos.</li> </ul>

	<p>? Realizar una investigación documental sobre los sistemas.</p> <p>? Propiciar una discusión colectiva para obtener conclusiones de los conceptos abordados en la investigación realizada.</p>	
--	---	--

## EVALUACIÓN

APRENDIZAJES	Evaluación Parcial	Evaluación Global
☒ Describe los antecedentes históricos del origen de la cibernética y personas que contribuyeron a su desarrollo.	15%	3%
☒ Comprende la cibernética como ciencia interdisciplinaria.	15%	3%
☒ Identifica los elementos de los sistemas.	15%	3%
☒ Compara los sistemas naturales y artificiales con el fin de comprender sus similitudes y diferencias.	30%	6%
☒ Diseña un sistema.	25%	4%
<b>TOTAL</b>	<b>100%</b>	<b>19 %</b>

## BIBLIOGRAFÍA BÁSICA

Jramoi, A. V, et al. *Introducción e Historia de la Cibernética*, México, Grijalbo, 1975.

Sluckin Wladyslaw. *La Cibernética. Cerebros y Máquinas*, Buenos Aires, Nueva Visión, 1971.

Wiener Norbert. *Cibernética y Sociedad*. Buenos Aires, Sudamericana, 1969.

Winer Norbert. *Cibernética ó El Control y Comunicación en Animales y Máquinas*, Barcelona, Tusquetes editores, 1998.

## **BIBLIOGRAFÍA COMPLEMENTARIA**

Ashby, W. Ross. *Introducción a la Cibernética*, Buenos Aires, Nueva Visión. 1977.

Campos Nordmann, Ramiro. *Cibernética y Planificación Económica*, Madrid, Pirámide, 1975.

Crosson, F. y Sayre K. *Filosofía y Cibernética*, México, Fondo de Cultura Económica, 1971.

García Santesmases, José. *Cibernética. Aspectos y Tendencias Actuales*, Madrid, Paraninfo, 1990.

George, F. H. *Fundamento de Cibernética*, Madrid, Pirámide, 1979.

Glushkov, V. M. *¿Qué es la Cibernética?*, Moscú, MIR. 1982.

Goudot Perrot. Andrée. *Cibernética y Biología*, Barcelona, Oikos-Tau, 1970.

Ruyer Raymond. *La Cibernética y el Origen de la Información*, México, Fondo de Cultura Económica, 1984.

## UNIDAD II. CIRCUITOS LÓGICOS

### Propósito:

- ☞ Al finalizar la unidad el alumno utilizará algunos elementos del álgebra de Boole y circuitos lógicos para el diseño, la construcción o simulación de algunos autómatas mediante el desarrollo de prácticas.

**TIEMPO:** 12 Horas

APRENDIZAJES	ESTRATEGIAS	TEMÁTICA
<p>El alumno:</p> <ul style="list-style-type: none"> <li>? Convierte números de un sistema de numeración a otro.</li> <li>? Realiza operaciones de adición, substracción, división y multiplicación en el sistema binario.</li> <li>? Realiza operaciones booleanas (disyunción, conjunción y negación).</li> <li>? Relaciona los operadores booleanos con las compuertas lógicas y las funciones.</li> <li>? Describe los conceptos de interruptor, compuerta lógica, circuito eléctrico y circuito lógico.</li> </ul>	<ul style="list-style-type: none"> <li>? Realizar una investigación documental sobre los sistemas de numeración, para describir las características de los sistemas de numeración binario, octal, decimal y hexadecimal.</li> <li>? Propiciar una discusión colectiva sobre la investigación realizada, para llegar a conclusiones.</li> <li>? Realizar conversiones entre los diferentes sistemas numéricos.</li> <li>? Realizar operaciones aritméticas en el sistema binario; adición y substracción en el octal y hexadecimal.</li> <li>? Relacionar los operadores aritméticos y booleanos.</li> <li>? Realizar analogías entre los circuitos eléctricos en serie, en paralelo y las compuertas lógicas.</li> </ul>	<p>Sistemas de numeración</p> <ul style="list-style-type: none"> <li>? Conversiones entre los sistemas de numeración.</li> <li>? Operaciones de adición, substracción, división y multiplicación en el sistema binario.</li> <li>? Operaciones de adición y substracción en los sistemas octal y hexadecimal.</li> </ul> <p>Elementos del álgebra de Boole</p> <ul style="list-style-type: none"> <li>? Conjunción, disyunción y negación.</li> <li>? Tablas de verdad de las funciones booleanas.</li> </ul>

<ul style="list-style-type: none"> <li>? Construye tablas de verdad de funciones booleanas y viceversa.</li> <li>? Construye o simula algunos autómatas.</li> </ul>	<ul style="list-style-type: none"> <li>? Representar en tablas de verdad, las entradas y salidas de los circuitos lógicos.</li> <li>? Señalar la importancia de la simplificación de funciones booleanas.</li> <li>? Realizar pruebas de circuitos lógicos con la tableta de experimentación o con un programa simulador de circuitos lógicos.</li> <li>? Diseñar y elaborar un circuito, por ejemplo, semisumador, sumador completo, contador binario o decimal.</li> <li>? Elaborar proyectos cibernéticos.</li> </ul>	<p>Compuertas y circuitos lógicos.</p> <ul style="list-style-type: none"> <li>? Componentes.</li> <li>? Circuito en serie y en paralelo.</li> <li>? Interruptores.</li> <li>? Compuertas lógicas.</li> <li>? Representación de las compuertas lógicas.</li> <li>? Funciones booleanas.</li> <li>? Representar la función booleana a partir de una tabla de verdad y/o circuito lógico.</li> <li>? Construcción o simulación de circuitos lógicos.</li> </ul>
---	--	--

## EVALUACIÓN

APRENDIZAJES	Evaluación Parcial	Evaluación Global
☒ Convierte números de un sistema de numeración a otro.	10%	2%
☒ Realiza operaciones de adición, sustracción, división y multiplicación en el sistema binario.	10%	2%
☒ Realiza operaciones booleanas (disyunción, conjunción y negación).	10%	2%
☒ Relaciona los operadores booleanos con las compuertas lógicas y las funciones.	10%	2%
☒ Describe los conceptos de interruptor, compuerta lógica, circuito eléctrico y circuito lógico.	10%	2%
☒ Construye tablas de verdad de funciones booleanas y viceversa.	20%	4%
☒ Construye o simula algunos autómatas.	30%	5%
<b>TOTAL</b>	<b>100%</b>	<b>19%</b>

## BIBLIOGRAFÍA BÁSICA

Kasatkin, V. *El ABC de la Cibernética*, Madrid, Paraninfo, 1976.

Morris Mano, M. *Lógica Digital y Diseño de Computadores*, México, Prentice-Hall Hispanoamericana, 1988.

Tocci, Ronald J. *Sistemas Digitales, Principios y Aplicaciones*, México, Prentice-Hall Hispanoamericana, 1987.

## BIBLIOGRÁFIA COMPLEMENTARIA

Floyd, T.L. *Fundamentos de Sistemas Digitales*, España, Prentice-Hall Internacional, 1998.

Heim, Klaus. *Álgebra de los circuitos lógicos*, España, DOSSAT, 1973.

Morris Mano, M. *Ingeniería Computacional, Diseño de Hardware*, México, Prentice-Hall Hispanoamericana. 1988.

Parr, E. A. *Circuitos Básicos del Ordenador*, Barcelona, CEAC, 1981.

## UNIDAD III. METODOLOGÍA DE SOLUCIÓN DE PROBLEMAS

**Propósito:**

- ✍ Al finalizar la unidad el alumno aplicará la metodología de solución de problemas mediante la construcción de algoritmos para formalizar el proceso de solución.

**TIEMPO:** 22 Horas

APRENDIZAJES	ESTRATEGIAS	TEMÁTICA
<p>El alumno:</p> <ul style="list-style-type: none"> <li>? Identifica los elementos que intervienen en los problemas y las relaciones entre ellos para obtener los resultados.</li> <li>? Describe las características de las etapas que intervienen en la solución de problemas.</li> <li>? Construye el algoritmo, el diagrama de flujo y el pseudocódigo para los problemas planteados.</li> <li>? Realiza la prueba de escritorio de los algoritmos desarrollados.</li> </ul>	<ul style="list-style-type: none"> <li>? Realizar una investigación documental sobre el ciclo de desarrollo de programas.</li> <li>? Propiciar una discusión colectiva sobre la investigación realizada, para identificar y describir las etapas del ciclo.</li> <li>? Plantear diferentes problemas para identificar los elementos que intervienen y las relaciones entre ellos para obtener los resultados.</li> <li>? Presentar propuestas de solución por parte de los alumnos.</li> <li>? Ejemplificar entre resolver un problema y utilizar una herramienta ya diseñada.</li> <li>? Representar el diagrama de flujo a pseudocódigo y viceversa.</li> <li>? Realizar la prueba de escritorio para validar el proceso lógico de solución del diagrama de flujo o del pseudocódigo.</li> </ul>	<p>Definiciones y conceptos generales</p> <ul style="list-style-type: none"> <li>? Problema.</li> <li>? Elementos y relaciones del problema.</li> <li>? Herramientas computacionales para la solución de problemas.               <ul style="list-style-type: none"> <li>✍ Hardware.</li> <li>✍ Sistema operativo.</li> <li>✍ Programas de aplicación.</li> <li>✍ Lenguajes de programación.</li> </ul> </li> </ul> <p>Ciclo de desarrollo de programas</p> <ul style="list-style-type: none"> <li>? Planteamiento del problema.</li> <li>? Análisis del problema.</li> <li>? Elaboración de algoritmos.</li> <li>? Codificación, edición y compilación.</li> </ul>

		<ul style="list-style-type: none"><li>? Ejecución y depuración.</li><li>? Documentación.</li><li>? Mantenimiento.</li></ul> <p>Expresiones y operadores</p> <ul style="list-style-type: none"><li>? Asignación.</li><li>? Operadores aritméticos.</li><li>? Operadores relacionales.</li><li>? Operadores lógicos.</li><li>? Precedencia de operadores y evaluación de expresiones.</li></ul> <p>Técnicas de desarrollo de algoritmos.</p> <ul style="list-style-type: none"><li>? Diseño descendente.</li><li>? Refinación progresiva de solución.</li><li>? Seudocódigo y diagrama de flujo.</li><li>? Prueba de escritorio.</li></ul>
--	--	--



## EVALUACIÓN

APRENDIZAJES	Evaluación Parcial	Evaluación Global
☞ Identifica los elementos que intervienen en los problemas y las relaciones entre ellos para obtener los resultados.	15%	5%
☞ Describe las características de las etapas que intervienen en la solución de problemas.	15%	5%
☞ Construye el algoritmo, el diagrama de flujo y el pseudocódigo para los problemas planteados.	35%	12%
☞ Realiza la prueba de escritorio de los algoritmos desarrollados.	30%	10%
<b>TOTAL</b>	<b>100%</b>	<b>34%</b>

## BIBLIOGRAFÍA BÁSICA

Cairó, Osvaldo. *Metodología de la Programación. Algoritmos, Diagramas de Flujo y Programas*, México, Alfaomega, 2003.  
Joyanes, Luis. *Fundamentos de la Programación. Algoritmos, Estructura de Datos y Objetos*, México, Mc. Graw-Hill, 2003.

## BIBLIOGRAFÍA COMPLEMENTARIA

Bores, Rosario. *Computación, Metodología, lógica Computacional y Programación*, México, Mc-Graw-Hill, 1995.  
Cairó, Osvaldo. *Metodología de la Programación, Algoritmos, Diagrama de Flujo y Programas*, México, Computec, ITAM, 1995.  
García Sánchez Luis y et al. *Construcción Lógica de programas. Teoría y Problemas Resueltos*. México, Alfaomega, 2003.  
Joyanes, Luis. *Problemas de Metodología de la Programación*, México, Mc Graw-Hill. 468 Problemas resueltos, 1990.  
Joyanes, Luis y et al. *Fundamentos de la Programación. Libro de Problemas en Pascal y Turbo Pascal*, México, Mc. Graw- Hill, 1997.  
Kruse, Robert L. *Estructura de Datos y Diseño de Programas*, México, Prentice Hall, 1988.  
Levine Gutiérrez, Guillermo. *Introducción a la Computación y a la Programación Estructurada*, México, Mc.Graw-Hill, 1989.

## UNIDAD IV. INTRODUCCIÓN A LA PROGRAMACIÓN Y AL LENGUAJE DE PROGRAMACIÓN PASCAL

### Propósito:

- ✍ Al finalizar la unidad el alumno utilizará de manera básica el lenguaje de programación Pascal, mediante la codificación de algoritmos a fin de comprender los fundamentos de la programación, destacando las características con otros lenguajes.

**TIEMPO:** 18 Horas

APRENDIZAJES	ESTRATEGIAS	TEMÁTICA
<p>El alumno:</p> <ul style="list-style-type: none"> <li>? Describe los antecedentes y características de los lenguajes de programación.</li> <li>? Describe las características del sistema operativo.</li> <li>? Utiliza sentencias básicas del lenguaje de programación Pascal.</li> <li>? Utiliza la sintaxis y semántica básica del lenguaje de programación.</li> <li>? Utiliza el ambiente de trabajo.</li> <li>? Codifica algoritmos básicos en Pascal.</li> </ul>	<ul style="list-style-type: none"> <li>? Realizar una investigación documental sobre los lenguajes de programación para describir la evolución, su clasificación, características, ventajas y desventajas, mediante cuadros sinópticos, tablas y esquemas.</li> <li>? Propiciar una discusión colectiva sobre la investigación realizada, para obtener conclusiones.</li> <li>? Realizar una práctica donde se utilicen los comandos para crear, cambiar, borrar, renombrar copiar directorios y dar formato.</li> <li>? Describir los elementos del lenguaje de programación: palabras reservadas, identificadores y sentencias básicas.</li> <li>? Ejemplificar algoritmos en los lenguajes de programación (máquina, ensamblador y alto nivel).</li> </ul>	<p>Lenguajes de programación</p> <ul style="list-style-type: none"> <li>? Clasificación de los lenguajes. <ul style="list-style-type: none"> <li>✍ Lenguaje máquina.</li> <li>✍ Lenguaje ensamblador.</li> <li>✍ Lenguaje de alto nivel: imperativo, funcional, declarativo, orientado a objetos, entre otros.</li> </ul> </li> <li>? Características básicas de cada nivel.</li> <li>? Sintaxis y semántica básicas.</li> </ul> <p>Sistema Operativo</p> <ul style="list-style-type: none"> <li>? Importancia</li> <li>? Características</li> <li>? Comandos.</li> </ul>

	<ul style="list-style-type: none"> <li>? Representar a las sentencias con el diagrama sintáctico y su codificación.</li> <li>? Describir los elementos del ambiente de trabajo del lenguaje de programación.</li> <li>? Codificar algoritmos básicos desarrollados en la unidad anterior y otros.</li> <li>? Editar, compilar y ejecutar los programas.</li> <li>? Ejemplificar el proceso de edición, compilación y ejecución de programas desarrollados en Pascal.</li> <li>? Resaltar las etapas de codificación, edición, almacenamiento, compilación y ejecución de un programa.</li> </ul>	<p>Elementos del lenguaje</p> <ul style="list-style-type: none"> <li>? Estructura de un programa.</li> <li>? Variables y constantes.</li> <li>? Identificadores.</li> <li>? Palabras reservadas.</li> <li>? Tipos de datos primitivos. <ul style="list-style-type: none"> <li>✗ Numéricos.</li> <li>✗ Caracter y cadena.</li> <li>✗ Lógicos.</li> </ul> </li> <li>? Sentencias: <ul style="list-style-type: none"> <li>✗ Lectura, asignación, escritura.</li> <li>✗ Estructura de la sentencia condicional simple (<i>If-then</i>).</li> </ul> </li> </ul> <p>Diagramas sintácticos básicos de las sentencias</p> <p>Ambiente de trabajo</p> <ul style="list-style-type: none"> <li>? Editor. <ul style="list-style-type: none"> <li>✗ Comandos básicos</li> <li>✗ Edición, compilación y generación de código.</li> <li>✗ Manejo de archivos.</li> </ul> </li> </ul>
--	--	---

## EVALUACIÓN

APRENDIZAJES	Evaluación	Evaluación
	Parcial	Global
Describe los antecedentes y características de los lenguajes de programación.	10%	2.8%
Describe las características del sistema operativo.	10%	2.8%
Utiliza sentencias básicas del lenguaje de programación Pascal.	15%	4.2%
Utiliza la sintaxis y semántica básica del lenguaje de programación.	15%	4.2%
Utiliza el ambiente de trabajo.	20%	5.6%
Codifica algoritmos básicos en Pascal.	30%	8.4%
<b>TOTAL</b>	<b>100%</b>	<b>28%</b>

## BIBLIOGRAFÍA BÁSICA

Cairó, Osvaldo. *Metodología de la Programación, Algoritmos, Diagrama de Flujo y Programas*, México, Computec, ITAM, 1995.

Levine Gutiérrez, Guillermo, *Introducción a la computación y a la programación estructurada*, McGraw-Hill.

Pratt, Terréense W. *Lenguajes de Programación. Diseño e Implementación*, México, Prentice-Hall Hispanoamericana, 1984.

Tucker, Allen B. *Lenguaje de Programación*, México, Mc.Graw-Hill, 1988.

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

## 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%
<b>TOTAL</b>	<b>100%</b>	<b>10 %</b>

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

## 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%
<b>TOTAL</b>	<b>100%</b>	<b>15 %</b>

## 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"> <li>? Comprende la importancia de dividir un programa en módulos.</li> <li>? Describe el concepto de programación modular, funciones, procedimientos y parámetros por valor, referencia y variable.</li> <li>? Identifica en la estructura de programas, la zona para la declaración e invocación de funciones y procedimientos.</li> <li>? Elabora programas utilizando la programación modular.</li> </ul>	<ul style="list-style-type: none"> <li>? Realizar una investigación documental sobre la programación modular.</li> <li>? Propiciar una discusión colectiva sobre la investigación realizada.</li> <li>? Resaltar la técnica de programación descendente o “divide y vencerás”</li> <li>? Ejemplificar programas sencillos donde se practiquen los procedimientos y funciones primero sin parámetros y de preferencia que los llamen varias veces.</li> <li>? Mostrar que el empleo de los procedimientos y funciones proporciona facilidad en el desarrollo de programas.</li> <li>? 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.</li> </ul>	<p>La modularidad como una herramienta para el manejo de programas</p> <ul style="list-style-type: none"> <li>? Concepto.</li> <li>? Importancia.</li> </ul> <p>Procedimientos y funciones</p> <ul style="list-style-type: none"> <li>? Concepto de procedimiento.</li> <li>? Concepto de función.                             <ul style="list-style-type: none"> <li>☞ Valor de regreso.</li> <li>☞ Tipos permitidos.</li> </ul> </li> <li>? Definición, declaración e invocación de procedimientos y funciones.</li> <li>? Parámetros por:                             <ul style="list-style-type: none"> <li>☞ Valor.</li> <li>☞ Referencia.</li> <li>☞ Variable.</li> </ul> </li> </ul>

<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"> <li>✗ Variables globales y locales.</li> <li>✗ Nombre de procedimientos y funciones.</li> </ul> <p>? Recursividad.</p> <ul style="list-style-type: none"> <li>✗ Concepto matemático</li> <li>✗ Implementación en el lenguaje.</li> <li>✗ Parte recursiva y parte terminal de un procedimiento recursivo.</li> </ul>
--	--	---

## 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%
<b>TOTAL</b>	<b>100%</b>	<b>15 %</b>

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

<p>? Elabora programas que involucran 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"> <li>✗ Campos y selección de elementos.</li> <li>✗ Almacenamiento de registros.</li> <li>✗ La sentencia With.</li> <li>✗ Registros variantes.</li> </ul> <p>Manejo de archivos</p> <ul style="list-style-type: none"> <li>? Medios de almacenamiento secundario.</li> <li>? Procesamiento secuencial. <ul style="list-style-type: none"> <li>✗ Archivos de tipo texto.</li> <li>✗ Funciones y procedimientos para el manejo de archivos de tipo texto.</li> </ul> </li> <li>? Procesamiento de tipo directo <ul style="list-style-type: none"> <li>✗ Archivos de acceso directo.</li> <li>✗ Funciones y procedimientos para el manejo de archivos de acceso directo.</li> </ul> </li> </ul> <p>Estructuras dinámicas</p> <ul style="list-style-type: none"> <li>? Apuntadores.</li> </ul>
--	--	---

## 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%
<b>TOTAL</b>	<b>100%</b>	<b>35 %</b>

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

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

## 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%
<b>TOTAL</b>	<b>100%</b>	<b>25 %</b>

## 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>.

**COMISIÓN DE REVISIÓN Y AJUSTE DE LOS PROGRAMAS DE CIBERNÉTICA Y COMPUTACIÓN**

Alfredo López Morales, Ignacio Vázquez Torre, Norma Angélica Andrade Díaz, Cecilio Rojas Espejo, Carlos Lorenzo Cuervo, Alfonso Anguiano Caballero, Asunción Reynoso Díaz, Verónica Lidya López Escobar, Claudia Durán Olmos, Horacio Colexcua García, Víctor Manuel Pérez Torres, Gilberto Fuentes Romero.

