UNIVERSIDAD NACIONAL EXPERIMENTAL DEL TÁCHIRA
VICE-RECTORADO ACADÉMICO
DECANATO DE DOCENCIA
INGENIERÍA INFORMÁTICA
PROGRAMA ANALÍTICO

Asignatura: Computación II **Departamento/Carreras:** Ingeniería Informática/Ingenierías: Electrónica y Mecánica **Código:** 0415201T

Unidad I: Programación Avanzada en Lenguaje C.

Objetivo General: Al finalizar la unidad, el estudiante será capaz de manipular técnicas de almacenamiento, codificación y decodificación para guardar, recuperar y transferir datos entre distintos sistemas computacionales.

Objetivos	Contenidos	Actividades	Evaluación	Recursos	Bibliografía
<p>Durante el desarrollo de la unidad el estudiante estará en capacidad de:</p> <p>1.1 Expresar de manera correcta lo que es un archivo construyendo su propia definición.</p> <p>1.2 Explicar la diferencia entre archivos de acceso secuencial y acceso directo en la manipulación de una lista de datos.</p> <p>1.3 Reconocer a través de ejercicios las diferencias entre los archivos de texto y binario para el acceso y recuperación de datos.</p>	<p>Definición de archivo.</p> <p>Organización de Archivos: Secuencial y Directo.</p> <p>Tipos de Archivo: Texto y Binario.</p>	<ul style="list-style-type: none"> • Lectura individual de la información suministrada en el material de apoyo. • Uso de la pizarra. • Exposición por parte del facilitador. • Interacción grupal para el análisis de los conceptos suministrados. • Mecánica de la pregunta. • Resolución de problemas propuestos por el facilitador en el aula (análisis, diseño y ejecución). 	<p>Prueba escrita de carácter individual. Consta de las siguientes partes:</p> <ul style="list-style-type: none"> • Corrida en frío. • Solución codificada de dos problemas. <p>Prácticas continuas en laboratorio.</p> <p>Valor del corte 30%:</p> <ul style="list-style-type: none"> • 75 puntos: Prueba escrita. • 25 puntos: Laboratorio. 	<ul style="list-style-type: none"> • Material de Apoyo. • Diapositivas o transparencias. • Marcadores. • Pizarra. • Borrador. • Computadora. 	<ul style="list-style-type: none"> • Deitel H., Deitel P. (1995). <i>Como Programar en C/C++ (2da. Edición)</i>. Prentice Hispanoamericana, S.A. México: D.F. • Guardati Caíro. (2002). <i>Estructuras de Datos</i>. Editorial: McGraw-Hill. México: D.F. • Schildt, Herbert. (1988). <i>Lenguaje C, Programación Avanzada</i>. Editorial: McGraw-Hill.

<p>1.4 Usar los distintos tipos de apertura existentes para el manejo de los archivos de texto o binario.</p>	<p>Modos de apertura.</p>	<ul style="list-style-type: none"> • Formulación de conclusiones por parte del estudiante y el facilitador. 			
<p>1.5 Expresar la respectiva sintaxis de las funciones para la lectura y escritura de archivos de texto o binario.</p>	<p>Funciones de escritura y lectura en un archivo.</p>	<ul style="list-style-type: none"> • Asignación de tópicos de investigación por parte del facilitador. 			
<p>1.6 Aplicar de manera correcta a un problema, el método de ordenación interna explicado en clase.</p>	<p>Ordenación interna.</p>	<ul style="list-style-type: none"> • Actividades prácticas dirigidas por el facilitador en la computadora. 			
<p>1.7 Aplicar de manera correcta a un problema, el método de ordenación externa explicado en clase.</p>	<p>Ordenación externa.</p>				
<p>1.8 Aplicar las operaciones básicas sobre archivos de texto o binario a un ejercicio que requieran de su uso.</p>	<p>Operaciones básicas sobre archivos (incluir, consultar, modificar, eliminar y listar).</p>				
<p>1.9 Emplear los archivos maestro e índice en la resolución de un problema planteado por el facilitador.</p>	<p>Archivos maestro e índice.</p>				

<p>1.10 Usar correctamente los operadores lógicos y desplazamiento para la resolución de operaciones a nivel de bits.</p>	<p>Operaciones a Nivel de Bits: Operadores lógicos y de desplazamiento.</p>				
<p>1.11 Convertir valores dados en un determinado sistema de numeración a otro sistema de representación.</p>	<p>Conversión entre sistemas de numeración.</p>				
<p>1.12 Aplicar correctamente la operación de enmascaramiento a casos particulares que requieran de su uso.</p>	<p>Enmascaramiento.</p>				
<p>1.13 Usar una estructura de datos con campos de bits en el desarrollo de un programa.</p>	<p>Campos de bits.</p>				

Unidad II: Programación Orientada a Objetos en C++.

Objetivo General: Al concluir la unidad, el estudiante estará en capacidad de construir modelos básicos del mundo real y programas informáticos basados en tales modelos de objetos.

Objetivos	Contenidos	Actividades	Evaluación	Recursos	Bibliografía
<p>Durante el transcurso explicativo de la unidad el estudiante será capaz de:</p> <p>2.1 Explicar los conceptos y fundamentos de C++ construyendo sus propias definiciones y ejemplificaciones.</p> <p>2.2 Manipular una clase y sus respectivos miembros en la solución de un problema facilitado por el docente.</p> <p>2.3 Establecer las diferencias para el uso correcto de objetos simples, arreglos de objetos y punteros de objetos en la solución de un problema.</p> <p>2.4 Establecer funciones constructoras y destructoras en el diseño de un programa, para la inicialización y eliminación</p>	<p>Conceptos y Fundamentos de C++: Objetos, métodos, mensajes, encapsulamiento, abstracción, polimorfismo y herencia.</p> <p>Definición de clase. Miembros: públicos y privados.</p> <p>Declaración de objetos simples, arreglos de objetos y punteros a objetos. Parámetros de objetos en funciones.</p> <p>Constructores y Destructores.</p>	<ul style="list-style-type: none"> • Inducción del tema a tratar a través de la exposición. • Uso de la pizarra. • Uso de ejemplos. • Presentación y discusión de conceptos. • Aplicación de los procedimientos planteados en la resolución de problemas. • Asignación de ejercicio como actividad de cierre para la retentiva de los conceptos explicados por el facilitador. • Resolución de problemas propuestos por el facilitador en el aula (análisis, diseño y ejecución). 	<p>Prueba escrita de carácter individual. Consta de las siguientes partes:</p> <ul style="list-style-type: none"> • Corrida en frío. • Solución codificada de dos problemas. <p>Prácticas continuas en laboratorio.</p> <p>Valor del corte 35%:</p> <ul style="list-style-type: none"> • 75 puntos: Prueba escrita. • 25 puntos: Laboratorio. 	<ul style="list-style-type: none"> • Material de Apoyo. • Diapositivas o transparencias. • Marcadores. • Pizarra. • Borrador. • Computadora. 	<ul style="list-style-type: none"> • Deitel H., Deitel P. (1995). <i>Como Programar en C/C++ (2da. Edición)</i>. Prentice Hispanoamericana, S.A. México: D.F. • Goldstein L. (1993). <i>Introducción a la Programación Orientada a Objetos</i>. Editorial: Prentice Hispanoamericana, S.A. México: D.F. • Jamsa Kris. (1997). <i>C++ Paso a Paso</i>. Editorial: Alfaomega. México: D.F.

<p>de objetos de una clase.</p> <p>2.5 Aplicar la sobrecarga de funciones constructoras en la inicialización de objetos de una clase.</p> <p>2.6 Programar correctamente funciones amigas en la solución de un problema en particular, para la restricción de acceso a los miembros privados de una clase.</p> <p>2.7 Ejecutar un programa dado en clase con archivos de texto donde se manejen los operadores de inserción y extracción de flujo de datos.</p> <p>2.8 Aplicar los tipos de herencia y sus respectivos miembros protegidos en la solución de un problema facilitado por el docente.</p> <p>2.9 Resolver un ejercicio donde un objeto puede cambiar de forma conforme se ejecuta el programa.</p>	<p>Sobrecarga de funciones constructoras.</p> <p>Funciones amigas.</p> <p>Archivos en C++: Operadores de inserción y extracción. Modos de apertura de archivos en C++.</p> <p>Herencia: Simple, múltiple y cadena de herencia. Miembros protegidos.</p> <p>Funciones virtuales y polimorfismo.</p>	<ul style="list-style-type: none"> • Actividades prácticas dirigidas por el facilitador en la computadora. 			<ul style="list-style-type: none"> • Joyanes Aguilar, Luis. (1998). <i>Programación Orientada a Objetos</i>. Editorial: McGraw-Hill. España: Madrid.
--	--	---	--	--	---

<p>2.10 Aplicar las plantillas de clases y funciones en un problema suministrado en clase con el propósito de no duplicar código cuyos objetos difieren únicamente en el tipo de dato.</p> <p>2.11 Elaborar diagramas de clase y la jerarquía de herencia, previa programación y ejecución de un problema.</p>	<p>Plantillas de clases y funciones.</p> <p>Introducción básica a UML: Diagramas de clase y jerarquía de herencia.</p>				
--	--	--	--	--	--

Unidad III: Programación Básica con C++ Builder.

Objetivo General: Al finalizar la unidad, el estudiante será capaz de realizar aplicaciones con el uso de interfaces gráficas diseñadas por el programador o predefinidas por el sistema operativo Windows.

Objetivos	Contenidos	Actividades	Evaluación	Recursos	Bibliografía
<p>Durante el desarrollo de la unidad el estudiante estará en capacidad de:</p> <p>3.1 Identificar cada elemento de la interfaz de C++ Builder, explicando su funcionamiento y uso en el diseño de un proyecto.</p> <p>3.2 Manejar elementos de la aplicación C++ Builder para la personalización de su interfaz.</p> <p>3.3 Realizar programas básicos de entrada y salida de datos sin el uso de componentes visuales de C++ Builder.</p> <p>3.4 Programar correctamente componentes receptores de texto, para la solución de problemas particulares que requieran su uso.</p>	<p>Fundamentos de Interfaz: Formularios, propiedades, eventos y métodos.</p> <p>Depósito de objetos. Gestión de proyectos. Gestión de tareas.</p> <p>Novedades de C++ Builder en el diseño de programas básicos: WinMain, AnsiString, c_str, ShowMessage, InputBox, new y delete.</p> <p>Componentes Receptores de Texto: Label, Edit, MaskEdit, Memo, ListBox, ComboBox y DrawGrid.</p>	<ul style="list-style-type: none"> • Lectura grupal de la información suministrada en el material de apoyo. • Uso de material audiovisual. • Exposición por parte del facilitador. • Aplicación de los propiedades, eventos y métodos de los componentes presentados por medio de la resolución de problemas. • Mecánica de la pregunta. • Formulación de conclusiones por parte del estudiante y el facilitador. • Asignación de tópicos de investigación por 	<p>Proyecto (trabajo práctico) de carácter grupal. Consta de las siguientes partes:</p> <ul style="list-style-type: none"> • Presentación del proyecto. • Defensa del proyecto. <p>Prácticas continuas en laboratorio.</p> <p>Valor del corte 35%:</p> <ul style="list-style-type: none"> • 75 puntos: Presentación y defensa del proyecto a través de una escala de estimación. • 25 puntos: Laboratorio. 	<ul style="list-style-type: none"> • Material de Apoyo. • Diapositivas o transparencias. • Marcadores. • Pizarra. • Borrador. • Computadora. 	<p>Charte Francisco. (2000). <i>Programación con C++ Builder 5</i>. Ediciones Anaya Multimedia. España: Madrid.</p>

<p>3.5 Programar correctamente componentes para iniciar órdenes, para la solución de problemas particulares que requieran su uso.</p>	<p>Componentes para Iniciar Órdenes: Button, BitBtn, SpeedButton, ToolBar y ImageList.</p>	<p>parte del facilitador.</p> <ul style="list-style-type: none"> • Actividades prácticas dirigidas por el facilitador en la computadora. 			
<p>3.6 Programar correctamente componentes de selección y control, para la solución de problemas particulares que requieran su uso.</p>	<p>Componentes de Selección y Control: CheckBox, RadioButton, GroupBox, RadioGroup, CheckListBox, Image, UpDown, Animate, DateTimePicker y ProgressBar.</p>				
<p>3.7 Emplear menús principales y emergentes en el diseño de un proyecto suministrado por el facilitador.</p>	<p>Creación de Menús (principales y emergentes): MainMenu, PopupMenu y ActionList.</p>				
<p>3.8 Realizar una aplicación que requiera el uso de varios formularios que interactúan entre sí para la entrada y salida de datos.</p>	<p>Proyectos Multiformulario.</p>				

<p>3.9 Manejar cuadros de diálogo predefinidos por el sistema operativo Windows en la simplificación de la solución de problemas particulares que requieran su uso.</p>	<p>Cuadros de Diálogo: Abrir y guardar archivos, atributos de fuente, opciones de color, imprimir, buscar y reemplazar texto.</p>				
<p>3.10 Usar herramientas para la construcción de archivos de ayuda para la documentación, uso y manejo de un proyecto asignado en clase.</p>	<p>Archivos de Ayuda.</p>				
<p>3.11 Usar el puerto de red para el intercambio de datos.</p>	<p>Componentes de Internet: ClientSocket y ServerSocket.</p>				
<p>3.12 Usar un procedimiento para el empaquetado de proyectos elaborados en C++ Builder.</p>	<p>Empaquetado de proyectos.</p>				

Aprobado por:

Jefe del Departamento

Jefe del Núcleo