



Universidad Centroccidental "Lisandro Alvarado"

Decanato de Ciencias y Tecnología

Departamento de Sistemas



PROGRAMA INSTRUCCIONAL

PROGRAMA: ANALISIS DE SISTEMAS		DEPARTAMENTO: SISTEMAS	
ASIGNATURA: INTRODUCCIÓN A LA COMPUTACION		AREA COORDINACIÓN: INTRODUCCION A LA COMPUTACION	
AREA CURRICULAR: CONOCIMIENTOS		EJE CURRICULAR: COMPUTACION	
CODIGO: C1	SEMESTRE: I	CARÁCTER: OBLIGATORIA	
HORAS TEORICAS: 2	HORAS PRACTICAS: 2	PRE-REQUISITOS: NINGUNA	
CREDITOS: 3			
PROFESORES: CAMACARO, VLADIMIR GIMENEZ, OSCAR PIÑERO, ERYs		COORDINADOR: PIÑERO, ERYs	
FECHA ELABORACION: 01- 2008	FECHA ULTIMA REVISION: 11 -2015	LAPSO ACADEMICO: 2015-1	

FUNDAMENTACIÓN

Introducción a la Computación es una materia básica en el área de computación y se imparte en el primer semestre del programa de Análisis de Sistemas. Forma parte de las asignaturas del eje curricular de computación. Es pre-requisito de la asignatura de Programación.

El curso tiene como finalidad formar al estudiante en los conceptos y herramientas necesarias para iniciarse en el entorno informático y en el desarrollo de programas mediante el análisis, diseño e implementación de modelos orientados a objetos.

Objetivo General

- Conocer la estructura y funcionamiento del computador.
- Manejar los conceptos, métodos y herramientas para la solución de problemas.
- Desarrollar en el estudiante habilidades y destrezas para la construcción de soluciones, a problemas reales, haciendo uso de la metodología orientada a objetos.

UNIDAD I: Componentes físicos y lógicos de un computador. Duración: 1 Semana Ponderación: 4,00%	OBJETIVO TERMINAL: Conocer el ambiente, los componentes y el funcionamiento básico de un computador,	
OBJETIVOS ESPECÍFICOS	CONTENIDO	ESTRATEGIAS DE ENSEÑANZA/APRENDIZAJE
<ol style="list-style-type: none"> 1. Manejar los conceptos básicos de la computación. 2. Identificar los componentes físicos y lógicos de un computador. 3. Identificar los pasos para la ejecución de un programa en un computador. 	<ol style="list-style-type: none"> 1. Definición de Computador. 2. Componentes Físicos (Hardware). <ol style="list-style-type: none"> a. Unidad Central de Procesamiento. b. Memoria y sus Tipos. c. Dispositivos Periféricos. d. Evolución del Hardware 3. Componentes Lógicos (Software) <ol style="list-style-type: none"> a. Clasificación según su naturaleza <ul style="list-style-type: none"> • Datos e Información. • Instrucciones y Programas b. Clasificación según su función. <ul style="list-style-type: none"> • Sistemas Operativos: • Lenguajes de Programación • Programas de Aplicación. c. Tipos de Programas. <ul style="list-style-type: none"> • Fuente y Ejecutable d. Tipos de Lenguajes de Programación. <ul style="list-style-type: none"> • De Máquina. • De Bajo Nivel - Ensamblador • De Alto Nivel – Intérpretes/Compilador e. Pasos para la Ejecución de un programa 	<ol style="list-style-type: none"> 1. Exposición del Profesor. 2. Discusión Grupal. 3. Trabajo de Investigación.

UNIDAD II: Algoritmos y Programación DURACIÓN: 1 Semana. Ponderación : 4,00%	OBJETIVO TERMINAL: Explicar los elementos básicos de la algoritmia y de la programación.	
OBJETIVOS ESPECÍFICOS	CONTENIDO	ESTRATEGIAS DE ENSEÑANZA/APRENDIZAJE
<ol style="list-style-type: none"> 1. Plantear algoritmos como solución a problemas cotidianos. 2. Conocer los elementos básicos de la Programación. 3. Comprender la interacción entre un Programa y el Computador 	<ol style="list-style-type: none"> 1. Algoritmos. <ol style="list-style-type: none"> a. Definición y Características. b. Planteamiento de algoritmos como solución a problemas cotidianos. 2. Programación. <ol style="list-style-type: none"> a. Definición. b. Tipos de Datos. c. Sistemas Numéricos. d. Operadores y su Orden de Evaluación. e. Expresiones Aritméticas y Lógicas. f. Identificadores, Constantes, Variables, Contadores y Acumuladores. g. Instrucciones: <ul style="list-style-type: none"> • De Entrada y Salida. • Secuenciales y Selectivas • Repetitivas. 3. Interacción Programa – Computador. <ul style="list-style-type: none"> • Instrucciones de programa y el CPU. • Área de Memoria para Datos • Área de Memoria para Instrucciones. 	<ol style="list-style-type: none"> 1. Exposición por parte del profesor. 2. Ejercicios prácticos. 3. Dinámicas.

UNIDAD III: Paradigma Orientado a Objeto para la resolución de problemas. Duración: 1 Semana Ponderación: 5,00%	OBJETIVO TERMINAL: Caracterizar el Paradigma Orientado a Objeto y sus beneficios como enfoque de programación.	
OBJETIVOS ESPECÍFICOS	CONTENIDO	ESTRATEGIAS DE ENSEÑANZA/APRENDIZAJE
<ol style="list-style-type: none"> 1. Conocer los fundamentos generales de la Orientación a Objetos. 2. Describir situaciones del mundo real como un conjunto de objetos relacionados entre si. 	<ol style="list-style-type: none"> 1. Definición de Programación Orientada a Objeto. 2. Ventajas de la Programación Orientada a Objeto. 3. Definición de Clase, Atributo, Método y Objeto. 4. Propiedades de la Programación Orientada a Objeto. <ol style="list-style-type: none"> a. Abstracción. b. Encapsulamiento / Ocultamiento c. Modularidad. 5. Clases, objetos y sus interrelaciones dentro del mundo real 	<ol style="list-style-type: none"> 1. Exposición del Profesor. 2. Ejemplos para identificar objetos, atributos y métodos.

UNIDAD IV. Metodología para la resolución de problemas bajo el enfoque orientados a objetos. Duración: 2 semana Ponderación: 12%	OBJETIVO TERMINAL: Construir modelos empleando el Análisis y el Diseño Orientados a Objetos como solución a problemas de índole computacional.	
OBJETIVOS ESPECÍFICOS	CONTENIDO	ESTRATEGIAS DE ENSEÑANZA/APRENDIZAJE
<ol style="list-style-type: none"> 1. Conocer los pasos para la resolución de problemas bajo el enfoque Orientado a Objeto. 2. Realizar el Análisis y el Diseño bajo el enfoque Orientado a Objeto. 	<ol style="list-style-type: none"> 1. Metodología para la construcción de algoritmos. <ol style="list-style-type: none"> a. Análisis: Identificación de clase, atributos y métodos. b. Diseño: Representación gráfica de la clase. 2. Desarrollo de Ejercicios, con una y dos clases de objetos, utilizando el Análisis y el Diseño Orientado a Objeto. 	<ol style="list-style-type: none"> 1. Exposición por parte del profesor. 2. Ejemplos aplicando la metodología.

<p>UNIDAD V: Implementación de modelos orientados a objetos utilizando estructuras secuenciales y selectivas. Duración: 5 Semanas Ponderación: 35,00%</p>	<p>OBJETIVO TERMINAL: Construir Modelos empleando el Análisis, el Diseño y la Implementación Orientados a Objetos, haciendo uso de estructuras Secuenciales y Selectivas.</p>	
<p>OBJETIVOS ESPECÍFICOS</p>	<p>CONTENIDO</p>	<p>ESTRATEGIAS DE ENSEÑANZA-APRENDIZAJE</p>
<ol style="list-style-type: none"> 1. Conocer las reglas sintácticas y semánticas de la herramienta de implementación de programas bajo el enfoque Orientado a Objeto 2. Desarrollar Modelos Orientados a Objeto, aplicando la metodología propuesta, que requieran la utilización de estructuras secuenciales. 3. Desarrollar Modelos Orientados a Objeto, aplicando la metodología propuesta, que requieran la utilización de estructuras selectivas simples y anidadas. 4. Comprobar el funcionamiento de los Modelos Elaborados. 	<ol style="list-style-type: none"> 1. Reglas sintácticas y semánticas de la herramienta de programación. <ol style="list-style-type: none"> a. Ejercicios simples utilizando la herramienta de programación propuesta. 2. Estructuras Secuenciales y Selectivas. <ol style="list-style-type: none"> a. Sintaxis de las estructuras Secuenciales. b. Sintaxis de las estructuras Selectivas Simple. c. Sintaxis de las estructuras Selectivas Anidadas. 3. Resolución de Problemas, con una y dos clases de objetos, utilizando estructuras secuenciales y selectivas. 4. Prueba y/o simulación de los Modelos elaborados. 	<ol style="list-style-type: none"> 1. Exposición por parte del profesor. 2. Resolución de los problemas en clases. 3. Ejercicios propuestos. 4. Talleres grupales.

<p>UNIDAD VI: Desarrollo de algoritmos orientados a objetos con estructuras iterativas o repetitivas Duración: 6 Semanas. Ponderación: 40%</p>	<p>OBJETIVO TERMINAL: Construir Modelos empleando el Análisis, el Diseño y la Implementación Orientados a Objetos, haciendo uso de estructuras iterativas o repetitivas</p>	
<p>OBJETIVOS ESPECIFICOS</p>	<p>CONTENIDO</p>	<p>ESTRATEGIAS DE ENSEÑANZA-APRENDIZAJE</p>
<ol style="list-style-type: none"> 1. Desarrollar Modelos Orientados a Objeto, aplicando la metodología propuesta, que requieran la utilización de estructuras repetitivas simples. 2. Desarrollar Modelos Orientados a Objeto, aplicando la metodología propuesta, que requieran la utilización de estructuras repetitivas dobles. 3. Comprobar el funcionamiento de los Modelos Elaborados. 	<ol style="list-style-type: none"> 1. Estructuras Repetitivas. <ol style="list-style-type: none"> a. Sintaxis de las Estructuras Repetitivas Simples. b. Sintaxis de las Estructuras Repetitivas Dobles c. Utilización de Contadores y Acumuladores d. Formas de Controlar los Ciclos. <ul style="list-style-type: none"> • Dato Centinela. • Contadores. • Interacción con el Usuario. • Ruptura de Control. 2. Resolución de Problemas, con dos clases de objetos, utilizando estructuras Repetitivas Simples. 3. Resolución de Problemas, con tres clases de objetos, utilizando estructuras Repetitivas Dobles 4. Prueba y/o simulación de los Modelos elaborados con estructuras Repetitivas. 	<ol style="list-style-type: none"> 1. Revisión de Literatura. 2. Exposición por parte del profesor. 3. Resolución de los problemas en clases. 4. Ejercicios propuestos. 5. Talleres grupales.

PLAN DE EVALUACIÓN

SEMANA	UNIDAD	OBJETIVO	ESTRATEGIAS DE EVALUACIÓN				PONDERACIÓN CORTE 1 - 25%
			TECNICAS	INSTRUMENTOS	ACTIVIDADES	TIPO DE EVALUACIÓN	
1 - 5	I III	1,2,3 1,2	Evaluación continua	<ul style="list-style-type: none"> • Intervención en clases. • Resolución de problemas. • Trabajo de investigación. 	Según Instrumento	Sumativa	5 %
6	II IV	1,2,3 1,2	Primera Prueba Larga	Prueba Escrita	Aplicación de Prueba	Sumativa	20 %

SEMANA	UNIDAD	OBJETIVO	ESTRATEGIAS DE EVALUACIÓN				PONDERACIÓN CORTE 2 - 35%
			TECNICAS	INSTRUMENTOS	ACTIVIDADES	TIPO DE EVALUACIÓN	
6 - 11	V	1..4	Evaluación continua	<ul style="list-style-type: none"> • Intervención en clases. • Resolución de problemas. • Trabajo de investigación. 	Según Instrumento	Sumativa	10 %
11	V	1..4	Segunda Prueba Larga	Prueba Escrita	Aplicación de Prueba	Sumativa	25 %

SEMANA	UNIDAD	OBJETIVO	ESTRATEGIAS DE EVALUACIÓN				PONDERACIÓN CORTE 3 - 40%
			TECNICAS	INSTRUMENTOS	ACTIVIDADES	TIPO DE EVALUACIÓN	
12 - 16	VI	3	Evaluación Continua	<ul style="list-style-type: none"> • Resolución de problemas. • Trabajo de investigación. • Taller 	Aplicación del Instrumento	Sumativa	10 %
14	VI	1	Prueba Corta (Quiz)	<ul style="list-style-type: none"> • Prueba Escrita 	Aplicación de Prueba	Sumativa	10 %
17	VI	2	Tercera Prueba Larga	Prueba Escrita	Aplicación de Prueba	Sumativa	20 %

REFERENCIAS

- GRADY BOOCH. Análisis y diseño orientado a objetos con aplicaciones. Addison-Wesley. 1996
- JOYANES AGUILAR, Luís. Fundamentos de la Programación. McGraw Hill. Tercera Edición.2003
- JOYANES Aguilar, LUIS. Programación en C++ Algoritmos, estructuras de datos y objetos. McGraw Hill. Segunda Edición. 2006
- RUMBAUGH. Modelado y diseño orientado a objeto. Prentice Hall.1996