



I. DATOS DE IDENTIFICACIÓN:

Proyecto de Carrera:		Ingeniería en Informática					
Programa de Estudio:		TSU	Licenciado	Ingeniero	X	Diplomado	
Unidad Curricular:		Fundamentos de la Informática					
Semestre	Código	Unidad Crédito	Horas Semanales		Horas Semestre		
I			6				

Componente de Formación:	General		Profesional Básica		Profesional Especializada	X
	Práctica Profesional:		Pasantía:			

Carácter de la Unidad Curricular:	Obligatoria	X	Electiva	
Requisitos para Cursar la Unidad Curricular (Prelaciones):				

Elaborado por: William Mercado, Julio Canelón			Fecha: 12 de diciembre 2013		
VºBº Coordinador (a) del Proyecto de Carrera:					
Nombre:	William Mercado		Firma:		
VºBº Coordinador (a) de Currículo:					
Nombre:	Holanda García Cova		Firma:		



II. PROPÓSITO:

El ingeniero en informática debe poseer la capacidad de análisis y diseño de soluciones a problemas basados en programación secuencial y estructurada, plasmando estas soluciones en algoritmos escritos en pseudocódigo como herramientas para la resolución de problemas. Para esto requiere comprender los conceptos básicos inherentes a la informática, comprender y aplicar, nociones y técnicas para el desarrollo de algoritmos y pseudocódigos.

III. COMPETENCIAS GENÉRICAS:

- * Capacidad para identificar, plantear y resolver problemas a través de diferentes métodos.
- * Planificar, controlar y evaluar proyectos en su organización o medio socio-cultural.
- * Respeto a la diversidad.
- * Identidad Institucional.
- * Capacidad crítica.
- * Comunicación escrita y verbal.
- * Comprender, decodificar e interpretar lenguaje formal y simbólico, y entender su relación con el lenguaje natural.
- * Manejo y aplicación de las nuevas tecnologías de la información y la comunicación.
- * Abstraer y modelar realidades del entorno.
- * Lectura comprensiva y crítica.

IV. COMPETENCIAS PROFESIONALES:

- * Desarrolla soluciones informáticas con base en productos de hardware y software para resolver problemas computacionales en el entorno empresarial y profesional de manera efectiva y eficiente.

V. COMPETENCIAS DE LA UNIDAD CURRICULAR:

- * Diseña soluciones algorítmicas en pseudocódigo, empleando técnicas de programación secuencial y estructurada con la finalidad de resolver problemas computacionales, matemáticos y lógicos de forma efectiva.

VI. VALORES Y ACTITUDES:

- * Respeto a la diversidad.
- * Desarrollo de la Identidad Institucional.
- * Conciencia ética y Honestidad
- * Responsabilidad social.

VII. TEMARIO:

- Conceptos de Informática
- Historia y evolución contemporánea de las tecnologías de información y comunicación -TIC-
- Resolución de problemas y diseño de algoritmos

VIII. CONTENIDO DETALLADO POR TEMA:

- Conceptos de Informática
 - Informática, Computación y Sistemas como ciencias. (b) Como áreas de desarrollo profesional. (c) Actividades específicas de cada especialidad.
 - Definición del hardware. Esquema básico del hardware: dispositivos de entrada y salida, bus de sistema (puerto de comunicación, almacenamiento), dispositivos de almacenamiento primario y secundario, unidad central de proceso (ALU, UC).



- Definición de software. Clasificación: de acuerdo al método de distribución (software libre, propietario, gratis, shareware). De acuerdo con el uso en el computador y necesidades de usuarios (sistemas aplicación, programación: traductores, compiladores e intérpretes; software base: sistemas operativos, clasificación según su interfaz y según **cantidad de usuarios**. Definición y diferencia entre dato e información.
- Los sistemas de numeración: romano, arábigo (binario, octal, decimal, hexadecimal); conversión entre bases numéricas. El sistema binario como origen de datos, información, y lenguaje básico en un computador. Codificación de la Información: bit, byte.
- Unidades de medida en el computador: bit, byte, kilobyte, megabyte, gigabyte, terabyte, pentabyte, zetabyte. Conversiones entre estas unidades.
- Unidades de medida para el procesamiento de información: herz, kilohertz, Mega**herz**
- Historia y evolución contemporánea de las tecnologías de información y comunicación -TIC-.
 - Prehistoria del computador (generación 0). Elementos que antecedieron y promovieron la computación.
 - Generaciones del computador: características, elementos de software y hardware precursores de cada generación, elementos que incidieron en el surgimiento de cada generación.
 - Tendencias actuales de la informática.
- Resolución de problemas y diseño de algoritmos
 - Identificadores (reglas genéricas para escribir un identificador)
 - Variables y constantes: diferencias, declaración.
 - Tipo de datos (básicos: entero, real, carácter, cadena, lógico)
 - Operadores (asignación, aritméticos: suma, resta, multiplicación, división entera, división real, módulo o resto; relacionales: mayor que, menor que, mayor igual que, menos igual que, diferente de, igual que; lógicas: AND, OR, NOT, tabla de la verdad). Precedencia de operadores.
 - Expresiones (aritméticas, relacionales, lógicas). Evaluación de expresiones.
 - Algoritmos: definición, tipos: gráficos (diagrama de flujo: símbolos y reglas), no gráficos (lenguaje común).
 - Metodología para la resolución de problemas en cuatro pasos: (1) Leer y entender el problema; (2) determinar entrada – proceso – salida; (3) escribir la solución; (4) considerar validaciones. (5) corrida en frío.
 - Programación: definición, programación secuencial. Estrategias de programación: bottom – up, top – down, divide y vencerás.
 - Pseudocódigo: definición, estructura general: identificación del pseudocódigo, declaración de constantes, declaración de variables, cuerpo del programa, inicialización de variables, procesos, fin de programa).
 - Programación estructurada:
 - Estructuras condicionales: simples, dobles, múltiples y anidadas.
 - Estructuras de control iterativas: para, mientras y repetir. Diferenciar en qué caso se debe usar en cada estructura.
 - Contadores y acumuladores.

IX. ESTRATEGIAS DIDÁCTICAS:

1. ESTRATEGIAS DOCENTES:

- * Uso de medios audiovisuales en clase (computador, video beam, video)
- * Comunicación masiva, entonación de voz y énfasis al explicar.
- * Discusiones grupales, Formulación de preguntas y repreguntas.
- * Ejercicios con explicación detallada, resolución de ejercicios guiados.



- * Uso herramientas Web 2.0 para la interacción fuera del aula, consultas, aclaratoria de dudas y entrega de material de estudio.
- * Realización de talleres y prácticas continuas.

2. ESTRATEGIAS DE APRENDIZAJES

- * Análisis de problemas. Problemas propuestos. Algoritmos resueltos.
- * Aprender – enseñando
- * Resolución de problemas en clase por parte de los alumnos (Problemas guiados y problemas supervisados)
- * Aprender haciendo, estudio de problemas y desarrollo de soluciones.
- * Realización de ejercicios entregados por el docente, trabajo en grupo.
- * Investigación.

X. PLAN DE EVALUACIÓN:

- * Pruebas contextualizadas escritas, donde se solicita información al alumno, para evaluar el área cognoscitiva. Estas preguntas requerirán la interpretación de la realidad, basándose en el contenido del programa de estudio.
- * Proyectos con estudio de casos, donde el alumno refleje los cambios producidos en el campo cognoscitivo y demuestre las habilidades que ha desarrollado y la información que ha integrado.
- * Generar una situación real o simulada, de forma que se observe en forma directa si el estudiante la identifica, realiza la adecuada selección de técnicas e instrumentos y la ejecución e integración del conocimiento adquirido en función del producto. Y así conocer, si es el caso, el origen de sus aciertos y errores.

XI. REFERENCIAS BIBLIOGRÁFICAS:

- * Beekman, George. (2000) Introducción a la Computación. Addison Wesley Longman.
- * Banahan, Mike; Brady, Declan y Doran, Mark (1991). The C Book (2nd ed.). Addison-Wesley.
- * Brassard, Gilles y Bratley Paul. (2000) Fundamentos de Algoritmia. Prentice Hall; 1st. Edition.
- * Cairó, Osvaldo. (2005) Metodología de la Programación: Algoritmos, Diagramas de Flujos y Programas. Tomo I. Alfaomega.
- * Farrel, Joyce. (2001) Introducción a la Programación. International Thomson Learning.
- * Joyanes, Aguilar Luis (2003) Fundamentos de programación. (3ra. Ed.) McGraw-Hill Interamericana.
- * Kernighan, Brian y Dennis, Ritchie (1988). The C Programming Language (2nd ed.). Prentice Hall.
- * King, K. N. (2008). C Programming: A Modern Approach (2nd ed.). W.W. Norton & Company.
- * Norton, Peter. (2001) Introducción a la Computación. Mc Graw Hill Interamericana.