

1.- DATOS DE LA ASIGNATURA

Nombre de la asignatura:	Curso propedéutico: Introducción a la Ing. en Sistemas Computacionales y al diseño orientado a objetos
Carrera:	Ingeniería en Sistemas Computacionales
Clave de la asignatura:	Propedéutico
Duración:	40 hrs.

2.- PRESENTACIÓN

Caracterización de la asignatura.

Este curso introduce al candidato a estudiar la carrera de ISC en el campo de la computación, mostrándole el objetivo de la carrera, el perfil del egresado, el plan de estudios, los módulos de especialidad y el campo de acción, con la finalidad de aclarar posibles dudas y confirmar sus aspiraciones de cursar esta carrera profesional.

El curso aporta al candidato la capacidad de desarrollar pensamiento lógico mediante el análisis y diseño de estrategias de solución a problemas sencillos a través de algoritmos computacionales orientados a objetos.

El curso proporciona al candidato una herramienta para resolver problemas de aplicaciones de la vida ordinaria y de aplicaciones de la ingeniería.

El contenido de este curso proporciona el soporte directo de las asignaturas: Fundamentos de Programación, Programación Orientada a Objetos, Estructuras de Datos, Tópicos Avanzados de Programación y de forma indirecta se relaciona con el Desarrollo de Sistemas de Software, Sistemas Operativos y Programación de Sistemas.

Intención didáctica.

- Es importante que el profesor explique claramente las actividades a desarrollar por un profesional de la Ing. en Sistemas Computacionales, el objetivo de la carrera, el perfil del egresado, el plan de estudios, los módulos de especialidad y el campo de acción.
- El curso debe llevarse de manera teórico-práctica, donde el profesor cubre completamente la parte teórica y plantea los ejercicios a resolver por parte de los estudiantes.

3.- COMPETENCIAS A DESARROLLAR

<p>Competencias específicas:</p> <p>El estudiante conocerá ampliamente la carrera y el campo de acción de la Ing. en Sistemas Computacionales.</p> <p>Conocerá el plan de estudios y los módulos de especialidad de la carrera.</p> <p>Desarrollará habilidad para resolver problemas a través de la computadora utilizando la metodología y sus etapas (análisis, diseño e implementación).</p>	<p>Competencias genéricas</p> <p>1- Competencias instrumentales:</p> <ul style="list-style-type: none">▪ Capacidad de análisis y síntesis▪ Capacidad de organizar y planificar▪ Conocimientos generales básicos▪ Conocimientos básicos de la carrera▪ Comunicación oral y escrita en su propia lengua▪ Habilidades básicas de manejo de la computadora▪ Habilidades de gestión de información (habilidad para buscar y analizar información proveniente de fuentes diversas)▪ Solución de problemas▪ Toma de decisiones. <p>2-Competencias interpersonales:</p> <ul style="list-style-type: none">• Capacidad crítica y autocrítica• Habilidades interpersonales• Compromiso ético <p>3-Competencias sistémicas:</p> <ul style="list-style-type: none">• Capacidad de aplicar los conocimientos en la práctica• Habilidades de investigación• Capacidad de aprender• Capacidad de adaptarse a nuevas situaciones• Capacidad de generar nuevas ideas (creatividad)• Liderazgo• Habilidad para trabajar en forma autónoma• Capacidad para diseñar y gestionar proyectos• Iniciativa y espíritu emprendedor• Preocupación por la calidad• Búsqueda del logro
---	---

4.- HISTORIA DEL PROGRAMA

Lugar y fecha de elaboración o revisión	Participantes	Observaciones
Instituto Tecnológico de Nuevo Laredo. Fecha: 13 de febrero del 2019.	<ul style="list-style-type: none">• Ing. Bruno López Takeyas, M.C.	

5.- OBJETIVO(S) GENERAL(ES) DEL CURSO (competencias específicas a desarrollar en el curso)

- Introducir al candidato a estudiante de la carrera de Ing. en Sistemas Computacionales en el área de las ciencias computacionales para aclarar posibles dudas y confirmar sus aspiraciones de cursar esta carrera.
- El estudiante conocerá el objetivo de la carrera, el perfil del egresado, el plan de estudios, los módulos de especialidad y el campo de acción para su futuro ejercicio profesional.
- Analizar y diseñar soluciones a problemas sencillos utilizando algoritmos computacionales orientados a objetos.
- Dominar la metodología para la resolución de problemas a través de la computadora, aplicando análisis y diseño de algoritmos utilizando el paradigma orientado a objetos.
- Desarrollar habilidades en el planteamiento de estrategias de solución a problemas cotidianos mediante la utilización de la computadora, las cuales se aplicarán de manera inmediata en las materias de Fundamentos de Programación, Programación Orientada a Objetos, Estructuras de Datos, Tópicos Selectos de Programación y posteriormente en las que se derivan de ellas.

6.- COMPETENCIAS PREVIAS

- **Competencias instrumentales**
 - Capacidad de análisis en general
 - Capacidad de pensamiento lógico y analítico
 - Resolución de problemas
 - Habilidades tecnológicas
 - Habilidades matemáticas (aritméticas y algebraicas)
 - Manejo adecuado de información

- **Competencias sistémicas**

- Capacidad de aplicar los conocimientos a la práctica
- Capacidad de aprender

7.- TEMARIO

Unidad	Temas	Subtemas
1	Introducción a la ISC	Definición de informática Carreras profesionales en informática y computación La ISC en el IT Nuevo Laredo Objetivo general de la carrera Perfil del egresado de ISC Plan reticular de ISC Módulos de especialidad <i>Programación de dispositivos móviles</i> <i>Redes y bases de datos</i>
2	Conceptos básicos de algoritmos	Definición de sistema Sistemas de procesamiento de información Elementos de un sistema de procesamiento de información Definiciones de <i>hardware</i> y <i>software</i> Esquema básico de una computadora (modelo de <i>Von Neumann</i>) Definición de algoritmo Características de los algoritmos Representación de los algoritmos
3	Metodología para resolver problemas a través de la computadora	Análisis del problema Definición del problema Especificaciones de entrada Especificaciones de salida Diseño del algoritmo Diseño descendente (<i>top-down</i>) Refinamiento por pasos Herramientas Pseudocódigo Diagramas de flujo de datos Implementación Codificación Ejecución Pruebas
4	Datos y tipos de datos	Conceptos de datos y tipos de datos Datos numéricos Enteros Reales Datos booleanos o lógicos

		<p>Datos alfanuméricos</p> <ul style="list-style-type: none"> Caracteres Cadenas (string) Variables Constantes
5	Expresiones aritméticas	<p>Representación de expresiones en forma algorítmica</p> <p>Operadores aritméticos</p> <ul style="list-style-type: none"> Suma (+) Resta (-) Multiplicación (*) División (/) División entera División real Cálculo del cociente (DIV) Cálculo del residuo (MOD) Exponenciación (** o ^ o ↑) <p>Reglas de prioridad de los operadores aritméticos</p> <p>Funciones matemáticas internas</p> <p>Evaluación de expresiones aritméticas</p> <p>Contadores</p> <p>Acumuladores</p>
6	Expresiones lógicas	<p>Operadores relacionales</p> <ul style="list-style-type: none"> Menor que (<) Mayor que (>) Menor o igual que (\leq) Mayor o igual que (\geq) Idéntico (= o ==) Diferente (< > o != o \neq) <p>Operadores lógicos</p> <ul style="list-style-type: none"> AND OR NOT <p>Tablas de verdad de los operadores lógicos</p> <p>Reglas de prioridad de los operadores aritméticos</p>
7	Control de flujo	<p>Instrucciones algorítmicas básicas</p> <p>Algoritmos secuenciales</p> <p>Estructuras selectivas</p> <ul style="list-style-type: none"> Simple (<i>if</i>) Doble (<i>if-else</i>) Múltiple (<i>switch-case</i>) <p>Estructuras iterativas, repetitivas o cíclicas</p> <ul style="list-style-type: none"> Repetir (<i>for</i>) Mientras (<i>while</i>) Hacer-mientras (<i>do-while</i>)
8	Métodos	<p>Definición de método</p> <p>Tipos de métodos</p>

		Procedimientos Funciones Envío de parámetros por valor Recibiendo el valor de una función
9	Clases y objetos	Introducción al diseño orientado a objetos Definiciones de clases y objetos Componentes de las clases Atributos (datos o campos) Métodos Mutators Accessors Modificadores de acceso Público Privado Diagramas de clases en UML
10	Diseño orientado a objetos en Raptor	Introducción a Raptor Descarga e instalación Estructura de un algoritmo en Raptor Sentencias y símbolos Variables Diseño de clases Entrada de datos Asignaciones Expresiones Salida Comentarios Sintaxis y semántica Estructuras de control Secuencial Estructuras selectivas Ciclos

8.- SUGERENCIAS DIDÁCTICAS (desarrollo de competencias genéricas)

Se recomienda que el estudiante realice las siguientes actividades:

- Propiciar actividades de búsqueda, selección y análisis de información en distintas fuentes.
- Propiciar el uso adecuado de conceptos y de terminología científico-tecnológica.
- Propiciar, en el estudiante, el desarrollo de actividades intelectuales de inducción-deducción y análisis-síntesis, las cuales lo encaminan hacia la investigación, la aplicación de conocimientos y la solución de problemas.

- Llevar a cabo actividades prácticas que promuevan el desarrollo de habilidades para la experimentación, tales como el diseño de algoritmos orientados a objetos aplicando la metodología de la resolución de problemas a través de la computadora.
- Documentar cada una de las etapas (incluyendo las subetapas) de la metodología.
- Proponer problemas que permitan al estudiante la integración de contenidos de este curso con las materias de Fundamentos de Programación, Programación Orientada a Objetos, Tópicos Selectos de Programación, Estructuras de Datos y otras que se derivan de ellas.
- Probar el correcto funcionamiento de los algoritmos diseñados mediante su ejecución en **Raptor** (software de diseño de algoritmos orientados a objetos).
- **La unidad 9 (Diseño de algoritmos en Raptor) se puede impartir paralelamente con las unidades 6, 7 y 8.**

9.- SUGERENCIAS DE EVALUACIÓN

- Se recomienda solicitar al estudiante que resuelva ejercicios planteados por el maestro, en la cual aplique las fases de análisis y diseño de la metodología para resolver problemas a través de la computadora.
- El maestro debe constatar el cumplimiento de las subetapas del análisis (definición del problema, especificaciones de entrada, especificaciones de salida) y del diseño (diseño descendente, refinamiento por pasos y diagrama de flujo) de cada ejercicio resuelto por el estudiante.
- Se sugiere el uso de software para diseño, ejecución y prueba de algoritmos orientados a objetos como Raptor, el cual puede descargarse de manera gratuita del sitio <http://raptor.martincarlisle.com/>
- Es recomendable aplicar un examen teórico y un examen práctico.

10.- UNIDADES DE APRENDIZAJE

Unidad 1: Introducción a la ISC

Competencia específica a desarrollar	Actividades de Aprendizaje
<p>Conocer el entorno de la carrera de ISC.</p> <p>Conocer el objetivo, perfil del egresado, el plan de estudios y el campo de acción de la carrera de ISC.</p>	<p>- Consultar la información relacionada con la carrera profesional de ISC en el Instituto Tecnológico.</p> <ul style="list-style-type: none"> ○ Objetivo ○ Plan de estudios ○ Perfil del egresado ○ Campo de acción

Reafirmar el interés por estudiar la carrera profesional de ISC	
---	--

Unidad 2: Elementos de un sistema computacional

Competencia específica a desarrollar	Actividades de Aprendizaje
Conocer los componentes de un sistema computacional. Entender el funcionamiento genérico de un equipo de cómputo.	<ul style="list-style-type: none"> - <i>Investigar el concepto de sistema.</i> - <i>Investigar el concepto de sistema de información.</i> - <i>Consultar los conceptos de hardware y software</i> - <i>Realizar un diagrama con el modelo computacional de Von Neumann</i>

Unidad 3: Conceptos básicos de algoritmos

Competencia específica a desarrollar	Actividades de Aprendizaje
Dominar la terminología relacionada con algoritmos	<ul style="list-style-type: none"> - <i>Investigar el concepto de algoritmo.</i> - <i>Identificar las características de los algoritmos.</i> - <i>Conocer las formas de representación de algoritmos</i> - <i>Elaborar un mapa conceptual de algoritmos.</i>

Unidad 4: Metodología para resolver problemas a través de la computadora

Competencia específica a desarrollar	Actividades de Aprendizaje
Conocer cada una de las etapas de la metodología para resolver problemas a través de la computadora.	<ul style="list-style-type: none"> - <i>Identificar cada una de las etapas para el desarrollo de sistemas computacionales (análisis, diseño e implementación)</i> - <i>Identificar cada uno de los pasos en las etapas indicadas por la metodología</i> - <i>Elaborar un mapa conceptual de la metodología.</i>

Unidades 5 – 8

Competencia específica a desarrollar	Actividades de Aprendizaje
<p>Diseñar propuestas de solución a problemas cotidianos sencillos mediante algoritmos.</p> <p>Aplicar la metodología para la resolución de problemas a través de la computadora.</p>	<p>Durante la elaboración del algoritmo deben cubrirse las siguientes actividades:</p> <ul style="list-style-type: none">- <i>Documentar la etapa de análisis identificando y definiendo el problema a resolver, los datos de entrada, de salida y auxiliares o de apoyo (con sus respectivos tipos de datos).</i>- <i>Elaborar el diseño del algoritmo, afinando sus pasos.</i>- <i>Elaborar un diagrama de flujo</i>- <i>Diseñar el conjunto de pruebas necesarias para asegurarse que el algoritmo ofrece los resultados deseados y considera todas las posibilidades.</i>

Unidad 9.- Diseño de algoritmos orientados a objetos usando Raptor

Competencia específica a desarrollar	Actividades de Aprendizaje
<p>Diseñar propuestas de solución a problemas cotidianos sencillos mediante algoritmos.</p> <p>Aplicar la metodología para la resolución de problemas a través de la computadora.</p> <p>Utilizar un equipo de cómputo y el software denominado Raptor para la elaboración del algoritmo orientado a objetos.</p>	<ul style="list-style-type: none">- <i>Descargar el software de manera gratuita del sitio http://raptor.martincarlisle.com/</i>- <i>Instalar Raptor</i>- <i>Diseñar algoritmos orientados a objetos utilizando Raptor</i>- <i>Ejecutar y probar el funcionamiento de los algoritmos en Raptor para asegurarse que ofrece los resultados deseados.</i>

11.- FUENTES DE INFORMACIÓN

Fuentes impresas (libros)

- López Takeyas, Bruno. (2019). "Introducción a la ISC y al Diseño Orientado a Objetos". México: Pearson. ISBN: 978-607-32-5014-6.

Fuentes electrónicas

- López Takeyas, Bruno. (2019). "Introducción a la ISC y al Diseño Orientado a Objetos". 20 de mayo del 2019, de Instituto Tecnológico de Nuevo Laredo. Sitio web: <http://www.itnuevolaredo.edu.mx/Takeyas/LibroISC/>

12.- PRÁCTICAS PROPUESTAS

Cada ejercicio planteado debe resolverse aplicando las fases de análisis y diseño de la metodología para resolver problemas a través de la computadora, documentando cada una de ellas (incluyendo sus subetapas); además se recomienda la comprobación de los algoritmos mediante Raptor.