

1.- DATOS DE LA ASIGNATURA

Nombre de la asignatura:	Lógica de Programación
Carrera:	Ingeniería en Nanotecnología
Clave de la asignatura:	NAF-0912
SATCA ¹	3 - 2 - 5

2.- PRESENTACIÓN

Caracterización de la asignatura.

Actualmente el uso de computadoras ha tenido gran impacto en áreas diseño; entre otras cosas, permite modelar sistemas, y procesar información a velocidades elevadas.

Sin importar el área de especialidad, se deben tener conocimientos básicos de programación para utilizar paquetes especializados o para desarrollar alguna aplicación para resolver problemas reales. Estos conocimientos también permiten comprender como la computadora lleva a cabo sus operaciones, y así poder proponer diferentes soluciones a dichos problemas.

Entre algunas aplicaciones para los Ingenieros en Nanotecnología se encuentra la caracterización y modelado molecular en Ciencia de los Materiales, modelado de estructuras de transición, etc.

La materia aporta conocimientos y habilidades para las materias de Análisis Numérico e Introducción al Modelado por Computadora.

Intención didáctica.

Los contenidos deben ser abordados de tal manera que el estudiante vaya sumando los conocimientos y habilidades en el desarrollo de competencias, que le permitan usar la computadora y la programación como herramienta para la resolución de problemas en su área de especialidad.

Se organiza el temario de manera que el estudiante haga una transición ágil y dinámica de los conceptos básicos de computación, a la programación.

El alcance que debe tener es que el estudiante comprenda como los paquetes de cómputo procesan operaciones, pueda hacer planteamiento de problemas para su solución con herramientas y técnicas de programación, y que pueda identificar variables y procesos para aplicarlos en software especializado de procesamiento y simulación para su área de especialidad.

Dentro de la estructura planteada en el contenido temático, la primera unidad tiene la

¹ Sistema de asignación y transferencia de créditos académicos

intención de contextualizar al estudiante, introducirlo a los conceptos básicos necesarios para que pueda desarrollar las competencias propuestas en el resto del plan de estudios.

El poder desarrollar sistemáticamente una propuesta solución para un problema, es la finalidad de la segunda unidad, para esto se introduce al estudiante en una mitología que le permita desarrollar competencias para identificar datos y procesos, mismos que son formalizados en la tercera unidad, en la cual se presentan métodos formales y generalizados para representar la solución a problemas.

Para poder resolver problemas computacionalmente, es necesario saber como son representados los datos en una computadora, y como es el esqueleto de un programa que pueda utilizar dichos datos para resolver un problema, estos conocimientos son desarrollados por el estudiante en la cuarta unidad.

En la quinta unidad es cuando el estudiante ya debe ser capaz de escribir programas, poder darle instrucciones a la computadora que esta nos pida y nos reporte datos. En esta unidad también se introducen los principios de programación que permiten al programador describir toma de decisiones y procesos iterativos.

En las actividades científico-tecnológicas siempre es necesario el uso de vectores, la sexta unidad presenta la versión computacional de este concepto, induciendo al estudiante para la elección correcta de estructuras que permiten la organización y procesamiento de información, sobre todo cuando esta viene en grandes volúmenes.

En el proceso natural de solución de problemas, generalmente optamos por hacer procesos pequeños a la vez, en lugar de hacer procesos largos y tediosos, este principio es incluido en la séptima unidad, donde además se presentan los conceptos y técnicas necesarios para la programación de funciones, permitiendo al estudiante implementar computacionalmente este concepto.

Es común que en la solución (sobre todo numérica) de problemas, se requiera de procesos que requieren del almacenamiento de grandes cantidades de información, misma que tiene una vigencia temporal puesto que es actualizada constantemente o simplemente desaparece o se vuelve innecesaria después de ciertos pasos o tiempo. Una metodología para optimizar el uso de memoria RAM para estos casos, es presentado en la octava unidad.

El profesor debe ser conocedor de la disciplina que está bajo su responsabilidad, conocer su origen y desarrollo histórico para considerar este conocimiento al abordar los temas. Desarrollar la capacidad para coordinar y trabajar en equipo; orientar el trabajo del estudiante y potenciar en él la autonomía, el trabajo cooperativo y la toma de decisiones.

3.- COMPETENCIAS A DESARROLLAR

Competencias específicas	Competencias genéricas
<p>Plantear metodológicamente la solución de problemas susceptibles de ser computarizados a través del manejo de técnicas estructuradas de diseño y formulación de algoritmos, así como realizar la documentación de manera pertinente.</p>	<p>Competencias instrumentales</p> <ul style="list-style-type: none">• Capacidad de análisis y síntesis• Capacidad de organizar y planificar• Conocimientos básicos de la carrera• Comunicación oral y escrita• Habilidades básicas de manejo de la computadora• Habilidad para buscar y analizar información proveniente de fuentes diversas• Solución de problemas• Toma de decisiones. <p>Competencias interpersonales</p> <ul style="list-style-type: none">• Capacidad crítica y autocrítica• Trabajo en equipo• Habilidades interpersonales <p>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 generar nuevas ideas (creatividad)• Habilidad para trabajar en forma autónoma• Búsqueda del logro

4.- HISTORIA DEL PROGRAMA

Lugar y fecha de elaboración o revisión	Participantes	Observaciones (cambios y justificación)
Instituto Tecnológico de Ciudad Juárez del 27 al 29 de Abril de 2009.	Representantes de los Institutos Tecnológicos de: Tijuana, Querétaro, Celaya, Saltillo, Ciudad Juárez, Superior de Irapuato, San Luis Potosí, Chihuahua.	Primera Reunión Nacional de diseño e innovación curricular para el desarrollo de competencias profesionales de las carreras de Ingeniería en Nanotecnología e Ingeniería Logística del SNEST.
Instituto Tecnológico de Puebla del 8 al 12 de Junio de 2009	Representantes de los Institutos Tecnológicos de: Tijuana, Querétaro, Celaya, Saltillo, Ciudad Juárez, Superior de Irapuato, San Luis Potosí, Chihuahua	Reunión de seguimiento de diseño e innovación curricular para el desarrollo de competencias profesionales de las carreras de Ing. en Nanotecnología, Gestión Empresarial, Logística, y asignaturas comunes del SNEST.
Instituto Tecnológico de Mazatlán del 23 al 27 de Noviembre de 2009	Representantes de los Institutos Tecnológicos de: Tijuana, Querétaro, Ciudad Juárez, Superior de Irapuato, San Luis Potosí, Chihuahua	Segunda Reunión de seguimiento de diseño e innovación curricular para el desarrollo de competencias profesionales de la carrera de Ing. en Nanotecnología, del SNEST.
Instituto Tecnológico de Villahermosa del 24 al 28 de Mayo de 2010	Representantes de los Institutos Tecnológicos de: Tijuana, Querétaro, Superior de Irapuato, Chihuahua, Saltillo.	Reunión de consolidación de diseño e innovación curricular para el desarrollo de competencias profesionales de la carrera de Ing. en Nanotecnología, del SNEST.

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

Plantear metodológicamente la solución de problemas susceptibles de ser computarizados a través del manejo de técnicas estructuradas de diseño y formulación de algoritmos, así como realizar la documentación de manera pertinente.

6.- COMPETENCIAS PREVIAS

- Utiliza equipo de cómputo y software básico.
- Aplica los conocimientos básicos de Matemáticas (Álgebra, Trigonometría y Geometría Analítica.)

7.- TEMARIO

Unidad	Temas	Subtemas
1	Conceptos básicos	1.1 Introducción 1.2 De los problemas a los programas 1.3 Definición de lenguaje 1.4 Definición de algoritmo 1.5 Algoritmos cotidianos 1.6 Definición de lenguajes algorítmicos
2	Metodología para la solución de problemas por medio de computadoras	2.1 Análisis de los datos 2.2 Diseño de la solución 2.3 Prueba y depuración
3	Técnicas para la formulación de algoritmos	3.1 Diagramas de flujo 3.2 Pseudocódigo
4	Principios de programación	4.1 Tipos de datos 4.2 Estructura básica de un programa
5	Estructuras Básicas de control	5.1 Secuenciales 5.2 Asignación 5.3 Entrada 5.4 Salida 5.5 Condicionales 5.5.1 Simple 5.5.2 Múltiple 5.6 Estructuras de repetición 5.6.1 Por contador 5.6.2 Por condición
6	Arreglos	6.1 Arreglos unidimensionales 6.2 Arreglos n-dimensionales
7	Manejo de módulos	7.1 Concepto y características de un módulo 7.2 Clasificación de los módulos

8	Apuntadores	<div>7.3 Funciones</div> <div>7.4 Procedimientos</div> <div>7.5 Operación de módulos y sus parámetros</div> <div>8.1 Concepto de apuntadores</div> <div>8.2 Lista simplemente Ligada</div>
---	-------------	--

8.- SUGERENCIAS DIDÁCTICAS (desarrollo de competencias genéricas)

- Diseñar y aplicar métodos de diagnóstico para determinar el nivel del conocimiento previo y propio del curso.
- Resolver problemas extra clase para que desarrollen habilidades de investigación de diversos métodos de solución y elaboración de algoritmos.
- Aplicación de los conocimientos de manera gradual durante el ciclo escolar, a partir del planeamiento de un proyecto al inicio del semestre.
- Realizar diversas actividades en equipo tanto en clase como fuera de ella, buscando incorporar en cada uno de los equipos por lo menos a un alumno destacado.
- Exposiciones en clase de los algoritmos diseñados de manera grupal o individual, buscando que se presenten diferentes algoritmos que implementen el mismo método, de tal forma que se propicie la discusión y comparación de los mismos.
- Formar grupos de asesoría con alumnos de semestres avanzados y/o docentes.
- Organizar talleres operativos con el apoyo de alumnos de semestres avanzados.
- Establecer los requerimientos mínimos que deben contener los trabajos entregados por los alumnos (elaborar una rúbrica para ello).
- Utilizar un lenguaje de programación básico.
- Realizar prácticas que permitan aplicar los conocimientos adquiridos en clase, así como para retroalimentar al docente.

9.- SUGERENCIAS DE EVALUACIÓN

La evaluación debe ser continua y cotidiana e imparcial, por lo que se debe considerar el desempeño en cada una de las actividades de aprendizaje, así como todo su contexto: autoevaluación, coevaluación y heteroevaluación.

- Autoevaluación (puede ser mediante un cuestionario breve)
 - Desempeño conceptual
 - Desempeño actitudinal y valores de responsabilidad
 - Desempeño en habilidades
- Coevaluación (puede ser mediante un cuestionario breve en donde los estudiantes evalúen a sus compañeros)
 - En desempeño actitudinal y valores en trabajo colaborativo
- Heteroevaluación
 - Diagnóstica
 - Desempeño conceptual
 - Desempeño en habilidades
 - Desempeño actitudinal y valores
- Reportes de las observaciones hechas durante las actividades, así como de las conclusiones obtenidas de dichas observaciones.
- Reportes de investigación documental que contengan los requisitos mínimos acordados en la clase.
- Exámenes escritos para comprobar el manejo de aspectos teóricos y declarativos y que no determine más del 60% de la calificación.

10.- UNIDADES DE APRENDIZAJE

Unidad 1: Conceptos básicos

Competencia específica a desarrollar	Actividades de Aprendizaje
Comprender los conceptos básicos de programación utilizados en la solución de problemas por medio de la computadora.	<ul style="list-style-type: none">• Comprender el significado y concepto de algoritmo.• Desarrollar y aplica el concepto de algoritmo en problemas reales aplicados en su entorno.

Unidad 2: Metodología para la solución de problemas por medio de computadoras

Competencia específica a desarrollar	Actividades de Aprendizaje
Aplicar de manera secuencial los pasos para proponer soluciones a problemas por medio de computadora.	<ul style="list-style-type: none">• Visualizar algún problema preferentemente del área de nanotecnología y metodológicamente diseñar una solución.

Unidad 3: Técnicas para la formulación de algoritmos

Competencia específica a desarrollar	Actividades de Aprendizaje
Seleccionar y utilizar de manera	<ul style="list-style-type: none">• Trabajar en grupos para discutir sobre la

correcta técnicas para la representación y formulación de algoritmos.	<p>técnica a seguir para representar y formular un algoritmo para un problema en específico.</p> <ul style="list-style-type: none"> • Diseñar algoritmos de casos específicos usando la secuencia de diagramas de flujo y pseudocódigos.
---	---

Unidad 4: Principios de programación

Competencia específica a desarrollar	Actividades de Aprendizaje
Traducir algoritmos a un lenguaje de programación para la resolución de problemas.	<ul style="list-style-type: none"> • Utilizar algoritmos desarrollados en las unidades 2 y 3 para identificar los diferentes tipos de datos. • Programar estos algoritmos en un lenguaje de programación de nivel alto.

Unidad 5: Estructuras básicas de control

Competencia específica a desarrollar	Actividades de Aprendizaje
Elaborar programas orientados a la aplicación de las diferentes estructuras de control.	<ul style="list-style-type: none"> • Resolver problemas a través de programas con diferentes estructuras y comentar las diferencias.

Unidad 6: Arreglos

Competencia específica a desarrollar	Actividades de Aprendizaje
Aplicar arreglos en programas para resolver problemas que requieran este tipo de estructura.	<ul style="list-style-type: none"> • Buscar aplicaciones similares a las expuestas por el profesor para su exposición. • Discutir en clase la solución a estos problemas. • Elaborar algoritmos y programas para la solución de problemas que involucren arreglos de una o más dimensiones.

Unidad 7: Manejo de módulos

Competencia específica a desarrollar	Actividades de Aprendizaje
Utilizar programación modular en la construcción de programas complejos y extensos.	<ul style="list-style-type: none"> • Explicar el concepto y la necesidad de la existencia de módulos como forma estructurada de diseño y manejo de problemas extensos, así como las formas de clasificación y modularización comunes. • Elaborar programas aplicando el concepto de módulos.

Unidad 8: Apuntadores

Competencia específica a desarrollar	Actividades de Aprendizaje
Programar apuntadores para realizar programas más eficientes y flexibles.	<ul style="list-style-type: none">• Explicar el concepto, objetivo y manipulación de apuntadores.• Elaborar programas para generar una lista simplemente ligada.

11.- FUENTES DE INFORMACIÓN

1. Aho, Hopcroft, Ullman, *Estructura de Datos y Algoritmos*, Ed. Sitesa.
2. Luis Joyanes Aguilar, *Fundamentos de Programación, Algoritmos y Estructura de Datos*.
3. Guillermo Levine, *Introducción a la Computación y la Programación Estructurada*, Ed. Mc. Graw Hill.
4. Goodman. Hedetmienic, *Introduction to the Design and Analysis of Algorithms*, Ed. Mc. Graw Hill.
5. Francisc Scheid, *Computers and Programing*, Ed. Mc. Graw Hill.
6. Tremblay Sorensen, *An Introduction to Data Structures With Applications*, Ed. Mc. Graw Hill.
7. Deitel, Java: *How To Program*, Pearson.
8. Deitel, C#: *How To Program*, Pearson.

12.- PRÁCTICAS PROPUESTAS

1. Desarrollar y aplicar el concepto de algoritmo en problemas reales aplicados en su entorno.
2. Desarrollar y aplicar el concepto de algoritmo y la metodología en problemas orientados a su área de conocimientos.
3. Elaborar diagramas de flujo y programas para la solución de problemas orientados a su área de conocimiento utilizando un lenguaje de programación.
4. Diseñar y programar la solución de problemas prácticos utilizando las estructuras de control.
5. Elaborar algoritmos y programas para la solución de problemas que involucren arreglos de una o más dimensiones.
6. Elaborar programas sencillos, pero aplicando el concepto de módulos que hace uso de un "menú" principal.
7. Elaborar programas para generar una lista simplemente ligada.