

1.- DATOS DE LA ASIGNATURA

| |
|--|
| Nombre de la asignatura: Programación |
| Carrera: Ingeniería Química |
| Clave de la asignatura: QUE – 0526 |
| Horas teoría-horas práctica-créditos: 2 2 6 |

2.- HISTORIA DEL PROGRAMA

| Lugar y fecha de elaboración o revisión | Participantes | Observaciones (cambios y justificación) |
|--|---|---|
| Instituto Tecnológico de Aguascalientes del 9 al 13 de Agosto de 2004. | Representantes de las Academias de Ingeniería Química de los Institutos Tecnológicos. | Reunión Nacional de Evaluación Curricular de la Carrera de Ingeniería Química |
| Instituto Tecnológico de Durango y Toluca. | Academias de la carrera de Ingeniería Química. | Análisis y enriquecimiento de las propuestas de los programas diseñados en la reunión nacional de evaluación curricular |
| Instituto Tecnológico de Durango del 22 al 26 de Noviembre de 2004. | Comité de Consolidación de la Carrera de Ingeniería Química. | Definición de los Programas de Estudio de la Carrera de Ingeniería Química. |

3.- UBICACIÓN DE LA ASIGNATURA

a). Relación con otras asignaturas del plan de estudio

| Anteriores | | Posteriores | |
|--------------------|--------------|---|------------------------------------|
| Asignaturas | Temas | Asignaturas | Temas |
| | | Métodos Numéricos | Solución de Ecuaciones Algebraicas |
| | | En todas las asignaturas posteriores donde sea necesario programar algoritmos | |

b). Aportación de la asignatura al perfil del egresado

- En la actualidad la industria química se encuentra altamente computarizada, tanto en sus procesos de fabricación como en el diseño, calculo y fabricación de equipo, lo mismo que en su administración, por lo que el Ingeniero Químico debe tener conocimiento de programación y computo, para poder entender y no quedar al margen, sino participar en la modernización de la Industria.

4.- OBJETIVO(S) GENERAL(ES) DEL CURSO

Adquirirá los conocimientos y las habilidades para utilizar la computadora y los lenguajes de programación como instrumento para la solución de problemas científicos y tecnológicos, entre otros.

5.- TEMARIO

| Unidad | Temas | Subtemas |
|---------------|--------------------------------|---|
| 1 | Introducción a la Computación | 1.1 Arquitectura de la Computadora. 1.2 Definición de Lenguaje 1.3 Tipos de lenguajes: natural, artificial, computacional. 1.4 Historia de los lenguajes computacionales 1.5 Paradigmas de los lenguajes computacionales 1.6 ¿Qué es un traductor? 1.7 Tipos de traductores 1.8 Estructura y funcionamiento de un compilador |
| 2 | Introducción a la Programación | 2.1 Análisis de algoritmos computacionales 2.2 Tipos de datos definidos por el lenguaje 2.3 Declaración de variables y constantes 2.4 Expresiones aritméticas, relacionales y lógicas 2.5 Asignación de expresiones 2.6 Estructura básica del lenguaje 2.7 Solución de problemas con expresiones |
| 3 | Estatutos | 3.1 De entrada y salida estándar 3.2 De control o selección if. 3.3 De ciclos: for, while 3.4 Otros estatutos 3.5 Solución de problemas aplicando estatutos de control y ciclos |
| 4 | Arreglos | 4.1 Declaración de variables dimensionadas 4.2 Operaciones con arreglos unidimensionales: inserción, remoción, búsqueda y modificación. 4.3 Aplicación de arreglos unidimensionales en algoritmos de búsqueda y ordenamiento 4.4 Arreglos multidimensionales 4.5 Operaciones con arreglos de 2 o más dimensiones: inserción, remoción, búsqueda y modificación 4.6 Aplicación de arreglos de 2 |

| | | |
|---|---|---|
| | | dimensiones para solución de problemas de matrices y grafos 4.7 Aplicación de arreglos de más de dos dimensiones en sistemas complejos |
| 5 | Funciones | 5.1 Declaración de funciones como subprogramas. 5.2 Llamada a una función 5.3 Pasos de parámetros, por valor y por referencia 5.4 Elaboración de programas y subprogramas |
| 6 | Tipos de Datos Abstractos definidos por el usuario (TDA) y Archivos | 6.1 Declaración de tipos de datos definidos por el usuario. 6.2 Operaciones para acceder, almacenar y modificar valores de un TDA. 6.3 Aplicación de TDA en problemas simples 6.4 Declaración de archivos. 6.5 Tipos de archivos 6.6 Operaciones con Archivos. 6.7 Aplicación de problemas con archivos y TDA's |

6.- APRENDIZAJES REQUERIDOS

7.- SUGERENCIAS DIDÁCTICAS

- Explicar al alumno programas completos de menor a mayor grado de dificultad y con base en cada una de las instrucciones que los componen, enseñar la sintaxis del lenguaje y la finalidad de cada una de ellas.
- Solicitar que corrobore la validez del mismo, ejecutándolo en la computadora.
- Organizar un taller de grupos cooperativos para la realización de programas similares, agregándoles algunas variantes. Suma, Resta, Multiplicación, división, exponenciación, entre otros
- Plantearle al alumno problemas para que los resuelva utilizando el algoritmo que lo lleve a la solución del mismo expresado en los lenguajes en cuestión, lo ejecute en la computadora y lo discuta en una sesión grupal.
- El maestro definirá en las reglas del juego los lineamientos mínimos de documentación que deberán contener las tareas relacionadas con el punto anterior.
- Solicitarle al alumno las propuestas de problemas a resolver y que sean significativas para él.

- Que el alumno experimente con nuevos programas encontrados en revistas y libros de la especialidad, que lo lleven a descubrir nuevos conocimientos.
- Desarrollar prácticas para el laboratorio con ejercicios ad-hoc a la especialidad.
- Promover la investigación de algoritmos relacionados con las ciencias químicas, como casos de aplicación.
- Utilizar como herramienta para el análisis de algoritmos, diagramas de flujo, pseudocódigo o lenguaje unificado de modelado (UML)

8.- SUGERENCIAS DE EVALUACIÓN

- Exámenes escritos(Opción múltiple, análisis de casos, etc)
- Elaboración de programas
- Trabajo en grupos de Internet.
- Trabajos de investigaciones.
- Tareas.
- Discusiones.
- Presentaciones.
- Exposiciones.

9.- UNIDADES DE APRENDIZAJE

Unidad 1.- Introducción a la Computación

| Objetivo Educativo | Actividades de Aprendizaje | Fuentes de Información |
|--|--|------------------------|
| El estudiante describirá los componentes principales de una computadora, los conceptos de lenguajes, lenguajes de programación y procesos de traducción. | <ul style="list-style-type: none"> • Investigar los componentes principales de una computadora. • Realizar prácticas en el Centro de Cómputo para conocer los componentes de una computadora. • Realizar la investigación acerca de la definición lenguaje y la estructura del mismo. • Investigar y experimentar con diferentes paradigmas de la programación a lo largo de la historia. • Investigar la definición de traductor y los diferentes tipos de traductores. • Utilizar el compilador que será utilizado para la programación en el curso. | 4, 10, 11, 19, 22 |

Unidad 2.- Introducción a la Programación

| Objetivo Educativo | Actividades de Aprendizaje | Fuentes de Información |
|--|--|---------------------------|
| Desarrollará la lógica en la solución de problemas y podrá resolver problemas relacionados con la evaluación de expresiones matemáticas. | <ul style="list-style-type: none"> • Desarrollará la lógica para la solución de problemas algorítmicos a través de diagramas de flujo, pseudocódigo y como mejor alternativa, mediante el lenguaje unificado de modelado (UML) • Conocerá los tipos de datos definidos por un lenguaje de programación • Conocerá los conceptos de variables y constantes utilizadas en un lenguaje de programación y aplicará los conceptos en la solución de problemas simples • Dominará la construcción de expresiones matemáticas para la solución de expresiones. • Comprenderá la sintaxis para la asignación de expresiones a variables • Desarrollará sus primeros programas y aplicando a soluciones simples de expresiones matemáticas. | 1, 2, 3, 4, 5, 6, 7, 8, 9 |

Unidad 3.- Estatutos

| Objetivo Educativo | Actividades de Aprendizaje | Fuentes de Información |
|---|---|---------------------------|
| Solucionará, resolverá, probará y depurará problemas relacionados con sentencias de control | <ul style="list-style-type: none"> • Aprender y aplicar las proposiciones de E/S del lenguaje estudiado • Aplicar las diferentes proposiciones de decisión <ul style="list-style-type: none"> ○ If-then, if then-else ○ If's anidados ○ Otros • Estructuras repetitivas básicas <ul style="list-style-type: none"> ○ Reconocer y aplicar las proposiciones de ciclos. • Resolver problemas planteados por el maestro, donde se puedan aplicar los conceptos vistos. | 1, 2, 3, 4, 5, 6, 7, 8, 9 |

Unidad 4.- Arreglos

| Objetivo Educativo | Actividades de Aprendizaje | Fuentes de Información |
|---|---|-------------------------------|
| Resolverá problemas, mediante la programación donde se aplique el concepto de Arreglos. | <ul style="list-style-type: none">• Definir y declarar tipos de datos de Arreglo• Aplicar los conceptos de búsqueda lineales y búsqueda binarias• Diseñar un algoritmo de clasificación• Resolver los problemas planteados por el maestro, donde se aplique el concepto de arreglos. | 1, 2, 3, 4, 5, 6, 7, 8, 9 |

Unidad 5.- Funciones

| Objetivo Educativo | Actividades de Aprendizaje | Fuentes de Información |
|--|---|-------------------------------|
| Aplicará las funciones en la solución de problemas específicos | <ul style="list-style-type: none">• Conocer la sintaxis de la declaración de funciones, subprogramas y procedimientos• Aplicara la sintaxis de los subprogramas para dar solución a un problema específico• Aplicara la sintaxis de los procedimientos para dar solución a un problema específico• Resolver problemas utilizando funciones, subprogramas y procedimientos. | 4, 5, 7 |

Unidad 6.- Tipos de Datos Abstractos definidos por el usuario (TDA) y archivos

| Objetivo Educativo | Actividades de Aprendizaje | Fuentes de Información |
|---|--|---------------------------|
| Resolverá problemas mediante la programación, aplicando el concepto de Registro y Archivos. | <ul style="list-style-type: none">• Reconocer y aplicar las proposiciones relacionadas con registros en el lenguaje de programación bajo estudio• Distinguir los diferentes tipos de archivos• Abrir, cerrar, escribir, leer, entre otros. En un archivo• Resolver, probar y depurar programas, para dar solución a problemas con inserción, actualización y borrado de registros, planteados por el maestro. | 1, 2, 3, 4, 5, 6, 7, 8, 9 |

10.- FUENTES DE INFORMACIÓN

1. Alcalde, E. y García, M. *Metodología de la Programación: Aplicaciones en Basic, Cobol y Pascal*. McGraw – Hill.
2. Joyanes Aguilar, Luis. *Fundamentos de Programación: Algoritmos y Estructura de Datos*. McGraw – Hill, 2nda. edición.
3. Cairó B., Osvaldo. *Metodología de la Programación: Algoritmos, Diagramas de flujo y Programas*. Alfa Omega.
4. Chapman, Stephen J. *Fortran 90/95 for Scientists and Engineers*. McGraw – Hill.
5. Nyhoff, Larry & Leestma, Sanford. *Introduction to Fortran 90 for Engineers and Scientists*. Prentice – Hall.
6. *Microsoft MS Dos*, Guía de Referencia para el Usuario.
7. The Student Edition of MATLAB: *The Language of Technical Computing The MATH WORKS Inc*. User's Guide Versión 5 o superior.
8. Joyanes Aguilar, Luis. *Programación Basic: Para Microcomputadoras*. Prentice – Hall
9. Joyanes Aguilar, Luis. *Programación Basic Avanzado*. Prentice – Hall

10. Nieves, Antonio, Domínguez, Federico. *Métodos Numéricos Aplicados a la Ingeniería*. CECSA
11. Sánchez, Sebastián. *Unix y Linux Guía Practica*. Alfa Omega RAMA.
12. *Como Funcionan las Computadoras*. Guía visual.
13. Martín, Nacho B. *Guía Visual de Introducción a la Informática*. 1999.
14. Cevallos, Francisco Javier. *EL Lenguaje de Programación Java*. Alfa Omega RAMA.
15. Cevallos, Francisco Javier. *Java: Curso de Programación*. Alfa Omega RAMA.
16. Baltazar Birnios, Mariano Bienios. *Manual de Visual Basic 6.0*.
17. Du Portier, Gustavo. *Bases de Datos en Ms Visual Basic 6.0*.
18. Mariano Bienios. *Manual de Visual Basic*. MP Ediciones.
19. <http://sunsite.informatik.rwth-aachen.de/fortran>
20. <http://astro.ft.uam.es/~gustavo/COMPUTACIONALES/programa/>
21. <http://www.star.le.ac.uk/%7Ecgp/fortran.html>
22. <http://www.cs.rit.edu/%7Eats/plc-2001-2/reports/fortran/>
23. <http://www.programatium.com/libros2/vb.htm>

11.- PRÁCTICAS (FRENTE A LA COMPUTADORA)

- 1 Identificación física de los componentes de una computadora
- 2 Codificación de problemas en un lenguaje de alto nivel en la computadora a partir de diagramas de flujo y pseudo códigos
- 3 Entrada y salida de información en la computadora para la solución de problemas sencillos
- 4 Solución de problemas con estructuras repetitivas
- 5 Solución de problemas con estructuras selectivas

- 6 Manipulación de los elementos en un arreglo
- 7 Paso de parámetros en funciones y subrutinas
- 8 Crear y manejar archivos(abrir, cerrar, leer, escribir, etc.)
- 9 Suma, Multiplicación, Resta de Matrices , manejo de Índices.
- 10 Cálculo de Cp promedios
- 11 Arreglos dinámicos
- 12 A través de una serie de datos obtener la función que el comportamiento y su gráfico correspondiente
- 13 Determinar el límite de una función, hallar la convergencia gráficamente de una serie de Taylor y Mclaurin
- 14 Acceso a bases de Datos
- 15 Comparación de Lenguajes de programación bajo un mismo programa
- 16 Presentar un ambiente grafico a las soluciones numéricas
- 17 Ampliación del programa con la incorporación de Métodos Numéricos