

1.- DATOS DE LA ASIGNATURA

Nombre de la asignatura: Tópicos selectos de programación
Carrera: Ingeniería en Sistemas Computacionales
Clave de la asignatura: SCM - 0435
Horas teoría-horas práctica-créditos 3-2-8

2.- HISTORIA DEL PROGRAMA

Lugar y fecha de elaboración o revisión	Participantes	Observaciones (cambios y justificación)
Instituto Tecnológico de Toluca del 18 al 22 agosto 2003.	Representantes de la academia de sistemas y computación de los Institutos Tecnológicos.	Reunión nacional de evaluación curricular de la carrera de Ingeniería en Sistemas Computacionales.
Instituto Tecnológico de: Cd. Cuauhtémoc, Chihuahua II, Nogales. 23 agosto al 7 de noviembre 2003.	Academia de sistemas y computación.	Análisis y enriquecimiento de las propuestas de los programas diseñados en la reunión nacional de evaluación.
Instituto Tecnológico de León 1 al 5 de marzo 2004.	Comité de consolidación de la carrera de Ingeniería en Sistemas Computacionales.	Definición de los programas de estudio de la carrera de Ingeniería en Sistemas Computacionales.

3.- UBICACIÓN DE LA ASIGNATURA

a). Relación con otras asignaturas del plan de estudio

Anteriores		Posteriores	
Asignaturas	Temas	Asignaturas	Temas
Programación Orientada a Objetos	Arreglos unidimensionales y multidimensionales Métodos y mensajes Constructor, destructor Sobrecarga Herencia	Sistemas operativos.	Administración de procesos y del procesador. Administración de entrada/salida Administración de memoria

b). Aportación de la asignatura al perfil del egresado

- Desarrolla sistemas de información, programación de redes de computadora, sistemas distribuidos, de base y aplicación.
- Desarrolla interfaces de software hombre-máquina, máquina-máquina.
- Aplica nuevas tecnologías a la solución de problemas de su entorno laboral.

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

El estudiante integrará tópicos avanzados de programación al desarrollo de aplicaciones que requieran multihilo, multimedia, interfaz grafica de usuario y comunicación con puertos.

5.- TEMARIO

Unidad	Temas	Subtemas
1	Tipos de datos definidos por el usuario.	1.1 Agregación de datos (struct). 1.2 Uniones de datos (union). 1.3 Registros variantes. 1.4 Tipos de datos enumerados. 1.5 Manejo de bits. 1.6 Campos de bits. 1.7 Operaciones con Bits (AND, OR, NOT, XOR).
2	Creación de componentes y librerías dinámicas	2.1 Creación de componentes. 2.2 Creación de librerías dinámicas.
3	Programación concurrente multihilo.	3.1 Concepto de Hilo. 3.2 Comparación de un programa de flujo único contra uno de flujo múltiple. 3.3 Creación y control de hilos. 3.3.1 Atributos de hilos. 3.3.2 Creación e Inicialización de hilos. 3.3.3 Arranque de hilos. 3.3.4 Manipulación de hilos. 3.3.5 Suspensión de hilos. 3.3.6 Parada de hilos. 3.4 Sincronización de hilos. 3.4.1 Mutex. 3.4.2 Semáforos. 3.4.3 Barreras (Barrier).
4	Interfaz Gráfica de Usuario (GUI).	4.1 Creación de interfaz gráfica para usuarios. 4.1.1 Librería de interfaz gráfica (API's). 4.1.2 Aplicaciones GUI. 4.2 Computación gráfica. 4.2.1 Área de dibujo. 4.2.2 Primitivas de dibujo (línea, arco, círculo, colores, rellenos, imágenes).

5.- TEMARIO (Continuación)

5	Multimedia.	<ul style="list-style-type: none">5.1 Introducción a la multimedia.5.2 Componentes de un sistema multimedia.5.3 Formatos de archivo multimedia.5.4 Creación y manipulación de objetos con formatos de archivo gráfico comunes (GIF, TIFF, JPEG, WMF).5.5 Creación y manipulación de objetos con formatos de animación y video comunes (AVI, QUICKTIME, MPEG, GIF animado).5.6 Creación y manipulación de objetos con formatos de audio (compresión de sonido, formato MIDI, MP3).5.7 Integración de los elementos multimedia.5.8 Interactividad.
6	Programación de puertos e interrupciones.	<ul style="list-style-type: none">6.1 Interrupciones.6.2 Generalidades de los puertos.6.3 Puerto serial y sus variantes.<ul style="list-style-type: none">6.3.1 USB.6.3.2 Infrarrojo.6.3.3 Fire wire.6.4 Paralelo y sus variantes.6.5 Interrupciones que intervienen en cada puerto.<ul style="list-style-type: none">6.5.1 Direcciones bases.6.5.2 Registros.6.6 Envío y recepción de los datos.6.7 Ejemplos de programación entre puertos y dispositivos.

6.- APRENDIZAJES REQUERIDOS

- Comprender la metodología de orientación a objetos.
- Diseñar soluciones orientadas a objetos.
- Desarrollar programas orientados a objetos.
- Comprender los conceptos de la arquitectura de computadoras.
- Desarrollar aplicaciones en lenguaje ensamblador.

7.- SUGERENCIAS DIDÁCTICAS

- Propiciar la búsqueda y selección de información de tecnologías emergentes sobre programación avanzada, mediante equipos analizar y discutir en clase.
- Realizar las prácticas propuestas para poder alcanzar el objetivo de la materia.
- Ejercicios extra clase.
- Analizar prácticas modelo para comprender su funcionamiento.
- Realizar búsquedas de información sobre temas afines.
- Elaborar reportes o informes de las prácticas.
- Realizar practicas de comunicación con dispositivos externos.
- Propiciar debate en clase sobre la optimización de los algoritmos y códigos de programación.
- Asignar proyectos finales que integren los temas de este programa de estudio.
- Exponer los proyectos finales

8.- SUGERENCIAS DE EVALUACIÓN

- Desempeño, dominio de los conceptos la capacidad de la aplicación de los conocimientos en problemas reales (trasferencia del conocimiento).
- Cumplimiento de los objetivos de las practicas y del proyecto final.
- Dar seguimiento al desempeño integral del estudiante en el desarrollo del programa.
- Exámenes prácticos y teóricos.
- Realización de programas de aplicación.
- Desarrollo de un proyecto final que integre todas las unidades de aprendizaje.

9.- UNIDADES DE APRENDIZAJE

UNIDAD 1.- Tipos de datos definidos por el usuario.

Objetivo Educativo	Actividades de Aprendizaje	Fuentes de Información
El estudiante creará tipos de datos complejos utilizando estructuras de datos definidas por el usuario	<ul style="list-style-type: none">• Buscar y seleccionar información sobre las opciones de tipos de datos definidos por el usuario de acuerdo a la sintaxis del lenguaje de programación empleado.• Elaborar programas sobre manejo de bits.	1, 2, 3, 4, 5, 6, 7, 8, 9

UNIDAD 2.- Creación de componentes y librerías dinámicas.

Objetivo Educativo	Actividades de Aprendizaje	Fuentes de Información
Desarrollará componentes y librerías dinámicas que respondan a las necesidades de desarrollo de aplicaciones	<ul style="list-style-type: none">• Buscar y seleccionar información para crear componentes y librerías dinámicas.• Elaborar programas simples creando componentes derivados de los ya existentes, introduciendo cada una de las propiedades que definen al componente.• Realizar programas para estudiar la metodología de creación de librerías dinámicas.	5, 6, 7, 8, 10

UNIDAD 3.- Programación concurrente multihilo.

Objetivo Educativo	Actividades de Aprendizaje	Fuentes de Información
Implementara aplicaciones multihilo compartiendo la misma arquitectura	<ul style="list-style-type: none">• Buscar y seleccionar información sobre hilos.• Organizar un debate sobre hilos vs. procesos.• Analizar uso e impacto de hilos en el desarrollo de aplicaciones.• Desarrollar programas para reforzar el uso de hilos con énfasis en los cambios de estado.• Desarrollar programas concurrentes que comparten recursos.• Debatir problemas de exclusión mutua y formas de resolverla.	1, 2, 3, 4, 6, 7, 9

UNIDAD 4.- Interfaz Gráfica de Usuario.

Objetivo Educativo	Actividades de Aprendizaje	Fuentes de Información
Implementará aplicaciones con interfaz gráfica de usuario.	<ul style="list-style-type: none">• Buscar y seleccionar librerías para creación de interfaz gráfica de usuario.• Desarrollar aplicaciones con interfaz gráfica de usuario, que utilice múltiples ventanas.• Buscar y seleccionar librerías de computación gráfica.• Desarrollar aplicaciones que permitan crear gráficos en tiempo de ejecución.	1, 2, 3, 4, 6, 7, 9

UNIDAD 5.- Multimedia.

Objetivo Educativo	Actividades de Aprendizaje	Fuentes de Información
Comprenderá que es la multimedia, cuales son sus ventajas y desarrollará una aplicación multimedia interactiva.	<ul style="list-style-type: none">• Buscar y seleccionar información para discutir sobre la multimedia.• Elaborar un mapa conceptual sobre los tipos de formatos gráficos más comunes.• Buscar y seleccionar información sobre las herramientas y clases que provee el lenguaje para crear y manipular objetos multimedia.• Desarrollar programas para aplicar alguna biblioteca de clases para el manejo de multimedia.	1, 6, 8

UNIDAD 6.- Programación de puertos e interrupciones.

Objetivo Educativo	Actividades de Aprendizaje	Fuentes de Información
Programará aplicaciones que requieran los puertos de comunicación.	<ul style="list-style-type: none">• Desarrollar programas que incluyan los parámetros programables que intervienen los puertos de comunicación.• Buscar y seleccionar información sobre la detección de dispositivos conectados a un puerto.• Elaborar aplicaciones para manejo de interrupciones y parámetros utilizados en programas de envío y recepción de datos.• Elaborar un mapa conceptual sobre los tipos de formatos gráficos más comunes.• Buscar y seleccionar información sobre las herramientas y clases que provee el lenguaje para crear y manipular objetos multimedia.• Desarrollar programas para aplicar alguna biblioteca de clases para el manejo de multimedia.	10

10. FUENTES DE INFORMACIÓN

1. Deitel y Deitel. *Como programar en Java*. Prentice Hall. 1998.
2. Mitchell will David. *Java sin errores*. Mc Graw Hill. 2001
3. Becerril C. Francisco. *Java a su alcance*. Mc Graw Hill. 1998
4. Lemay Laura & Perkins Charles L. *Aprendiendo Java en 21 días*. Prentice Hall. 1996.
5. Smiley John. *Learn to program with Java*. Mc Graw Hill. 2002.
6. Naughton Patrick . *The Java Handbook*. Berkeley, CA: Osborne-McGraw Hill. 1996.
7. Schildt Herbert. *The Complete Reference Java 2. Fourth Edition*. McGraw-Hill. 2001.
8. Case Bradley Julia & Millspaugh Anita C. Mt. San Antonio College. *Programming With Java W/CD-ROM*. Mc Graw Hill. 2002.
9. Arnow David M. & Weiss Gerald. *Introducción a la programación con Java. Un enfoque orientado a objetos*. Addison Wesley. 2001.
10. Elliotte Rusty Harold. *Java I/O O*. O'Reilly & Associates. 1999.
11. Scott Oaks and Henry Wong. *Java Threads, second Edition*. O'Reilly & Associates. 1999.

Referencias en Internet

- [1] www.bibitec.org.mx
- [2] www.programacion.com/java/tutorial/java_basico/
- [3] www.java.com/

11. PRÁCTICAS

Unidad	Práctica	
1	1	Desarrollar un programa que implemente el uso de tipos definidos por el usuario que respondan a un problema real.
2	1	Crear varios componentes y librerías dinámicas que respondan a requerimientos generales.
	2	Desarrollar un programa que utilice los componentes y librerías dinámicas creados en la práctica anterior.
3	1	Desarrollar un programa que permita la creación y control de hilos, incluyendo cambios de estado de cada uno de ellos..
	2	Desarrollar un programa que integre varios hilos que compartan recursos entre ellos e implemente un mecanismo de control de recursos.
	3	Desarrollar un programa que incluya la planeación, asignación de prioridades, comunicación con hilos y monitorización.
	4	Desarrollar un programa de ordenamiento de cajón (Bucket sort) usando múltiples hilos.
4	1	Desarrollar un programa que tenga una interfaz grafica de usuario.
	2	Desarrollar un programa que permita la creación de gráficos en tiempo de ejecución.
5	1	Desarrollar un programa que cuente con todos los elementos de multimedia (audio, video, gráficos, texto) y que sea interactivo.
6	1	Desarrollar un programa que permita leer y escribir en los distintos puertos de la computadora.
	2	Desarrollar un programa que permita detectar un dispositivo conectado a uno de los puertos e indique el estado del mismo..
	3	Desarrollar una aplicación para manejo de interrupciones y parámetros utilizados en un programa de envío y recepción de datos.