



Técnicas avanzadas de programación

Clave	IA3P03
Horas teoría/semana	3
Horas práctica/semana	1
Duración semanas	32
Total de horas anuales	128
Número de créditos	8
Requisitos	CI2P02

Objetivo: El alumno diseñará sistemas de cómputo con interfaces gráficas de usuario, con base en el paradigma de la programación orientada a objetos y programación concurrente. El alumno elaborará programas eficientes, con un desempeño a prueba de fallas y que sean amigables con el usuario. Se utilizará como lenguaje de referencia a C++.

Temario	Horas
1. Importancia del software en la mecatrónica.	2
2. Metodología de la programación orientada a objetos en C++.	20
3. Desarrollo de sistemas de cómputo orientados a objetos.	12
4. Interfaces gráficas de usuario.	30
5. Programación concurrente.	32
Actividades prácticas	32
Total	128

1. Importancia del software en la mecatrónica. Diferencia entre programación y codificación. El lugar que ocupa el software en los sistemas mecatrónicos. Clasificación de los lenguajes de programación según su: nivel de abstracción (bajo nivel, nivel medio, alto nivel), propósito (general, específico), generación o evolución histórica (1GL, 2GL, 3GL, 4GL, 5GL), manera de ejecutarse (interpretados, compilados). Ejemplos de los distintos tipos de lenguajes de programación y sus áreas de aplicación.

2. Metodología de la programación orientada a objetos en C++. Clases y objetos. Constructores, atributos y métodos. Encapsulación, herencia y polimorfismo. Sobrecarga de funciones. Sobrecarga de operadores. Manejo de errores y de excepciones. Arreglos y colecciones. Implementación de interfaces. Manejadores de eventos. Construcción de bibliotecas y reutilización de código. Almacenamiento, actualización y eliminación de información en base a estructuras. Manejo de archivos (escritura, lectura, acceso secuencial, acceso aleatorio).

3. Desarrollo de sistemas de cómputo orientados a objetos. Análisis y definición de los requerimientos del usuario y del sistema. Diseño y modelado de sistemas, utilizando el Lenguaje de



Modelado Unificado (UML). Desarrollo de los subsistemas. Integración del sistema. Verificación, validación y pruebas del software. Mantenimiento del software (depuración, actualización, evolución).

4. Interfaces gráficas de usuario. Diseño y construcción de interfaces gráficas. Reglas y metodología para el diseño de interfaces claras, concisas, eficientes y fáciles de usar. Uso y configuración de controles predefinidos: formularios, botones de comando, cajas de texto, etiquetas, botones de opción, casillas de verificación, listas, timers, menús, barras de herramientas. Diseño y construcción de controles personalizados. Uso de la interfaz de dispositivos gráficos para realizar dibujos. Utilización del ambiente Qt para el desarrollo de interfaces gráficas en C++.

5. Programación concurrente. Introducción a la concurrencia. Necesidad de concurrencia. Diferencias entre procesos e hilos. Manejo y sincronización de procesos: procesos padre y atributos de los procesos; creación y destrucción de procesos hijos; asignación dinámica de memoria; memoria compartida.; Semáforos. Manejo de hilos: creación de hilos; Tratamiento de errores; Atributos de un hilo; Cancelación de hilos; Funciones seguras y reentrantes; Planificación y sincronización de hilos. Comunicación entre procesos e hilos: Señales; Tuberías; Sockets.

Bibliografía básica:

- Bronson, Gary J. C++ for Engineers and Scientists. Third edition. Cengage Learning. 2010.
- CAIRÓ, Osvaldo. Estructuras de datos. México. McGraw-Hill, 2006.
- Francisco M. Márquez. UNIX: Programación avanzada, 3ra edición Alfaomega 2004.

Bibliografía complementaria

- BOOCH, Grady. El lenguaje unificado de modelado. Manual de referencia. 2a edición. España. Pearson, 2007.
- BRAUDE, Eric J. Ingeniería de software. Una perspectiva orientada a objetos. México. Alfaomega, 2007.
- CAIRÓ, Osvaldo. Estructuras de datos. México. McGraw-Hill, 2006.
- LÓPEZ ROMÁN, Leobardo. Metodología de la programación orientada a objetos. México. Alfaomega, 2007.
- PRESSMAN, Roger S. Ingeniería del Software. Un enfoque práctico. México. McGraw-Hill, 2005.
- SCHMULLER, Joseph. Aprendiendo UML en 24 horas. México. Prentice-Hall, 2000.
- Milan Milenkovic. Sistemas Operativos: conceptos y diseño. 2da edición. McGraw-Hill 1996.
- Abraham Silberschatz. Fundamentos de Sistemas Operativos. 7ma edición McGraw-Hill 2006



Sugerencias didácticas:

Exposición oral	X	Uso de plataformas educativas	X
Exposición audiovisual	X	Lecturas obligatorias	X
Ejercicios dentro de clase	X	Trabajo de investigación	X
Ejercicios fuera de clase	X	Prácticas de laboratorio	X
Seminarios		Búsqueda especializada en internet	X
Uso de software especializado	X	Uso de redes sociales con fines académicos	

Sugerencias de evaluación:

Exámenes parciales	X	Elaboración de informes técnicos o proyectos	X
Exámenes finales	X	Participación en clase	X
Tareas fuera del aula	X	Asistencia a prácticas	X

Perfil profesional de quienes pueden impartir la asignatura:

Licenciatura en Ingeniería en Computación. Deseable haber realizado estudios de posgrado, contar con experiencia docente o haber participado en cursos o seminarios de iniciación en la práctica docente.