

PLAN DE ESTUDIOS (PE):Licenciatura en Ciencias de la Computación /
Ingeniería en Ciencias de la Computación

AREA:Ciencias de la Computación /Ingeniería en Computación

ASIGNATURA:Programación Concurrente y Paralela

CÓDIGO:CCOM-252

CRÉDITOS:5

FECHA:29de Junio 2012



1. DATOS GENERALES

| | |
|---|---|
| Nivel Educativo: | Licenciatura |
| Nombre del Plan de Estudios: | Licenciatura en Ciencias de la Computación |
| Modalidad Académica: | Presencial |
| Nombre de la Asignatura: | Programación Concurrente y Paralela |
| Ubicación: | Nivel Formativo |
| Correlación: | |
| Asignaturas Precedentes: | Estructuras de Datos |
| Asignaturas Consecuentes: | Programación Distribuida, Sistemas Operativos II |
| Conocimientos, habilidades, actitudes y valores previos: | <p>Conocimientos</p> <ul style="list-style-type: none"> • De la sintaxis de un lenguaje de programación. • De análisis y diseño de Algoritmos. • De estructuras de Datos. <p>Habilidades</p> <ul style="list-style-type: none"> • Creatividad en la búsqueda de soluciones a problemas concretos. • Capacidad creativa y analítica, inteligencia práctica, perseverancia y compromiso social. • Capacidad de observación, iniciativa y disposición al trabajo en equipo. • Habilidad para la lectura y comprensión de textos. • Para el desarrollo de Aplicaciones en algún lenguaje de Programación. <p>Actitudes y valores</p> <ul style="list-style-type: none"> • Una actitud favorable para adquirir nuevos conocimientos y realizar innovaciones. • Aplicar los conocimientos adquiridos en problemas reales. • De respeto y empatía con las personas. • De honestidad y responsabilidad en la entrega de trabajos. • De liderazgo y humanismo. |



| | |
|--|--|
| | <ul style="list-style-type: none"> • Actitud participativa. • Empatía, apertura al dialogo, comprensión y tolerancia hacia la diversidad cultural. |
|--|--|

2. CARGA HORARIA DEL ESTUDIANTE (Ver matriz 1)

| Concepto | Horas por periodo | | Total de horas por periodo | Número de créditos |
|---|-------------------|-----------|----------------------------|--------------------|
| | Teoría | Práctica | | |
| Horas teoría y práctica (16 horas = 1 crédito) | 48 | 32 | 80 | 5 |
| Total | 48 | 32 | 80 | 5 |

3. REVISIONES Y ACTUALIZACIONES

| | | |
|--|--|---|
| Autores: | Rafael De la Rosa Flores Leticia Mendoza Alonso Carmen Cerón Garnica Graciano Almanza Cruz Darnes Vilariño Ayala Hilda Castillo Zacatelco Mireya Tovar Vidal | Luis Enrique Colmenares Guillén Mario Anzures García Beatriz Beltrán Martínez Mario Rossainz López José Andrés Vázquez Flores Mariano Larios Gómez |
| Fecha de diseño: | 1 de Junio 2009 | |
| Fecha de la última actualización: | 12 de Junio 2012 | |
| Fecha de aprobación por parte de la academia de área | 29 de enero de 2013 | |
| Fecha de aprobación por parte de CDESCUA | 15-febrero-2013 | |
| Fecha de revisión del Secretario Académico | 18-febrero-2013 | |
| Revisores: | Carmen Cerón Garnica Mario Rossainz López Rafael de la Rosa Flores Luz A. Sánchez Gálvez Mario Anzures García Miguel Rodríguez Hernández Darnes Vilariño Ayala Mireya Tovar Vidal | |
| Sinopsis de la revisión y/o actualización: | Se cambió un tema de la unidad 2 (Especificación formal, añadiendo grafos de precedencias y sentencias Cobegin/Coend). | |

| | |
|--|---|
| | <p>Se añadió a la unidad 3 el tema de Aplicaciones reales a nivel usuario y herramientas para el análisis de programas</p> <p>A la unidad 4 se le agregó el tema de arquitectura multikernel.</p> <p>A la unidad 5 se cambió el 5.4 por programación con multikernel y se aumentó herramientas para análisis de programas</p> |
|--|---|

4. PERFIL DESEABLE DEL PROFESOR (A) PARA IMPARTIR LA ASIGNATURA:

| | |
|--------------------------|----------------------------|
| Disciplina profesional: | Ciencias de la Computación |
| Nivel académico: | Posgrado en Ciencias |
| Experiencia docente: | Mínima de 2 años |
| Experiencia profesional: | Mínima de 1 año |

5. OBJETIVOS:

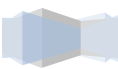
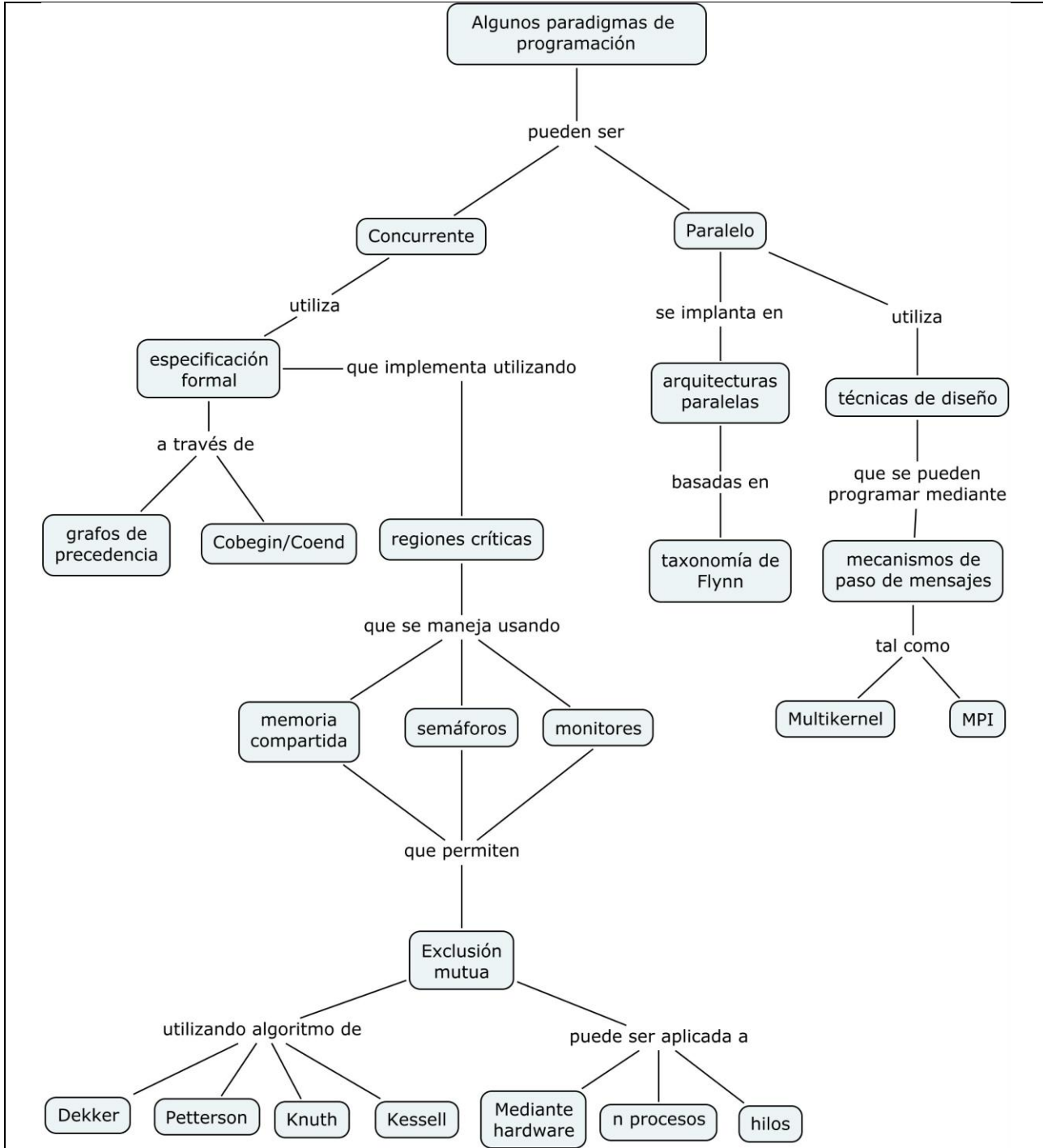
5.1 General: El estudiante desarrollará aplicaciones concurrentes y paralelas

5.2 Específicos: El estudiante será capaz de:

- Identificar el concepto de programación concurrente.
- Interpretar la definición de exclusión mutua y las condiciones de sincronización.
- Identificar la corrección de aplicaciones concurrentes.
- Distinguir la diferencia entre programación concurrente, programación distribuida y programación paralela.
- Identificar la especificación formal de sistemas concurrentes.
- Implementar algoritmos que permitan la especificación formal de sistemas concurrentes.
- Implementar aplicaciones concurrentes que permitan manejar el concepto de exclusión mutua y las condiciones de sincronización utilizando regiones críticas, semáforos y monitor.
- Conocer las herramientas que permiten el análisis y el diseño de las aplicaciones concurrentes.
- Reconocer los conceptos básicos de programación paralela.
- Manejar las diferentes arquitecturas.
- Conocer las herramientas que le permiten el análisis y el diseño de programas paralelos.
- Utilizar los mecanismos de comunicación mediante paso de mensajes para el desarrollo de aplicaciones en paralelo.
- Desarrollar aplicaciones utilizando MPI.
- Implementar algoritmos que se ejecuten sobre arquitecturas multi-kernel.



6. REPRESENTACIÓN GRÁFICA DE LA ASIGNATURA:



7. CONTENIDO

| Unidad 1 | Objetivo Específico | Contenido Temático/Actividades de aprendizaje | Bibliografía | |
|---|--|---|--|---|
| | | | Básica | Complementaria |
| Introducción | Identificar el concepto de programación concurrente | 1.1 Concepto de Concurrencia | Ben-Ari, M. (1990). Concurrent and Distributed Programming. (3ra ed). España: Prentice Hall Burns, A. & Davis, G. (1993) Concurrent Programming. (2da Ed) España: Prentice Hall | Gregory R. Andrews. (2000). Foundations of Multithreaded, Parallel, and Distributed Programming. México: Addison-Wesley |
| | | 1.2 Exclusión Mutua y Sincronización | | |
| | | 1.3 Corrección en Sistemas Concurrentes | | |
| | Interpretar la definición de exclusión mutua y las condiciones de sincronización | 1.4 Consideraciones sobre el Hardware | | |
| | | 1.5 Sistemas Distribuidos | | |
| | | 1.6 Sistemas de Tiempo Real | | |
| | | 1.7 Procesamiento Paralelo | | |
| Identificar la corrección de aplicaciones concurrentes | | | | |
| Distinguir la diferencia entre programación concurrente, programación distribuida y programación paralela | | | | |

| Unidad 2 | Objetivo Específico | Contenido Temático/Actividades de aprendizaje | Bibliografía | |
|--|---|--|---|--|
| | | | Básica | Complementaria |
| Especificación Formal de Sistemas Concurrentes | Identificar la especificación formal de sistemas concurrentes | 2.1. Condiciones de Bernstein / Aplique las condiciones de Bernstein para determinar las instrucciones que | Murata, T. (1999) Petri Nets: Properties, Analysis and Applications. España: Prentice | Michael G. Hinchey and Stephen A. Jarvis. (1995). Concurrent Systems: Formal Development in CSP. McGraw Hill |

| Unidad 2 | Objetivo Específico | Contenido Temático/Actividades de aprendizaje | Bibliografía | |
|----------|---|---|--------------|----------------|
| | | | Básica | Complementaria |
| | Implementar algoritmos que permitan la especificación formal de sistemas concurrentes | <p>se pueden ejecutar concurrentemente</p> <p>2.2. Grafos de Precedencia/Construya grafos de precedencia a partir de programas COBEGIN-COEND</p> <p>2.3 Sentencias COBEGIN-COEND/Desarrolle programas concurrentes a partir de grafos de precedencias</p> | Hall | Book Company. |

| Unidad 3 | Objetivo Específico | Contenido Temático/Actividades de aprendizaje | Bibliografía | |
|----------------------------|--|--|---|--|
| | | | Básica | Complementaria |
| Sistemas con memoria común | <p>Implementar aplicaciones concurrentes que permitan manejar el concepto de exclusión mutua y las condiciones de sincronización utilizando regiones críticas, semáforos y monitor</p> <p>Conocer las herramientas que permiten el análisis y el</p> | <p>3.1 El Problema de la Exclusión Mutua</p> <p>3.2 Concepto de Sección Crítica</p> <p>3.3 Refinamiento Sucesivo</p> <p>3.4 Algoritmo de Dekker</p> <p>3.5 Algoritmos de Peterson, Hyman, Knuth y Kesell</p> <p>3.6 Exclusión Mutua para N procesos</p> <p>3.7 Exclusión Mutua mediante Hardware</p> | <p>Ben-Ari, M. (1990). Concurrent and Distributed Programming. (3ra edición). España: Prentice Hall</p> <p>Burns, A. & Davis, G. (1993) Concurrent Programming. (2da Edición) España: Prentice Hall</p> | <p>Gregory R. Andrews. (1991). Concurrent Programming Principles and Practice. The Benjamin/Cummings Publishing Company.</p> <p>Robbins. (1997) .Unix Programación Práctica. Guía para la concurrencia, la comunicación y los multihilos. Prentice</p> |

| Unidad 3 | Objetivo Específico | Contenido Temático/Actividades de aprendizaje | Bibliografía | |
|----------|---|---|--------------|----------------|
| | | | Básica | Complementaria |
| | diseño de las aplicaciones concurrentes | 3.8 Semáforos, Regiones Críticas y Monitores | | Hall |
| | | 3.9 Aplicaciones reales a nivel de usuario | | |
| | | 3.10 Herramientas para el análisis de programas | | |

| Unidad 4 | Objetivo Específico | Contenido Temático/Actividades de aprendizaje | Bibliografía | |
|---|--|---|---|--|
| | | | Básica | Complementaria |
| Introducción a la Programación Paralela | Reconocer los conceptos básicos de programación paralela | 4.1 Concepto de Programación Paralela | Michael J. Quinn.(1994). Parallel Computing . Theory and Practice. (SecondEdition). McGraw- Hill. | Selim G. Akl (1992). Diseño y Análisis de Algoritmos Paralelos. RA-MA. Prentice Hall |
| | | 4.2 Arquitecturas paralelas: clasificación de Flynn | | |
| | 4.3 El Modelo PRAM | | | |
| | 4.4 Computadores MIMD | | | |
| | 4.5 Arquitectura Multikernel | | | |
| | Manejar las diferentes arquitecturas | | | |

| Unidad 5 | Objetivo Específico | Contenido Temático/Actividades de aprendizaje | Bibliografía | |
|---|--|---|---|---|
| | | | Básica | Complementaria |
| Técnicas de Diseño de Programas en Paralelo mediante el modelo de paso de | Conocer las herramientas que le permiten el análisis y el diseño de programas paralelos. | 5.1 Análisis y diseño de algoritmos en Paralelo (particionamiento, aglutinamiento, mapeo, agrupamiento) | Michael J. Quinn.(1994). Parallel Computing . Theory and Practice. (SecondEdition). McGraw- Hill. | Selim G. Akl(1992). Diseño y Análisis de Algoritmos Paralelos. RA-MA. Prentice Hall |

| Unidad 5 | Objetivo Específico | Contenido Temático/Actividades de aprendizaje | Bibliografía | |
|-----------|--|--|--------------|----------------|
| | | | Básica | Complementaria |
| mensajes. | <p>Utilizar los mecanismos de comunicación mediante paso de mensajes para el desarrollo de aplicaciones en paralelo</p> <p>Desarrollar aplicaciones utilizando MPI</p> <p>Implementar algoritmos que se ejecuten sobre arquitecturas multi-kernel.</p> | <p>5.1.1 Herramientas para el análisis de programas</p> <p>5.2 Programación básica utilizando paso de mensajes</p> <p>5.3 Programación con MPI</p> <p>5.4 Programación con multikernel</p> | | |

8.CONTRIBUCIÓN DEL PROGRAMA DE ASIGNATURA AL PERFIL DE EGRESO

| Asignatura | Perfil de egreso (anotar en las siguientes tres columnas, cómo contribuye la asignatura al perfil de egreso) | | |
|-------------------------------------|---|--|--|
| | Conocimientos | Habilidades | Actitudes y valores |
| Programación Concurrente y Paralela | <p>En los fundamentos matemáticos de la ciencias de la computación.</p> <p>De los conceptos principales y las teorías relacionadas con la ciencia de la computación.</p> <p>Necesarios para incorporarse a empresas o institutos de investigación, los cuales demanden el</p> | <p>Para analizar y generar modelos matemáticos que impliquen soluciones a problemas computacionales</p> <p>Para desarrollar y aplicar metodologías para el análisis, diseño e implementación de sistemas de cómputo.</p> <p>Para comunicar sus ideas y transferir conocimiento</p> | <p>Mostrará una actitud positiva y favorable a los cambios científico – tecnológicos.</p> <p>Mantendrá una actitud favorable para la actualización permanente en la disciplina.</p> <p>Estará preparado para incorporarse en el marco de la globalización.</p> <p>Estará preparado</p> |

| Asignatura | Perfil de egreso (anotar en las siguientes tres columnas, cómo contribuye la asignatura al perfil de egreso) | | |
|------------|--|-------------|--|
| | Conocimientos | Habilidades | Actitudes y valores |
| | <p>análisis y diseño de nuevas alternativas del uso de tecnologías de la computación.</p> <p>Para continuar con estudios de posgrado</p> | | <p>paratrabajar en equipo, emprender, liderar proyectos e incidir en la transformación sustentable de la realidad.</p> <p>Será un profesional responsable, solidario, crítico, ético y comprometido con la sociedad y con el medio ambiente. .</p> |

9. Describa cómo el eje o los ejes transversales contribuyen al desarrollo de la asignatura

| Eje (s) transversales | Contribución con la asignatura |
|--|--|
| Formación Humana y Social | Promover la ética en el desarrollo de programas como un elemento de formación integral, así como el interés por el cuidado del medio ambiente. |
| Desarrollo de Habilidades en el uso de las Tecnologías de la Información y la Comunicación | Desarrollo de aplicaciones concurrentes y paralelas mediante la programación de algoritmos. |
| Desarrollo de Habilidades del Pensamiento Complejo | Desarrollo de la habilidad de programación concurrente y en paralelo |
| Lengua Extranjera | Lectura de artículos especializados en el área de la programación concurrente y paralela, para favorecer la comprensión de la sintaxis de los lenguajes de programación. |
| Innovación y Talento Universitario | Planteamiento de programas usando técnicas de resolución de problemas que puedan ser utilizados en otras disciplinas de la ciencia. |
| Educación para la Investigación | Fomentar el hábito de investigar diversas técnicas de programación y compararlas con algunas otras. |



10. ORIENTACIÓN DIDÁCTICO-PEDAGÓGICA.

| Estrategias y Técnicas de aprendizaje-enseñanza | Recursos didácticos |
|---|---|
| <p>Estrategias de aprendizaje:</p> <ul style="list-style-type: none"> • Lectura y comprensión, • Reflexión, • Comparación, • Resumen. <p>Estrategias de enseñanza:</p> <ul style="list-style-type: none"> • ABP, • Aprendizaje activo, • Aprendizaje cooperativo, • Aprendizaje colaborativo, • Basado en el descubrimiento. <p>Ambientes de aprendizaje:</p> <ul style="list-style-type: none"> • Aula, • Laboratorio, • Simuladores. <p>Actividades y experiencias de aprendizaje:</p> <ul style="list-style-type: none"> • Asistencia a congresos <p>Técnicas</p> <ul style="list-style-type: none"> • grupales, • de debate, • del diálogo, • de problemas, • de estudio de casos, • cuadros sinópticos, • mapas conceptuales, • para el análisis, • comparación, • síntesis, • mapas mentales, • lluvia de ideas, • analogías, • portafolio, • exposición. | <p>Plumón y pizarrón, Ejercicios Prácticas de Laboratorio Libros, fotocopias Artículos científicos Antologías Materiales audiovisuales Programas informáticos (CD u on-line) educativos. Aplicaciones Multimedia Páginas Web Webquest correo electrónico chats foros</p> |

11. CRITERIOS DE EVALUACIÓN

| Criterios | Porcentaje |
|----------------------------|------------|
| • Exámenes | 35% |
| • Participación en clase | 15% |
| • Tareas | 15% |
| • Prácticas de laboratorio | 15% |
| • Proyectos Finales | 20% |
| Total | 100% |

12. REQUISITOS DE ACREDITACIÓN

| |
|---|
| Estar inscrito como alumno en la Unidad Académica en la BUAP |
| Asistir como mínimo al 80% de las sesiones |
| La calificación mínima para considerar un curso acreditado será de 6 |
| Cumplir con las actividades académicas y cargas de estudio asignadas que señale el PE |

13. Anexar (copia del acta de la Academia y de la CDESCUA con el Vo. Bo. del Secretario Académico)

