

Programa de asignatura por competencias de educación superior

Sección I. Identificación del Curso

Tabla 1. Identificación de la Planificación del Curso.

| | | | | | |
|--------------------------|---------------------------------------|-----------------------|-----------------------------------|------------------------|----------|
| Actualización: | Junio 20, 2022 | | | | |
| Carrera: | Ingeniería en Tecnologías de Software | Asignatura: | Computación Paralela | | |
| Academia: | Ciencias Computacionales Avanzadas / | Clave: | 19SCATS0704 | | |
| Módulo formativo: | Cómputo de Alto Desempeño | Seriación: | 19SCATS0807 - Sistemas Expertos | | |
| Tipo de curso: | Modalidad mixta | Prerrequisito: | 19SCATS0603 - Sistemas Operativos | | |
| Semestre: | Séptimo | Créditos: | 4.50 | Horas semestre: | 72 horas |
| Teoría: | 2 horas | Práctica: | 0 horas | Trabajo indpt.: | 2 horas |
| | | | | Total x semana: | 4 horas |

Sección II. Objetivos educacionales

Tabla 2. Objetivos educacionales

| Objetivos educacionales | | Criterios de desempeño | Indicadores |
|-------------------------|---|--|---|
| 1 | Solucionará problemas con sólidas bases científicas y fundamentos tecnológicos que le permitirán comprender, analizar, diseñar, organizar, producir, operar y dar soluciones prácticas a problemas relacionados con las áreas de Organización de Sistemas Computacionales e Ingeniería en Software para el sector productivo y social, promoviendo los principios de ética, responsabilidad y trabajo colaborativo. | El egresado implementará las diferentes etapas del ciclo de vida del software contemplando la protección de datos y prevención de desastres, salvaguardando con ética la seguridad de la información. | 50 % Egresados trabajarán en cualquier proceso del desarrollo de software o áreas afines a los sistemas computacionales, promoviendo los principios de ética, responsabilidad y trabajo colaborativo. |
| 2 | Aportará soluciones innovadoras y sustentables en el área de la electrónica en el que establezca el análisis, diseño, implementación, selección de componentes de hardware de uso específico, el software asociado y su conectividad a través de redes de comunicación para el sector productivo y social. | El egresado implementará las diferentes técnicas de análisis y diseño de circuitos electrónicos que den una solución innovadora sustentable a problemas con el hardware. | 20% Egresados trabajarán en cualquier proceso de creación y aplicación de hardware o áreas afines en el sector productivo y social. |
| 3 | Implementará soluciones innovadoras y sustentables con tecnologías de información que sean acordes a las necesidades, a las tecnologías disponibles y emergentes, para lograr un aprovechamiento óptimo de los recursos humanos y financieros en el sector productivo y social. | El egresado implementará las diferentes tecnologías emergentes en equipos multidisciplinarios que den una solución innovadora y sustentable a las necesidades que se presenten en el ámbito productivo y social. | 20 % Egresados trabajarán en la aplicación de Tecnologías de la información o áreas afines en el sector productivo o social. |



| Atributos de egreso de plan de estudios | | Criterios de desempeño | Componentes |
|---|--|--|---|
| 1 | Aplicar una experimentación adecuada con apoyo de metodologías y juicio ingenieril que permitan interpretar datos para obtener conclusiones que den solución a problemáticas en un contexto determinado. | <ul style="list-style-type: none"> - Será capaz de modelar los requerimientos de performance y determinar las características del hardware de servidores para atender volúmenes de transacciones. | <ul style="list-style-type: none"> 1. Performance. <ul style="list-style-type: none"> 1.1 Ley de Amdahl. 1.2 Diferencias de performance dependiendo de las tecnologías de almacenamiento utilizadas: HDD vs SSD. |
| 2 | Identificar la necesidad de actualizarse constantemente para innovar y desarrollar la tecnología de software que sea amigable con el medio ambiente. | <ul style="list-style-type: none"> - Será capaz de escribir códigos que corran simultáneamente, para procesar volúmenes altos de transacciones. - Será capaz de programar núcleos del procesador y lograr un trabajo colaborativo entre ellos. | <ul style="list-style-type: none"> 2. Programación Concurrente. <ul style="list-style-type: none"> 2.1 Hilos y procesos. 2.2 Monitores, Mutex, Semáforos. 2.3 Región crítica, Inanición, Condición de carrera, Algoritmos voraces, Abrazo mortal. 3. Cómputo Paralelo. <ul style="list-style-type: none"> 3.1 Algoritmos paralelizables. 3.2 OpenMP. 3.3 Deadlock, Regiones críticas, Work units, Distribución de cargas. |

Sección III. Atributos de la asignatura

Tabla 3. Atributos de la asignatura

| Problema a resolver | | |
|--|---|---|
| Modelar con base a los requerimientos de performance para dar solución a volúmenes de transacciones concretos utilizando las herramientas del cómputo concurrente cuando las características de un sistema en concreto lo requieran o las herramientas del cómputo paralelo, cuando las características de un sistema en concreto lo requieran. | | |
| Atributos (competencia específica) de la asignatura | | |
| Modelar con base a los requerimientos de performance de sistemas, siendo capaz de utilizar la Ley de Amdahl y de determinar las características de hardware necesarias en un sistema concreto; Codificar programas concurrentes, así como resolver los problemas de la sincronización de procesos derivados del diseño de sistemas de software; Codificar problemas sencillos de cómputo paralelo, utilizando el framework libre OpenMP. | | |
| Aportación a la competencia específica | | Aportación a las competencias transversales |
| Saber | Saber hacer | Saber Ser |
| - Identificar los requerimientos de performance de sistemas, siendo capaz de utilizar la Ley de Amdahl y de determinar las características de hardware necesarias en un sistema concreto. | <ul style="list-style-type: none"> - Modelar los requerimientos de performance y determinar las características del hardware de servidores para atender volúmenes de transacciones. - Desarrollar códigos que corran simultáneamente, para procesar volúmenes altos de transacciones. - Programar núcleos del procesador y lograr un trabajo colaborativo entre ellos. - Resolver los problemas de la sincronización de procesos derivados del diseño de sistemas de software utilizando programas concurrentes. - Resolver problemas sencillos de cómputo paralelo, utilizando el framework libre OpenMP. | <ul style="list-style-type: none"> - Aporta puntos de vista con apertura a aprender de los otros y considera los de otras personas de manera reflexiva y respetuosa. - Participa activamente en la construcción de su aprendizaje y en la resolución de problemas, colaborando de manera productiva en espacios y equipos de trabajo. - Cumple en tiempo y forma en sus obligaciones como estudiante, siguiendo las indicaciones y considerando los criterios de evaluación. |
| Producto integrador de la asignatura, considerando los avances por unidad | | |
| Portafolio de las prácticas realizadas en las unidades de aprendizaje. | | |

Sección IV. Desglose específico por cada unidad formativa

Tabla 4.1. Desglose específico de la unidad "Performance"

| Número y nombre de la unidad: 1. Performance | | | | | | | |
|--|---|--|---|---|---------|--------------------------|--------|
| Tiempo y porcentaje para esta unidad: | | Teoría: | 8 horas | Práctica: | 8 horas | Porcentaje del programa: | 22.22% |
| Aprendizajes esperados: | | Modelar los requerimientos de performance y determinar las características del hardware de servidores para atender volúmenes de transacciones. | | | | | |
| Temas y subtemas (secuencia) | Criterios de desempeño | Estrategias didácticas | Estrategias de evaluación | Producto Integrador de la unidad (Evidencia de aprendizaje de la unidad) | | | |
| 1. Performance. 1.1 Ley de Amdahl. 1.2 Diferencias de performance dependiendo de las tecnologías de almacenamiento utilizadas: HDD vs SSD. | Saber: - Identificar los requerimientos de performance para el modelado y determinar las características del hardware de servidores para atender volúmenes de transacciones. Saber hacer: - Modelar los requerimientos de performance y determinar las características del hardware de servidores para atender volúmenes de transacciones. Ser: - Aporta puntos de vista con apertura a | - Identificación de conocimientos previos. - Exposición didáctica. - Debates. - Mapas conceptuales. | Evaluación diagnóstica: - Cuestionario de conocimientos previos. Evaluación formativa: - Debates y actividades en clase. Evaluación sumativa: - Examen. | Portafolio de evidencias: - Actividades y ejercicios. - Examen de conocimientos de los temas de esta unidad. | | | |



Continuación: Tabla 4.1. Desglose específico de la unidad "Performance"

| Temas y subtemas (secuencia) | Criterios de desempeño | Estrategias didácticas | Estrategias de evaluación | Producto Integrador de la unidad |
|--|---|------------------------|---------------------------|----------------------------------|
| | <p>aprender de los otros y considera los de otras personas de manera reflexiva y respetuosa.</p> <ul style="list-style-type: none"> - Participa activamente en la construcción de su aprendizaje y en la resolución de problemas, colaborando de manera productiva en espacios y equipos de trabajo. - Cumple en tiempo y forma en sus obligaciones como estudiante, siguiendo las indicaciones y considerando los criterios de evaluación. - Utiliza la tecnología para apoyar su aprendizaje y para el desarrollo de habilidades metacognitivas, el aprendizaje autónomo y el longlife learning. | | | |
| Bibliografía | | | | |
| <ul style="list-style-type: none"> - Wilkinson, B.; Allen, M. (2005). Parallel Programming: Techniques and Applications using Networked Workstations and Parallel Computers. USA: Pearson: Prentice Hall - Quinn, M.J. (2004). Parallel Programming in C with MPI and OpenMP. USA: McGraw Hill. - Andrews, G.R. (2000). Foundations of Multithreaded, Parallel, and Distributed Programming. USA: Addison Wesley. - Pacheco, P.S. (1997). Parallel Programming with MPI. USA: Morgan Kaufmann. - Tanenbaum, A.S.; Van Steen, M. (2002). "Distributed Systems: Principles and Paradigms. USA: Prentice Hall. | | | | |

Sección IV. Desglose específico por cada unidad formativa

Tabla 4.2. Desglose específico de la unidad "Programación Concurrente."

| Número y nombre de la unidad: 2. Programación Concurrente. | | | | |
|--|--|--|---|---|
| Tiempo y porcentaje para esta unidad: | | Teoría: 8 horas | Práctica: 8 horas | Porcentaje del programa: 22.22% |
| Aprendizajes esperados: Desarrollar códigos que corran simultáneamente, para procesar volúmenes altos de transacciones. | | | | |
| Temas y subtemas (secuencia) | Criterios de desempeño | Estrategias didácticas | Estrategias de evaluación | Producto Integrador de la unidad (Evidencia de aprendizaje de la unidad) |
| 2. Programación Concurrente. 2.1 Hilos y procesos. 2.2 Monitores, Mutex, Semáforos. 2.3 Región crítica, Inanición, Condición de carrera, Algoritmos voraces, Abrazo mortal. | Saber: - Comprender las características de los códigos que corran simultáneamente, para procesar volúmenes altos de transacciones. Saber hacer: - Desarrollar códigos que corran simultáneamente, para procesar volúmenes altos de transacciones. Ser: - Aporta puntos de vista con apertura a aprender de los otros y considera los de otras personas de manera reflexiva y respetuosa. | - Aprendizaje basado en estudio de casos. - Exposición didáctica. | Evaluación formativa: - Análisis de estudios de caso. - Actividades realizadas en clase. Instrumento: Lista de cotejo. Evaluación sumativa: - Práctica y su reporte. | Portafolio de evidencias: - Codificación y ejecución individual de un programa que corra simultáneamente para procesar volúmenes de alto de transacciones |



Continuación: Tabla 4.2. Desglose específico de la unidad "Programación Concurrente."

| Temas y subtemas (secuencia) | Criterios de desempeño | Estrategias didácticas | Estrategias de evaluación | Producto Integrador de la unidad |
|--|--|------------------------|---------------------------|----------------------------------|
| | <ul style="list-style-type: none"> - Participa activamente en la construcción de su aprendizaje y en la resolución de problemas, colaborando de manera productiva en espacios y equipos de trabajo. - Cumple en tiempo y forma en sus obligaciones como estudiante, siguiendo las indicaciones y considerando los criterios de evaluación. - Utiliza la tecnología para apoyar su aprendizaje y para el desarrollo de habilidades metacognitivas, el aprendizaje autónomo y el longlife learning. | | | |
| Bibliografía | | | | |
| <ul style="list-style-type: none"> - Wilkinson, B.; Allen, M. (2005). Parallel Programming: Techniques and Applications using Networked Workstations and Parallel Computers. USA: Pearson: Prentice Hall - Quinn, M.J. (2004). Parallel Programming in C with MPI and OpenMP. USA: McGraw Hill. - Andrews, G.R. (2000). Foundations of Multithreaded, Parallel, and Distributed Programming. USA: Addison Wesley. - Pacheco, P.S. (1997). Parallel Programming with MPI. USA: Morgan Kaufmann. - Tanenbaum, A.S.; Van Steen, M. (2002). "Distributed Systems: Principles and Paradigms. USA: Prentice Hall. | | | | |

Sección IV. Desglose específico por cada unidad formativa

Tabla 4.3. Desglose específico de la unidad "Cómputo Paralelo."

| Número y nombre de la unidad: 3. Cómputo Paralelo. | | | | | | | |
|---|---|--|---|--|---------|--------------------------|--------|
| Tiempo y porcentaje para esta unidad: | | Teoría: | 8 horas | Práctica: | 8 horas | Porcentaje del programa: | 22.22% |
| Aprendizajes esperados: Programar núcleos del procesador para lograr un trabajo colaborativo entre ellos. | | | | | | | |
| Temas y subtemas (secuencia) | Criterios de desempeño | Estrategias didácticas | Estrategias de evaluación | Producto Integrador de la unidad (Evidencia de aprendizaje de la unidad) | | | |
| 3. Cómputo Paralelo. 3.1 Algoritmos paralelizables. 3.2 OpenMP. 3.3 Deadlock, Regiones críticas, Work units, Distribución de cargas. | Saber: - Identificar como programar núcleos del procesador para el logro de un trabajo colaborativo entre ellos. Saber hacer: - Ser capaz de programar núcleos del procesador y lograr un trabajo colaborativo entre ellos. Ser: - Aporta puntos de vista con apertura a aprender de los otros y considera los de otras personas de manera reflexiva y respetuosa. | - Aprendizaje basado en estudio de casos. - Trabajo colaborativo. | Evaluación formativa: - Estudio de casos. Instrumento: Lista de cotejo. Evaluación sumativa: - Prácticas y su reporte. Instrumento: Lista de cotejo. | Portafolio de evidencias: El proceso elaborado en equipo debe incluir: Prácticas aplicadas a la programación de núcleos. | | | |



Continuación: Tabla 4.3. Desglose específico de la unidad "Cómputo Paralelo."

| Temas y subtemas (secuencia) | Criterios de desempeño | Estrategias didácticas | Estrategias de evaluación | Producto Integrador de la unidad |
|--|--|------------------------|---------------------------|----------------------------------|
| | <ul style="list-style-type: none"> - Participa activamente en la construcción de su aprendizaje y en la resolución de problemas, colaborando de manera productiva en espacios y equipos de trabajo. - Cumple en tiempo y forma en sus obligaciones como estudiante, siguiendo las indicaciones y considerando los criterios de evaluación. - Utiliza la tecnología para apoyar su aprendizaje y para el desarrollo de habilidades metacognitivas, el aprendizaje autónomo y el longlife learning. | | | |
| Bibliografía | | | | |
| <ul style="list-style-type: none"> - Wilkinson, B.; Allen, M. (2005). Parallel Programming: Techniques and Applications using Networked Workstations and Parallel Computers. USA: Pearson: Prentice Hall - Quinn, M.J. (2004). Parallel Programming in C with MPI and OpenMP. USA: McGraw Hill. - Andrews, G.R. (2000). Foundations of Multithreaded, Parallel, and Distributed Programming. USA: Addison Wesley. - Pacheco, P.S. (1997). Parand Programming with MPI. USA: Morgan Kaufmann. - Tanenbaum, A.S.; Van Steen, M. (2002). "Distributed Systems: Principles and Paradigms. USA: Prentice Hall. | | | | |



V. Perfil docente

Tabla 5. Descripción del perfil docente

| Perfil deseable docente para impartir la asignatura |
|---|
| <p>Carrera(s): - Ingeniería en Tecnologías de software.</p> <p>- Ingeniería en Informática, Computación.</p> <p>- Licenciatura en Informática o Sistemas Computacionales o Maestría relacionada con el área de conocimiento o carreras afín. o carrera afín</p> <p>- Tenga una formación sólida en computación paralela.</p> <p>Esté familiarizado con las aplicaciones de la computación paralela y su aplicación en la resolución de problemas reales de la sociedad.</p> <p>Tenga disposición para incorporar el empleo de recursos educativos interactivos y multimedia, disponibles en línea, durante el proceso de aprendizaje</p> <p>- Experiencia mínima de dos años</p> <p>- Licenciatura o Ingeniería. Deseable Maestría.</p> |