

| | | |
|--|--|---|
| <p style="text-align: center;">UNIVERSIDAD AUTÓNOMA DE CHIHUAHUA</p>  <p style="text-align: center;">UNIVERSIDAD AUTÓNOMA DE CHIHUAHUA</p> <p style="text-align: center;">PROGRAMA ANALÍTICO DE LA UNIDAD DE APRENDIZAJE: CONCEPTOS AVANZADOS DE PROGRAMACIÓN</p> | DES: | INGENIERÍA |
| | Programa Educativo | Ingeniería en Ciencias de la Computación |
| | Tipo de materia (Obli/Opta): | Obligatoria |
| | Clave de la materia: | CI274 |
| | Semestre: | 2 |
| | Área en plan de estudios (G, E): | Ciencias de la Ingeniería |
| | Total, de horas por semana: | 4 |
| | <i>Teoría: Presencial o Virtual</i> | 2 |
| | <i>Laboratorio o Taller:</i> | 2 |
| | <i>Prácticas:</i> | 0 |
| | <i>Trabajo extra-clase:</i> | 0 |
| | Créditos Totales: | 4 |
| | Total, de horas semestre (x 16 sem): | 64 |
| | Fecha de actualización: | Febrero 2023 |
| | CONCEPTOS AVANZADOS DE PROGRAMACIÓN | <i>Prerrequisito (s):</i> |
| | <i>Realizado por:</i> | Comité de Rediseño Curricular |
| PROPÓSITO DEL CURSO | | |
| <p>El curso proporciona conceptos y técnicas de programación mediante las cuales el estudiante pueda escribir programas de cómputo adecuados, utilizando los diferentes paradigmas y lenguajes, así como el uso de las diversas plataformas de desarrollo. La aplicación de los diversos lenguajes, los conocimientos y las técnicas de programación, promueven en el estudiante la habilidad para desarrollar aplicaciones avanzadas y complejas en las que entren a formar parte aspectos avanzados de concurrencia, subprograma y programación defensiva.</p> | | |
| COMPETENCIAS (Tipo Y Nombre de la competencias que nutre la materia y a las que contribuye) | DOMINIOS COGNITIVOS (Objetos de estudio, temas y subtemas) | RESULTADOS DE APRENDIZAJE. (Por objeto de estudio). |
| El curso promueve las siguientes competencias: | I. EJECUCION DE PROGRAMAS | Distingue entre lenguajesde |

BÁSICAS:

SOLUCION DE PROBLEMAS

- Aplica las diferentes técnicas de observación para la solución de problemas.
- Distingue los diversos tipos de sistemas

- 1.1 Interpretación Vs. Compilación
- 1.2 Ejecución nativa
- 1.3 Máquinas virtuales
- 1.4 Esquema de memoria en tiempo de ejecución: call-stack, heap, static data
- 1.5 Concepto de administración de memoria: allocating, de-allocating, garbage collection.

programación interpretados y compilados.

Identifica los ambientes y plataformas para la ejecución de programas.

Ejecuta programas escritos en plataformas interpretadas y compiladas.

| | | |
|---|---|--|
| <ul style="list-style-type: none"> • Aplica la tecnología a la solución de problemáticas • Aplica el enfoque sistémico en diversos contextos • Crea soluciones innovadoras y utiliza formas no convencionales en la solución de problemas | | <p>Escribe programas simples que aplican los conceptos de administración de memoria.</p> |
| <p>PROFESIONALES:</p> <p>CIENCIAS FUNDAMENTALES DE LA INGENIERÍA</p> | <p>II. SUBPROGRAMAS</p> <p>2.1 Concepto de subprograma 2.2 Referenciamiento local, alcance y tiempo de vida de parámetros. 2.3 Sobrecarga de operadores 2.4 Bloques y anidamiento</p> | <p>Define el concepto de subprograma</p> <p>Escribe programas simples usando subprogramas</p> |
| <ul style="list-style-type: none"> • Utiliza las matemáticas como herramientas para solución de problemas en ingeniería | <p>III. MANEJO DE EVENTOS Y EXCEPCIONES</p> <p>3.1 Concepto de excepción 3.2 Concepto de evento 3.3 Programación defensiva 3.3.1 Captura de excepciones 3.3.2 Aserciones</p> | <p>Distingue entre los conceptos de excepción y evento.</p> <p>Escribe programas simples utilizando los principios de programación defensiva</p> |
| <p>ESPECIFICAS:</p> <p>FUNDAMENTOS DE CIENCIAS DE LA COMPUTACIÓN</p> | <p>IV. CONCURRENCIA.</p> <p>4.1 Concepto de concurrencia 4.1.1 Semáforos 4.1.2 Monitores 4.1.3 Hilos de control</p> | <p>Explica el concepto de concurrencia.</p> <p>Escribe programas simples con hilos de ejecución independientes.</p> |
| <ul style="list-style-type: none"> • Diseña y aplica algoritmos, estructuras y representación de datos para soluciones computacionales • Distingue los fundamentos de un sistema computacional • Aplica las bases de los lenguajes de programación para generar aplicaciones óptimas • Aplica los tópicos de paradigmas de programación: estructurado, orientado a objetos. Lógico, funcional, entre otros para la creación de aplicaciones óptimas | <p>V. PARADIGMAS DE PROGRAMACIONY LA EVOLUCION DE LOS LENGUAJES DE PROGRAMACION</p> <p>5.1 Generaciones de lenguajes de programación 5.2 Paradigma estructurado 5.3 Paradigma funcional 5.4 Paradigma lógico 5.5 Paradigma Orientado a Objetos 5.6 Combinación de paradigmas</p> | <p>Identifica los diferentes paradigmas de programación y los relaciona con su aplicación</p> <p>Explica la mezcla de paradigmas para aplicaciones robustas.</p> |

| OBJETO DE ESTUDIO | METODOLOGIA (Estrategias, secuencias, recursos didácticos) | EVIDENCIAS DE APRENDIZAJE. |
|--|--|--|
| I. EJECUCIÓN DE PROGRAMAS II. SUBPROGRAMAS III. MANEJO DE EVENTOS Y EXCEPCIONES IV. CONCURRENCIA V. PARADIGMAS DE PROGRAMACIÓN Y LA EVOLUCIÓN DE LOS LENGUAJES DE PROGRAMACIÓN | Aprendizaje interactivo (exposición del profesor) Grupo de discusión. Auto aprendizaje (búsqueda y análisis de información) Inductivo <ul style="list-style-type: none"> • Observación • Comparación Deductivo <ul style="list-style-type: none"> • Aplicación • Comprobación • Demostración Sintético <ul style="list-style-type: none"> • Recapitulación • Definición • Resumen • Esquemas • Modelos matemáticos • Conclusión. Material de Apoyo didáctico: Recursos <ul style="list-style-type: none"> • Manual de Instrucción • Materiales gráficos: artículos, libros, diccionarios, etc. • Cañón • Pintarrones • Equipo de computo | <ul style="list-style-type: none"> • Tareas y ejercicios. • Escritura de programas de cómputo. • Exámenes escritos. |

| FUENTES DE INFORMACIÓN (Bibliografía, Direcciones electrónicas) | EVALUACIÓN DE LOS APRENDIZAJES (Criterios e instrumentos) |
|--|--|
| 1. Concepts of Programming Languages. Robert W. Sebesta. Cualquier edición. | INSTRUMENTOS: <ul style="list-style-type: none"> • Exámenes • Solución de ejercicios (aplicación de conocimientos) • Lista de cotejo (Respeto y participación al trabajo dentro del salón de clase, interés por la asignatura) CRITERIOS DE DESEMPEÑO: <ul style="list-style-type: none"> • Los programas desarrollados deberán estar completos. • Las tareas se deberán entregar en tiempo y forma |

| | |
|--|--|
| | <ul style="list-style-type: none"> • Exámenes escritos: Se realizan 3 exámenes escritos durante el semestre y las fechas se establecen por la secretaría académica <p>Se toma en cuenta para integrar calificaciones parciales: Exámenes 60% Tareas y practicas 40%</p> <p>Fecha de exámenes parciales: 1º. Parcial: 2º. Parcial: 3er Parcial:</p> <p>LAS ACTIVIDADES NO REALIZADAS EN TIEMPO Y FORMA SE CALIFICAN CON CERO.</p> |
|--|--|

Cronograma del Avance Programático

S e m a n a s

| Unidades de aprendizaje | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|
| I. Ejecución de programas | | | | | | | | | | | | | | | | |
| II. Subprogramas | | | | | | | | | | | | | | | | |
| III. Manejo de eventos y excepciones | | | | | | | | | | | | | | | | |
| IV. Concurrencia | | | | | | | | | | | | | | | | |
| V. Paradigmas de programación y la evolución de los lenguajes de programación | | | | | | | | | | | | | | | | |