##### Escudo

##### UNIVERSIDAD DEL BÍO-BÍO

##### VICERRECTORÍA ACADÉMICA – DIRECCION DE DOCENCIA

# **PROGRAMA DE ASIGNATURA**

##### I. IDENTIFICACIÓN

|  |  |
| --- | --- |
| Nombre Asignatura: Fundamentos de Ciencias de la Computación | Período de Vigencia:2015 - 2017 |
| **Código**: 620437 |
| **Tipo de Curso**: Obligatorio, Formación de Especialidad. Ingeniería aplicada |

|  |  |  |  |
| --- | --- | --- | --- |
| **Carrera:** Ingeniería Civil en Informática | **Departamento**: Ciencias de la Computación y Tecnologías de Información.  Sistemas de Información | | **Facultad**: Ciencias Empresariales |
| **Nº Créditos SCT:** 6 | **Total de horas:**  **Cronológicas:**162  **Pedagógicas:**252 | | **Año/ semestre:**  3/1 |
| **Horas presenciales:**108  **HT:**4  **HP:** 2  **HL**: 0 | | **Horas trabajo autónomo:**144  **HT:** 4  **HP:**4  **HL:**0 | |
| **Prerrequisitos**:  Asignatura: Estructuras de Datos  Código: 620435 | | **Correquisitos:**  No tiene | |

##### II.- DESCRIPCIÓN

**ll.1 Presentación: Relación de la Asignatura con las Competencias del Perfil de Egreso**

|  |
| --- |
| Asignatura teórico – práctica, impartida en el tercer año,primer semestre,entrega los fundamentos de las ciencias de la computación que permiten al estudiante discriminar las potencialidades y limitaciones de los computadores, métodos y lenguajes computacionales.  Tributa a las competencias específicas:   * Resolver problemas de programación utilizando lenguajes de programación y modelado de acuerdo a reglas y estándares existentes, y aplicando estrategias que aseguren la generación de soluciones eficientes. * Construir aplicaciones de software, probando su funcionalidad y eficiencia, mediante el uso de arquitecturas, modelos, patrones, técnicas y herramientas de programación pertinentes para distintas plataformas.   Así también tributa a las competencias genéricas:   * Manifestar una actitud permanente de búsqueda y actualización de sus aprendizajes, incorporando los cambios sociales, científicos y tecnológicos en el ejercicio y desarrollo de su profesión. * Comunicar ideas y sentimientos en forma oral y escrita para interactuar efectivamente en el entorno social y profesional en su lengua materna y en un nivel inicial en un segundo idioma. |

##### ll.2 Descriptor de competencias (metas de la asignatura)

|  |
| --- |
| Analizar diferentes modelos de computación y los problemas que estos son capaces de solucionar.  Resultados de aprendizaje:   1. Reconoce lenguajes regulares y sus representaciones en forma de autómatas finitos y expresiones regulares para aplicarlos a situaciones prácticas. 2. Utiliza representaciones en forma de autómatas de pila y gramáticas libres del contexto para aplicar la teoría de parsing. 3. Analiza los lenguajes decidibles y aceptables para comprender que existen problemas que no se pueden resolver por computador. 4. Utiliza Máquinas de Turing como modelo de computación para determinar si un problema se puede resolver con los computadores actuales. 5. Maneja el concepto de NP-completitud para determinar las clases de problemas existentes. |

**ll.3 Aprendizajes Previos**

|  |
| --- |
| * Utiliza estructuras de datos básicas. * Implementa algoritmos sencillos. |

**III. RESULTADOS DE APRENDIZAJE**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Resultados de Aprendizaje** | **Metodología** | **Criterios de Evaluación** | **Contenidos conceptuales, procedimentales y actitudinales.** | **Tiempo estimado** |
| 1. Reconoce lenguajes regulares y sus representaciones en forma de autómatas finitos y expresiones regulares para aplicarlos a situaciones prácticas. | * Clases expositivas con desarrollo de ejercicios de manera socializada. * Trabajo grupal e individual. | * 1. Resuelve problemas simples asociados Lenguajes y Gramáticas Regulares utilizando losconceptosteóricos.   2. Describe lenguajes regulares utilizando Expresiones Regulares.   3. Reconoce las palabras pertenecientes a Lenguajes Regulares específicos a través de la construcción de AFD, AFND.   4. Genera lenguajes regulares a través del diseño deGramáticas Regulares.   5. Convierte representaciones de un Lenguaje Regular entre ellas: AFD-GR, GR-AFD, AFD-ER, ER-AFD.   6. Aplica Lema de Bombeo para demostrar si un lenguaje específico es o no es regular.   7. Aplica propiedades algorítmicas de los lenguajes regulares para interiorizarlas correctamente. | **Conceptuales:**  **Conceptos de Lenguajes Regulares**   1. Expresiones Regulares (ERs) 2. Tipos de Autómatas 3. Autómatas Finitos Determinísticos (AFDs) 4. Autómatas Finitos No Determinísticos (AFNDs) 5. Conversión de ER a AFND 6. Conversión de AFND a AFD 7. Conversión de AFD a ER 8. Propiedades de Clausura 9. Lema del Bombeo 10. Propiedades Algorítmicas de Lenguajes Regulares   **Conceptos de Gramáticas Regulares:**   1. Clasificación de Gramáticas 2. Definición de Gramática Regular. 3. Conversión GR-AFD 4. Conversión AFD-GR   **Procedimentales:**   1. Diseño de AFD, AFND, AFDpara reconocimiento de lenguajes regulares. 2. Diseño de gramáticas Regulares para generación de lenguajes Regulares. 3. Diseño de Expresiones Regulares para describir un lenguaje Regular. 4. Conversión entre representaciones: de ER a AFND; de AFND a AFD; de AFD a ER; de GR a AFD 5. Reconocimiento de Equivalencias entre diferentes representaciones de un mismo LR. 6. Uso de jFlag para formular soluciones a problemas.   **Actitudinales:**   * Rigurosidad en la búsqueda y actualización de información * Responsabilidad y colaboración en el trabajo en equipo | Horas presenciales:  HT: 16  HP: 8  HL: 0  Horas de trabajo autónomo:  HT: 16  HP: 16  HL:0 |
| 1. Utiliza representaciones en forma de autómatas de pila (AP) y gramáticas libres del contexto (GLL) para aplicar la teoría de parsing. | * Clases expositivas con resolución socializada de ejercicios. * Trabajo grupal e individual. | * 1. Identifica conceptos de Lenguajes y Gramáticas Libres del Contexto (LLC y GLC) diferenciándolos de los lenguajes Regulares.   2. Reconoce palabras pertenecientes a LLC a través de la construcción de Autómatas de Pila (AP)   3. Genera LLC bien definidos a través de la construye de GLC adecuadas.   4. Convierte representaciones de un LLC a GLC y viceversa.   5. Reconocesintáxis válidas de Lenguajes de programación a través de su descripción BNF o con diagramas sintácticos.   6. Reconoce, en definiciones formales, la existencia del problema de ambigüedad inherente a GLC y propiedades de clausura de LLC.   7. Aplica Algoritmos de decisión para determinar si un LLC es vacío, finito oinfinito. | **Conceptuales:**   1. Autómatas Apiladores 2. Gramáticas Libres de Contexto (GLC). 3. Configuración de las Gramáticas Libres de Contexto 4. Árboles de Derivación Simplicación de GLC. 5. Formas Normales 6. Equivalencia entre Lenguajes Libres del contexto (LLC) y Autómatas Apiladores (AP). 7. Ambiguedad Inherente 8. Gramáticas BNF y BNF-extendida. 9. Diagramas Sintácticos. 10. Lema de Bombeo para LLC. 11. Propiedades de Clausura 12. Algoritmos de Decisión   **Procedimentales:**   1. Diseño de Autómatas de Pila para reconocimiento de LLC. 2. Diseño deGLC para generación de LLC. 3. Conversión de representaciones de AP a GLC y viceversa. 4. Reconocimiento de Equivalencias entre diferentes representaciones de un mismo LLC. 5. Implementación de las etapas de análisis léxico y sintáctico de un traductor simple. 6. Uso de Flex, Yacc y jFlag para formular soluciones a problemas.   **Actitudinales:**   * Rigurosidad en el trabajo de programación * Rigurosidad en la búsqueda y actualización de información * Responsabilidad y colaboración en el trabajo en equipo | Horas presenciales:  HT: 16  HP: 08  HL: 0  Horas de trabajo autónomo:  HT: 16  HP:16  HL:0 |
| 1. Analiza los lenguajes decidibles y aceptables para comprender que existen problemas que no se pueden resolver por computador. | * Clases expositivas con resolución conjunta de problemas. * Trabajo grupal e individual. | * 1. Reconoce Lenguajes Recursivamente enumerables a través de la construcción de Máquinas de Turing (MT).   2. Analiza la decibilidad de los lenguajes utilizando una MT. | **Conceptuales:**   1. Definición de Lenguajes Decidibles y Semidecidibles 2. Algoritmos 3. Máquina de Turing (MT) 4. Técnicas para la construcción de MT’s. 5. Almacenamiento en el Control Finito 6. Pistas Multiples   **Procedimentales:**   1. Diseño deMT’spara reconocimiento de Lenguajes Recursivos (LR). 2. Análisis de decibilidad de lenguajes. 3. Uso de jFlag para formular soluciones a problemas.   **Actitudinales:**   * Rigurosidad en la búsqueda y actualización de información * Responsabilidad en el trabajo en equipo | Horas presenciales:  HT: 16  HP: 08  HL: 0  Horas de trabajo autónomo:  HT: 16  HP: 16  HL:0 |
| 1. Utiliza Máquinas de Turing (MT) como modelo de computación para determinar si un problema se puede resolver con los computadores actuales. | * Clases expositivas con resolución socializada de problemas. * Trabajo grupal colaborativo e individual | * 1. Reconoce Lenguajes Recursivamente enumerables a través de la construcción de MT.   2. Analiza la computabilidad de los lenguajes a través de MT’s.   3. Construye MT para generar lenguajes formales de nivel 0. | **Conceptuales:**   1. Lenguajes y Funciones Computables 2. Extensiones al Modelo 3. Cinta Infinita en Ambas Direcciones 4. MT con Varias Cintas 5. Movidas No Determinísticas 6. Máquinas Multidimensionales 7. Máquinas de Varias Cabezas 8. Hipótesis de Church 9. MT como Generadores   **Procedimentales:**   1. Diseño deMT para reconocimiento de LLC. 2. Diseño de GLC para generación de LLC. 3. Conversión de representaciones de AP a GLC y viceversa. 4. Reconocimiento de Equivalencias entre diferentes representaciones de un mismo LLC.   **Actitudinales:**   * Rigurosidad en la búsqueda de información y actualización de conocimientos * Responsabilidad en el trabajo en equipo   . | Horas presenciales:  HT: 16  HP: 08  HL: 0  Horas de trabajo autónomo:  HT: 16  HP: 16  HL:0 |
| 1. Maneja el concepto de NP-completitud para determinar las clases de problemas existentes. | * Clases expositivas con demostraciones y ejercicios * Trabajo colaborativo * Trabajo individual * Exposición de resultados monográficos. | * 1. Identifica conceptos de NP-Completitud.   2. Aplica conceptos de NP-Completitud para determinar si un problema presentado es o no NP-Completo.   3. Explica conceptos y problemas asociados a la complejidad de problemas computacionales. | **Conceptuales:**   1. Problemas, algoritmos y complejidad 2. Problema computacional 3. Problemas de decisión 4. Algoritmos de tiempo polinómico y problemas intratables 5. Clases de complejidad 6. Máquinas de Turing deterministas y la clase P 7. Computación no determinista y la clase NP 8. NP-Completitud 9. Problemas NP-completos   **Procedimentales:**   1. Aplicación de conceptos de problemas NP y P en la resolución de problemas.   **Actitudinales:**   * Rigurosidad en la búsqueda y actualización de información * Responsabilidad en el trabajo en equipo * Disposición para comunicar ideas en forma oral y escrita interactuando efectivamente con sus compañeros y profesor | Horas presenciales:  HT: 8  HP: 4  HL: 0  Horas de trabajo autónomo:  HT: 8  HP: 8  HL:0 |

**IV. SISTEMA DE EVALUACIÓN**

|  |  |
| --- | --- |
| **RESULTADOS DE APRENDIZAJE** | **EVIDENCIAS DE APRENDIZAJE**  **(proceso y producto)** |
| 1. Reconoce lenguajes regulares y sus representaciones en forma de autómatas finitos y expresiones regulares para aplicarlos a situaciones prácticas. | * Test 1 * Informe de Tarea grupal Lenguajes Regulares |
| 1. Utiliza representaciones en forma de autómatas de pila y gramáticas libres del contexto para aplicar la teoría de parsing. | * Test 2 * Informe de Tarea grupal AP, GLC. * Certamen 1 * Informe y Software de Proyecto de programación. |
| 1. Analiza los lenguajes decidibles y aceptables para comprender que existen problemas que no se pueden resolver por computador. | * Test 3 * Informe de Tarea grupal Decibilidad |
| 1. Utiliza Máquinas de Turing como modelo de computación para determinar si un problema se puede resolver con los computadores actuales. | * Test 4 * Informe de Tarea grupal M. Turing |
| 1. Maneja el concepto de NP-completitud para determinar las clases de problemas existentes. | * Test5 * Certamen 2 * Monografía y exposición tema. |
| La evaluación de la asignatura considera Promedio de Tests, Certámenes, Tareas, proyecto de desarrollo de Software , Monografía y Exposición Oral de la misma en un total de 100 % distribuído como sigue:  Promedio de Test 20%  Promedio de Certámenes 50%  Promedio de Tareas (obligatorias, promedio ≥ 4.0) 20%  Monografía + Exposición oral 10%  Promedio de Tareas: Promedio Lineal Tareas \*.25 + Trabajo Grupal\*.75  Si ( Promedio de Tareas <4.0) calificación final = NCR. | |

**V. BIBLIOGRAFÍA**

|  |
| --- |
| **Fundamental**  Kelley, D. (1995). *Teoría de autómatas y lenguajes formales*: Prentice-Hall.  John, M.(2004). *Lenguajes formales y teoría de la computación*(3ª ed.): McGraw Hill.  Hopcroft, J., Motwani, R. y Ullman, J. (2002). *Introducción A La Teoría De Autómatas, Lenguajes Y Computación*: Pearson. |
| **Complementaria**  Brookshear, J. (1993). *Teoría de la computación*: Addison Wesley.  Sudkamp, T. (1994). *Languajes and machines (an introduction to the theory of computer science)*: Addison Wesley. |