***SYLLABUS DE FUNDAMENTOS DE PROGRAMACION***

**Fecha de Actualización: 05/03/2017**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **a. DATOS GENERALES DE LA ASIGNATURA** | | | | | | | | | | | | |
| **Nombre de la Asignatura** | | Fundamentos de Programación | | | | | | | | | | |
| **Nro. Créditos** | | **Código SIA** | | | **Horas de trabajo directo con el docente** | | | | **Horas de trabajo autónomo del estudiante** | | | |
| 2 | |  | | | 32 | | | | 64 | | | |
| **b. DATOS GENERALES DEL PROGRAMA O ÁREA QUE LA OFERTA** | | | | | | | | | | | | |
| **Programa académico al que corresponde la asignatura** | | | | | Ingeniería de Sistemas e Ingeniería Electrónica | | | | | | | |
| **Programa o Área que oferta la asignatura** | | | | | Ingeniería de Sistemas | | | | | | | |
| **Correo electrónico del Programa o Área que oferta la asignatura** | | | | | prog\_sistemas@cun.edu.co | | | | | | | |
| **c. PROPÓSITO DE FORMACIÓN Y COMPETENCIAS** | | | | | | | | | | | | |
| **Propósito de formación:** | | | | | Como su nombre lo indica; la asignatura Fundamentos de Programación tiene como propósito principal desarrollar competencias básicas en el estudiante relacionadas con la adquisición de la lógica de programación y el desarrollo de software. De igual manera busca orientar al estudiante hacia la solución de problemas reales empleando el pensamiento lógico y la algoritmia como herramientas principales. | | | | | | | |
| **Problemas (preguntas) que determinan el propósito de formación en la asignatura:** | | | | | * ¿Cómo abordar problemas y plantear soluciones utilizando el pensamiento Lógico? * ¿De qué forma la lógica aporta en la comprensión, planeación y resolución de problemas? * ¿Cuáles son los factores que se deben tener en cuenta al diseñar un algoritmo para la resolución de un problema? * ¿Cómo el uso de la estructura secuencial, iterativa y condicional, en un algoritmo permite representar una idea? * ¿Cuáles son los elementos que se deben tener en cuenta para diseñar una prueba de escritorio? * ¿Cuál es la estructura de un programa básico en un lenguaje de programación estructurado? * ¿De qué forma se debe aplicar un algoritmo a un lenguaje de programación? | | | | | | | |
| **Competencias** | | | | | * Generar habilidades que permitan al estudiante analizar, sintetizar, formular, plantear y dar solución a problemas utilizando pensamiento Lógico, representándolo en algoritmos. * Diseñar pruebas de escritorio que permitan evaluar el algoritmo y medir su productividad. * Implementar un algoritmo de programación que resuelva un problema aplicando el pensamiento lógico para luego ser implementado en cualquier lenguaje de programación. | | | | | | | |
| **d. NIVEL Y PRE-REQUISITOS** | | | | | | | | | | | | |
| **Del Nivel** | | **Asignaturas pre-requisitos** | | | | | | | | | | |
| Técnico profesional | | No aplica | | | | | | | | | | |
| Tecnológico | |  | | | | | | | | | | |
| Profesional | |  | | | | | | | | | | |
| Posgrado | |  | | | | | | | | | | |
| **e. BIBLIOGRAFÍA Y CIBERGRAFÍA** (usar norma APA) | | | | | | | | | | | | |
| **Bibliografía** | | Joyanes Aguilar, L., Castillo Sanz, A., & Sanchez Garcia, L. (2005). *C Algortimos, programación y estructuras de datos.* España: McGraw- Hill.  Joyanes Aguilar, L., Rodriguez Baena, L., & Matilde, F. (2003). *Fundamentos de Programación: Libro de problemas. Algoritmos. Estructura de datos y Objetos (2a. ed).* España: McGraw-Hill. | | | | | | | | | | |
| **Cibergrafía** | | Facultad de Informática. Universidad Nacional de la Plata (2017). Tutorial CODE::BLOCKS. Recuperado de: http://weblidi.info.unlp.edu.ar/catedras/TallerLeng1/Tutorial%20Code-Blocks.pdf | | | | | | | | | | |
| f. PERFIL DOCENTE-TUTOR | | | | | | | | | | | | |
| **Académico** | | | Profesional con título de pregrado y/o postgrado en sistemas, informática, electrónica o área afín | | | | | | | | | |
| **Experiencia** | | | Profesional con dos años de experiencia en desarrollo de aplicaciones informáticas y/o docencia universitaria en asignaturas relacionadas con algoritmos de programación, técnicas de programación, estructura de datos, programación web, ingeniería de software. | | | | | | | | | |
| **Observaciones** | | | Profesional con un alto nivel de compromiso, gusto por el aprendizaje permanente y excelente relación con estudiantes y compañeros | | | | | | | | | |
| **g. PLANEACIÓN DEL PROCESO DE FORMACIÓN** (Plan de Trabajo) | | | | | | | | | | | | |
| **Sesión** | **Propósito de Formación de la Sesión** | | | | | **Acciones a desarrollar** | | | | **Tiempos de trabajo por Créditos** | | |
| **T\*** | **TA\*** | **TC\*** |
| 1. Fundamento de la programación y conceptos básicos | * Comprender la terminología relacionada con los algoritmos. * Representación. Simbología de los algoritmos Diagramas * Identificar los elementos Lógicos de un algoritmo | | | | | * Clase Magistral * Ejercicios Prácticos | | | | 2 | 4 |  |
| 2. Variables y tipos de datos | * Identificar claramente los tipos de datos * Asimilar los elementos para la declaración de una variable * Diferenciar entre variables globales y locales * Conocer los tipos de datos * Representar variables usando diagramas de datos y / o Pseudocódigo | | | | | * Clase Magistral * Ejercicios Prácticos | | | | 2 | 4 |  |
| 3. .Operadores logicos, aritmeticos, relacionales, Mod,div, jerarquía | * Identificar claramente los tipos de operadores * Aplicar los operadores según el requerimiento * Representar los operadores usando diagramas de datos y / o Pseudocódigo. | | | | | * Clase Magistral * Ejercicios Prácticos utilizando Pseudocódigo | | | | 2 | 4 |  |
| 4. Estructura condicional simple, doble y múltiple o anidada | * Conocer las diferentes estructuras algorítmicas condicionales (simple, doble y múltiple) * Aplicar las estructuras condicionales como componentes básicos de los diagramas y aplicar la combinación de ellas para el desarrollo de algoritmos y/O pseudocódigo más complejos. | | | | | * Clase Magistral * Ejercicios Prácticos utilizando herramientas como Pseudocódigo * Aplicaciones a proyecto de aula | | | | 2 | 4 |  |
| 5. Estructura de repetición ciclo condicionado al inicio, mientras-hacer y para, según sea haga | * Diseñar eficazmente ciclos repetitivos para el desarrollo de multiprocesos. | | | | | * Clase Magistral * Ejercicios Prácticos Pseudocódigo * Aplicaciones a proyecto de aula | | | | 2 | 4 |  |
| **6. Primer parcial** | **Primer parcial 30%** | | | | | **Primer parcial 30%** | | | |  |  |  |
| 7.Arreglos Unidimensionales | * Reconocer la diferencia entre el almacenamiento de una variable y un Vector de almacenamiento. * Almacenar datos en vectores a través de las estructuras de repetición mostrar los datos almacenados | | | | | * Clase Magistral * Ejercicios Prácticos Pseudocódigo * Aplicaciones a proyecto de aula | | | | 2 | 4 |  |
| 8. Aplicaciones con Arreglos Unidimensionales | * Reconocer la diferencia entre el almacenamiento de una variable y un Vector de almacenamiento. Almacenar datos en vectores y a través de las estructuras de repetición mostrar los datos almacenados | | | | | * Clase Magistral * Ejercicios Prácticos utilizando Pseudocódigo * Aplicaciones a proyecto de aula | | | | 2 | 4 |  |
| 9. Formulación de algoritmos en DFD. Diagramas lógicos, prueba de escritorio y seudocódigo | * Diseñar algoritmos basados en las características del problema utilizando la herramienta DFD, Seudocódigo. * Seguimiento de algoritmo por medio de prueba de escritorio | | | | | * Clase Magistral * Ejercicios Prácticos | | | | 2 | 4 |  |
| 10. Aplicaciones de refuerzo usando sentencias de control y arreglos | * Diferenciar claramente el uso y los beneficios de una estructura secuencial de asignación, usando la herramienta DFD, Seudocódigo, Prueba de escritorio. | | | | | * Clase Magistral * Ejercicios Prácticos | | | | 2 | 4 |  |
| **11. Segundo parcial** | **Segundo parcial 30%** | | | | | **Segundo parcial 30%** | | | |  |  |  |
| 12. lenguajes de programación | * Conocer las generalidades de los lenguajes de programación. * Diferenciar los tipos de Lenguajes de Programación * Reconocer el ambiente de desarrollo CODEBLOCK | | | | | * Clase Magistral | | | | 2 | 4 |  |
| 13. Manejo de condicionales en Code::Blocks | * Aplicar las estructuras condicionales utilizando el ambiente de desarrollo CODEBLOCK | | | | | * Clase Magistral * Ejercicios Prácticos | | | | 2 | 4 |  |
| 14. Manejo de sentencias repetitivas en Code::Blocks | * Aplicar las estructuras repetitivas utilizando el ambiente de desarrollo CODEBLOCK | | | | | * Clase Magistral * Ejercicios Prácticos | | | | 2 | 4 |  |
| 15. Manejo de arreglos en Code::Blocks | * Aplicar arreglos utilizando el ambiente de desarrollo CODEBLOCK | | | | | * Clase Magistral * Ejercicios Prácticos | | | | 2 | 4 |  |
| **16. Examen Final 40%** | **Examen Final 40%** | | | | | **Examen Final 40%** | | | |  |  |  |
| \* T: Tutoría, TA: trabajo autónomo, TC: trabajo colaborativo | | | | | | | | | | | | |
| **h. SISTEMA DE EVALUACIÓN** | | | | | | | | | | | | |
| **Criterios** | | | | | | | **Descripción** | | | | | |
| Evaluación diagnostica: | | | | | | | Para establecer el nivel de conocimientos que el estudiante tiene acerca del tema | | | | | |
| Evaluación formativa: | | | | | | | Le permite al docente y al estudiante detectar las fortalezas y debilidades | | | | | |
| Evaluación sumativa: | | | | | | | de acuerdo con la exigencia de la institución para cualificar el nivel de competencias y está compuesta por tres cortes, primer corte 30%,segundo corte 30% y tercer corte 40% y la escala de las mismas es de 1 a 5 | | | | | |
|  | | | | | | | Lo anterior debe estar directamente relacionado con la metodología, los acuerdos pedagógicos logrados al inicio del curso y lo consagrado en el reglamento estudiantil | | | | | |
|  | | | | | | |  | | | | | |
|  | | | | | | |  | | | | | |
| **i. DISTRIBUCIÓN DE NOTAS** (Calificación para los programas presenciales) | | | | | | | | | | | | |
| **Prueba parcial 1** | | | | **Prueba parcial 2** | | | | **Prueba final** | | | | |
| Evaluación principal: 15%  1º Trabajo: 5%  2º Trabajo 5%  3ºTrabajo 5% | | | | Evaluación principal: 15%  1º Trabajo: 5%  2º Trabajo 5%  3ºTrabajo 5% | | | | Evaluación principal: 20%  1º Trabajo: 6.66%  2º Trabajo 6.66%  3ºTrabajo 6.66% | | | | |
| **Total 30%** | | | | **Total 30%** | | | | **Total 40%** | | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **CONTROL DE APROBACIÓN** | | | |
| **Realizado por** | **Validado por** | **Aprobado por** | **Fecha de Aprobación** |
| Ing. Magda Fernández | Ing. Yesid Luengas | Ing. Sandra Castiblanco | 02/01/2014 |

|  |  |  |
| --- | --- | --- |
| **CONTROL DE ACTUALIZACIÓN DE CONTENIDO** | | |
| **Fecha de Actualización** | **Descripción del Cambio** | **Aprobado Por** |
| 05/06/2017 | Actualización de formato |  |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| **CONTROL DE CAMBIOS**  **(Espacio exclusivo para el Equipo de Calidad)** | | |
| **FECHA** | **VERSIÓN** | **DESCRIPCIÓN DEL CAMBIO** |
|  |  |  |
|  |  |  |