|  |  |
| --- | --- |
| **Nombre Corto de la Asignatura** | **Introducción a la programación** |
| **Nombre Largo de la Asignatura** | **Introducción a la programación** |
| **Código de la asignatura** | **33698** |
| **Grado** | Pregrado |
| **Descripción** | La asignatura es una introducción a las estrategias para solucionar problemas utilizando el computador y a los conceptos de Programación, para estudiantes sin experiencia previa. Se desarrollan las habilidades para pensar algorítmicamente y aplicar métodos básicos de la programación para resolver problemas. El curso utiliza clases magistrales interactivas, talleres y trabajo en equipo para desarrollar un proyecto. |
| **Número de Créditos** | 3 |
| **Condiciones Académicas de Inscripción (Pre-requisitos)** | Sin condiciones |
| **Período Académico de Vigencia** | 2430 |

|  |
| --- |
| **Objetivos de Formación** |
| * Presentar estrategias para identificar información útil para caracterizar un problema y enfrentar la solución del mismo empleando el computador como herramienta. * Brindar herramientas para la definición y desarrollo de algoritmos, así como para la representación de los mismos en un lenguaje de programación para implementar las soluciones |

|  |
| --- |
| **Resultados de Aprendizaje Esperados (RAE)** |
| Al finalizar el curso el estudiante estará en capacidad de:   * Identificar información que permite caracterizar un problema informático de complejidad baja,  mediante el uso de los principios básicos de Lógica e Ingeniería de Sistemas (Disciplinar 1) (CDIO 2.1) * Conocer las construcciones básicas de programación para diseñar soluciones a problemas informáticos de complejidad baja, mediante algoritmos, utilizando el pensamiento computacional. (Disciplinar 1,2) (CDIO 4.4) * Conocer los procedimientos para diseñar la organización de los datos requeridos para solucionar el problema utilizando colecciones de datos y archivos (Disciplinar 1,3,4) (CDIO 4.4) * Determinar las herramientas necesarias para implementar, en un lenguaje de programación, los algoritmos diseñados, permitiendo su modificación  (Disciplinar 1,2,3,4) (CDIO 4.5) |

|  |
| --- |
| **Contenidos temáticos** |
| 1. Solución de problemas  - Ciclo de vida  - Pensamiento Computacional  - Abstracción  - Algoritmos  2. Construcciones básicas de programación  - Elementos básicos de un programa  Variables, tipos de datos, instrucciones, expresiones, asignación, E/S  - Estilo de programación  Convenciones, comentarios, escritura de código  - Manejo de lenguaje de programación  sintaxis, errores, ¿debugging¿  - Flujo de Control  Condicional, Iteración  - Funciones  Parámetros, retorno, llamada por valor  3. Manejo de memoria con colecciones de datos  - Memoria estática  - Arreglos  - Otras colecciones de datos  4. Manejo de datos en archivos  - Secuenciales  - Aleatorios |

|  |
| --- |
| **Estrategias Pedagógicas** |
| Durante el curso se utilizarán 4 estrategias. La primera de ellas será el aprendizaje directivo mediado mediante clases magistrales. La segunda el aprendizaje basado en problemas mediante talleres, iniciando con talleres individuales, y culminando con talleres en grupo, donde se expone a los estudiantes a problemas de complejidad sencilla, pero de refuerzo de los conceptos. La tercera será el aprendizaje por proyectos, donde se propone la elaboración de una solución que responda a un problema acotado. Por último, se utilizará el aprendizaje colaborativo, en el cual, a través de grupos de trabajo se logrará un intercambio de conocimientos por pares para reforzar el aprendizaje significativo. |

|  |
| --- |
| **Evaluación** |
| Las estrategias de evaluación están centradas en la valoración de los resultados de aprendizaje esperado de la asignatura; las cuales pueden ser formativas que suscitan la comprensión y construcción de conocimiento, y sumativas que incluyen porcentajes de evaluación con el fin de corroborar el logro de los aprendizajes y el desarrollo de las competencias en los estudiantes. Las estrategias de evaluación de la asignatura son:  Primer Parcial  20%  Segundo Parcial 20%  Tercer Parcial  20%  Proyecto 25%      Primera Entrega 7%      Entrega Final - Programa   8%      Sustentación  - 10%  Quices y Talleres 15% |

|  |
| --- |
| **Recursos Bibliográficos** |
| Solución de problemas   * Beecher Karl, Computational Thinking, BCS, 2018 * Curzon Paul, The Power of Computational Thinking, World Scientific,2017 * Peña, Monica, El Problema * Polya, George, Como Plantear y Resolver Problemas, Editorial Trillas * Wod, Larry E, Estrategias de Pensamiento, Ejercicios de Agilidad Mental, Editorial Labor * Rueda Serrano, Jairo, Matemática divertida, Sistemas & Computadores * Bolt, Brian, Actividades matemáticas, Editorial Labor * Bolt, Brian, Divertimentos matemáticos, Editorial Labor * Bolt, Brian,¿ Más actividades matemáticas, Editorial Labor * Bolt, Brian, Aún más actividades matemáticas, Editorial Labor * Mataix Lorda, Mariano, Fácil, menos fácil y difícil : 100 problemas de lógica y matemáticas, Editorial Marcombo * Castro Ch, Iván, El arte de razonar, Pontificia Universidad Javeriana 2003   Algoritmos   * Cairó Osvaldo, Metodología de la programación, Segunda Edición,AlfaOmega , 2003 * Joyanes Aguilar, Luis Fundamentos de Programación, Ed. McGraw Hill * Joyanes Luis, Problemas de Metodología de la Programación, Mc GrawHill * Becerra, César Algoritmos: Conceptos Básicos , 1995   Solución de problemas utilizando un Lenguaje de Programación   * Deitel H.M., Como Programar en C/C++, Segunda Edición. Prentice Hall. 1995 * Joyanes, Luis, Programación en C++, McGraw Hill. 2000 * Savitch Walter, Resolución de Problemas con C++, Ed. Pearson * Hetland Magnus Lie, Beginning Python, SpringerLink , 2005 * Lutz Mark, Learning Python, OReilly Media, 2013 * Python Software Foundation, [en línea] https://www.python.org/ * Joyanes Luis, Fundamentos de Programación, McGraw Hill. 2008 * Mata Toledo, Ramón Introducción a la programación, McGraw Hill,2001 |