|  |  |
| --- | --- |
| **Nombre Corto de la Asignatura** | **Estructuras de Datos** |
| **Nombre Largo de la Asignatura** | **Estructuras de Datos** |
| **Código de la asignatura** | **4196** |
| **Grado** | Pregrado |
| **Descripción** | En la mayoría de los ambientes reales de desarrollo de software se procesan grandes cantidades de información, que está representada en datos que tienen sentido, secuencia y magnitud. Entonces, los correctos diseños e implementaciones de estructuras de datos y algoritmos son vitales para el funcionamiento adecuado de los sistemas de cómputo y almacenamiento en la vida real.  Este curso se centra en el estudio de las estructuras contenedoras de datos básicas y los algoritmos relacionados con ellas, sin perder de vista su uso en proyectos de desarrollo de software de mediana complejidad que resuelvan problemas de cómputo reales.  El curso utiliza clases magistrales interactivas, talleres guiados y trabajo en equipo para desarrollar un proyecto |
| **Número de Créditos** | 3 |
| **Condiciones Académicas de Inscripción (Pre-requisitos)** | Requisitos de inscripción: Programación avanzada /y/ (Lógica y Matemáticas Discretas /o/  Matemáticas Discretas Sistemas /o/ Fundamentos de matemáticas II /o/ Matemáticas II /o/  001297 Cálculo Integral) |
| **Período Académico de Vigencia** | 2430 |

|  |
| --- |
| **Objetivos de Formación** |
| 1. Presentar los principios básicos de la complejidad algorítmica como criterio de calidad de algoritmos.  2. Brindar al estudiante las herramientas básicas para el diseño de tipos abstractos de datos, particularmente en el diseño de estructuras contenedoras.  3. Mostrar estrategias de implementación de estructuras de datos y algoritmos asociados a ellas en un lenguaje de programación (C++).  4. Exponer al estudiante a ambientes de desarrollo de software dónde se cubran las primeras etapas de dicha actividad: concepción, diseño, implementación y pruebas. |

|  |
| --- |
| **Resultados de Aprendizaje Esperados (RAE)** |
| Al finalizar el curso el estudiante estará en capacidad de:   * Explicar la complejidad de los algoritmos usando conceptos del cálculo asintótico y el impacto en el uso de recursos (CDIO 2.1). * Conocer estrategias para el diseño de algoritmos usando diferentes estructuras de datos (CDIO 4.4). * Aplicar estrategias para la implementación de algoritmos usando diferentes estructuras de datos (CDIO 4.5) * Conocer los procedimientos necesarios para desarrollar un plan de pruebas unitarias para un algoritmo (CDIO 4.5). |

|  |
| --- |
| **Contenidos temáticos** |
| 1. Bases de complejidad algorítmica. 2. Estructuras lineales. 3. Estructuras recurrentes. 4. Estructuras no lineales. |

|  |
| --- |
| **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 individuales, 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:  Parciales:  Parcial 1: estructuras lineales (15%).  Parcial 2: estructuras jerárquicas (15%).  Parcial 3: estructuras no lineales (15%).  Talleres:  Taller 1: conceptos básicos de análisis asintótico (8%).  Taller 2: estructuras lineales (4%).  Taller 3: árboles de búsqueda (4%).  Taller 4: grafos (4%).  Proyecto de desarrollo a lo largo del semestre:  Proyecto  diseño (15%)  Proyecto  implementación (10%)  Proyecto  plan de pruebas (10%) |

|  |
| --- |
| **Recursos Bibliográficos** |
| * Nyhoff, L.R. TADs, Estructuras de datos y resolución de problemas con C++. Pearson Education, Inc., 2 edition, 2006. * Weiss, M.A. Data Structures and Algorithm analysis in C++. Addison-Wesley, 4th edition, 2014. * Joyanes Aguilar, L. y Zahonero Martínez, I. Algoritmos y Estructuras de Datos: Una perspectiva en C. McGraw Hill, 2004. * Villalobos, J.A. Diseño y manejo de estructuras de datos en C. McGraw Hill, 1996. * Brassard, G., Bratley, P. Fundamentos de Algoritmia. Prentice Hall, 1997. * Ford, W. Data structures with C ++ using STL. Prentice Hall, 2002. * Bergin, J. Data Structure programming with the standard template library in C+ +.Springer, 1998. * Rumbaugh, J. Modelado y diseño orientado a objetos, metodología OMT. Prentice Hall,1996. * Vandevoorde, D. C++ Templates the Complete guide. Addison-Wesley, 2003. * Frienman, F.L. Problem solving, Abstraction, and design using C++. Addison-Wesley, 1997. |