|  |  |
| --- | --- |
| **Nombre Corto de la Asignatura** | **Análisis y diseño de software** |
| **Nombre Largo de la Asignatura** | **Análisis y diseño de software** |
| **Código de la asignatura** | **34805** |
| **Grado** | Pregrado |
| **Descripción** | La asignatura aborda los temas iniciales de análisis y diseño como son captura de requisitos, análisis y diseño de clases en niveles básicos; abarca también todos los conceptos de programación orientada a objetos haciendo una integración con el componente de análisis y diseño enmarcados en un proceso de desarrollo de software. El objetivo es que el estudiante desarrolle habilidades de diseño y codificación de aplicaciones orientadas usando el paradigma de orientación a objetos. La metodología de la asignatura está centrada en clases magistrales y desarrollo de proyectos para aplicar los conceptos. |
| **Número de Créditos** | 3 |
| **Condiciones Académicas de Inscripción (Pre-requisitos)** | Requisitos de inscripción: Programación avanzada |
| **Período Académico de Vigencia** | 2430 |

|  |
| --- |
| **Objetivos de Formación** |
| 1. Establecer un mecanismo de recolección de necesidades para crear modelos de requerimientos que articulan la posterior creación de los modelos de análisis y diseño.  2. Presentar los conceptos y las herramientas para crear modelos de análisis y diseño usando el paradigma orientado a objetos.  3. Presentar los conceptos y las herramientas para codificación de programas usando el paradigma orientado a objetos.  4. Ilustrar la utilización de las herramientas para garantizar la calidad de un producto de software.  5. Ilustrar una estrategia de integración de los modelos de análisis y diseño con los modelos de programación para realizar la trazabilidad entre los diferentes modelos abordados en clase. |

|  |
| --- |
| **Resultados de Aprendizaje Esperados (RAE)** |
| Al finalizar el curso el estudiante estará en capacidad de:  1. Identificar las diferentes aproximaciones para diseñar e implementar un sistema de baja y media complejidad. (Disciplinar 1) (CDIO 4.4)  2. Conocer los conceptos fundamentales de análisis y diseño del paradigma orientado a objetos. (Disciplinar 2) (CDIO 4.4)  3. Conocer la forma de implementar el diseño de un modelo orientado a objetos a través de las construcciones básicas de programación orientada a objetos. (Disciplinar 3) (CDIO 4.3)  4. Conocer las notaciones y herramientas que soporten el proceso de diseño de una solución de software. (Disciplinar 4) (CDIO 4.4)  5. Describir los modelos de las distintas etapas del desarrollo de software usando lenguajes y diagramas que sean estándares de la industria. (Disciplinar 5) (CDIO 2.1)  6. Conocer las estrategias de identificación de roles y responsabilidades de los miembros de un equipo de un Proyecto de software (Transversal) (CDIO 3.1) |

|  |
| --- |
| **Contenidos temáticos** |
| 1. Captura de requisitos  2. Análisis Orientado a Objetos  3. Diseño Orientado a Objetos  4. Implementación de modelos orientados a Objetos |

|  |
| --- |
| **Estrategias Pedagógicas** |
| Las estrategias pedagógicas que se utilizarán en el curso se fundamentan en:  1. Exposiciones en clase para presentar los contenidos  2. Talleres dirigidos para mecanizar los conceptos de clase  3. Componente práctico: Los contenidos del curso se irán trabajando grupalmente fuera del salón de clase en equipos de estudiantes que resolverán en entregas (correspondientes al avance temático del curso) un proyecto cuyo enunciado es dado por el profesor.  Para la construcción de modelos y la generación de código, los estudiantes cuentan con los laboratorios de la Carrera que podrán usar fuera de las horas presenciales.  4. Se realizará un taller semanal de los temas vistos en clase, donde, con la asistencia del profesor, se estudiará o resolverá un caso en equipos y para desarrollar en la sesión de clase. |

|  |
| --- |
| **Evaluación** |
| Las estrategias de evaluación se dividen en dos categorías: formativas y sumativas. Las evaluaciones formativas - que permiten construir relacionamiento - se llevarán a cabo en las como retroalimentación en los talleres de clase.  Las evaluaciones sumativas ¿ que permiten corroborar el logro de los aprendizajes y competencias en los estudiantes- se distribuyen de la siguiente forma:  Parciales (50%)  1. Primer Parcial (15%)  2. Segundo Parcial (20%)  3. Tercer Parcial (15%)  Trabajo en Clase (20%)  1. Talleres en Clase  2. Quices y tareas  Proyecto (30%)  1. Primera entrega (Requisitos y Análisis):  8%  2. Segunda entrega (Análisis y Diseño): 11%  3. Tercera entrega (Diseño e Implementación -codificación-): 11% |

|  |
| --- |
| **Recursos Bibliográficos** |
| Análisis y Diseño de Software:  1. Bernd bruegge, allen h. Dutoit. ¿object oriented software engineering using UML, pattterns, and java¿, tercera edición, prentice hall, 2010.  2. Bennet, s, mcrobb, s y farmer, r. ¿análisis y diseño orientado a objetos de sistemas¿, mc cgra-hill, tercera edición, 2007.  3. Booch, g., rumbaugh, j. Y jacobson, i. ¿el lenguaje unificado de modelado uml¿, pearson educación, segunda edición 2006.  4. Fowler m. ¿uml distilled: a brief guide to the standard object modeling language¿, third edition sep.2003  5. Bruegge, b. Y dutoit, a. ¿ingeniería de software orientado a objetos¿. Prentice hall, 2002.  6. Hooks, ivy, ¿writing good requirements¿. Proceedings of the third ncose symposium, vol. 2, 1993.  7. Wiegers, karl e., ¿writing quality requirements¿, process impact, www.processimpact.com  8. Cockburn, ¿A Basic use case template¿. Sep. 2006  9. Eckel b. ¿thinking in java, thinking in patterns¿, sep.2006  10 Material de apoyo en uml en internet:  10.1 IBM\_Clases  http://www.ibm.com/developerworks/rational/library/content/RationalEdge/nov03/t\_modelinguml\_db.pdf  10.2 The Object Primer, Copyright 2003-2006 Scott W. Ambler http://www.agilemodeling.com/artifacts/classDiagram.htm#Associations  10.3 UML 2.0 http://www.uml.org/index.htm#UML2.0  Programación Orientada a Objetos  1. Java how to program early objects 11th edition. Deitel, paul j. 2018.  2. Bernd bruegge, allen h.dutoit. ¿object oriented software engineering  using uml, pattterns, and java¿, tercera edición, prentice hall, 2010.  3. Programación en java introducción a la programación orientada a objetos autor wu, c. Thomas. Mcgrawhill. 2008.  4. An introduction to object-oriented programming with java, wu, c. Thomas. 2006.  5. Eckel, bruce. Piensa en java, cuarta edición, prentice hall, 2007  6. Java for programmers autor    deitel, paul j.  7. Introduction to java programming comprehensive version. Liang, y. Daniel. 2007.  8. Object oriented programming through java autor radha krishna, p.  9. Java programación orientada a objetos 1a ed. Autor Juárez, Andrés  10. Orientación A Objetos Con Java y UML 2a ed. Autor Fontela, Carlos Moisés  11. Programación en java 6 algoritmos, programación orientada a objetos e interfaz gráfica de usuario Autor: Joyanes Aguilar, Luis  12. Java in two semesters. Charatan, Quentin. 2006.  13. Java generics and collections. Naftalin, Maurice. 2007.  14. Java the complete reference, J2SE 5 edition. Schildt, Herbert.2005.  15. Java an introduction to problem solving & programming.2005.  16. Savitch, Walter. Resolución de Problemas con C++, Segunda Edición. Prentice Hall, 2000  17. Larman, Craig, UML y Patrones, Segunda Edición, Pearson Educación, 2002.  18. Booch, G., Rumbaugh, J. y Jacobson, I. El Lenguaje Unificado de Modelado UML,  Pearson Educación, Segunda Edición 2006.  19. Joyanes Luis, Programación Orientada a Objetos, McGraw Hill, 1998  20. Barker Jacquie, Beginning Java Objects: From concepts to Code, Wrox, 2000. ISBN 1861004176.  21. Terrence W. Pratt, Marvin V. Zelkowitz, "Lenguajes de Programación. Diseño e Implementación". Prentice Hall. 3ra. Ed.1996  Libros Electrónicos:  1. Java Programming Fundamentals Problem Solving Through Object Oriented Analysis and Design.  2. Java programming fundamentals [Recurso electrónico] problem solving through object oriented analysis and design Autor    Nair, Premchand S., 1956- # clasif.    005.117 N158 Editor Edición Fecha de pub     2009  3. ECOOP 2005 - Object-Oriented Programming [Recurso electrónico] / 19th European Conference, Glasgow, UK, July 25-29, 2005. Proceedings  5. Java for Artists : The Art, Philosophy, and Science of Object-oriented Programming.Miller, Rick; Kasparian, Raffi. Falls Church, Va : Pulp Free Press. 2006. eBook.,  6. Wicked Cool Java : Code Bits, Open-source Libraries, and Project Ideas.  Sitios Web:  1. https://courses.edx.org/courses: edX Home Page UBCx: SoftConst2x Software Construction: Object Oriented Design.  2. https://courses.edx.org/courses: HKUSTx: COMP102.1x Introduction to Java Programming  3. http://highered.mheducation.com/sites/0073523305/student\_view0/index.html  4. https://www.guru99.com/java-oops-concept.html  5. Animaciones en línea de programas en java libro Daniel Lang http://www.cs.armstrong.edu/liang/animation/web/java10e/AnimationForInteractiveEbook.html  6. Ver Ejemplos de Código en línea de programas en java libro Daniel Lang https://media.pearsoncmg.com/ph/esm/ecs\_liang\_introjavaprog\_1\_ap/cw/content/source-code.php#code\_examples. |