|  |  |
| --- | --- |
| **Nombre Corto de la Asignatura** | **Arquitectura de Software** |
| **Nombre Largo de la Asignatura** | **Arquitectura de Software** |
| **Código de la asignatura** | **4185** |
| **Grado** | Pregrado |
| **Descripción** | Este curso propicia un ambiente de aprendizaje en el cual se aplican metodologías, herramientas y estrategias enfocadas en apoyar el diseño e implementación de arquitecturas de sistemas grandes cumpliendo con los atributos de calidad. Lo cual tiene como objetivo que el estudiante adquiera las competencias necesarias para analizar, diseñar e implementar arquitecturas de software que permitan incrementar la productividad y competitividad de las organizaciones. La asignatura cubre temáticas como atributos de calidad, estilos y patrones arquitectónicos, procesos de diseño, arquitecturas de referencia y documentación, entre otros. La asignatura tiene tres componentes metodológicos para el desarrollo de los temas: aprendizaje directivo, auto-aprendizaje y un componente práctico. Adicionalmente, los estudiantes tendrán la oportunidad de realizar talleres en clase con el fin de resolver dudas y afianzar los conceptos. |
| **Número de Créditos** | 3 |
| **Condiciones Académicas de Inscripción (Pre-requisitos)** | Requisitos de inscripción: Desarrollo web /y/ Computación móvil /y/ Introducción distribuidos /o/ (Para Ing. Sistemas Plan Anterior: Ingeniería de Software /o/ Fundamentos Ingeniería de SW) |
| **Período Académico de Vigencia** | 2430 |

|  |
| --- |
| **Objetivos de Formación** |
| 1. Presentar los conceptos fundamentales de arquitectura de software para contextualizar las problemáticas que aborda, y su relación con otras temáticas de la ingeniería.  2. Presentar los modelos de calidad de software que se aplican a la arquitectura de software para que los estudiantes seleccionen un modelo de referencia.  3. Presentar los estilos y patrones arquitectónicos relacionándolos con los atributos de calidad que ofrecen, para identificar diseños de referencia y sus atributos de calidad.  4. Presentar los métodos de diseño arquitectural enfatizando en uno de ellos, para que sea usado en los procesos de construcción de software.  5. Presentar marcos de trabajo, herramientas y técnicas de implementación de arquitecturas de referencia para desarrollar un proyecto de aplicación |

|  |
| --- |
| **Resultados de Aprendizaje Esperados (RAE)** |
| Al finalizar el curso la/el estudiante estará en capacidad de:  1. Modelar una arquitectura de software usando atributos de calidad, estructuras, estilos, patrones y vistas de representación de un problema real. (Disciplinar 1, 2, 3) (CDIO 2.1)  2. Aplicar un método de diseño arquitectural en la construcción, documentación y validación de una arquitectura de software de un problema real (Disciplinar 3, 4, 5) (CDIO 4.3 y 4.4)  3. Emplear los elementos y consideraciones de las tecnologías de desarrollo en el diseño e implementación de arquitecturas de software. (Disciplinar 5) (CDIO 4.5)  4. Utilizar los procedimientos necesarios para la integración del sistema desarrollado con otros sistemas, protocolos, aplicaciones y plataformas. (Disciplinar 3 - 5) (CDIO 4.4) |

|  |
| --- |
| **Contenidos temáticos** |
| 1. Conceptos fundamentales de AS  2. Atributos de calidad y estilos arquitecturales  3. Proceso de diseño de una arquitectura de software  4. Validación de arquitecturas de software  5. Implementación de AS |

|  |
| --- |
| **Estrategias Pedagógicas** |
| Durante el curso se utilizarán 3 estrategias. La primera será el aprendizaje directivo mediado por clases magistrales interactivas. La segunda será el aprendizaje basado en problemas mediante talleres individuales o grupales, donde se expone a los estudiantes a problemas de diferente nivel de complejidad, que refuerzan los conceptos y son útiles para la elaboración de los proyectos del curso. La tercera será el aprendizaje por proyectos, donde los contenidos del curso se irán trabajando grupalmente en equipos de estudiantes que resolverán en entregas 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. |

|  |
| --- |
| **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, las cuales 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:  Evaluación de Conocimientos Teóricos en Arquitecturas de Software:  1. Parcial  No.1                         20%  2. Parcial  No.2                         20%  Evaluación de Conocimientos Prácticos en Arquitecturas de Software:  1. Proyecto Semestre:  1.1 Entrega No.1: SAD arquitectura lógica sin tecnología                                 15%  1.2 Entrega No.2: (SAD + JEE + .NET) 10% + Implementación (15%)             25%  Evaluación de Preparación, seguimiento de la clase y responsabilidad:  1. Tareas, Quices, y Laboratorios           20% |

|  |
| --- |
| **Recursos Bibliográficos** |
| Textos Básicos:  1. Software Archit0ecture in Practice, Third Edition.2013. Len Bass, Paul Clements, Rick Kazman  2. Essential Software Architecture. Ian Gorton. Springer.2006  3. The Art of Software Architecture. Stephen Albin.Wiley. 2003.  4. Documenting software architectures views and beyond 2nd ed., Clements, Paul. Addison-Wesley. 2011  5. The process of Software Architecting. Peter Eeles, Peter Cripps. Addison Wesley.2010  6. Software Systems Architecture: Working with Stakeholders Using Viewpoints and Perspectives. Nick Rozanski y Eoin Woods  7. Microsoft Application Architecture Guide. 2nd Edition. David Hill. Patterns and practices. Disponible en: http://www.microsoft.com/en-us/download/details.aspx?id=16236  8. Beginning Java EE 7 Platform with GlassFish 3. From Novice to Professional Second Edition. Antonio Goncalves, |