



No.	Descripción															
.	Código 785	Créditos 4														
1	Escuela Ciencias y Sistemas	Área a la que pertenece Software	Vigencia Segundo semestre 2019													
2	Horas por semana 4	Horario Martes de 7:10 a 8:50 y jueves 7:10 a 8:50														
3	Pre-requisitos:283															
4	Postrequisitos:															
5	Sección: A-															
6	<p>I. Descripción General</p> <p>En el curso pre-requisito de análisis y diseño de sistemas 2, el objetivo fue introducir al estudiante en la ingeniería de software, que describiera métodos y metodologías para desarrollar software y la aplicación de formas de identificación de los requerimientos del cliente, primordialmente los funcionales, en este curso el enfoque va dirigido a la disciplina de diseño de software, es decir la exploración de técnicas para definir cómo lograr alcanzar los requerimientos de los clientes, dando siempre un enfoque de ingeniería de software.</p> <p>Al finalizar el curso, se espera que el estudiante tenga una visión integral de los elementos de diseño que se deben considerar para el desarrollo de un sistema de software. Los conceptos de patrones de diseño, arquitectura de software y patrones de arquitectura permiten tener una visión general de las estructuras propuestas para conformar un sistema. El alcanzar los diferentes atributos de calidad de un sistema se logrará a través de la aplicación de puntos de vista, vistas y perspectivas.</p> <p>II. Objetivos</p> <p>Al finalizar el curso de análisis y diseño de sistemas 2, el estudiante:</p> <ul style="list-style-type: none"> • Explica y describe lo que es el diseño de software: los fundamentos, principios, el proceso y distingue entre diseño de alto nivel y diseño de bajo nivel. • Identifica y describe los conceptos básicos de arquitectura de software para el desarrollo de sistemas de software. • Identifica y describe al menos 6 atributos de calidad del software y tácticas aplicables para alcanzarlos. • Identifica y describe al menos 6 patrones de arquitectura. • Aplica al menos 3 patrones de diseño de alto o bajo nivel utilizando tecnologías emergentes en un proyecto de clase. • Utiliza la metodología Scrum para el desarrollo de un proyecto de clase e implementa prácticas de Integración Continua y Entrega Continua apoyándose de herramientas tecnológicas de actualidad. <p>III. Contenido</p> <table border="1"> <thead> <tr> <th>Tópico</th> <th>Descripción</th> <th>Recursos</th> </tr> </thead> <tbody> <tr> <td rowspan="2">Fundamentos de diseño de software</td> <td>Conceptos generales de diseño de software Contexto del diseño de software El proceso de diseño de software</td> <td>Libro: Software Design, David Budgen Capítulos 1, 2 y 3</td> </tr> <tr> <td>Principios de diseño de software</td> <td>Libro: Software Engineering 9th Edition, Ian Sommerville Capítulos 6, 7 y 21</td> </tr> <tr> <td rowspan="2">Aspectos clave en el diseño de software</td> <td>Concurrencia Event Handling and Control Persistencia de datos</td> <td>Libro: Software Engineering 9th Edition, Ian Sommerville Capítulos 9, 18 y 21</td> </tr> <tr> <td>Distribución de componentes Error and Exception Handling Tolerancia a fallos Seguridad</td> <td>Libro: Software Engineering 9th Edition, Ian Sommerville Capítulos 12, 16, 18</td> </tr> </tbody> </table>			Tópico	Descripción	Recursos	Fundamentos de diseño de software	Conceptos generales de diseño de software Contexto del diseño de software El proceso de diseño de software	Libro: Software Design, David Budgen Capítulos 1, 2 y 3	Principios de diseño de software	Libro: Software Engineering 9th Edition, Ian Sommerville Capítulos 6, 7 y 21	Aspectos clave en el diseño de software	Concurrencia Event Handling and Control Persistencia de datos	Libro: Software Engineering 9th Edition, Ian Sommerville Capítulos 9, 18 y 21	Distribución de componentes Error and Exception Handling Tolerancia a fallos Seguridad	Libro: Software Engineering 9th Edition, Ian Sommerville Capítulos 12, 16, 18
Tópico	Descripción	Recursos														
Fundamentos de diseño de software	Conceptos generales de diseño de software Contexto del diseño de software El proceso de diseño de software	Libro: Software Design, David Budgen Capítulos 1, 2 y 3														
	Principios de diseño de software	Libro: Software Engineering 9th Edition, Ian Sommerville Capítulos 6, 7 y 21														
Aspectos clave en el diseño de software	Concurrencia Event Handling and Control Persistencia de datos	Libro: Software Engineering 9th Edition, Ian Sommerville Capítulos 9, 18 y 21														
	Distribución de componentes Error and Exception Handling Tolerancia a fallos Seguridad	Libro: Software Engineering 9th Edition, Ian Sommerville Capítulos 12, 16, 18														

Introducción a la arquitectura de software	Definición Importancia de arquitectura de software Conceptos de arquitectura	Libro: Software systems Architecture, Working with Stakeholders Using viewpoints and Perspectives, Nick Rozanski, Eoin Woods Capítulos 1 y 2
	Puntos de Vista y Vistas Stakeholders	Libro: Software systems Architecture, Working with Stakeholders Using viewpoints and Perspectives, Nick Rozanski, Eoin Woods Capítulo3: 3
	Patrones (Estilos) de Arquitectura PoEAA (Martin Fowler)	Libro: Documenting Software Architecture, Clemments et al Capítulos: 1, 2, 3, 4, 5
Diseño de bajo nivel	Principios de diseño (SOLID)	http://butunclebob.com/ArticleS.UncleBob.PrinciplesOfOod
	Patrones de diseño (GoF)	Libro: Head First Design Patterns, Eric Freeman & Elisabeth Freeman
Diseño de software, análisis de calidad y evaluación	Atributos de Calidad (puntos de vista, vistas, perspectivas)	Libro: Software Design, David Budgen Capítulo 4
	Análisis de calidad y técnicas de evaluación	Libro: Software systems Architecture, Working with Stakeholders Using viewpoints and Perspectives, Nick Rozanski, Eoin Woods Part 3: The Viewpoint Catalog Part 4: The Perspective Catalog
	Descripciones Estructurales (Vistas Estáticas) Descripciones de Comportamiento (Vistas Dinámicas)	
	Proceso para definir una arquitectura	Libro: Essential software architecture, Ian Gorton
Arquitectura de Software	Descripciones Estructurales (Vistas Estáticas) Descripciones de Comportamiento (Vistas Dinámicas)	Libro: Software systems Architecture, Working with Stakeholders Using viewpoints and Perspectives, Nick Rozanski, Eoin Woods Part 3: The Viewpoint Catalog Part 4: The Perspective Catalog
Arquitectura de Software Patrones/Estilos	Documentando la arquitectura de software	Libro: Documenting Software Architecture, Clemments et al
	Arquitectura de software en ambientes ágiles	http://martinfowler.com/articles/microservices.html

	Toward Agile Architecture: Insights from 15 Years of ATAM Data, http://www.infoq.com/articles/atam-quality-attributes	Measure and improve code quality, http://www.infoq.com/news/2016/01/measure-improve-code-quality?utm_source=infoq&utm_medium=popular_widget&utm_campaign=popular_content_list&utm_content=homepage
Estrategias y métodos de diseño de Software	Estrategias generales Diseño orientado a funcionalidad (Estructurado) Diseño orientado a objetos Diseño centrado en la estructura de los datos Diseño basado en componentes El arquitecto de software	Libro: Software Design, David Budgen Capítulos 8,10, 13, 14, 15, 16, 17

*** La planificación con fechas de entrega de las diferentes actividades de clase se compartirá en la Universidad Virtual (dtt)

IV. Metodología:

El curso se basa principalmente en clases magistrales, pero también se desarrollarán otras actividades en clase como casos de estudio o exposiciones. Las discusiones en clase, de conceptos y ejemplos buscarán complementar los conceptos teóricos desarrollados. También se desarrollará 1 proyecto en clase.

En laboratorio se complementa el curso con practicas que servirán de apoyo a la elaboración de proyecto de clase, el programa y planificación de laboratorio será proporcionado por el tutor académico que asigne la escuela de ciencias y sistemas.

V. Evaluación:

Evaluaciones parciales

Se realizarán 3 evaluaciones parciales individuales (agosto 8, septiembre 19 y Octubre 29). Las evaluaciones se realizarán en clase, de forma escrita y evaluarán el contenido cubierto hasta la clase anterior a la evaluación.

Proyecto de Clase

El proyecto consta de 3 fases que se detallan en el enunciado de proyecto adjunto a este programa de clase.

Todos los grupos deben utilizar la metodología Scrum, el primer sprint da inicio con la entrega de la primera fase, serán sprints de 2 semanas y el tutor académico junto con la catedrática de curso realizar revisiones a mitad de los sprints para asegurar que el proyecto se este elaborando sin complicaciones.

a continuación, se detallan las fechas de entrega para referencia del estudiante.

Objetivo	Fecha de entrega	Descripción
Definición de herramientas tecnológicas	Jueves 1 de agosto	Demostración de las herramientas tecnológicas a utilizar en el proyecto
Diseño de bajo nivel	Jueves 12 de septiembre	Presentación en vivo del Scrum Ceremonies luego de finalizar el 3er Sprint, énfasis en aplicación de principios y patrones de diseño (SOLID, GoF)
Diseño de arquitectura	Jueves 31 de octubre	Presentación del 6to Sprint, énfasis en la identificación de requerimientos de

		arquitectura y en la selección y aplicación de patrones de arquitectura.
<p>Laboratorio El tutor académico del curso llevará a cabo actividades de apoyo al desarrollo del proyecto de clase, el enfoque principal del tutor es apoyar al estudiante a elaborar su proyecto de clase.</p> <p>Tareas y cortos En las clases durante el semestre se asignan lecturas para fomentar discusión, así como exámenes cortos y tareas para reforzar lo aprendido en clase.</p> <p>Evaluación</p> <ul style="list-style-type: none"> ● Evaluaciones parciales (30 puntos, 10 puntos cada parcial) ● Proyectos de Clase (25 puntos) <ul style="list-style-type: none"> ○ Fase 1, 5 puntos ○ Fase 2, 8 puntos ○ Fase 3, 12 puntos* ● Actividades en clase (7 puntos) ● Laboratorio (8 puntos) ● Tareas y cortos (5 puntos) ● Examen final (25 puntos) <p>VI. Observaciones:</p> <ul style="list-style-type: none"> ● Es obligatorio aprobar el proyecto de clase para tener derecho a examen final, se aprueba con 61/100 ● Es obligatorio aprobar el laboratorio para tener derecho a examen final, se aprueba con 61/100 ● Es obligatorio aprobar el curso de SQM de los laboratorios de la India. ● Solo se calificarán exámenes y proyectos de estudiantes asignados en el curso. NO se agregan estudiantes en actas ● Los equipos para trabajar los proyectos de clase son de 5 personas y deberán estar formados a más tardar el 25 de julio del 2018, si algún integrante se queda sin grupo luego de una entrega puede adherirse a otro grupo luego de discutirlo con la catedrática y el grupo al que se va a adherir. NO SE CALIFICARÁN TRABAJOS INDIVIDUALES EN NINGUNA CIRCUNSTANCIA. 		
7	Bibliografía	<ul style="list-style-type: none"> ● Head first design patterns, Elisabeth Freeman, Eric Freeman, Bert Bates, Kathy Sierra ● Software systems architecture, working with stakeholders using viewpoints and perspectives, Nick Rozanski, Eoin Woods ● Essential software architecture, Ian Gorton ● Software Engineering 9th Edition, Ian Sommerville ● Documenting Software Architecture, Clemments ● Patterns of Enterprise Application Architecture - Martin Fowler
8	No. De Secciones	2
9	Catedráticos titulares y auxiliares	Inga. Ivonne Aldana
1 1	Director de Escuela	Ing. Carlos Alonzo