

<b>Código:</b>	785	<b>Créditos:</b>	5
<b>Escuela:</b>	CIENCIAS Y SISTEMAS	<b>Área a la que pertenece:</b>	Software
<b>Pre requisito:</b>	283-Análisis y Diseño de Sistemas 1	<b>Post requisito:</b>	780-Software avanzado
<b>Categoría:</b>	Obligatorio	<b>Semestre:</b>	1er Semestre 2025
<b>Docente:</b>	Ivonne Aldana	<b>Auxiliar:</b>	Angel Barrera
<b>Edificio:</b>	-	<b>Sección:</b>	B
<b>Salón del curso:</b>		<b>Salón de laboratorio:</b>	meet
<b>Horas por semana del curso:</b>	4	<b>Horas por semana del laboratorio:</b>	2
<b>Días que se imparte el curso:</b>	Martes y jueves	<b>Días que se imparte el laboratorio:</b>	Miércoles
<b>Horario del curso:</b>	7:10 - 8:40	<b>Horario del laboratorio:</b>	15:40 - 17:20

### 1. Descripción del curso

El curso está diseñado para que el estudiante identifique y aplique los patrones aplicables para el diseño de componentes de una aplicación, es decir patrones de diseño. El uso de dichos patrones, de patrones de arquitectura, técnicas para definir una arquitectura que satisfaga los requerimientos funcionales y no funcionales, comprenden el resto del curso. Se aplicarán los elementos anteriores en un proyecto de curso para poder contar con una experiencia real en el uso de los patrones y técnicas presentados.

### 2. Objetivos

**General**

**Específicos**

Aplicar procesos en el diseño del software, de manera que los mismos consideren los diferentes tipos de requerimientos que pueden impactar en la definición de la estructura de un sistema de software.

1. Identificar y describir los conceptos de arquitectura de software para el desarrollo de sistemas informáticos.
2. Identificar y describir los conceptos de atributos de calidad y tácticas aplicables para alcanzarlos.
3. Evaluar y adaptar el diseño de un sistema de software de acuerdo con los diferentes tipos de requerimientos de un proyecto.
4. Aplicará técnicas para medición cuantificable del rendimiento y escalabilidad de una aplicación

### 3. Metodología

- 1) El curso se impartirá a través de clases magistrales **virtuales** dos días por semana, con duración de dos periodos cada día.
- 2) El laboratorio se impartirá **de manera virtual** una vez por semana, con duración de dos períodos cada día.
- 3) Durante el semestre, se asignará un proyecto dividido en tres fases a realizarse en grupos; así como tareas, ejercicios y pruebas cortas.
- 4) Se desarrollarán investigaciones y exposiciones de conferencias recientes relacionadas con el curso.

### 4. Observaciones

1. Es obligatorio acumular el 80% de asistencia antes de cada parcial (de lo contrario no se tendrá derecho a examen).
2. El laboratorio se calificará sobre 100, y será equivalente a 30 puntos de zona.
3. Habrá 1 proyecto dividido en tres fases que se realizará en grupo.
4. El catedrático revisará las notas obtenidas en el curso y el laboratorio. Podrá decidir sí es necesaria una segunda revisión a cada proyecto y considerar nuevamente la ponderación obtenida en cada proyecto.
5. Las notas de cada proyecto serán publicadas por el catedrático del curso en el transcurso del semestre, el estudiante tendrá 8 días como máximo para pedir revisión de proyecto.
6. El laboratorio debe aprobarse con 61 puntos sobre 100.
7. Es obligatorio ganar el laboratorio para tener derecho a evaluación final del curso.
8. No habrá proyecto de retrasada, ni reposición de nota de laboratorio. Al final del semestre, no se asignarán trabajos extra para recuperar puntos de zona.
9. El curso se aprueba con 61 puntos.
10. Las entregas fuera de fecha no son aceptadas.
11. Debe existir respeto por las opiniones de los demás.
12. Como estudiantes universitarios, se espera que sepan y entiendan las normas de educación, respeto, ética y plagio relacionadas con trabajos de otros autores y con el desarrollo del curso.

5. Contenido temático del curso	
Unidad	Tema
1. Principios de Diseño de Arquitectura de Software	<ol style="list-style-type: none"> <li>1. ¿Qué es Arquitectura de Software?                             <ol style="list-style-type: none"> <li>1.1. Concepto</li> <li>1.2. ¿Por qué es importante?</li> <li>1.3. Beneficios de una Arquitectura de SW</li> <li>1.4. Pasos para la definición de una Arquitectura</li> <li>1.5. ¿Qué se tiene en cuenta para el diseño?</li> <li>1.6. ¿Cómo se modela?</li> <li>1.7. Estructuras y Vistas                                     <ol style="list-style-type: none"> <li>1.4.1 Categorías de Estructuras</li> </ol> </li> <li>1.8. Generos y Estilos Arquitectónicos                                     <ol style="list-style-type: none"> <li>1.8.1 Arquitectura Candidata</li> <li>1.8.2 La Arquitectura de Referencia</li> <li>1.8.3 Diseño Arquitectónico</li> <li>1.8.4 Catálogo de Estilos   <ol style="list-style-type: none"> <li>1.8.4.1 Cliente-Servidor</li> <li>1.8.4.2 Centrada en Datos</li> <li>1.8.4.3 En o por capas</li> <li>1.8.4.4 Centrada en el flujo de datos</li> <li>1.8.4.5 Llamada y retorno</li> <li>1.8.4.6 Basada en eventos</li> <li>1.8.4.7 Modelo-Vista-Controlador</li> </ol> </li> </ol> </li> </ol> </li> </ol>
2. Calidad del Software	<ol style="list-style-type: none"> <li>1. <b>Conceptos Generales</b></li> <li>2. <b>¿Por qué es importante?</b></li> <li>3. <b>Normas ISO</b></li> <li>4. <b>Atributos de Calidad</b> <ol style="list-style-type: none"> <li>4.1. Funcionalidad</li> <li>4.2. Fiabilidad</li> <li>4.3. Usabilidad</li> <li>4.4. Eficiencia</li> <li>4.5. Mantenibilidad</li> <li>4.6. Portabilidad</li> <li>4.7. Otros atributos de calidad no observables via ejecución</li> </ol> </li> </ol>
3. Arquitectura en el Ciclo de Vida	<ol style="list-style-type: none"> <li>1. <b>Arquitectura y Requerimientos</b></li> <li>2. <b>Arquitectura en las Metodologías Ágiles (SCRUM)</b></li> <li>3. <b>Diseño Arquitectónico</b> <ol style="list-style-type: none"> <li>3.1. Estrategia</li> <li>3.2. Estructuras</li> <li>3.3. Patrones y Estilos Arquitectónicos</li> </ol> </li> <li>4. <b>Documentando la Arquitecturas de Software</b></li> <li>5. <b>Implementación y Pruebas</b></li> <li>6. <b>Evaluación de la Arquitectura</b></li> </ol>
4. Arquitectura y Negocios	<ol style="list-style-type: none"> <li>1. <b>Aplicaciones con casos</b></li> </ol>

## 6. Evaluación del rendimiento académico

Según el Reglamento General de Evaluación y Promoción del Estudiante de la Universidad de San Carlos de Guatemala, la zona tiene un valor de 75 puntos, la nota mínima de promoción es de 61 puntos y la zona mínima para optar a examen final es de 36 puntos.

Procedimiento de evaluación		Ponderación
Clase	Tareas y/o cortos	04 pts.
	Exposición	05 pts.
	Primer parcial	10 pts.
	Segundo parcial	13 pts.
	Tercer parcial	13 pts.
<b>Total de clase</b>		<b>45 pts.</b>
Laboratorio	Proyecto Fase I	9 pts.
	Proyecto Fase II	9 pts.
	Proyecto Fase III	9 pts.
	Prácticas	3 pts..
<b>Total de laboratorio</b>		<b>30 pts.</b>
Zona		75 pts.
Examen Final		25 pts.
Nota de promoción		100 pts.

## 7. Cronograma de actividades

Tema principal	Fecha
Principios de Diseño de Software (hasta 1.8.3)	8 Clases (20 períodos)
<b>Primer parcial.</b>	<b>18 de febrero</b>
Principios de Diseño de Software (1.8.4)	2 Clases (4 períodos)
Calidad del Software	5 Clases (12 períodos)
<b>Segundo Parcial.</b>	<b>18 de Marzo</b>
Calidad del Software	1 Clase (2 períodos)
Arquitectura en el ciclo de vida	4 Clases (8 períodos)
<b>Tercer Parcial</b>	<b>22 de abril</b>
Arquitectura y Negocios	Días para exposiciones y repaso

## 8. Bibliografía

- Essential software architecture, Ian Gorton
- Just enough software architecture, George Fairbanks
- 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
- Software performance and scalability: a quantitative approach, Henry H. Liu



## 9. Normas para la clase virtual

- Todas las Comunicaciones con el profesor y los auxiliares deben ser por los correos electrónicos que se indiquen en clase.
- En toda comunicación escrita se debe mostrar respeto y no utilizar mensajes en mayúsculas.
- Las comunicaciones enviadas por correo electrónico serán atendidas en un máximo de 3 días hábiles.
- Durante las clases los estudiantes deben encender su cámara siempre que el profesor o el auxiliar les hagan una pregunta directa, o bien, cuando el estudiante realice alguna consulta.
- Durante las clases los estudiantes pueden hacer consultas por el chat del curso o por la opción de Questions / Answers, según lo indique el profesor, teniendo el cuidado de ser respetuoso y mantener las reglas de cortesía durante la escritura.