

## PROGRAMA DE CURSO INGENIERÍA DE SOFTWARE

### A. Antecedentes generales del curso:

Departamento	Ciencias de la Computación					
Nombre del curso	Ingeniería de software	Código	CC4401	Créditos	6	
Nombre del curso en inglés	<i>Software Engineering</i>					
Horas semanales	Docencia	3	Auxiliares	1,5	Trabajo personal	5,5
Carácter del curso	Obligatorio	X		Electivo		
Requisitos	CC3002: Metodologías de diseño y programación, CC3201: Bases de Datos					

### B. Propósito del curso:

El propósito del curso Ingeniería de Software es que los y las estudiantes construyan, con su equipo de trabajo, un sistema web con dos a tres perfiles de usuario, considerando el diseño de interfaces gráficas, el dominio de negocio y el uso de frameworks para desarrollar aplicaciones web, según el problema a resolver.

Por otra parte, manejan técnicas de especificación de requisitos funcionales y no funcionales, así como estrategias de diseño basadas en patrones arquitectónicos, y técnicas de mantención, evolución, verificación y validación de software.

El curso tributa a las siguientes competencias específicas (CE) y genéricas (CG):

CE5: Concebir, diseñar y construir soluciones de software, siguiendo un proceso sistemático y cuantificable, acorde a los fundamentos, eligiendo el paradigma y las técnicas más adecuadas.

CE6: Desarrollar software en una amplia variedad de plataformas y lenguajes de programación.

CE7: Gestionar proyectos de diseño, desarrollo, implementación y evolución de soluciones de software, contemplando tanto los procesos involucrados como el producto esperado, su calidad y respuesta efectiva al problema que aborda.

CG1: Comunicación académica y profesional

Comunicar en español de forma estratégica, clara y eficaz, tanto en modalidad oral como escrita, puntos de vista, propuestas de proyectos y resultados de investigación fundamentados, en situaciones de comunicación compleja, en ambientes sociales, académicos y profesionales.

CG3: Compromiso ético

Actuar de manera responsable y honesta, dando cuenta en forma crítica de sus propias acciones y sus consecuencias, en el marco del respeto hacia la dignidad de las personas y el cuidado del medio social, cultural y natural.

**CG4: Trabajo en equipo:**

Trabajar en equipo, de forma estratégica y colaborativa, en diversas actividades formativas, a partir de la autogestión de sí mismo y de la relación con el otro, interactuando con los demás en diversos roles: de líder, colaborador u otros, según requerimientos u objetivos del trabajo, sin discriminar por género u otra razón.

**C. Resultados de aprendizaje:**

Competencias específicas	Resultados de aprendizaje
CE5	RA1: Construye con su equipo un sistema web, con dos a tres perfiles de usuario, considerando el diseño de interfaces gráficas, a fin de satisfacer las necesidades y requerimientos de un cliente.
CE6	RA2: Utiliza frameworks para el desarrollo de aplicaciones web, considerando el dominio de negocio, a fin de construir una aplicación donde se reutilizan funcionalidades comunes del manejo de usuario.
CE7	RA3: Gestiona un proyecto de desarrollo de software de mediana complejidad, que satisfaga los requerimientos del usuario/cliente, considerando requisitos funcionales y no funcionales, así como diseño arquitectónico basado en patrones.
Competencias genéricas	Resultados de aprendizaje
CG1	RA4: Elabora, con lenguaje objetivo y preciso, informes técnicos para reportar, en forma oral o escrita, propuestas de interfaces gráficas, especificación de requisitos y evaluación del código legado.
	RA5: Utiliza estrategias de lectura tales como identificar y sintetizar información relevante en textos científicos, artículos, y documentos técnicos, para relacionar y aplicar conceptos a su propuesta de solución en el ámbito de la ingeniería de software.
CG3, CG4	RA6: Realiza, de manera responsable y honesta, las tareas, cumpliendo, organizadamente, con su equipo los plazos correspondientes, sin incurrir en plagio, copia o suplantación de identidad respecto del entregable.

**D. Unidades temáticas:**

Número	RA al que tributa	Nombre de la unidad	Duración en semanas
1	RA2, RA4, RA5, RA6	Conceptos de Ingeniería de Software	1 semana
Contenidos		Indicador de logro	
1.1. Conceptos generales de ingeniería de software. 1.2. Atributos de calidad mínima de un software para su desarrollo.		La/el estudiante:  1. Utiliza conceptos de software, ingeniería, ciencias de la computación e ingeniería de software,	

	<p>considerando los atributos de calidad mínima de un software.</p> <ol style="list-style-type: none"> <li>Identifica las etapas involucradas en el desarrollo de un software.</li> <li>Cumple obligaciones y acuerdos, respetando los compromisos adquiridos en sus actividades académicas.</li> <li>Considera la opinión de los demás, aun cuando es divergente, aportando a la construcción de un clima de tolerancia en el contexto de las actividades del equipo.</li> <li>Lee textos científicos, académicos y profesionales en ingeniería de software para el uso de conceptos aplicables a soluciones para el desarrollo de software.</li> <li>Produce textos breves acerca de conceptos generales que se usan en la ingeniería de software, usando lenguaje objetivo y preciso.</li> </ol>
<b>Bibliografía de la unidad</b>	[1]

Número	RA al que tributa	Nombre de la unidad	Duración en semanas
2	RA3, RA5, RA6	Metodologías de Desarrollo Software	1 semana
Contenidos		Indicador de logro	
<p>2.1. Modelos de ciclo de vida del desarrollo de software.</p> <p>2.2. Modelo de cascada, iterativo-incremental, espiral, Scrum.</p>		<p>La/el estudiante:</p> <ol style="list-style-type: none"> <li>Identifica y analiza las etapas para el desarrollo de un software, considerando modelos del ciclo de vida de este.</li> <li>Organiza las etapas de desarrollo de un software en los distintos modelos de ciclo de vida (cascada, iterativo, incremental, espiral, Scrum).</li> <li>Lee diversos textos para identificar y sintetizar información específica sobre metodologías aplicables al desarrollo de software.</li> <li>Planifica organizadamente su trabajo y tiempo para cumplir con las tareas asignadas dentro del equipo.</li> </ol>	
<b>Bibliografía de la unidad</b>		[2][3]	

Número	RA al que tributa	Nombre de la unidad	Duración en semanas
3	RA1, RA2, RA5, RA6	Requisitos del Software	2 semanas
Contenidos		Indicador de logro	
3.1. Requisitos para el desarrollo de un software. 3.2. Técnicas de elicitación. 3.3. Técnicas para especificación de requisitos.		La/el estudiante: <ol style="list-style-type: none"> <li>Utiliza técnicas para especificar requisitos funcionales y no funcionales asociados al desarrollo</li> <li>Analiza casos de uso para identificar tareas del sistema, considerando aspectos técnicos, así como los actores relevantes que interactuarán en dichas tareas.</li> <li>Lee diversos textos, relacionando información extraída de estos con otros conocimientos del área de la computación aplicables al desarrollo de software.</li> <li>Define y respeta acuerdos comunes para dar cumplimiento a la meta, compartiendo y analizando diversas ideas.</li> </ol>	
Bibliografía de la unidad		[1][2][3]	

Número	RA al que tributa	Nombre de la unidad	Duración en semanas
4	RA2, RA3, RA4, RA6	Diseño del Software	5 semanas
Contenidos		Indicador de logro	
4.1. Diseño arquitectónico del software. 4.2. Patrones de arquitectura. 4.3. Diseño de interfaces con el usuario. 4.4. Evaluación de interfaces.		La/el estudiante: <ol style="list-style-type: none"> <li>Aplica patrones de diseño para elaborar la arquitectura del sistema a construir, tomando en cuenta requisitos funcionales y no funcionales.</li> <li>Diseña interfaces gráficas, tomando en cuenta principios de diseño de interacciones de usuario.</li> <li>Evalúa propuestas de interfaces, usando distintas técnicas, por ejemplo, estudios de usabilidad, y prototipos de papel, entre otros.</li> <li>Comparte al equipo información, conocimientos y experiencias de forma clara y precisa, para aportar al logro de los objetivos comunes.</li> <li>Produce reportes técnicos sobre el diseño de un software, usando lenguaje objetivo y preciso.</li> </ol>	
Bibliografía de la unidad		[1][2][3]	

Número	RA al que tributa	Nombre de la unidad	Duración en semanas
5	RA1, RA3, RA6	Mantenimiento y Evolución del Software	3 semanas
<b>Contenidos</b>		<b>Indicador de logro</b>	
5.1. Tipos de mantención. 5.2. Estrategias de cambio. 5.3. Proceso de cambio. 5.4. Herramientas de integración continua, issue tracking, bug tracking.		La/el estudiante: <ol style="list-style-type: none"> <li>1. Evalúa el diseño del software, mediante una reflexión acerca de su implementación, considerando qué funcionó o qué no, así como aspectos éticos derivados.</li> <li>2. Modifica código ajeno, para ajustarlo a nuevas necesidades y requisitos, o para corregir desperfectos.</li> <li>3. Determina la funcionalidad e importancia de las herramientas para hacer seguimiento al proceso de desarrollo (issue tracking, bug tracking).</li> <li>4. Identifica y analiza hechos o situaciones que conllevan dilemas éticos, en relación a acciones reñidas con las buenas prácticas respecto al desarrollo de software.</li> </ol>	
<b>Bibliografía de la unidad</b>		[1][2][3]	

Número	RA al que tributa	Nombre de la unidad	Duración en semanas
6	RA1, RA6	Verificación y Validación de Software	3 semanas
<b>Contenidos</b>		<b>Indicador de logro</b>	
6.1. Conceptos de verificación y validación de software. 6.2. Pruebas funcionales, estructurales y de integración. 6.3. Verificación estática del software. 6.4. Criterios de cobertura.		La/el estudiante: <ol style="list-style-type: none"> <li>1. Utiliza el Modelo V para verificación del software.</li> <li>2. Define pruebas en base a criterios de cobertura, que permitan evaluar si el producto construido satisface su especificación.</li> <li>3. Maneja pruebas de caja negra, pruebas de caja blanca, pruebas de integración.</li> <li>4. Realiza un análisis manual de código fuente, a fin de determinar la calidad del mismo.</li> <li>5. Considera la opinión de los demás, también cuando ella es divergente, aportando a la construcción de un clima de tolerancia en el contexto de las actividades del equipo.</li> </ol>	
<b>Bibliografía de la unidad</b>		[1][2][3]	

### E. Estrategias de enseñanza:

El curso considera diversas estrategias de enseñanza - aprendizaje:

- **Clases de expositivas:** se presentan los conceptos fundamentales de cada unidad. Los estudiantes analizan ejemplos y trabajan en problemas y ejemplos fundamentales para la ingeniería de software, así como el uso de técnicas y herramientas para abordarlos.
- **Uso de demos tecnológicos:** se trabaja con demos de herramientas para aplicar los aprendizajes al desarrollo de software.
- **Trabajo de proyecto:** trabaja en diversas entregas, donde los y las estudiantes avanzan en el desarrollo de un software, según las especificaciones, requisitos, y funcionalidad de este.

*En las clases auxiliares los estudiantes trabajan con demos de herramientas, y se presentan avances en relación con de tareas y actividades; asimismo, se aclaran dudas de las metodologías vistas.*

### F. Estrategias de evaluación:

Al inicio de cada semestre, el cuerpo académico informará sobre la cantidad y tipo de evaluaciones, así como las ponderaciones correspondientes.

El curso considera distintas instancias de evaluación:

Tipo de evaluación	Resultado de aprendizaje asociado a la evaluación
Dos controles	Con estas actividades se evalúan: Control 1: evalúa los RA1, RA6. Control 2: evalúa los RA3, RA6.
Examen final	Evalúa conceptos generales sobre Ingeniería de Software. Se evalúan los RA1, RA2, RA3.
Un proyecto dividido en 2 a 3 iteraciones	<ul style="list-style-type: none"> <li>- Informes evalúa RA5, RA6.</li> <li>- Especificación de requisitos evalúa RA1, RA4, RA5, RA6.</li> <li>- Diseño de interfaces evalúa RA1, RA4, RA6.</li> <li>- Programación y pruebas evalúa RA1, RA2, RA3, RA4, RA5, RA6.</li> </ul>

## G. Recursos bibliográficos:

### Bibliografía obligatoria:

- [1] Fundamentals of Software Engineering (2nd Edition), Carlo Ghezzi, Mehdi Jazayeri, Dino Mandrioli, Prentice Hall; September 29, 2002.
- [2] Software Engineering: (Update) (8th Edition), Ian Sommerville; Addison Wesley, June 4, 2006.
- [3] Software Engineering: A Practitioner's Approach / (7ª edición), Roger Pressman; McGraw-Hill Science/Engineering/Math, January 20, 2009.

## H. Datos generales sobre elaboración y vigencia del programa de curso:

Vigencia desde:	Otoño, 2021
Elaborado por:	Jocelyn Simmonds
Validado por:	Revisión académico par: Sergio Ochoa Validación CTD de Computación
Revisado por:	Área de Gestión Curricular