

1. Información general

NOMBRE DEL CURSO: Análisis y Diseño de Sistemas 1

CÓDIGO:	283	CRÉDITOS:	4
ESCUELA:	Ciencias y Sistemas	ÁREA A LA QUE PERTENECE:	Software
PRE REQUISITO:	Sistemas Administrativos de Bases de Datos 1	POST REQUISITO:	Análisis y diseño de sistemas 2
CATEGORÍA:	Obligatorio		
CATEDRÁTICO (A):	Edgar Francisco Rodas Robledo	AUXILIAR:	Ludwing Gabriel Paz Hernandez
EDIFICIO:	Virtual	SECCIÓN:	B
SALÓN DEL CURSO:	Meet	SALÓN DEL LABORATORIO:	Meet
HORAS POR SEMANA DEL CURSO:	4	HORAS POR SEMANA DEL LABORATORIO:	2
DÍAS QUE SE IMPARTE EL CURSO:	Sabado	DÍAS QUE SE IMPARTE EL LABORATORIO:	Sábado
HORARIO DEL CURSO:	7:10 a 12:10	HORARIO DEL LABORATORIO:	12:10 a 13:50

2. Descripción del curso

El curso busca describir todas las actividades, documentación y roles involucrados en la ingeniería de software que son importantes en el desarrollo de software, en el contexto de las metodologías establecidas para ello. En el curso se presentan diferentes metodologías de desarrollo de software aplicables en diferentes contextos y tipos de software, además los conceptos y técnicas relacionadas con crear modelos que representen al negocio, para utilizarlos como base en la definición de un sistema informático. Por último, el curso busca que estudiante comprenda la ingeniería de requerimientos para la construcción de un sistema informático, considerando la satisfacción de las necesidades de los usuarios.

3. Vinculación con las competencias del Perfil de Egreso

1.	• Utiliza una definición de lo que es una metodología de desarrollo de software tradicionales y ágiles para aplicarlo al contexto organizacional e implementación de software
2.	• Implementa y desarrolla modelos de negocio para entender cómo se relaciona el software con la organización.
3.	• Analiza requerimientos e implementa soluciones de software aplicando las buenas prácticas de la ingeniería de requerimientos.

4. Unidad de aprendizaje N° 1 Título: Introducción a la Ingeniería de Software

Comprender, analizar, e implementar conceptos de confiabilidad, disponibilidad, seguridad y calidad en productos y servicios de Software

Competencias de la unidad

- Implementa software para organizaciones de acuerdo al concepto de desarrollo de Software Profesional
- Implementa diferentes tipos de software para diferentes modelos de negocio a partir del conocimiento de la ingeniería de software como un medio que le ayudará a la implementación
- Implementa diferentes tipos de software para diferentes modelos de negocio a partir del conocimiento de lo que es la diversidad de la Ingeniería de Software
- Define correctamente un software para implementaciones en tiempo y calidad a partir del conocimiento del concepto de ingeniería de Software y la Web
- Desarrolla e implementa software de acuerdo a sus capacidades y conocimientos en organizaciones de diferente dominio aplicando los conceptos de ética en la ingeniería de Software.

Criterios de desempeño

Saber hacer	Saber conocer	Saber ser
Resolver problemas de Ingeniería de Software identificando y describiendo los conceptos de arquitectura de software para el desarrollo de sistemas informáticos.	Analizar problemas para plantear una solución de Ingeniería de Software. Comprender la importancia de la Ingeniería de Software y su aplicación en los diferentes tipos de soluciones de software.	Crea soluciones a problemas de la vida real, utilizando la ingeniería de Software. Integra los conocimientos adquiridos y los utiliza para crear respuestas a problemas de Software.

<p>Desarrollar soluciones de Software tomando en cuenta las definiciones de Calidad, agilidad y ética.</p> <p>Planifica e implementa técnicas de medición cuantificable para el rendimiento y escalabilidad de una solución.</p>		
--	--	--

4.1 Evidencia de aprendizaje

Plantea la propuesta para resolver el primer caso de Estudio denominado Práctica No.1

Analiza críticamente el enunciado de la práctica para proponer posibles soluciones de Análisis y Diseño.

4.2 Instrumento de evaluación

Hoja de Calificación de la primera práctica:

- Análisis
- Diseño
- Conclusiones
- Recomendaciones

5. Unidad de aprendizaje N° 2

Título: Procesos de Software

Comprender, analizar, e implementar Modelos de procesos de Software, actividades del proceso y como enfrentar el cambio.

Competencias de la unidad

- Define, desarrolla e implementa software con altos estándares de calidad para mejorar los procesos en las organizaciones por medio de los modelos de Procesos de Software.

	<ul style="list-style-type: none"> ● Implementa software en tiempo y calidad para organizaciones con diferentes modelos de negocio aplicando todas las actividades del proceso de ingeniería de software ● Implementa software para organizaciones con requerimientos cambiantes en el tiempo por medio de mecanismos que ayuden a enfrentar el cambio.
--	---

Criterios de desempeño		
Saber hacer	Saber conocer	Saber ser
<p>Resolver problemas de Ingeniería de Software utilizando metodologías ágiles como SCRUM.</p> <p>Adaptar la implementación de un problema de Desarrollo de Software a un marco de trabajo ágil.</p> <p>Construir soluciones de Análisis y Diseño de Software a través de marcos de trabajo iterativos e incrementales.</p>	<p>Analizar problemas desde un marco de trabajo ágil para plantear una solución de Ingeniería de Software.</p> <p>Comprender los diferentes roles que se tienen en marcos de trabajo ágil.</p> <p>Conocer los diferentes marcos de trabajo ágil, su teoría, valores, roles y eventos.</p>	<p>Crea soluciones a problemas de la vida real, utilizando marcos de trabajo ágil.</p> <p>Cooperar con algún rol en la implementación de soluciones de software utilizando marcos de trabajo ágil.</p> <p>Colaborar en la adaptación de marcos de trabajo ágil para un proyecto de Software.</p>

5.1 Evidencia de aprendizaje

Plantea la propuesta para resolver caso de Estudio denominado Práctica No.2
 Analiza críticamente el enunciado de la práctica para proponer posibles soluciones de análisis y diseño utilizando marcos de trabajo ágil.

5.2 Instrumento de evaluación

Hoja de Calificación de la primera práctica:

- Propuesta de solución utilizando marcos de trabajo ágil.
- Definición de Roles
- Definición de Artefactos
- Definición de Eventos

6. Unidad de aprendizaje N° 3
Título: Desarrollo ágil de Software

Analizar, desarrollar e implementar software utilizando las buenas prácticas de las metodologías dentro del contexto organizacional

Competencias de la unidad

- Desarrolla software ágil para cumplir con los tiempos de entrega y calidad del software por medio del conocimiento de los métodos de desarrollo ágil de software, el manifiesto ágil, así como las diferencias entre el desarrollo ágil y el dirigido por un plan.
- Desarrolla software de forma extrema para entregas de software en períodos cortos utilizando las prácticas clave en la programación extrema y conociendo cómo se relacionan con los principios generales de los métodos ágiles.
- Desarrolla y administra el software para cumplir con entregas periódicas en tiempos cortos utilizando el enfoque de Scrum.
- Diseña sistemas de información de gran magnitud para resolver problemas complejos y automatizar procesos conociendo los conflictos y problemas de escalar los métodos de desarrollo ágil para el diseño de sistemas de software grandes.

Criterios de desempeño

Saber hacer	Saber conocer	Saber ser
<p>Desarrollar e implementar software utilizando métodos de desarrollo ágil y programación extrema.</p> <p>Planificar e implementar software utilizando un enfoque de Scrum.</p> <p>Resolver los conflictos o problemas de escalar los métodos ágiles para diseño de sistemas grandes.</p>	<p>Comprender y analizar las buenas prácticas de cada una de las metodologías ágiles aplicables al contexto organizacional.</p> <p>Conocer las prácticas clave de la programación extrema y como se relacionan con los principios generales de los métodos ágiles.</p>	<p>Proponer la metodología correcta a utilizar de acuerdo al proyecto de software y el contexto organizacional.</p> <p>Organizar al equipo a su cargo para la implementación del software.</p> <p>Integrar conocimientos para la organización e implementación exitosa del software.</p>

--	--	--

6.1 Evidencia de aprendizaje

Planificación y desarrollo de 1 o 2 iteraciones del proyecto de software.

6.2 Instrumento de evaluación

Hoja de Calificación de la práctica:

- Propuesta de iteraciones del proyecto de clase.
- Planificación de las 2 iteraciones
- Desarrollo de las 2 iteraciones
- Evidencia de las ceremonias del proyecto
- Implementación de las 2 iteraciones

7. Unidad de aprendizaje N° 4

Título: Ingeniería de requerimientos

Descubrir y administrar los requerimientos de usuario utilizando los procesos y la documentación adecuada.

Competencias de la unidad

- Descubre los requerimientos del sistema para mejor entendiendo y desarrollo de los mismos utilizando los conceptos de requerimientos del usuario y del sistema,

	<p>así como por qué tales requerimientos se deben escribir en diferentes formas.</p> <ul style="list-style-type: none"> ● Analiza y especifica los requerimientos funcionales y no funcionales del sistema para poder planificar la forma de resolverlos en tiempo, calidad y costos a partir de la comprensión de las diferencias entre requerimientos de software funcionales y no funcionales. ● Desarrolla un documento de requerimientos de software para especificar, entender y organizar los requerimientos a partir de una definición de los mismos y el conocimiento de la estructura del documento, esto con el fin de desarrollar lo que el cliente solicita y pueda estar satisfecho con el sistema. ● Define los requerimientos para llevar un control de las etapas de los mismos a partir de las principales actividades de la ingeniería de requerimientos: adquisición, análisis y validación, así como las relaciones entre dichas actividades. ● Administra los requerimientos para no sobrepasar los recursos definidos para el sistema (tiempo, costo y calidad) a partir del conocimiento de la administración de requerimientos y cómo ésta apoya otras actividades de la ingeniería de requerimientos.
--	---

Criterios de desempeño

Saber hacer	Saber conocer	Saber ser
<p>Organizar los requerimientos del sistema y requerimientos de usuario.</p> <p>Organizar y conocer los requerimientos funcionales y no funcionales que afecta el proyecto de desarrollo de software.</p> <p>Analizar, entender y priorizar los requerimientos del sistema.</p> <p>Definir los alcances y las pruebas de cada</p>	<p>Conocer los procesos de la ingeniería de requerimientos</p> <p>Identificar las diferencias entre requerimientos de usuario y requerimientos del sistema, así como la diferencia entre requerimientos funcionales y no funcionales.</p> <p>Reconocer la importancia de la administración de los requerimientos.</p>	<p>Cooperar con el equipo de desarrollo de software para implementar el alcance de cada requerimiento.</p> <p>Cooperar en resolver los problemas y conflictos con los usuarios y las ambigüedades de los requerimientos.</p> <p>Colaborar en diseñar y ejecutar las pruebas de cada requerimiento del sistema.</p> <p>Mostrar cómo los requerimientos funcionales y no funcionales pueden ocasionar atrasos en las entregas, por lo tanto organiza</p>

<p>requerimiento del sistema.</p>		<p>y prioriza.</p> <p>Mostrar cómo utilizar el lenguaje para descubrir los requerimientos del usuario.</p>
-----------------------------------	--	--

7.1 Evidencia de aprendizaje

Desarrollo del documento de Requerimientos de software.

7.2 Instrumento de evaluación

Hoja de Calificación de la práctica:

- Propuesta del proyecto de clase
- Requerimientos de usuario
- Requerimientos del sistema
- Requerimientos funcionales
- Requerimientos no funcionales
- Requerimientos en lenguaje natural estructurado
- Escenarios de cada requerimiento
- Documento final de requerimientos de software

Título: Modelado del Sistema

Utilizar los distintos tipos de modelo de sistema como parte de la ingeniería de requerimientos y los procesos de diseño del sistema.

Competencias de la unidad

- Desarrolla modelos gráficos de software para entender el funcionamiento del sistema a partir de la comprensión de los diferentes modelos para utilizados para representar los sistemas de software.
- Desarrolla diferentes tipos de modelos de software para entender las interacciones internas y externas del sistema a partir de la comprensión de los diferentes tipos de modelo, así como las perspectivas fundamentales de contexto, interacción, estructura y comportamiento del modelado de sistemas.
- Desarrolla algunos de los tipos de diagrama en el Lenguaje de Modelado Unificado (UML) para conocer cómo se utilizan dichos diagramas en el modelado del sistema a partir del uso de UML.
- Desarrolla las ideas que subyacen en la ingeniería dirigida por modelo, donde un sistema se genera automáticamente a partir de modelos estructurales y de comportamiento.

Criterios de desempeño

Saber hacer	Saber conocer	Saber ser
<p>Desarrollar diferentes tipos modelos gráficos para representar los sistemas de software.</p> <p>Aplicar Lenguaje de modelado unificado (UML) para desarrollar diagramas de modelado de sistemas</p>	<p>Identificar los diferentes modelos gráficos para representar los sistemas de software.</p> <p>Identificar los diferentes tipos de diagramas utilizados en el lenguaje de modelado unificado.</p>	<p>Cooperar con el equipo de desarrollo para diseñar y desarrollar software utilizando los modelos gráficos.</p> <p>Mostrar el uso de los diagramas y modelos del sistema para implementar y desarrollar de manejar acertada los requerimientos del usuario.</p>

--	--	--

8.1 Evidencia de aprendizaje

Desarrollar los diferentes modelos y diagramas del proyecto de software.

8.2 Instrumento de evaluación

Hoja de Calificación de la primera práctica:

- Propuesta de proyecto de clase.
- Desarrollo de modelos de contexto
- Desarrollo de modelos de interacción
- Desarrollo de modelos estructurales
- Desarrollo de modelos de comportamiento

9. Evaluación del curso

Unidad de aprendizaje	Evidencia de aprendizaje	Instrumento evaluación	Fecha	Valoración
No.1	Proyecto de clase	Documento de análisis y requerimientos de		20
No.2	Examen Parcial 1	Evaluación		15
No.3	Examen Parcial 2	Evaluación		15
No.4	Laboratorio	Proyecto y examen final		15
No.5	Tareas, cortos, asistencia y participación	Documento de tareas de cada unidad		10
No.6	Examen Final	Evaluación		25

10. Referencias bibliográficas

Ingeniería de Software 9ª. Novena Edición, Ian Sommerville, Pearson

Scrum from the trenches – 2nd edition, Henrik Kniberg, <http://www.infoq.com/minibooks/scrum-xp-from-the-trenches-2>

Essentials of software engineering, Frank Tsui, Orlando Karam, Barbara Bernal

Software Engineering - modern approaches, Eric J. Braude, Michael E. Bernstein

Writing effective use cases. Alistair Cockburn.

User stories applied: for agile software development. Mike Cohn

Extreme programming pocket guide. O'Really Media.