



CONTENIDO

Presentación de la asignatura virtual 2

 Perfil de la asignatura virtual 3

 Metodología 5

 Evaluación 6

 Cronograma 7

 Proyecto final 12

 Referencias 13

 Perfil del docente 14





Presentación de la asignatura virtual

Ingeniería de software.



[Software. Freepik.com. \(CCBY\)](https://www.freepik.com)

Este curso se presentan los conceptos que guíen el proceso de análisis y diseño para la creación de software. Se basa en conceptos, técnicas de la ingeniería de software presentando al final el Documento visión con los aspectos del software a desarrollar.

El curso se desarrolla a través de cinco módulos. Se inicia con aspectos de definición de la Ingeniería de software, desafíos y conceptos, actividades de desarrollo, así como modelos de ciclo de vida, y metodologías de desarrollo, finalizando con Lenguaje de Modelado Unificado (UML) correspondientes a los Fundamentos de la Ingeniería de software del módulo 1.

EL módulo 2 aborda temas de modelo de negocio, técnicas de recopilación de información. Seguido por el análisis del problema, visión del sistema y la matriz de trazabilidad del módulo 3. Siguiendo con la elaboración de especificación de requisitos en el módulo 4 y culminando con el módulo 5 con temas de riesgo en el desarrollo de software.



Nota: El contenido presentado se basa en plan de contenido actualizado en el 2021.

Perfil de la asignatura virtual

Ingeniería de software

Módulo I: Fundamentos de la Ingeniería de software

Objetivo: Examinar los modelos del ciclo de vida de desarrollo de software a través de las actividades de desarrollo de la ingeniería de software para conocer las metodologías de desarrollo de la ingeniería software.

OBJETIVO GENERAL

Desarrollar una propuesta de desarrollo de sistemas que considere problemas y necesidades del cliente, aplicando los conceptos de la Ingeniería de software presentado a través de una especificación de desarrollo de software.

Unidad I. Definición de la Ingeniería de software
Objetivo: revisar las definiciones de la Ingeniería de software explorando diferentes autorías para generar una definición sobre la misma.

Unidad II. Desafíos de la Ingeniería de software
Objetivo: Contrastar las actividades en la ingeniería de software que permita inferir los retos que involucra a través del análisis de experiencias documentadas

Unidad III. Conceptos de Ingeniería de software
Objetivo: Examinar los conceptos de ingeniería de software que permita integrarlos para la comprensión de las actividades que se desarrollan

Unidad IV. Actividades de desarrollo de la Ingeniería de software
Objetivo: Contrastar las actividades para identificar las etapas a realizarse en la ingeniería de software.

Unidad V. Modelos del ciclo de vida de desarrollo de software
Objetivo: Revisar los modelos de ciclo vida que permita organizar el ciclo de desarrollo del software.

Unidad VI. Metodologías de desarrollo de software
Objetivo: Comparar las metodologías de desarrollo que permita identificar aquella que guíe el proceso de desarrollo del software.

Unidad VII. Lenguaje de Modelado Unificado (UML)
Objetivo: Examinar UML para inferir la representación de un modelo de desarrollo del software.

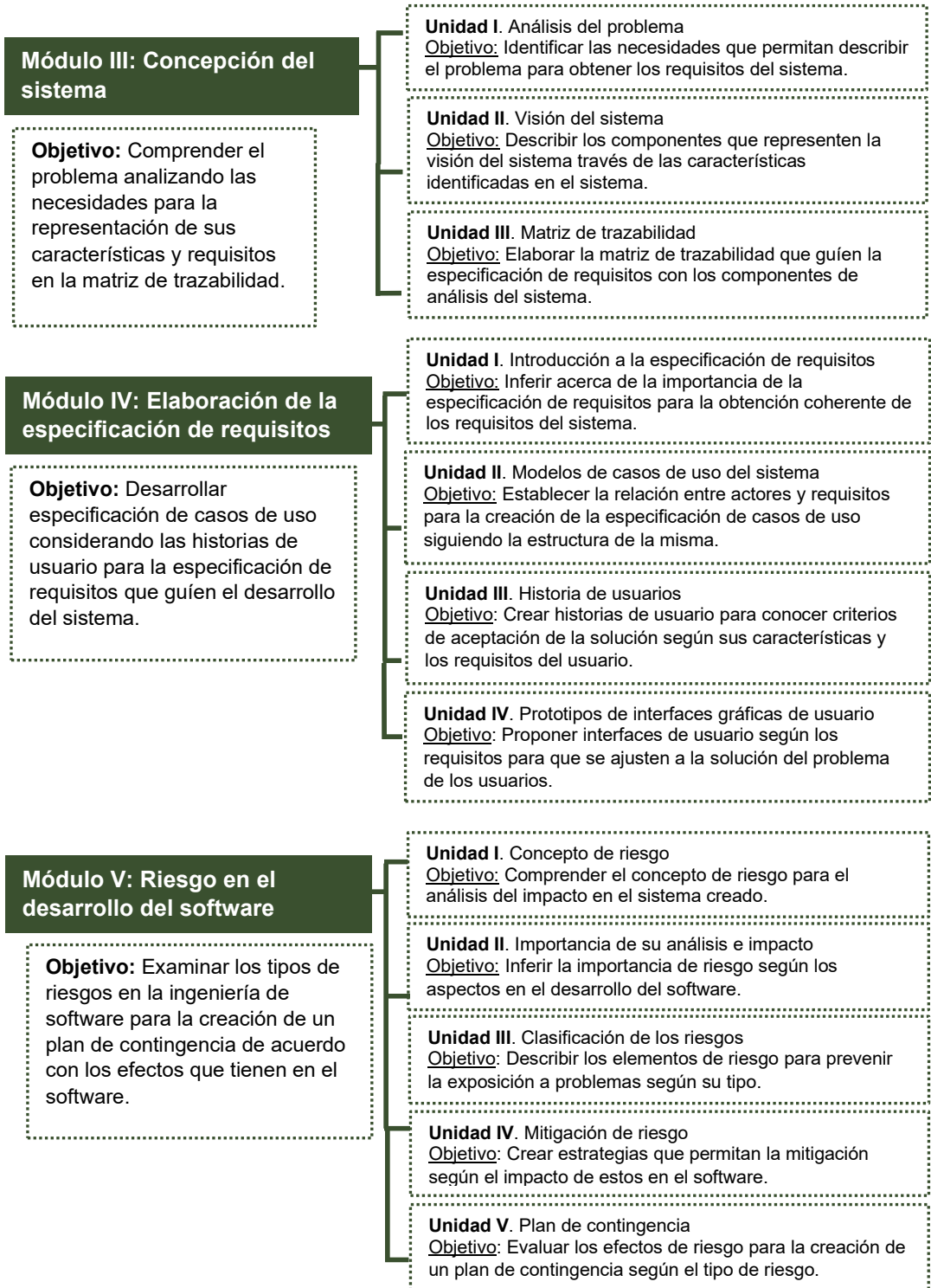
Módulo II: Modelado del negocio

Objetivo: Representar el modelo de negocios de acuerdo con las oportunidades de mejora que se identifican con las técnicas de recopilación de información.

Unidad I. Entendimiento del entorno e identificación de oportunidades de mejora
Objetivo: Explorar el entorno del negocio para identificar aspectos de mejora revisando las técnicas de recopilación de información.

Unidad II. Técnicas de recopilación de información
Objetivo: Analizar las técnicas de recopilación de información para describir el modelo de negocio según el tipo información que se obtenga con estas.

Unidad III. El modelo de negocios
Objetivo: Reconocer los elementos del modelo de negocio que guíe su representación a través de un caso práctico.





Metodología

Este curso se basa en los siguientes aspectos:



Teacher. Freepik License

- **Revisión semanal del material por unidad.** Corresponderán a las unidades de los módulos del curso.
- **Seguimiento continuo.** Se asignarán actividades que van desde ejercicios cortos, investigaciones, actividades varias individuales o grupales, entregables de proyectos y parciales. Debe cumplir con las fechas de entrega correspondiente.
- **Proyecto final.** Se trabajará en grupo y se realizarán revisiones continuas con actividades que le ayuden a completar el

mismo para ser entregado al finalizar el semestre.

Importante: Todas las actividades se realizarán en la plataforma Moodle. Comunicación vía correo e interacciones en Moodle y la plataforma OneDrive del Office 365 cuando se indique.



Evaluación

Detalle la evaluación



[Freepik License \(CCBY\).](#)

Una cantidad significativa de su trabajo será el trabajo en equipo y otras actividades individuales según se asigne.

Criterios de Evaluación	Porcentaje
Asignaciones individuales	10
Asignaciones grupales	15
Ejercicios cortos	5
Parciales	30
Portafolio	7
Semestral	33
Total	100





Cronograma

Módulo I: FUNDAMENTOS DE LA INGENIERÍA DE SOFTWARE

CONTENIDO	ESTRATEGIAS DIDÁCTICAS	FECHA (semana)	ACTIVIDADES
Unidad 1. Definición de ingeniería software 1.1 Definición Unidad 2. Desafíos de la ingeniería de software 1.2 Desafíos	Investigaciones Lecturas videos	8 agosto – 12 agosto (semana 1)	Presentación del Curso Mod1- U1 y U2 Fundamentos de la Ingeniería de Software Conceptos de la Ingeniería de Software Actividades de desarrollo
Unidad 3. Conceptos de la ingeniería de software 1.3 Conceptos de Ingeniería de Software 1.3.1 Participantes y papeles 1.3.2 Sistemas y modelos 1.3.3 Productos de trabajo 1.3.4 Actividades, tareas y recursos 1.3.5 Objetivos, requerimientos y restricciones 1.3.6 Notaciones, métodos y metodologías 1.3.7 Concepto de ciclo de vida del software	Lecturas	15 agosto – 19 agosto (semana 2)	Mod1- U3
Unidad 4. Actividades de desarrollo de la ingeniería de software 1.4 Actividades de desarrollo de la Ingeniería de Software 1.4.1 Obtención de requerimientos 1.4.2 Análisis de los requerimientos 1.4.3 Diseño del software 1.4.4 Desarrollo del software 1.4.5 Pruebas y aseguramiento de la calidad del software 1.4.6 Despliegue o puesta en marcha 1.4.7 Mantenimiento de software 1.4.7.1 Mantenimiento correctivo 1.4.7.2 Mantenimiento preventivo 1.4.7.3 Mantenimiento adaptativo 1.4.7.4 Mantenimiento perfectivo	Trabajo por proyecto (ABP) Resolución de ejercicios en grupos	22 agosto – 26 agosto (semana 3)	Mod1- U4 Ver obtención de requisitos para proyecto Qué busca un usuario al utilizar el cajero/sist matrícula (foro) fábula-A greatWarrior Ejercicio corto 1



<p>Unidad 5. Modelos de ciclos de vida de desarrollo de software</p> <p>1.5 Modelos de ciclos de vida de desarrollo de software</p> <p>1.5.1 Clasificación de los ciclos de vida del software</p> <p>1.5.1.1 Modelo en Cascada</p> <p>1.5.1.2 Modelo Incremental</p> <p>1.5.1.3 Modelo Iterativo</p> <p>1.5.1.4 Modelo Espiral</p>	Resolución de ejercicios en grupos	29 agosto – 2 septiembre (semana 4)	<p>Mod1- U5</p> <p>fábula(theSultan/TheFox)</p>
<p>Unidad 6. Metodologías de desarrollo de software</p> <p>1.6 Metodología de desarrollo de software</p> <p>1.6.1 Concepto de metodología de desarrollo de software</p> <p>1.6.2 Metodologías tradicionales</p> <p>1.6.2.1 Proceso Unificado (Rational Unified Process – RUP)</p> <p>1.6.2.1.1 Concepto</p> <p>1.6.2.1.2 Fases y disciplinas</p> <p>1.6.3 Metodologías ágiles</p> <p>1.6.3.1 XP (Extreme Programming)</p> <p>1.6.3.2 FDD (Feature Driven Development)</p> <p>1.6.3.3 Scrum</p> <p>1.6.3.4 Crystal Methodology</p> <p>1.6.3.5 Dynamic Systems Development Method (DSDM)</p> <p>1.6.3.6 Lean Software Development</p>	Resolución de ejercicios en grupos	5 septiembre 9 septiembre (semana 5)	<p>Mod1- U6</p> <p>ProyFinal-1-necesidad-problema</p> <p>Ejercicio corto 2</p> <p>Act- Desc problema (cuadro)</p>
<p>Unidad 7. Lenguaje de Modelado Unificado (UML)</p> <p>1.7 Lenguaje de Modelado Unificado (UML)</p>	Resolución de ejercicios en grupos	12septiembre 16septiembre (semana 6)	<p>ProyFinal-Revisión de tema de propuesta (idea gral qué busca atender)</p> <p>Seguimiento proyecto</p> <p>---Parcial 1--</p>



Módulo II: MODELADO DEL NEGOCIO

CONTENIDO	ESTRATEGIAS DIDÁCTICAS	FECHA	ACTIVIDADES
Unidad 1. Entendimiento del entorno 2.1 Entendimiento del entorno Unidad 2. Identificación de oportunidades de mejora 2.2 Identificación de oportunidades de mejora	Trabajo en equipo Lecturas	19septiembre 23septiembre (semana 7)	Mod2- U1 y U2 fábula(theSultan Act1- cuadro comp EO-Act- Desc problema- plantilla (cuadro)
Unidad 3. Técnicas de recopilación de información 2.3. Técnicas de recopilación de información 2.3.1.Introducción 2.3.1.1. Lluvia de ideas 2.3.1.2. Entrevistas 2.3.1.3. Presentaciones (storyboards) 2.3.1.4. Encuestas 2.3.1.5. Cuestionarios 2.3.1.6. Intercambio de roles 2.3.1.7. Revisión y análisis de documentos 2.3.1.8. Observaciones de campo 2.3.1.9. Prototipo 2.3.1.10.Focus group	Trabajo en equipo Lecturas Videos	26septiembre 29septiembre (semana 8)	Mod2- U3 Act- técnica recolección fábula- theHouses/TheLions Act-Describir requisitos- práctica Ejercicio corto3
Unidad 4. El modelo de negocio 2.4. El modelo de negocio 2.4.1.Concepto 2.4.2.Elementos del modelo de negocio 2.4.3.Modelo de Casos de Uso del Negocio 2.4.3.1. Actores 2.4.3.2. Casos de uso del negocio 2.4.3.2.1. Reglas del negocio 2.4.3.3. Diagrama de casos de uso del negocio 2.4.4.Modelo de objetos o análisis del negocio 2.4.4.1. Trabajadores del negocio (coworkers) 2.4.4.2. Entidades del negocio 2.4.4.3. Realización de casos de uso del negocio 2.4.4.3.1. Diagrama de actividades 2.4.4.4. Representación de un Modelo de Negocio - Caso Práctico.	Trabajo en equipo Lecturas	3 octubre 7 octubre (semana 9)	Mod2- U4 Act- modelo de CU negocio Act- Caso de uso –práctica



Módulo III: CONCEPCIÓN DEL SISTEMA

CONTENIDO	ESTRATEGIAS DIDÁCTICAS	FECHA	ACTIVIDADES
Unidad 1. Análisis del problema 3.1. Análisis del problema 3.1.1. Definición del problema 3.1.2. Descripción de stakeholders y usuarios 3.1.3. Necesidades 3.1.3.1. Derivación de las necesidades de usuario a partir del modelado del negocio. 3.1.4. Características 3.1.5. Requerimientos 3.1.5.1. Requerimientos funcionales 3.1.5.2. Requerimientos no funcionales 3.1.5.3. Restricciones de diseño 3.1.5.4. Aspectos de calidad de software	Lecturas Trabajo en equipo	10 de octubre 14 de octubre (semana 10) 17 de octubre 21 de octubre (semana 11)	Parcial 2 (módulo2) Mod3- U1 y U2 E1-Act- Análisis de plantilla del Documento Visión Documento Visión -Problema -Requisitos Act-RequeF&NF E3-Requisitos Funcionales y No Funcionales (Seguimiento de proyecto) Act-Matriz de trazabilidad

Módulo IV: ELABORACIÓN DE LA ESPECIFICACIÓN DE REQUERIMIENTOS

CONTENIDO	ESTRATEGIAS DIDÁCTICAS	FECHA	ACTIVIDADES
Unidad 1. Introducción a la especificación de requerimientos 4.1 Especificación de requerimientos 4.1.1 Concepto 4.1.2 Importancia	Lectura	24 de octubre 28 de octubre (semana12)	E3- Matriz de trazabilidad (Seguimiento de proyecto) Mod4- U1
Unidad 2. Modelo de casos de uso del sistema 4.2 Modelo de casos de uso del sistema 4.2.1 Actores 4.2.2 Casos de uso 4.2.3 Relaciones entre casos de uso 4.2.4 Diagrama de casos de uso 4.2.5 Especificación de casos de uso	Lectura	31 de octubre 4 de noviembre (semana13)	Mod4- U2 y U3 Act-Escenario de usuario
Unidad 3. Historias de usuario 4.3 Historias de usuarios 4.3.1 Concepto 4.3.2 Características 4.3.3 Formato	Trabajo en equipo		



4.3.4 Criterios de aceptación Unidad 4. Prototipos de interfaces gráficas 4.4 Prototipos de interfaces gráficas de usuario		7 -11 de noviembre (semana14)	E4- Lista casos de uso y escenario de proyecto Mod4- U5 ----Parcial 3--- E5- Casos de uso del proyecto
--	--	--------------------------------------	--

Módulo V: RIESGO EN EL DESARROLLO DEL SOFTWARE

CONTENIDO	ESTRATEGIAS DIDÁCTICAS	FECHA	ACTIVIDADES
Unidad 1. Riesgo en el desarrollo 5.1. Riesgo en el desarrollo 5.2. Significado de Riesgo 5.3. Importancia de su análisis e impacto 5.4. Clasificación de los riesgos 5.5. Mitigación de riesgos 5.6. Plan de contingencia	Lecturas	14 -18 de noviembre (semana15) 21 -25 de noviembre (semana16)	E6- Interfaces del proyecto Mod5- U1 E5-Entregable final



Proyecto final

El proyecto consistirá en la aplicación de los conceptos adquiridos para la solución de un caso práctico propuesto por el profesor o estudiante, de naturaleza real de alguna organización o contexto de nuestro medio.

Los alumnos trabajarán para crear su propio proyecto tomando en cuenta los temas vistos desde el inicio del



[Proyecto. Freepik License \(CCBY\)](#)

semestre. Considere su tema de proyecto para presentar los avances que se indiquen conforme se desarrollen los temas involucrados hasta su presentación al final

Los entregables corresponden a principales puntos del documento visión. Algunos son los siguientes:

- Descripción del problema y necesidades
- Descripción de requisitos funcionales y no funcionales
- Matriz de trazabilidad
- Lista casos de uso y escenario
- Casos de uso de la solución
- Prototipo de interfaces



Referencias



[Referencias. Freepik License \(CCO\)](#)

1. Project Management Institute, “Guía de los fundamentos para la dirección de proyecto (PMBOK: A Guide to the Project Management Body of Knowledge)”. Sexta Edición. 2017
2. Ludmila Rinaudo, Guillermo Pantaleo, “Ingeniería de Software”, Editorial Ink 2016.
3. Sebastián R. Gómez Palomo. y Eduardo A. Moraleda Gil, “Aproximación a la Ingeniería del software” Editorial universitaria Ramón Areces. 2da Edición. 2020.
4. Daniel Ramos, Raúl Noriega, José Rubén Láinez, Alicia Durango
5. Salvador Sánchez, Miguel Ángel Sicilia y Daniel Domínguez. Ingeniería del Software, un enfoque desde la guía SWEBOK. Editorial Alfaomega, México, 2012.
6. José Rubén Láinez Fuentes, 2da. edición, (2015).Desarrollo de Software ÁGIL: Extreme Programming y Scrum
7. I. Jacobson, G. Booch, J Rumbaugh, (2000). El proceso unificado de desarrollo de software.
8. J. Canós, P. Letelier, MC Penadés. (2003). Metodologías ágiles en el desarrollo de software. Universidad Politécnica de Valencia, academia.edu. (Citado por 450 artículos)
9. Arias Chaves, Michael. La ingeniería de requerimientos y su importancia en el desarrollo de proyectos de software.InterSedes: Revista de las Sedes Regionales, vol. VI, núm. 10, 2005, pp. 1-13. Universidad de Costa Rica. (Citado por 122 artículos relacionados)
10. UPV. Ejemplo de desarrollo software utilizando la metodología RUP: Desarrollo de un sistema para la gestión de artículos deportivos.. DOI: [//users.dsic.upv.es/asignaturas/facultad/lsi/ejemplorup/](https://users.dsic.upv.es/asignaturas/facultad/lsi/ejemplorup/) Accesado el 28 de febrero de 2011.



Perfil del docente



Erika Quintero-Rivas

Profesor tiempo parcial egresado de la FISC. Pertenece al Departamento de Ingeniería de software y ha impartido el curso de Ingeniería de software 1, Ingeniería de software 2 entre otros.

Valora mucho el esfuerzo de los estudiantes, por lo que se le invita a trabajar y esforzarse en cada actividad para obtener buen resultado al final del curso. Le gusta que sus estudiantes estén atentos a la clase, sigan indicaciones, sean creativos, tengan su pensamiento crítico, para el desarrollo de lo que se realice en el curso.