



**PROGRAMA DEL CURSO  
IF7100 - INGENIERÍA DE SOFTWARE  
I CICLO 2020**

## **1. Datos Generales**

- Sigla: IF7100
- Nombre: Ingeniería de Software
- Tipo de curso: Teórico - práctico
- Créditos: 4
- Horas lectivas: 6
- Requisitos: IF6100 Análisis y Diseño de Sistemas
- Correquisitos: Ninguno
- Ubicación en el plan de estudio: VII Ciclo
- Suficiencia: No
- Tutoría: No
- Modalidad: Regular - bajo virtual

## **2. Descripción**

Este curso proporciona al y la estudiante conocimientos sobre conceptos, métodos, metodologías y herramientas de la ingeniería de software empleadas en el diseño, desarrollo, pruebas, operación y mantenimiento de productos de software. El curso pone especial énfasis en aspectos de modelado arquitectónico del software, herramientas CASE de soporte al proceso y en la gestión de la calidad del producto a través de pruebas exhaustivas de distinta índole. Adicionalmente, el curso le proporciona al y la estudiante un espacio de aplicación a través del desarrollo de un proyecto de software.



### 3. Objetivo General

Adquirir una visión integral de los procesos de la ingeniería de software así como desarrollar sus conocimientos y habilidades a nivel de buenas prácticas de la ingeniería de software para producir aplicaciones de software de alta calidad.

### 4. Objetivos Específicos

Al finalizar el curso el o la estudiante estará en capacidad de:

- Reconocer el carácter transdisciplinario de la ingeniería de software.
- Proponer el modelo de diseño de los productos de software que construya.
- Proponer una arquitectura de software que atienda los atributos calidad de la misma en función del contexto del proyecto de software.
- Realizar la implementación del producto de software en función del diseño y la arquitectura que haya formulado.
- Vivenciar el proceso integración de productos intermedios y de módulos en un proyecto de desarrollo de software.
- Diseñar estrategias de pruebas pertinentes al producto de software que se esté desarrollando.
- Administrar los procesos de implantación de las aplicaciones que desarrolle.
- Gestionar el proceso de administración de configuración del software.
- Emplear eficazmente herramientas CASE modernas para soportar los procesos de desarrollo de software de manera más estructurada y sistematizada.
- Adoptar una actitud crítica ante la diversidad de temáticas asociadas con la ingeniería de software.

### 5. Contenidos

1. INTRODUCCIÓN INGENIERÍA DE SOFTWARE (1/2 SEM)
2. REQUERIMIENTOS DE SOFTWARE (2 SEM)



- 2.1 Fundamentos de requerimientos
- 2.2 Procesos de requerimientos
- 2.3 Obtención de requerimientos
- 2.4 Análisis de requerimientos
- 2.5 Especificación de requerimientos
- 2.6 Validación de requerimientos
- 3. DOCUMENTACIÓN Y MODELADO DE PROCESOS (1/2 SEM)**
  - 3.1 Conceptos: proceso, características de los procesos
  - 3.2 Tipos de procesos: operativos, de apoyo y de gestión
  - 3.3 Formas alternativas para documentar los procesos
  - 3.4 Relación de los procesos con la ingeniería de software
  - 3.5 Técnicas para la definición y diseño de procesos: diagramas de actividad UML y diagramas de flujos de proceso (flujogramas)
  - 3.6 Notación y simbología utilizada para modelado de procesos
  - 3.7 Herramientas CASE para el modelado de procesos
- 4. DISEÑO DE SOFTWARE (2 SEM)**
  - 4.1 Fundamentos de diseño
  - 4.2 Análisis y evaluación de la calidad del diseño
  - 4.3 Notación de diseño software
  - 4.4 Estrategias y métodos de diseño
- 5. ARQUITECTURA DEL SOFTWARE (2 SEM)**
  - 5.1 Definiciones: arquitectura versus diseño, arquitectura de software, elementos-relaciones y propiedades de la arquitectura del software
  - 5.2 Razones por las cuales es necesaria la arquitectura de software
  - 5.3 Atributos de calidad de la arquitectura de software: performance, exactitud, seguridad, mantenibilidad, portabilidad
  - 5.4 Estilos arquitectónicos: estilo módulo, estilo componente y conector, estilo de aloación (despliegue)



- 5.5 Diagrama de Contexto
- 5.6 Vistas arquitectónicas 4+1
- 5.7 Modelado arquitectónico del software
- 5.8 Arquitectura Física: N-Tier Architecture: Tipos ( Cliente-Servidor, 3-Tier )
- 5.9 Beneficios Arquitectura Física: Escalabilidad, seguridad y tolerancia a fallas
- 5.10 Arquitectura Lógica: N-Layer Architecture: Tipos ( 3-layers, n-layers )
- 5.11 Beneficios: mantenibilidad, reutilización, distribución del trabajo, flexibilidad, robustez, entre otros
- 5.12 Capas lógicas de implementación (aplicación, negocios y servicio de datos)
- 5.13 Subsistemas
- 6. PATRONES DE DISEÑO (1 SEM)**
  - 6.1 Descripción de los patrones
  - 6.2 Clasificación
  - 6.3 Creacionales, Estructurales, De Partición, De Comportamiento
- 7. CONSTRUCCIÓN DE SOFTWARE (1 SEM)**
  - 7.1 Fundamentos de construcción
  - 7.2 Gestión de construcción
- 8. GESTIÓN DE CALIDAD DEL PRODUCTO (1/2 SEM)**
  - 8.1 Concepto de Calidad
  - 8.2 Prevención versus Detección
  - 8.3 Clientes de la calidad: interno, externo y oculto
  - 8.4 Calidad del proyecto versus calidad del producto
  - 8.5 Verificación versus Validación
  - 8.6 Aseguramiento y Control de la Calidad del Software
- 9. PRUEBAS DEL SOFTWARE (2 SEM)**
  - 9.1 Propósitos de la disciplina de pruebas



- 9.2 Actividades de la evaluación de la calidad del software: pruebas funcionales (caja blanca), pruebas estructurales (caja negra), análisis estático y análisis dinámico
  - 9.3 Flujo de trabajo general de la disciplina: Identificación del objetivo de las pruebas, selección de las entradas, definición de salidas esperadas, configuración del ambiente de ejecución, ejecución del programa, análisis de los resultados.
  - 9.4 Implementación de Pruebas Pruebas de Unidad Pruebas de Integración Pruebas de Validación Pruebas del Sistema: funcionalidad, seguridad, robustez, cargado, estabilidad, de estrés, performance y fiabilidad Pruebas de Aceptación Pruebas Funcionales: Casos de prueba, matriz de cobertura de pruebas, matriz de casos de prueba, procedimientos de prueba, escenarios de prueba, scripts de prueba
  - 9.5 Herramientas CASE para efectuar pruebas del software
- 10. IMPLANTACIÓN (1 SEM)**
- 10.1 Propósitos de la disciplina
  - 10.2 Estrategias para la implantación del producto
  - 10.3 Migración de datos
  - 10.4 Capacitación de usuarios
- 11. MANTENIMIENTO DEL SOFTWARE (1 SEM)**
- 11.1 Propósitos de la disciplina
  - 11.2 Elementos claves del mantenimiento: usuario, ambiente, ambiente operativo, ambiente organizacional, proceso de mantenimiento, producto de software, personal de mantenimiento.
  - 11.3 Cambios del software: correctivos, adaptivos, perfectivos, preventivos
  - 11.4 Procesos de mantenimiento
  - 11.5 Técnicas para mantenimiento
- 12. GESTIÓN DE INGENIERÍA DE SOFTWARE (1 SEM)**
- 12.1 Definición inicial y alcances
  - 12.2 Planificación proyectos



12.3 Revisión y evaluación

12.4 Cierre proyectos

### 13. ADMINISTRACIÓN DE LA CONFIGURACIÓN DEL SOFTWARE (ACS) (1 SEM)

13.1 Definición de la Administración de la Configuración del Software

13.2 Actividades de la disciplina: identificación, almacenamiento, control del cambio y reporte del estado

13.3 Mejores prácticas de la disciplina.

13.4 Conceptos: metadata, línea base, roles y responsabilidades, ítem de configuración, árbol de versiones (versión y revisión), repositorio

13.5 El proceso de la Administración de Configuración de Software

13.6 Control de versiones, Control del Cambio, Reporte Estado, Auditoría de Configuración

13.7 Herramientas CASE para la ACS

### 14. METODOLOGÍAS DE DESARROLLO DEL SOFTWARE (1 SEM)

14.1 Concepto de metodología

14.2 Modelo de procesos de desarrollo empleados en las metodologías (cascada, iterativo-incremental)

14.3 Aspectos fundamentales de las metodologías: fases, disciplinas e hitos (milestones), flujos de trabajo, artefactos y productos de entrada y salida de las fases, roles y responsabilidades

14.4 Metodología estructurada: Proceso Unificado de Rational

14.5 Metodologías ágiles: Extreme Programming y SCRUM

## 6. Metodología

El curso presenta un eje de desarrollo teórico-práctico.

El profesorado desarrolla las clases soportado en diapositivas, modelos UML, plantillas de artefactos y código fuente. Algunas clases se realizarán en el laboratorio y estarán acompañadas de ejercicios prácticos sobre las temáticas siendo estudiadas.



Los materiales didácticos y el programa del curso estarán disponibles en el entorno Mediación Virtual. Para registrarse en el curso deben visitar la dirección electrónica <https://mediacionvirtual.ucr.ac.cr>. La persona docente del curso les proveerá la clave de acceso.

En el curso se promueve, en algunas semanas, la realización de lecturas previas a la clase.

Los y las estudiantes desarrollan un proyecto de software. El mismo lo gestionarán mediante el marco de trabajo Proceso Unificado o metodologías ágiles. Los requerimientos técnicos y las políticas de evaluación del proyecto estarán consignadas en un documento de proyecto que se entregará de forma oportuna. El proyecto tendrá puntos de control en los cuales los y las estudiantes entregarán los elementos o artefactos definidos en el enunciado del proyecto.

Además, los y las estudiantes realizarán exposiciones acerca de temáticas de interés que complementen el desarrollo del curso. Los temas de exposición serán entregados en la segunda semana de clase, así como las fechas y los aspectos por ser evaluados. Algunas exposiciones serán de carácter práctico (herramientas CASE). Los y las estudiantes serán responsables de instalar y poner en funcionamiento versiones de prueba así como mostrar ejemplos básicos de uso sobre las mismas.

Los y las estudiantes resolverán casos de estudio, en clase o extra-clase, en donde se plantean escenarios asociados con procesos de la ingeniería de software.

## 7. Evaluación

DESCRIPCIÓN	PORCENTAJE
I Parcial	20 %
II Parcial	20 %
Evaluaciones cortas, casos de estudios y tareas	10 %
Proyecto	40 %
Exposiciones	10 %

### 7.1. Consideraciones sobre la evaluación

- Según lo establecido en las resoluciones VD-R-8458-2009 y VD-R-9374-2016, se utilizará el entorno virtual de aprendizaje institucional Mediación Virtual (<https://mediacionvirtual.ucr.ac.cr>). El mismo se empleará para la entrega del programa del curso, material, enunciados de evaluaciones, entre otros, por



parte del profesorado. En el caso del estudiantado, para el envío de entregables y/o realización de evaluaciones asociadas al curso.

- Según lo establecido en la resolución R-2664-2012, que establece el correo institucional con el dominio @ucr.ac.cr como la herramienta oficial para las comunicaciones de toda la comunidad universitaria. Se utilizará el correo institucional como medio oficial de comunicación entre docentes y estudiantes, por lo cual el estudiantado deberá tenerlo activo y revisarlo continuamente.
- Durante las evaluaciones, el uso de teléfonos celulares, tabletas o cualquier otro dispositivo de comunicación está totalmente prohibido - a excepción que el profesor indique lo contrario - dentro y fuera del aula mientras el estudiante no haya hecho entrega de su evaluación. Dichos dispositivos deberán permanecer apagados y guardados en su bolso o bulto.
- Los criterios de calificación de cada evaluación serán especificados en el enunciado de la misma.
- Toda evaluación será comunicada al estudiantado del curso al menos 5 días hábiles antes de realizarse, a excepción de las pruebas cortas o “quices”, de acuerdo con lo especificado en los artículos 15 y 18 del Reglamento de Régimen Académico Estudiantil.
- En caso de ausencia a alguna evaluación, se procederá según lo establecido en el Artículo 24 del Reglamento de Régimen Académico Estudiantil.
- Ante la detección de una posible copia o plagio, total o parcial, en cualquier evaluación, se procederá de acuerdo con lo establecido en el Reglamento de Orden y Disciplina Estudiantil.
- Como parte de las lecturas de apoyo a los temas que se desarrollarán en clase, se utilizará al menos dos lecturas en idioma inglés. El objetivo principal de este aspecto es impulsar la comprensión de lectura. Debido a que hay estudiantes con diferente nivel lingüístico, los reportes y presentaciones para revisar el material leído se deben realizar en idioma español.
- Las fechas del cronograma están sujetas a cambio dependiendo del avance en los contenidos.





## 8. Docentes del curso

GRUPO	AULA	DOCENTE	HORARIO	CONSULTA
<b>SEDE DEL SUR</b>				
01	K06 J08	Lic Eddie Bermúdez Carmona eddie.bermudez@ucr.ac.cr	K 09 a 12:00 J 13 a 16:00	K 13 a 16:00
<b>SEDE DEL ATLÁNTICO, RECINTO DE GUÁPILES</b>				
01	Lab 3	Mgtr Kenneth Pereira Hernández kenneth.pereira@ucr.ac.cr	S 08 a 11:50 S 13 a 13:50	J 19 a 20:50 S 14 a 15:00
<b>SEDE DEL ATLÁNTICO, RECINTO DE PARAÍSO</b>				
01	Lab14 A18	MGP Álvaro Mena Monge alvaro.mena@ucr.ac.cr	L 14 a 16:50 J 13 a 15:50	M 08 a 11:00
<b>SEDE DEL ATLÁNTICO, RECINTO DE TURRIALBA</b>				
01	07	MSc Juan José Quesada Sánchez juanjose.quesada@ucr.ac.cr	K 19 a 20:50 S 08 a 11:50	S 13 a 15:50
<b>SEDE DEL CARIBE, RECINTO DE LIMÓN</b>				
01	06 LAB 2	Lic Delia Smith Paul delia.smith@ucr.ac.cr	K 09 a 11:50 J 09 a 11:50	K 13 a 15:50
<b>SEDE DEL CARIBE, RECINTO DE SIQUIRRES</b>				
01	3	Lic Delia Smith Paul delia.smith@ucr.ac.cr	L 13 a 15:50 M 09 a 11:50	M 13 a 15:50
<b>SEDE DE GUANACASTE, RECINTO DE LIBERIA</b>				
01	Lab.	Lic Iván Alonso Chavarría Cubero alonso182@hotmail.com	M 07 a 11:50	M 14 a 15:50
<b>SEDE DE OCCIDENTE, RECINTO DE GRECIA</b>				
02	Lab C	MAP Verny Fernández Castro verny.fernandez@ucr.ac.cr	L 17 a 19:50 V 17 a 19:50	L 20 a 21:00 V 20 a 21:00
<b>SEDE DE OCCIDENTE, RECINTO DE SAN RAMÓN</b>				
01	201	MCI Juan C. Miranda Sánchez	L 17 a 20:50	L 16 a 16:50



GRUPO	AULA	DOCENTE	HORARIO	CONSULTA
		juancarlos.miranda@ucr.ac.cr	K 07 a 8:50	K 9 a 10:50 Virtual
<b>SEDE DEL PACÍFICO</b>				
01	18	MATI Wendy Vargas Hernandez wendy.vargas.h@ucr.ac.cr	V 14 a 16:50 J 14 a 16:50	V 08 a 11

## 9. Cronograma

SEM	FECHA	TEMA O ACTIVIDAD
01	09 - 14 MAR	Entrega y Lectura carta del estudiante Introducción ingeniería de software
02	16 - 21 MAR	Documentación y modelado de procesos Metodologías de desarrollo del software Lectura 1: "Agile Scrum deep dive" de (Bibik, 2018, p. 15-29)
03	23 - 28 MAR	Requerimientos de software
04	30 MAR- 04 ABR	Requerimientos de software
05	06 - 11 ABR	Semana santa
06	13 - 18 ABR	Diseño de software
07	20 - 25 ABR	Semana U. Diseño de software
08	27 ABR - 02 MAY	Arquitectura del software Lectura 2: Capítulo 1 - "Introduction" Capítulo 4. "Architecture Characteristics Define" Capítulo 5. "Identifying Architectural Characteristics" de (Ford y Richards, 2020)
09	04 - 9 MAY	Examen Parcial I Construcción de software Lectura 3: Capítulo 1 - "Clean Code" de (Martin, 2009)



SEM	FECHA	TEMA O ACTIVIDAD
10	11 - 16 MAY	Patrones de diseño
11	18 - 23 MAY	Gestión de calidad del producto Lectura 4: Capítulo 2 "Quality Assurance Framework" de (Lewis, 2017)
12	25 - 30 MAY	Pruebas del software Lectura 5: Capítulo 3: "Overview of Testing Techniques" de (Lewis, 2017)
13	01 - 06 JUN	Implantación
14	08 - 13 JUN	Mantenimiento del software
15	15 - 20 JUN	Gestión de ingeniería de software
16	22 - 27 JUN	Administración de la configuración del software
17	29JUN - 04 JUL	Examen Parcial II

## 10. Acreditación

La Carrera Bachillerato en Informática Empresarial está acreditada por el Sistema Nacional de Acreditación de la Educación Superior (SINAES) en el periodo comprendido entre el 10 de diciembre del 2019 al 3 de diciembre del 2023 (ACUERDO-CNA-400-2019) en las siguientes Sedes y Recintos:

- Sede Regional del Atlántico, Recinto de Guápiles
- Sede Regional del Atlántico, Recinto de Paraíso
- Sede Regional del Atlántico, Recinto de Turrialba
- Sede Regional del Caribe, Recinto de Limón
- Sede Regional de Guanacaste, Recinto de Liberia
- Sede Regional de Occidente, Recinto de Grecia
- Sede Regional de Occidente, Recinto de San Ramón
- Sede Regional del Pacífico





## Referencias obligatorias

- Bibik, I. (2018). How to kill the scrum monster: Quick start to agile scrum methodology and the scrum master role. Apress. Descargado de <https://doi-org.ezproxy.sibdi.ucr.ac.cr/10.1007/978-1-4842-3691-8>
- IEEE. (2014). *Swebok-guide to the software engineering body of knowledge* (Inf. Téc.).
- Joshi, B. (2016). *Beginning solid principles and design patterns for asp. net developers*. Springer. Descargado de <https://doi-org.ezproxy.sibdi.ucr.ac.cr/10.1007/978-1-4842-1848-8>
- Pressman, R. (2010). *Ingeniería del software: Un enfoque práctico*. Mc Graw Hill.
- SEI. (2019). Publications. *SEI Publications*. Carnegie Mellon University.
- Sommerville, I. (2011). *Ingeniería del software* (9.<sup>a</sup> ed.). Pearson Education.

## Referencias secundarias

- Brown, S. (2019). Software architecture for developers-volume 2, visualise, document and explore your software architecture. *Victoria, British Columbia, Canada: Leanpub*.
- de la Torre Llorente, C., Zorrilla Castro, U., Ramos Barroso, M. A., y Calvarro Nelson, J. (2010). *Guía de arquitectura n-capas orientada al dominio con .net*. Microsoft Ibérica.
- Ford, N., y Richards, M. (2020). *Fundamentals of software architecture*. O'Reilly Media, Inc.
- Kazman, R., y Cervantes, H. (2016). *Designing software architectures: A practical approach*. Pearson Education.
- Lewis, W. E. (2017). *Software testing and continuous quality improvement* (3.<sup>a</sup> ed.). Auerbach Publications.
- Martin, R. C. (2009). *Clean code: a handbook of agile software craftsmanship*. Pearson Education.
- Paradkar, S. (2017). *Mastering non-functional requirements*. Packt Publishing Ltd.
- Pylayeva, D. (2017). Introduction to devops with chocolate, lego and scrum game. Apress. Descargado de <https://doi-org.ezproxy.sibdi.ucr.ac.cr/10.1007/978-1-4842-2565-3>