PROGRAMA DEL CURSO
IF7100 - INGENIERÍA DE SOFTWARE
I CICLO 2024

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

1.1. Modalidad

Modalidad

Sede	Recinto	Modalidad
Atlántico	Guápiles	Presencial
	Paraíso	Presencial
	Turrialba	Presencial
Caribe	Limón	Presencial
	Siquirres	Presencial
Guanacaste	Liberia	Presencial
Occidente	Grecia	Bimodal
	San Ramón	Bimodal

Continúa de la página previa

Sede	Recinto	Modalidad
Pacífico		Alto virtual
Sur		Presencial

2. Descripción

Este curso proporciona a la persona 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 a la persona 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 alocación (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 (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, código fuente y mediante ejercicios prácticos sobre las temáticas estudiadas.

Los materiales didácticos y el programa del curso estarán disponibles en el entorno Mediación Virtual y/o TEAMS. Para registrarse en el curso deben visitar la dirección electrónica https://mv1.mediacionvirtual.ucr.ac.cr.

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 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 investigaciones acerca de temáticas de interés que complementen el desarrollo del curso. Los temas de investigación serán entregados en la segunda semana de clase, así como las fechas y los aspectos por ser evaluados. Algunas investigaciones serán de carácter práctico (herramientas CASE) por lo que las y los 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.

En el caso de los grupos que se impartirán en la modalidad, bimodal y alto virtual, las sesiones de clase se desarrollarán de forma sincrónica y asincrónica según la estrategia didáctica que se seleccione para hacer el abordarje de los contenidos del curso. Con respecto a las actividades asincrónicas, entre ellas casos de estudio y

proyectos, el profesorado dará el seguimiento correspondiente vía correo institucional, vía mensajería instantánea o mediante videollamadas u otras herramientas similares.

7. Evaluación

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

7.1. Consideraciones sobre la evaluación

- Según lo establecido en las resoluciones VD-R-8458-2009 y VD-11502-2020, se utilizará el entorno virtual de aprendizaje institucional Mediación Virtual (https://mv1.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, se 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.

- Cuando las evaluaciones sean en la modalidad presencial, durante las evaluaciones, el uso de teléfonos celulares, tabletas o cualquier otro dispositivo de comunicación está totalmente prohibido - a excepción que la persona docente indique lo contrario - dentro y fuera del aula mientras la persona 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. En caso de utilizar inteligencia artificial generativa se debe proceder a referenciarla correctamente.
- 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	DOCENTE	HORARIO	CONSULTA
	SEDE DEL ATLÁNTICO, RECINTO DE GUÁPILES		
31	Lic. Geber A. Guillén Berrocal geber.guillen@ucr.ac.cr	K 17 a 19:50 V 17 a 19:50	S 16 a 19
	SEDE DEL ATLÁNTICO, RECINTO DE PARAÍSO		
21	MSc. Leonardo Camacho Navarro jose.camacho@ucr.ac.cr	K 17 a 19:50 J 17 a 19:50	M 18 a 21
	SEDE DEL ATLÁNTICO, RECINTO DE TURRIALBA		
01	Mag. Juan José Quesada Sánchez juanjose.quesada@ucr.ac.cr	K 19 a 20:50 S 08 a 11:50	S 13 a 17

GRUPO	DOCENTE	HORARIO	CONSULTA
	SEDE DEL CARIBE, RECINTO DE LIMÓN		
01	Msc. Carlos Morales Castro carlos.moralescastro@ucr.ac.cr	S 09 a 11:50 S 13 a 15:50	S 16 a 19
	SEDE DEL CARIBE, AULA DE SIQUIRRES		
01	Lic. Delia Smith Paul delia.smith@ucr.ac.cr	L 09 a 11:50 M 09 a 11:50	K 13 a 16
	SEDE DEL GUANACASTE, RECINTO DE LIBERIA		
01	Lic Iván A. Chavarría Cubero ivan.chavarriacubero@ucr.ac.cr	M 13 a 16:50 V 13 a 14:50	M 08 a 11
	SEDE DE OCCIDENTE, RECINTO DE GRECIA		
02	Mag. Verny Fernandez Castro verny.fernandez@ucr.ac.cr	J 17 a 19:50 V 17 a 19:50	L 17 a 20
	SEDE DE OCCIDENTE, RECINTO DE SAN RAMÓN		
10	Mag Juan Carlos Miranda juancarlos.miranda@ucr.ac.cr	L 17 a 20:50 K 07 a 8:50	L 16 a 17 K 09 a 11
	SEDE DEL PACÍFICO		
01	Mag. Raquel Porras Soto raquel.porrassoto@ucr.ac.cr	L 13 a 15:50 J 13 a 15:50	M 08 a 11
	SEDE DEL SUR		
01	Msc. María José Peralta Varela maria.peraltavarela@ucr.ac.cr	K 18 a 19:50 S 08 a 11:50	S 13 a 16

9. Cronograma

SEM	FECHA	TEMA O ACTIVIDAD
01	11 - 16 MAR	Entrega y lectura carta del estudiante Presentación y discusión del video "¿Querés conocer acerca del hostigamiento sexual y la reforma al Reglamento de la UCR en su contra?" del Centro de Investigación de Estudios de la Mujer, UCR (2021). Introducción ingeniería de software
02	18 - 23 MAR	Documentación y modelado de procesos Metodologías de desarrollo del software Lectura 1: "From waterfall to agile" , "Overview of Agile Methodologies" y "Agile Scrum deep dive" de (Bibik, 2018, p. 1-29)
03	25 - 30 MAR	Semana Santa
04	01 - 06 ABR	Requerimientos de software
05	08 - 13 ABRR	Requerimientos de software
06	15 - 20 ABR	Diseño de software
07	22 - 27 ABR	Diseño de software **Semana universitaria**
08	29 ABR - 04 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	06 - 11 MAY	Examen Parcial I Construcción de software Lectura 3: Capítulo 1 - "Basics of clean C++" y "Modularization"de (Roth, 2021)
10	13 - 18 MAY	Patrones de diseño

SEM	FECHA	TEMA O ACTIVIDAD
11	20 - 25 MAY	Gestión de calidad del producto Lectura 4: Capítulo 2 "Quality Assurance Framework" de (Lewis, 2017)
12	27 MAY - 01 JUN	Pruebas del software Lectura 5: Capítulo 3: "Overview of Testing Techniques" de (Lewis, 2017)
13	03 - 08 JUN	Implantación
14	10 - 15 JUN	Mantenimiento del software
15	17 - 22 JUN	Gestión de ingeniería de software
16	24 - 29 JUN	Administración de la configuración del software Defensa final del proyecto Examen Parcial II
17	01 - 06 JUL	Examen Parcial II
18	08 - 13 JUL	Exámen de ampliación

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

Baumgartner, M., Klonk, M., Mastnak, C., Pichler, H., Seidl, R., y Tanczos, S. (2021). *Agile testing: The agile way to quality.* Springer. (Acceso en bases de datos del SIBDI)

Brown, S. (2019). *Software architecture for developers:visualise, document and explore your software architecture* (Vol. 2). Leanpub.

Centro de Investigación de Estudios de la Mujer, UCR. (2021). *¿Querés conocer acerca del hostigamiento sexual y la reforma al reglamento de la ucr en su contra?* (`https://youtu.be/dzKMV8FNpks`)

Ford, N., y Richards, M. (2020). *Fundamentals of software architecture.* O'Reilly Media, Inc. (Biblioteca del Recinto de Paraíso)

Pressman, R. S., y Maxin, B. R. (2020). *Software engineering: A practitioner's approach* (9.ª ed.). Mc Graw Hill.

Roth, S. (2021). *Clean c++20.* Apress. (Acceso en bases de datos del SIBDI)

SEI. (2019). Publications. *SEI Publications. Carnegie Melon University.*

Sommerville, I. (2019). *Software engineering* (10.ª ed.). Pearson Education Limited.

Referencias secundarias

Bibik, I. (2018). How to kill the scrum monster: Quick start to agile scrum methodology and the scrum master role. Apress. (Acceso en bases de datos del SIBDI)

Bourque, P., y Fairley, R. E. (2014). *Swebok-guide to the software engineering body of knowledge* (3.ª ed.). IEEE Computer Society. (`http://www.swebok.org`)

Bryant, D., y Marín-Pérez, A. (2019). *Continuous delivery in java: essential tools and best practices for deploying code to production*. O'Reilly Media.

Davis, J., y Daniels, R. (2016). *Effective devops: building a culture of collaboration, affinity, and tooling at scale*. O'Reilly Media, Inc.

de la Torre Llorente, C., Zorrilla Castro, U., Ramos Barroso, M. A., y Calvarrro Nelson, J. (2010). *Guuía de arquitectura n-capas orientada al dominio con .net*. Microsoft Ibérica.

Ford, N., Richards, M., y Sadalage, D. Z., Pramod. (2021). *Software architecture: The hard parts*. O'Really.

Joshi, B. (2016). *Beginning solid principles and design patterns for asp. net developers*. Springer. (Acceso en bases de datos del SIBDI)

Kazman, R., y Cervantes, H. (2016). *Designing software architectures: A practical approach*. Pearson Education.

Khorikov, V. (2020). *Unit testing: Principles, practices and patterns*. Manning Publications.

Lewis, W. E. (2017). *Software testing and continuous quality improvement* (3.ª 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. (Acceso en bases de datos del SIBDI)

Ritter, F. (2014). *Foundations for designing user-centered systems*. Springer.

Rubin, K. S. (2012). *Essential scrum: A practical guide to the most popular agile process*. Addison-Wesley.

Stull, E. (2018). *Ux fundamentals for non-ux professionals: User experience principles for managers, writers, designers, and developers*. Apress.

Troelsen, A. W., y Japikse, P. (2022). *Pro c# 10 with. net 6: Foundational principles and practices in programming*. Springer.

Whitesell, S., Richardson, R., y Groves, M. (2022). *Pro microservices in .net 6*. Springer.

Winters, T., Manshreck, T., y Hyrum, W. (2022). *Ingeniería de software en google*. Marcombo.