

PROGRAMA CURSO: **IF7100 INGENIERÍA DEL SOFTWARE**  
I Semestre, 2018

**Datos Generales**

---

**Sigla:** IF7100  
**Nombre del curso:** Ingeniería del Software  
**Tipo de curso:** Teórico – Práctico  
**Número de créditos:** 4  
**Número de horas semanales presenciales:** 6  
**Requisitos:** IF-6100 Análisis y Diseño de Sistemas  
**Co-requisitos:** Ninguno  
**Ubicación en el plan de estudio:** VII Ciclo  
**Tutoría:** (NO)  
**Suficiencia:** (NO)  
**Modalidad:** (Presencial, bajo virtual)

**Horario del curso:**

M 8:00 a 11:50
J 13:00 a 14:50

**Datos del Profesor**

---

**Nombre:** Verny Fernández Castro  
**Correo Electrónico:** [verny.fernandez@ucrsi.info](mailto:verny.fernandez@ucrsi.info)

**Horas consulta presencial:** Días de Clase, reuniones de seguimiento de proyectos, o a solicitud del estudiante en horario de consulta virtual.

**Horas consulta Virtual:** Jueves 8:00 a.m a 11:00 am.

**Medio oficial para consulta virtual:** Hangouts [verny.fernandez@ucrsi.info](mailto:verny.fernandez@ucrsi.info)

**1. Descripción del curso**

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.

---

## Objetivos

### 2. Objetivo General

El curso le permitirá al y la estudiante:

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.

---

### 3. 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

---

## 4. Contenidos

### 1. INTRODUCCIÓN INGENIERÍA DE SOFTWARE

### 2. GESTIÓN DE INGENIERIA DE SOFTWARE

- 2.1 Definición inicial y alcances
- 2.2 Planificación proyectos
- 2.3 Revisión y evaluación
- 2.4 Cierre proyectos

3. REQUERIMIENTOS DE SOFTWARE
  - 3.1 Fundamentos de requerimientos
  - 3.2 Procesos de requerimientos
  - 3.3 Obtención de requerimientos
  - 3.4 Análisis de requerimientos
  - 3.5 Especificación de requerimientos
  - 3.6 Validación de requerimientos
  
4. DOCUMENTACIÓN Y MODELADO DE PROCESOS
  - 4.1 Conceptos: proceso, características de los procesos
  - 4.2 Tipos de procesos: operativos, de apoyo y de gestión
  - 4.3 Formas alternativas para documentar los procesos
  - 4.4 Relación de los procesos con la ingeniería de software
  - 4.5 Técnicas para la definición y diseño de procesos: diagramas de actividad UML y diagramas de flujos de proceso (flujogramas)
  - 4.6 Notación y simbología utilizada para modelado de procesos
  - 4.7 Herramientas CASE para el modelado de procesos
  
5. DISEÑO DE SOFTWARE
  - 5.1 Fundamentos de diseño
  - 5.2 Análisis y evaluación de la calidad del diseño
  - 5.3 Notación de diseño software
  - 5.4 Estrategias y métodos de diseño
  
6. ARQUITECTURA DEL SOFTWARE
  - 6.1 Definiciones: arquitectura versus diseño, arquitectura de software, elementos-relaciones y propiedades de la arquitectura del software
  - 6.2 Razones por las cuales es necesaria la arquitectura de software
  - 6.3 Atributos de calidad de la arquitectura de software: performance, exactitud, seguridad, mantenibilidad, portabilidad
  - 6.4 Estilos arquitectónicos: estilo módulo, estilo componente y conector, estilo de alocação (despliegue)
  - 6.5 Diagrama de Contexto
  - 6.6 Vistas arquitectónicas 4+1
  - 6.7 Modelado arquitectónico del software
  - 6.8 Arquitectura Física: N-Tier Architecture: Tipos ( Cliente-Servidor, 3-Tier )
  - 6.9 Beneficios Arquitectura Física: Escalabilidad, seguridad y tolerancia a fallas
  - 6.10 Arquitectura Lógica: N-Layer Architecture: Tipos ( 3-layers, n-layers )
  - 6.11 Beneficios: mantenibilidad, reutilización, distribución del trabajo, flexibilidad, robustez, entre otros
  - 6.12 Capas lógicas de implementación (aplicación, negocios y servicio de datos)
  - 6.13 Subsistemas

7. PATRONES DE DISEÑO
  - 7.1 Descripción de los patrones
  - 7.2 Clasificación
  - 7.3 Creacionales, Estructurales, De Partición, De Comportamiento
  
8. CONSTRUCCIÓN DE SOFTWARE
  - 8.1 Fundamentos de construcción
  - 8.28.2 Gestión de construcción
  
9. GESTIÓN DE CALIDAD DEL PRODUCTO
  - 9.1 Concepto de Calidad
  - 9.2 Prevención versus Detección
  - 9.3 Clientes de la calidad: interno, externo y oculto
  - 9.4 Calidad del proyecto versus calidad del producto
  - 9.5 Verificación versus Validación
  - 9.6 Aseguramiento y Control de la Calidad del Software
  
10. PRUEBAS DEL SOFTWARE
  - 10.1 Propósitos de la disciplina de pruebas
  - 10.2 Actividades de la evaluación de la calidad del software: pruebas funcionales (caja blanca), pruebas estructurales (caja negra), análisis estático y análisis dinámico
  - 10.3 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.
  - 10.4 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
  - 10.5 Herramientas CASE para efectuar pruebas del software
  
11. IMPLANTACIÓN
  - 11.1 Propósitos de la disciplina
  - 11.2 Estrategias para la implantación del producto
  - 11.3 Migración de datos
  - 11.4 Capacitación de usuarios
  
12. MANTENIMIENTO DEL SOFTWARE
  - 12.1 Propósitos de la disciplina

- 12.2 Elementos claves del mantenimiento: usuario, ambiente, ambiente operativo, ambiente organizacional, proceso de mantenimiento, producto de software, personal de mantenimiento.
  - 12.3 Cambios del software: correctivos, adaptivos, perfectivos, preventivos
  - 12.4 Procesos de mantenimiento
  - 12.5 Técnicas para mantenimiento
13. ADMINISTRACIÓN DE LA CONFIGURACIÓN DEL SOFTWARE (ACS)
- 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
- 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

---

## 5. Metodología

El curso presenta un eje de desarrollo teórico-práctico.

El profesor 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 estarán disponibles en un grupo virtual creado para el curso. Visitar y registrarse en: dirección URL

Los y las estudiantes realizan lecturas semanales y presentan resúmenes. De esta forma se pretende que durante la clase los y las estudiantes tengan conocimiento del tema para poder participar. Los resúmenes serán de carácter individual, únicamente.

Los y las estudiantes desarrollan un proyecto de software. El mismo lo gestionarán mediante el marco de trabajo denominado Proceso Unificado. 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 a ser evaluados. Algunas exposiciones serán de carácter práctico.

## 6. Evaluación

Descripción	Porcentaje
I Parcial	20%
II Parcial	20%
Evaluaciones cortas y Tareas	10%
Proyecto	40%
Exposiciones	10%

### Consideraciones sobre la evaluación

- La plataforma oficial que se utilizará en el curso es la plataforma institucional de Mediación Virtual (<https://mediacionvirtual.ucr.ac.cr>). La entrega de tareas, proyectos y otros evaluables será mediante el aula virtual del curso, el día y la hora indicados previamente por el docente. No se recibirán por ningún otro medio ni fuera de la fecha y hora establecida.
- Para comunicación entre profesores y estudiantes se utilizarán únicamente las cuentas de correo de la universidad (@ucr.ac.cr) y/o de la carrera (@ucrso.info) por lo cual el estudiante debe tenerlo activo y revisarlo continuamente, ya que es el medio por el cual se les informará sobre actualizaciones en el aula virtual, evaluaciones, asignación de tareas, entre otros.
- Durante las lecciones debe mantener los celulares en modo silencioso.
- Durante las evaluaciones el uso de teléfonos celulares, tabletas o cualquier otro dispositivo de comunicación está totalmente prohibido dentro y fuera del aula mientras el estudiante no haya hecho entrega de su evaluación. Todos dispositivos que traiga consigo deberán permanecer apagados y guardados en su bolso o salveque. El uso o aparición de algún dispositivo anulará automáticamente la evaluación de todos los involucrados.

- Los rubros de calificación de cada evaluación serán entregados junto con la especificación de la misma.
- Toda evaluación será comunicada al estudiante al menos 5 días hábiles antes de realizarse, a excepción de las pruebas cortas o “quizes”, según lo establece el Reglamento de Régimen académico.
- Las evaluaciones deben realizarse únicamente dentro de las instalaciones de la sede o recinto donde se imparte el estudiante, y no en otro lugar aunque sea parte de la UCR.
- En caso de ausencia a alguna evaluación se procederá de acuerdo a lo establecido en el Artículo 24 del Reglamento de Régimen Académico Estudiantil de la UCR para su reposición.
- La detección de una posible copia o plagio, total o parcial, en cualquier evaluación se anulará automáticamente a los involucrados y se elevará el caso a la Dirección de la Sede de Occidente para que se aplique el proceso y las sanciones correspondientes según el Reglamento de Orden y Disciplina Estudiantil.
- Todo trabajo escrito deberá entregarse en formato de documento abierto (odf) o en formato de documento portátil (pdf).
- Para todas las tareas o investigaciones solamente se permite el uso de sitios web de corte académico, como por ejemplo artículos o libros de revistas indexadas, enciclopedias Web, sitios de noticias, así como el uso de libros y periódicos. No se permite utilizar material de Internet de fuentes de información que se pueden corroborar o sean poco confiables, de lo contrario no se tomará como bibliografía o referencia válida y la nota de la evaluación será cero.
- Dentro del aula/laboratorio sólo pueden permanecer estudiantes matriculados en el curso y grupo, no se permiten “oyentes”, estudiantes matriculados en otros grupos, administrativos o terceras personas salvo que hayan sido invitados explícitamente por el profesor para fines del curso.
- El uso del aula y/o del laboratorio es exclusivo para fines académicos propios del curso. Cualquier otro uso está prohibido.
- Durante la clase velar por el orden y aseo, principalmente cuando se dan lecciones en los laboratorios, está prohibido consumir alimentos dentro de los laboratorios y no deben dejar basura de ningún tipo al terminar la clase. También asegurar que todos los equipos queden apagados al igual que los aires acondicionados y los pupitres ordenados.
- El examen de ampliación contendrá todos los contenidos del curso.
- Los proyectos de investigación presentados en clase se consideran dentro de los temas a evaluar en los exámenes del curso.
- Los laboratorios, investigaciones, exposiciones, tareas programadas y cualquier asignación grupal se realizarán en grupos que el profesor definirá, según la cantidad de estudiantes del curso.
- Las tareas programadas cuentan con un porcentaje de comprobación individual con el fin de determinar la participación y conocimientos de cada integrante.
- Las tareas programadas deben ir acompañadas de la respectiva documentación interna. No se recibirá la misma posterior a la entrega.
- El no entregar cualquier evaluación a tiempo provocará que se pierda el 100% de la nota. En casos muy justificados y previamente aprobados por el profesor se podrá recibir un trabajo hasta 24 horas después de la hora de entrega, en este caso perderán el 40% del valor del trabajo.
- Para todas las entregas sólo se recibirán aquellas cuyo contenido sea exclusivamente de la asignación respectiva.
- Si algún estudiante es expulsado(a) de su grupo de trabajo debido a incumplimiento de labores justificadas adecuadamente, perderá los puntos respectivos y no podrá continuar con el trabajo. Las

- pruebas respectivas para expulsar a un compañero(a) deben ser presentadas por escrito y con copia al o los involucrados y firmadas por los miembros, donde se exponga la situación presentada.
- La no presentación de código fuente en tareas o proyectos programados, se calificará con nota cero. Aquellos programas que no cumplan con los requerimientos mínimos establecidos en el enunciado o que presenten errores de compilación se calificará de igual manera con nota cero.
  - El valor de cada prueba corta, tarea y/o laboratorio depende de la cantidad total que se realicen durante el curso, sacando un promedio de las notas obtenidas para obtener el porcentaje correspondiente.
  - Las fechas de las evaluaciones del cronograma están sujetas a cambio, previamente comentados en clase, dependiendo del avance de los temas en el cronograma.
  - Las pruebas cortas podrán contemplar temas abarcados en tareas cortas, tareas programadas, laboratorios, exposiciones, lecturas, presentaciones y también cualquier materia vista durante la clases.
  - La redacción y ortografía se tomarán en cuenta como parte de los rubros de evaluación.
  - El material utilizado en el desarrollo del curso podrá ser tanto en el idioma español como en inglés.
  - Los exámenes podrán ser teóricos, prácticos o teórico/prácticos.
  - Se calificará trabajo en equipo, por lo tanto la falta de trabajo, investigación y aportes individuales serán evaluados como grupo, si algún integrante se comprueba que no realizó lo correspondiente a la investigación o proyecto programado, se le restarán puntos al grupo completo.

## 7. Cronograma

Semana 1 - 12 al 16 Marzo	Actividades
<b>Inicio de clases</b>	Entrega y Lectura carta del estudiante Tema: Introducción ingeniería de software
Semana 2 - 19 al 23 Marzo	Actividades
	Tema: Introducción a la Administración de Proyectos
Semana 3 – 26 al 30 Marzo	Actividades
	<b>SEMANA SANTA</b>
Semana 4 - 02 Abril al 06 Abril	Actividades
	<b>Exposición 1: Metodologías de desarrollo del Software</b> Tema: Requerimientos de Software
Semana 5 – 09 al 13 Abril	Actividades
	Tema: Requerimientos de Software (Cont) <b>Exposición 2: Modelado de Procesos</b> <b>Entregable #1 Proyecto: Carta constitución Proyecto</b>
Semana 6 – 16 al 20 de Abril	Actividades
	<b>Primer Examen Parcial</b>
Semana 7 – 23 al 27 de Abril	Actividades

	<b>Semana Universitaria</b> <b>Exposición 3: Herramientas visuales para Análisis de requerimientos (Mockups)</b> Tema: Diseño de Software
<b>Semana 8 – 30 Abril al 04 Mayo</b>	<b>Actividades</b>
	<b>Entregable #2 Proyecto: Análisis de Requerimientos</b> <b>Exposición 4: Patrones de Diseño</b> Tema: Arquitectura de Software
<b>Semana 9 – 7 al 11 Mayo</b>	<b>Actividades</b>
	Tema: Construcción de Software – Gestión de Calidad del Producto
<b>Semana 10 – 14 al 18 Mayo</b>	<b>Actividades</b>
	Tema: Pruebas del Software <b>Entregable #3 Proyecto: Diseño</b> <b>Exposición 4: Herramientas de automatización de pruebas</b>
<b>Semana 11 – 21 al 25 Mayo</b>	<b>Actividades</b>
	Tema: Administración de la configuración del software <b>Exposición 5: Repositorios para Control de Configuración y versionamiento</b>
<b>Semana 12 – 28 Mayo al 01 Junio</b>	<b>Actividades</b>
	Tema: Proceso de Implantación de Producto <b>Exposición 6: User Experience (Mejores prácticas desarrollo de Interfaz de usuarios)</b> <b>Punto de Control #1 Construcción Producto</b>
<b>Semana 13 – 04 al 08 Junio</b>	<b>Actividades</b>
	Tema: Mantenimiento del Software <b>Exposición 7: In Memory Computing</b>
<b>Semana 14 – 11 al 15 Junio</b>	<b>Actividades</b>
	<b>SEGUNDO EXAMEN PARCIAL</b>
<b>Semana 15 – 18 al 22 Junio</b>	<b>Actividades</b>
	<b>Entregable 5 Proyecto: Calidad y Pruebas</b> <b>Punto de Control #2 Construcción Producto</b>
<b>Semana 16 – 25 al 29 Junio</b>	<b>Actividades</b>
	<b>Punto de Control #3 Construcción Producto</b>
<b>Semana 17 – 02 al 06 Julio</b>	<b>Actividades</b>
	<b>PRESENTACION y ENTREGA FINAL DE PROYECTOS</b>
<b>Semana 18 – 09 al 13 Julio</b>	<b>Actividades</b>
	<b>ENTREGA PROMEDIOS</b>
<b>Semana 19 – 16 al 20 Julio</b>	<b>Actividades</b>
	<b>AMPLIACION</b>

---

## 8. Normativa del Curso

Se utilizará la plataforma Mediación Virtual para colocar el material en formato digital y para entregar tareas, entre otros.

Cada estudiante debe matricularse utilizando su cuenta institucional (@ucr.ac.cr)

<https://mediacionvirtual.ucr.ac.cr/login/index.php>

Curso: IF-7100, grupo 002, Sede de Occidente, Tacaes

Contraseña: ingenieria02

## 9. Bibliografía

- IEEE. (2004). *Guide to the Software Engineering Body of Knowledge*. Institute of Electrical and Electronics Engineers. Recuperado el 9 de marzo de 2018 de: <https://www.computer.org/web/swebok>
- IEEE. (2018). *Transactions on Software Engineering*. Institute of Electrical and Electronics Engineers. Recuperado el 9 de marzo de 2018 de: <https://www.computer.org/web/tse>
- CMU. (2018). *Publications*. Carnegie Mellon University – Software Engineering Institute. Recuperado el 9 de marzo de 2018 de: <https://www.sei.cmu.edu/publications/index.cfm>
- Pressman, Roger. (2005). *Ingeniería del Software: un Enfoque Práctico*. Editorial Mc Graw Hill. México.
- Piattini Velthuis, Mario Gerardo. (2004). *Análisis y diseño de aplicaciones informáticas de gestión: una perspectiva de ingeniería de software*. Editorial Alfaomega. México.