

Programa del curso CE4101

Especificación y Diseño de Software

Área Ingeniería en Computadores
Licenciatura Ingeniería en Computadores

[Última revisión de la plantilla: 31 agosto de 2017]

I Parte: Aspectos relativos al plan de estudios

1 Datos generales

Nombre del curso:	Especificación y Diseño de Software
Código:	CE-4101
Tipo de curso:	Teórico – Práctico
Electivo o no:	No
Nº de créditos:	4
Nº horas de clase por semana:	4
Nº horas extra clase por semana:	8
% de las áreas curriculares:	Ciencias de Ingeniería (30%), Diseño en Ingeniería (70%)
Ubicación en el plan de estudios:	7mo Semestre
Requisitos:	CE-3103: Bases de Datos
Co-requisitos:	No
El curso es requisito de:	CE-4302: Arquitectura de Computadores II CE-4303: Principios de Sistemas Operativos
Asistencia:	Obligatorio
Suficiencia:	No
Posibilidad de reconocimiento:	Sí
Vigencia del programa:	II Semestre 2022

2 Descripción general

En este curso el estudiante aprenderá y pondrá en práctica técnicas y medios para la recopilación y documentación de los requerimientos de software, así como también desarrollará las etapas, instrumentos y técnicas sistemáticas referentes al diseño, siendo consciente de la importancia de estos pasos, como parte fundamental de la ingeniería de software, comprendiendo los procesos concernientes a producción de soluciones tecnológicas.

La metodología del curso busca además reforzar ciertas habilidades no técnicas como el trabajo en equipo y la creatividad, así como los valores de responsabilidad, respeto y tolerancia. El desarrollo de estas habilidades y valores permitirá una ejecución óptima de las diferentes actividades de manera individual y grupal.

Al finalizar el curso el estudiante será capaz de realizar la captura y validación de los requerimientos de un sistema, aplicar técnicas de análisis y modelado para la creación de una especificación de requerimientos de software y a partir de dicha especificación, diseñar una solución de software mediante un proceso de diseño sistemático, utilizando tecnologías actuales y el conocimiento adquirido en los cursos anteriores para el desarrollo de sistemas y aplicar los conceptos fundamentales de diseño de bases de datos del curso Bases de Datos.

Este curso es de vital importancia para el resto de los cursos que se imparten en la carrera, ya que desarrolla las habilidades en el estudiante para ser capaz de realizar una adecuada gestión de los requerimientos y a partir de esto diseñar sistemas que permitan brindar soluciones a los retos específicos de cada curso.

En caso de ser requerido se brindará atención de necesidades educativas especiales en cumplimiento con la ley 7600.

Los atributos del graduado relacionados a este curso son los siguientes:

- ◆ **Diseño (Medio):** Capacidad para elaborar y comparar propuestas de diseño de ingeniería para sistemas, componentes o procesos, de acuerdo con requerimientos establecidos.
- ◆ **Trabajo individual y en equipo (Avanzado):** Capacidad para evaluar el trabajo individual y de equipo.
- ◆ **Administración de proyectos y finanzas (Inicial):** Capacidad para comprender los conceptos básicos en finanzas y administración de proyectos asociados a la práctica de la ingeniería.

3 Objetivos

Objetivos específicos	Atributos correspondientes
Comparar y poner en práctica las diferentes metodologías de desarrollo en ingeniería de software.	TE(A), AF(I)
Implementar un conjunto de métodos, técnicas y herramientas para el diseño y la especificación de un producto de software.	DI(M), TE(A), AF(I)
Experimentar con métodos para recopilar y validar información que sea transformada a requerimientos de software.	TE(A)
Usar las herramientas de ingeniería de software para definir y documentar los requerimientos de software de un usuario; funcionales y no funcionales.	TE(A)
Identificar, estudiar y poner en práctica los diferentes niveles a los que se puede aplicar tareas y herramientas de diseño relacionadas con el proceso de ingeniería de software tomando aspectos como: orientación a objetos, patrones, casos de estudio, redes, arquitectura de software, relación sistema usuario.	DI(M), TE(A)
Comprender los diferentes niveles de abstracción en que deben expresarse las soluciones de problemas de diseño.	DI(M), TE(A)
Reconocer los diferentes aspectos relacionados con UE y UI.	DI(M), TE(A),

Ejemplificar y aplicar las herramientas necesarias para poder diseñar una arquitectura de software	DI(M), TE(A)
Generar un producto de software a partir de la puesta en práctica artefactos del diseño de software	DI(M), TE(A), AF(I)
Juzgar las diferentes técnicas para garantizar la calidad del producto o solución tecnológica tanto en la especificación de requerimientos como en el diseño de la solución.	DI(M), TE(A)
Reconocer las tecnologías actuales y las tendencias tecnológicas que influyen en los diseños del software.	DI(M), TE(A)
Desarrollar habilidades no técnicas como la comunicación, el trabajo en equipo y la creatividad, así como los valores de responsabilidad, respeto y tolerancia.	TE(A)

4 Contenidos

→ Proceso de desarrollo de software (1.5 semanas)

- ◆ El ciclo de vida de los proyectos de software.
- ◆ Modelos de desarrollo de software.

→ Ingeniería de requerimientos (1 semana)

- ◆ Concepto de requerimiento.
- ◆ Conceptos de la ingeniería de requerimientos.
- ◆ Impacto de la mala toma de requerimientos.
- ◆ Estándares y documentos para la especificación y/o mantenimiento de los requerimientos.
- ◆ Herramientas para la administración de requerimientos.

→ Obtención y Análisis de Requerimientos (**1.5 semanas**)

- ◆ Proceso de obtención.
- ◆ Problemas y situaciones a considerar.
- ◆ Stakeholders, relación usuario/cliente.
- ◆ Herramientas y técnicas para obtener requerimientos.
- ◆ Validación de los requerimientos.
- ◆ Cuantificación y detalle de los requerimientos.
- ◆ Documentación de las reuniones, minutas y otros.
- ◆ Análisis y seguimiento de requerimientos según la metodología.
- ◆ Dimensionando el tiempo y el costo de los requerimientos.

→ Herramientas de Ingeniería de requerimientos (**2.5 semanas**)

- ◆ Casos de uso y sus diagramas.
- ◆ Modelo conceptual.
- ◆ Diagramas de actividades.
- ◆ Diagramas de secuencia.
- ◆ Diccionario de datos.
- ◆ Diagrama de bloques.
- ◆ Requerimientos no funcionales.
- ◆ Requerimientos en TDD y BDD.
- ◆ Requerimientos de UI y UE.

→ La etapa de diseño de software (**1 semana**)

- ◆ De la especificación al diseño.
- ◆ Pasos para diseñar software.
- ◆ Artefactos del diseño de software.
- ◆ Intervención del diseño en las N capas del software.

→ Diseño de UI y UE (**2 semanas**)

- ◆ Usabilidad.
- ◆ Técnicas y guías para el diseño de interfaces.
- ◆ Diseño de la parte funcional de la interface y componentes gráficos.
- ◆ Validación de datos.
- ◆ Diseño de reportes.
- ◆ Interacción humano computador.
- ◆ Otros factores como móvil y realidad aumentada.
- ◆ Ergonomía.
- ◆ Diseño de dispositivos físicos.
- ◆ Uso del color y la tipografía.
- ◆ Dominio del espacio y el peso.
- ◆ Resistencia del dispositivo o el software al uso.

→ Diseño de la arquitectura del software (**2 semanas**)

- ◆ Entender el problema.
- ◆ Identificar elementos y sus relaciones.
- ◆ Diseño con componentes.
- ◆ Diseño con servicios.
- ◆ Diseño de la integración de sistemas.
- ◆ Requerimientos no funcionales en el diseño de arquitectura.
- ◆ Impacto del networking y tecnologías para las N capas.
- ◆ Arquitecturas empresariales.
- ◆ Tendencias tecnológicas en arquitectura de software.

→ Artefactos del diseño de software (**3.5 semanas**)

- ◆ Diseño de clases.
- ◆ Patrones de diseño.
- ◆ Diseño de la base de datos.
- ◆ Diseño de la topología.
- ◆ Patrones arquitectónicos.
- ◆ Diseño de la seguridad en el software.
- ◆ Uso de herramientas y productos existentes.
- ◆ Bocetos, diagramas y fotografías.
- ◆ Estándar y forma del código.
- ◆ Incidencia de los diseños en el tiempo y el costo.
- ◆ Validación del diseño contra los requerimientos.

→ Preparándose para la fase de construcción (**1 semana**)

- ◆ Administrando el código fuente versus los requerimientos y su trazabilidad.
- ◆ Uso de procedimientos documentados y controlados.
- ◆ El administrador de proyectos.
- ◆ Ambiente de desarrollo, pruebas y producción.
- ◆ Control de calidad y sus diversas técnicas.
- ◆ Test unitario de caja blanca y caja negra.
- ◆ Test de stress y de carga.
- ◆ Revisiones de código e inspecciones.

→ Otros temas

- ◆ Herramientas para el control de requerimientos y diseños.
- ◆ Prácticas para la creación de contratos.
- ◆ Redactando propuestas de proyectos de tecnología.
- ◆ Estimación de tamaño de proyectos.
- ◆ Defensa en profundidad.
- ◆ Charlas de buenas prácticas empresariales.
- ◆ Principio de usar y adaptar.

- ◆ Tendencias tecnológicas y de diseño en el mercado.
-

II parte: Aspectos operativos

5 Metodología de enseñanza y aprendizaje

Exposición magistral de los temas por parte del profesor, análisis de casos de estudio, ejercicios y proyectos prácticos para afianzar los conocimientos, desarrollar habilidades y destrezas del trabajo en equipo.

El profesor realizará sesiones prácticas en el aula al finalizar cada tema para aplicar los conocimientos desarrollados en las exposiciones magistrales.

Se realizarán evaluaciones escritas individuales (quices y examen) sobre los contenidos del curso y tareas cortas (individuales y grupales) cada semana.

Se desarrollará un proyecto práctico semestral grupal, el cual estará dividido en diferentes etapas para poner en práctica lo aprendido en el curso y que permita al estudiante la aplicación de los instrumentos y técnicas de la ingeniería de software a un problema “real”, tanto dentro como fuera de clase.

Se realizarán dos investigaciones grupales de temas complementarios a la materia del curso.

Las herramientas básicas para poder enfrentar este curso son:

- Una computadora con mínimo 4GB de RAM y CPU Core i5 o similar.
- Sistema Operativo GNU Linux o Windows.

Si un estudiante no cuenta con estos recursos puede hacer uso de los diferentes laboratorios disponibles en el campus central del Instituto Tecnológico de Costa Rica.

Durante el curso se estarán usando herramientas de gestión de requerimientos y de diseño que en su momento serán indicadas por el profesor.

El curso presenta una modalidad híbrida en donde de ser necesario el profesor podría citar a los estudiantes para una sesión presencial previa notificación.

6 Evaluación

Evaluaciones escritas	20%
Casos de estudio e investigaciones	10%
Proyecto semestral	55%
Quices	7,5%
Tareas	7,5%
TOTAL	100%

Se realizarán dos evaluaciones escritas, cada una con el mismo valor, que permitirá validar el conocimiento teórico relacionado con los contenidos del curso.

Los estudiantes tendrán que resolver al menos dos casos de estudio y elaborar al menos dos investigaciones. La nota de estos cuatro entregables se promediarán para obtener el rubro de Casos de estudio e investigaciones a menos de que se indique de otra manera en los enunciados de las asignaciones.

El proyecto será desarrollado a lo largo de todo el semestre y tendrá varios entregables que se establecerán en su respectiva especificación.

Con respecto a los quices, todos tendrán el mismo valor, es decir, se promediará la nota de todos los quices para obtener el rubro de Quices. Esto mismo aplicaría para el rubro de Tareas.

7 Bibliografía

Obligatoria

- Albin, Stephen. 2003. The Art of Software Architecture: Design Methods and Techniques. John Wiley & Sons
- Bass L; Clements P; Kazman R. Software Architecture in Practice. Third Edition. Addison-Wesley, 2013
- Booch. Análisis y Diseño Orientado a Objetos Con Aplicaciones. Addison-Wesley/Díaz De Santos, 1996
- Booch, Grady, Rumbaugh, James, Jacobson, Ivar. El lenguaje unificado de modelado. Manual de referencia Madrid, España. Addison-Wesley. 2000.
- Budgen, David. Software Design, 2nd. Ed. Addison-Wesley, 2003
- IEEE Recommended Practice for Software Requirements Specifications, 1993.
- Larman, Craig. UML y Patrones: Una introducción al análisis y diseño orientado a objetos y al proceso unificado. Tercera edición. Madrid, España. Prentice-Hall. 2005.
- Sommerville. Ingeniería de Software. Addison-Wesley, 2015
- Wieggers, Karl. Software Requirements. Second Edition. Washington, USA. Microsoft Press. 2003
- Yourdon, Edward; Argila, Carl. Case Studies in Object-Oriented Analysis and Design. NJ, USA. Yourdon Press Computing Series, 1996.
- Documentos de internet

Bibliografía Adicional

- Beatty J; Chen Anthony. Visual Models for Software Requirements. Microsoft Press
- Fowler, M. UML Gota a Gota. Addison-Wesley, 1997
- Gorton, Ian. Essential Software Architecture. Springer-Verlag, 2006.
- Monahan; Shaw. Model-based specifications. En McDermid (ed.), Software engineer's reference book. Butterworth-Heinemann, 1991, 1993
- Rozanski, N; Woods E. Software Systems architecture, Second Edition. Addison-Wesley, 2012
- Wordsworth. Software engineering with B. Addison-Wesley, 1996

8 Cuerpo docente

Daniel Madriz Huertas

Egresado del Instituto Tecnológico de Costa Rica y de la Universidad de Costa Rica.

Trece años como profesor del Instituto Tecnológico de Costa Rica.

Contacto

Email: daniel.madriz.h@gmail.com

Teléfono móvil: 8324-1684 (WhatsApp, Telegram)

Oficina: Edificio F2, Oficina 22

Consultas

Horario y lugar de consulta a convenir con cada profesor.

Tutorías

Horario y lugar de tutorías se indicará una vez iniciado el curso lectivo debido a que se debe esperar si el curso contará con tutoría. Si así fuera se debe definir horarios entre tutores y estudiantes.

9 Instrucciones generales

Medios de comunicación

Cada profesor establecerá los medios de comunicación con los estudiantes y los medios para hacer entrega de las evaluaciones y para la entrega de materiales.

Entregables

Cualquier documento que sea solicitado como entregable debe ser en formato PDF. En caso de ser necesario otro formato será indicado por el profesor.

Cada proyecto establece las reglas de entrega.

Sobre el proceso de acreditación de la carrera

Dado que la carrera Ingeniería en Computadores está en proceso de re-acreditación, el profesor podrá dejarse algunos de los entregables y quices, eliminando la información del estudiante.

Otros

Está terminantemente prohibido realizar grabaciones de las lecciones sin autorización del profesor.