

Programa del curso IC-7841

Proyecto de Ingeniería de Software

Escuela de Computación
Carrera de Ingeniería de Computación, Plan 410.

I parte: Aspectos relativos al plan de estudios

1 Datos generales

Nombre del curso:	Proyecto de Ingeniería de Software
Código:	IC-7841
Tipo de curso:	Teórico - Práctico
Nº de créditos:	3
Nº horas de clase por semana:	4
Nº horas extracласe por semana:	5
Ubicación en el plan de estudios:	Curso del 7º semestre de la carrera de Ingeniería en Computación
Requisitos:	IC-4302 Bases de Datos II. IC-6831 Aseguramiento de la Calidad del Software.
Correquisitos:	AE-4208 Desarrollo de Emprendedores.
El curso es requisito de:	IC-8842 Práctica profesional.
Asistencia:	Obligatoria
Suficiencia:	No.
Posibilidad de reconocimiento:	Sí.
Vigencia del programa:	I semestre 2012.

2 Descripción general

El curso introduce los conceptos y técnicas necesarios para la creación de productos con tecnologías de tendencia reciente, así como los impactos arquitectónicos que estos puedan tener. El curso presta atención al enfoque de diseño centrado en el usuario, atendiendo especialmente lo concerniente al diseño de la experiencia del usuario. Los estudiantes trabajan en equipos que siguen un enfoque ágil que integra procesos, herramientas y técnicas de análisis, diseño, construcción, verificación y validación de software aprendidos a lo largo de la carrera, para lograr un producto expresado en tecnologías modernas que componen una arquitectura adecuada.

Aplica de manera práctica un proceso de desarrollo de software completo, pasando por todas las etapas – desde el estudio del problema y las necesidades de un cliente real, pasando por el diseño de una solución informática, hasta construir, integrar y probar el producto. Se desarrolla un sistema planteado por los estudiantes del curso, en grupos de 2 o 3 estudiantes.

Este curso resume e integra las mejores prácticas en el desarrollo de sistemas y prepara al estudiante en una de las actividades que puede llegar a desarrollar en el futuro inmediato como profesional en Ingeniería de Software.

3 Objetivos

Objetivo General

Desarrollar un sistema que integre conocimientos y mejores prácticas como profesional en Ingeniería de Software.

Objetivos Específicos

- Gestionar un proyecto de desarrollo de software siguiendo un enfoque ágil en el proceso que enmarque las actividades técnicas.
- Analizar un problema de desarrollo de software e modelar el dominio en el cual se circunscribe.
- Describir textualmente los problemas y los requerimientos, tomando en cuenta el contexto, dominio, y limitaciones/restricciones que presenten los usuarios y los interesados.
- Modelar las clases, las entidades, las relaciones, las operaciones y los comportamientos claves del dominio del problema.
- Diseñar una arquitectura para solucionar el problema, evaluando alternativas que busquen favorecer los principales atributos de calidad identificados como requerimientos no-funcionales.
- Aplicar patrones apropiados en los distintos niveles de diseño de la solución.
- Especificar los componentes de la arquitectura de software propuesta.
- Construir programas y componentes para satisfacer sus especificaciones.
- Realizar revisiones sobre documentos técnicos y de código de programación.
- Planear, diseñar y aplicar pruebas a las unidades, componentes y subsistemas, en todos niveles requeridos, para evaluar su funcionamiento.
- Integrar los elementos y probar su integración para ensamblar un sistema completo de software que satisfaga los requerimientos según los alcances establecidos.
- Trabajar en equipos de manera colaborativa, para lograr un producto de alta calidad al final del proyecto.
- Producir documentación técnica de la especificación, el análisis, el diseño, los componentes, las pruebas y los manuales de usuario.
- Mantener los documentos técnicos y administrativos que permitan dar visibilidad al trabajo del equipo, controlar sus avances y atender los riesgos.
- Gestionar rigurosamente las versiones y administrar la configuración de los elementos de software, la documentación y demás recursos que se generen como parte del proyecto.

4 Contenidos

Enfoques ágiles para el desarrollo de software

Definición de los usuarios y dominios para el desarrollo del proyecto

Repertorio de tecnologías de software

- Tecnologías de software para aplicaciones Web
- Tecnologías de software para aplicaciones con dispositivos móviles
- Tecnologías de software para construcción de servicios distribuidos y cómputo en la Nube

Diseño de la experiencia del usuario

- Principios de la experiencia del usuario
- Arquitectura de la información
- Paradigmas de interacción
- Diseño de experiencias de usuario

Arquitecturas de aplicaciones de software

- Estilos y patrones arquitectónicos
- Arquitecturas multi-capas
- Arquitecturas de aplicaciones Web
- Servicios y arquitecturas orientadas a servicios
- Arquitecturas con dispositivos móviles
- Otras consideraciones entre arquitectura y tecnología

Pruebas del software

Transición e implantación del producto

- Integración y prueba final del producto
- Construcción de ayudas en línea
- Manuales de usuario
- Manuales técnicos
- Capacitación del usuario

Gestión calidad del software

Gestión de la configuración

II parte: Aspectos operativos

5 Metodología de enseñanza y aprendizaje

La metodología del curso consistirá en la combinación de clases magistrales por parte del profesor, sesiones de seguimiento grupal, exposiciones de avance y de investigaciones, y el trabajo de campo de los estudiantes con sus usuarios.

Las tecnologías y arquitecturas por usar en el curso podrán ser variadas periódicamente, según determine el Consejo de la Escuela de Ingeniería en Computación.

Al inicio del curso se definirán los dominios sobre los cuales los estudiantes realizarán el proyecto.

6 Evaluación

- La evaluación está dividida en las diferentes etapas para realizar un producto completo de ingeniería de software tomando con gran interés la interacción con el usuario final así como la planificación y seguimiento de cada estudiante.

Evaluación	
Rubro calificación	Puntaje
Project charter	5%
Visión	5%
Prototipo	5%
Iteraciones (1,2 y 3)	60%
Pruebas aceptación del sistema	5%
Manual técnico	7%
Calificación final usuario y evidencias seguimiento semanal con el usuario	8%
Paper	5%
Total:	100%

Cronograma de Trabajo		
Semana semestre	Actividad	Fecha aproximada
1.	Programa de curso, acuerdo de horario	Semana 1 4 de febrero
2.	Primera reunión con cliente	Semana 2 11 de febrero
3.	Project charter Listado y propuesta de casos de uso a entregar para cada iteración	Semana 3 18 de febrero
4.	Visión del proyecto Aprobación de casos de uso por iteración	Semana 4 25 de febrero
5.	Prototipo completo y definición de temas de paper	Semana 5 4 de marzo
6.	Iteración 1- ERS 1 y plan pruebas	Semana 6 11 de marzo
7.	Iteración 1- SAD	Semana 7 18 de marzo
8.	Iteración 1- ejecución, manual	Semana 8 25 de marzo
9.	Iteración 2- ERS 2 y plan pruebas 2	Semana 9 1 de abril

10.	Iteración 2 y entrega de paper	Semana 10 8 de abril
11.	Iteración 2- Ejecución, manual	Semana 11 22 de abril
12.	Iteración 3- ERS 1 y plan pruebas	Semana 12 29 de abril
13.	Iteración 3- SAD	Semana 13 6 de mayo
14.	Iteración 3- Ejecución, manual	Semana 14 13 de mayo
15.	Pruebas sistema	Semana 15 20 de mayo
16.	Manual técnico	Semana 16 27 de mayo
17.	Calificación usuario y evidencias	Semana 17 3 de junio
Calificación iteración 1,2 y 3		
Rubro calificación		Valor
ERS		10
Plan pruebas		5
SAD		10
Ejecución del programa y evidencias del avance personal de cada persona		50
Informe ejecución de pruebas		5
Manual de usuario		10
Carta usuario aceptación		10
Total nota iteración		100

Disposiciones generales

El curso trata de la aplicación práctica de un proceso ágil de desarrollo de software para desarrollar software útil para usuarios reales.

Se brindan los conceptos tecnológicos y metodológicos que complementan el bagaje de conocimientos de los estudiantes, para que puedan construir un producto de software con tecnologías de

tendencia reciente.

El proceso de software abarca desde el análisis de un problema hasta la implementación de su solución. El curso requiere que los estudiantes tengan una gran capacidad de investigación y aprendizaje autónomo.

El profesor les guía en ese proceso, pero es esperable que haya mucha diversidad tecnológica y el profesor no será experto en cada posible tecnología por utilizar.

El proyecto por desarrollar deber tener un alcance definido que no sobrepase las 16 semanas de trabajo con una dedicación de 9 horas de trabajo por semana por cada miembro de equipo.

1. El curso se aprueba con nota de 67
2. NO hay examen de reposición
3. La entrega de cada documento del curso, se debe colocar en el sitio de TEC digital del curso, a más tardar el día asignado antes de la medianoche.
4. PAPER: Cada alumno deberá realizar un paper con los temas relacionados con los temas del proyecto y del curso, los cuales se asignarán en la semana 2 del período de verano y se entregará a más tardar en semana 5 del mismo.
5. PROYECTO: El tamaño se mide por casos de uso. Cada alumno debe desarrollar al menos 5 casos de uso. Uno de los cuales uno debe ser para dispositivo móvil. Para esta medición no se toman en cuenta los casos de uso de gestión de usuarios, seguridad ni login.
Se desarrollar en forma iterativa. Las iteraciones 1 y 3 son de trabajo grupal y la iteración 2 es de trabajo individual
6. PROJECT CHARTER y VISION: se debe entregar al profesor en digital
7. PROTOTIPO: debe ser realizado preferiblemente en algún software libre. Debe ser multiplataforma, se entregará un prototipo navegable, no solo mockups.
En el desarrollo del prototipo se debe abarcar todos los casos de uso del proyecto, adjuntar carta aprobación usuario. Esta carta valdrá 30 puntos de la calificación del prototipo y colocar estos entregables en

el TECdigital antes de la medianoche de la fecha asignada.

8. ITERACION: Los documentos ERS, SAD, plan e informe pruebas deben ser diferentes pues varían los casos de uso de cada iteración.

9. ITERACION 2: Es estrictamente individual. Cada alumno del grupo debe presentar su programa, documentos y carta del usuario. Deben ser diferentes a los de sus compañeros de equipo, pues varían los casos de uso.

10. MINUTAS Y/O EVIDENCIAS DE COMUNICACIÓN CON EL USUARIO : deben tener el formato dado al inicio del curso para una minuta o el mecanismo formal de interacción con el usuario, para las entregas si debe venir con firma y sello del usuario. Se debe presentar solamente una minuta o evidencia por semana del semestre.

A las reuniones con el usuario deben asistir todos los integrantes del grupo, por lo que las minutas deben venir firmadas por el usuario y por cada estudiante.

Estas evidencias deben corresponder a reuniones con el usuario, en los que se traten temas del proyecto.

Minutas para firmar documentos no se toman en cuenta dentro del mínimo total por entregar.

Se debe generar un pdf evidencia semanal de contacto con el usuario se deben subir al TEC Digital cada semana, si se utiliza otro medio debe indicarse.

11. EVIDENCIAS DE AVANCE SEMANAL INDIVIDUAL: la bitácora individual almacenarse en GitHub, debe haber evidencia del avance de cada miembro según tareas asignadas, este rubro es de suma importancia para la calificación de cada iteración.

12. CALIFICACION FINAL USUARIO: es indispensable para aprobar el proyecto. Debe ser un email o un documento impreso de parte del usuario, con una calificación del 1 al 10 y con observaciones para oportunidades de mejora que justifiquen la nota asignada

13. CALIFICACION GRUPAL: es una calificación de parte de cada

alumno hacia sus compañeros de grupo de trabajo. El promedio de esta calificación se multiplicará por la calificación obtenida en cada iteración grupal, para definir la nota de la iteración.

14. ENTREGA DE CALIFICACION GRUPAL: Debe ser enviada por correo electrónico al profesor, el mismo día que se entrega el manual técnico

15. CALIFICACION ITERACION: La calificación de las dos iteraciones grupales será el producto de la calificación de los compañeros de grupo multiplicado por la nota obtenida por el grupo en los tópicos de la iteración

16. Correo electrónico: todo correo electrónico del curso debe llevar al inicio del "asunto" lo siguiente:

IC-7841-Proy-", nombre del documento o punto de control, y nombre del estudiante

Debe ir dirigido a la cuenta de la profesora del curso: mariaestrada.s@gmail.com

17. SOBRE DOCUMENTOS IMPRESOS: todo entregable será alojado en la fecha de entrega en el TECdigital, al finalizar el curso según requerimiento del usuario final se imprimirán los documentos, de lo contrario se entregará todo el proyecto completo y documentos en algún dispositivo portable para entregarse al usuario.

18. LUGAR REVISION: se realizará por vía digital o presencial en el espacio que se coordine de previo por parte de la profesora del curso.

7 Bibliografía

Lethbridge, Timothy; Laganière, Robert. Object-Oriented Software Engineering. McGraw Hill. 2001. ISBN 0-07-710908-2.

Augustine, S. (2005). Managing Agile Projects. Prentice-Hall.

Rosenberg, Doug, Stephens, Matt (2007). Use Case Driven Object Modeling with UML: Theory and Practice. Springer-Verlag, New York, USA.

8 Profesor

Profesora: Ing. María Estrada Sánchez MSc.

Campus Central Cartago

Horario: L 17:00-20:50 B306

Horario de consulta: martes 1pm -2:30 pm, (coordinar previamente para confirmar espacio físico), adicional por vía digital

Otros medios de consulta:

Telegram: <https://telegram.me/MariaEstrada>

Skype: mariaestrada.s

Correo electrónico mariaestrada.s@gmail.com

Teléfono: 8323-2214