



PROGRAMA DE ESTUDIOS

CODIGO	TICS431
ASIGNATURA	Taller de Ingeniería de Software
CREDITOS	3
REQUISITOS	Programación, Probabilidades y Estadística, Tecnologías Digitales/Información, todos aprobados
FACULTAD DE ORIGEN - Sede	Ingeniería y Ciencias
DISTRIBUCIÓN HORARIA	140 mins. clases/semana 70 mins.hrs. /taller/ejercicios/semana
PROFESOR DE TALLER	Vicky Latorre Cristián Astudillo
AÑO Y SEMESTRE	2016-2

1 Introducción

El desarrollo de software se ha convertido en una de las áreas importantes de la ingeniería moderna, y donde es necesario aplicar métodos sistemáticos y herramientas que permitan generar productos adecuados a las necesidades requeridas de los usuarios y del negocio, con altos estándares de calidad. Este curso se orienta al ejercicio práctico de cada una de las metodologías necesarias para el diseño y construcción de software.

Hoy día se puede decir que el objetivo común de todas las organizaciones que realizan proyectos ya sea de software u otros es que deben generar valor. Para ello, uno de los medios que las empresas realizan es invertir en mejoramiento de procesos y, dentro del desarrollo de aplicaciones de software esto implica invertir en aplicar un proceso o definir un marco de trabajo que pueda ser repetible y medido de forma que los desarrolladores puedan hacer una introspección de manera tal, que les permita detectar sus fallas, las buenas prácticas en el desarrollo de aplicaciones y aprender del pasado para mejorar los futuros desarrollos

2 Objetivos

Al aprobar este curso de Taller el alumno estará capacitado para aplicar metodologías en el desarrollo de software de alta productividad, disponiendo del dominio de las técnicas modernas para ello. Así, el alumno dispondrá de un dominio apropiado de aplicación práctica de las técnicas de ingeniería en todas las fases del desarrollo y mantenimiento de software, así como de los procesos de soporte asociados, y de aseguramiento de calidad en la producción de software .

3 Contenidos del Curso

El contenido del curso ha sido estructurado en 6 unidades, las cuales cubren los aspectos relativos al desarrollo moderno de software. Estas unidades son las siguientes.

PRIMERA UNIDAD: Fundamentos. Software e ingeniería de software. Proceso de software y paradigmas de desarrollo. Revisión práctica de los paradigmas.

SEGUNDA UNIDAD: Análisis de Requerimientos de Información. Identificación del problema. Determinación de los requerimientos y especificación basadas en procesos, cómo se evalúan los requerimientos en un contexto más integral en la empresa..

- Concepción
- Comprensión de los requerimientos
- Modelo de casos de uso: especificación de los requerimientos en su contexto
- Identificación de otros requerimientos
- Transición desde la concepción a la elaboración

TERCERA UNIDAD: Diseño de Software. Aplicación de Métodos, técnicas y herramientas de modelamiento y representación del software.

- Modelo de casos de uso: diagramas de secuencia del sistema
- Modelo conceptual: conceptos, asociaciones, atributos
- Transición desde los requerimientos al diseño
- Diagramas de interacción
- Uso de patrones para el diseño
- Modelo de diseño: realizaciones de casos de uso con patrones, visibilidad, diagramas de clases del diseño
- Refinamiento del modelo conceptual, diseño de la arquitectura, diseño de frameworks

CUARTA UNIDAD: Construcción de Software. Aplicación de métodos, técnicas y herramientas de modelamiento y representación del software.

- Modelo de implementación: mapeo del diseño a código
- Desarrollo iterativo, herramientas, notación avanzada de UML
- Gestión de proyectos de software

QUINTA UNIDAD: Calidad de software. Aplicación del Aseguramiento de calidad de software. Revisiones e inspecciones. Pruebas.

SEXTA UNIDAD: Soporte. Aplicación de la Gestión de configuración. Herramientas y ambientes de desarrollo. Estándares. Documentación.

4 Metodología

Durante el semestre los alumnos aplican Ingeniería de Software mediante el desarrollo de un proyecto acotado.

Para el desarrollo del proyecto se formarán "equipos" de alumnos, donde cada equipo de trabajo tendrá asignadas tareas que deberá realizar, guiado por pautas del Profesor de Taller.

Cada integrante del equipo de trabajo cumplirá un rol. Además, cada equipo de trabajo deberá autoadministrar su desarrollo, registrar su quehacer, generar la documentación correspondiente y coordinarse con los otros equipos de trabajo para lograr la integración de las partes y construir los subproductos que finalmente constituirán el producto final.

Basado en este proyecto, se estructurará la aplicación de Ingeniería de Software, similar a una empresa de desarrollo de software. Se preparará por cada sesión un Laboratorio, que los alumnos deberán seguir y al fin de éste presentarán el resultado alcanzado, con la posibilidad de corregirlo, de acuerdo a las observaciones encontradas.

5 Evaluación

La evaluación del curso será en base a lo siguiente:

Laboratorios (Informes o Exposiciones) (L):	35%
Proyecto (P):	55%
Evaluación de desempeño grupal e individual (G):	10 %

Normas

La Nota Final (NF) se calculará de la siguiente forma:

Se calculará una Nota presentación a Examen: $NPE = 0,35 L + 0,55 P + 0,1 G$

Si $NPE \geq 5,0$ y L, P y $G > 4,5$, entonces $NF = NPE$.

De lo contrario, el alumno deberá rendir un examen oral basado en el Proyecto. Si la nota obtenida en el examen (NE) es mayor o igual a $3,0$, entonces $NF = 0,30 NE + 0,70 NPE$. Si NE es menor a $3,0$, entonces $NF = NE$.

Asistencia a Laboratorios: 75 %.

La inasistencia a exposiciones del Proyecto se calificará con un $1,0$. La justificación aceptada por la Facultad dará derecho a rendir el examen final del curso, cuya evaluación valdrá por las evaluaciones de pruebas no rendidas y también como evaluación del examen.

La entrega de todos los informes de laboratorio es obligatoria. La no entrega de un informe de laboratorio implica que la nota respectiva sea $1,0$.

6 Bibliografía

1. Larman, C. "Applying UML and patterns: an introduction to O-O Analysis and Design". Prentice Hall, 2005.
2. Pressman, R. "Software Engineering: a practical approach". Prentice Hall, 2010.
3. Ian F. Darwin, "Java Cookbook", O'Reilly Media, 2014.
4. David Flanagan, Benjamin J Evans, "Java in a Nutshell", O'Reilly Media, 2014.