



ANÁLISIS Y DISEÑO DE SISTEMAS 1

MARTES 7:10 A 8:50 (SALÓN 49) – JUEVES 17:20 A 19:00 (SALÓN 9)

PROGRAMA DEL CURSO

ING. RICARDO MORALES

EMAIL: JRMPRADO@PROTONMAIL.COM

DESCRIPCIÓN GENERAL

¿ACERCA DE QUÉ ES EL CURSO?

El desarrollo de software se ha convertido en los últimos años en uno de los trabajos con mayor crecimiento. En cuanto al uso de smartphones, las estadísticas indican un crecimiento acelerado en la adopción de los mismos. Algunos autores han dicho que “la interacción diaria con otros está tan asociada a la tecnología que no se puede discutir de manera significativa la sociedad humana o el desarrollo económico sin incorporar el impacto de la tecnología”. El software está presente en diferentes campos como salud, educación, manufactura y otros.

El curso es acerca de procesos utilizados para el desarrollo de software, considerando diferentes enfoques actuales para ello. Se aplicarán metodologías ágiles y tradicionales, con el objetivo de contrastar sus características y evaluar las posibilidades de uso dependiendo del tipo de proyecto. Lo anterior, tomando en cuenta perspectivas como requerimientos del software, administración de proyectos de software, prácticas de codificación y prácticas de calidad aplicadas durante el proceso de codificación.

¿POR QUÉ ES IMPORTANTE EL CURSO?

Dada la presencia del software en casi todos los aspectos de la sociedad, su confiabilidad,

disponibilidad y calidad son factores críticos de éxito, tanto para un producto de software, como para la comunidad de usuarios que lo utilice.

Para alcanzar niveles de calidad acorde a las expectativas de los clientes, es necesario aplicar procesos en la construcción del software, de manera que los mismos consideren los factores de alcance del proyecto, tiempo requerido para el desarrollo, costos y calidad.

OBJETIVOS

¿QUÉ DEBO SABER Y PODER HACER AL FINAL DEL CURSO?

Identificar y describir las fases del ciclo de vida del desarrollo de software

Aplicar y evaluar prácticas de manejo de requerimientos de software en proyectos de desarrollo de software

Aplicar la metodología Scrum en la construcción de una aplicación de software

Aplicar la metodología Kanban en la construcción de una aplicación de software

Comparar y contrastar las metodologías Scrum y Kanban en el contexto de un proyecto de desarrollo de software

Conocer y evaluar la aplicación de las prácticas de las metodologías Scrum y Kanban

CONTENIDO

Introducción a la ingeniería de software

Procesos de desarrollo de software

Enfoques para requerimientos

Metodologías ágiles

- Scrum
- XP
- Kanban

Calidad y pruebas en metodologías ágiles

METODOLOGÍA

Se impartirá clase magistral 2 veces por semana.

Así mismo se formarán grupos para el desarrollo de un proyecto de construcción de software donde se aplicarán las metodologías presentadas en el curso, dando solución a un problema real, lo cual se complementará con tareas y evaluaciones periódicas.

Se desarrollarán investigaciones y exposiciones de conferencias recientes relacionadas con los temas del curso.

EVALUACIÓN

EVALUACIONES PARCIALES

Se realizarán 3 evaluaciones parciales (20 agosto, 22 septiembre y 15 de octubre). Las evaluaciones se realizarán de forma escrita y electrónica, evaluando el contenido cubierto hasta la clase anterior a la evaluación.

PROYECTO

Los estudiantes deben formar grupos para el desarrollo de un proyecto de curso que consistirá en la construcción de un sistema de software utilizando las metodologías y prácticas que están en el contenido del curso.

El proyecto está dividido en 4 fases:

- Fase 1 (14 agosto): Definir requerimientos (sistema a desarrollar), definir arquitectura a utilizar, seleccionar y configurar herramientas de apoyo al proyecto. (6)
- Fase 2 (04 septiembre), en esta fase se desarrollarán al menos dos iteraciones del proyecto elegido utilizando SCRUM y algunas prácticas de XP. (7)
- Fase 3 (26 septiembre). en esta fase se desarrollarán al menos dos iteraciones del proyecto elegido utilizando SCRUM y algunas prácticas de XP. (8)
- Fase 4 (30 octubre), en esta fase se desarrollarán al menos dos iteraciones del proyecto elegido utilizando Kanban y algunas prácticas de XP. (8)

Para la evaluación de cada fase se definirá una rúbrica que contendrá detalle los aspectos a calificar.

EXPOSICIÓN TEMAS

Cada grupo debe realizar una exposición en el curso de una conferencia de InfoQ, que será asignada en los primeros días de clase.

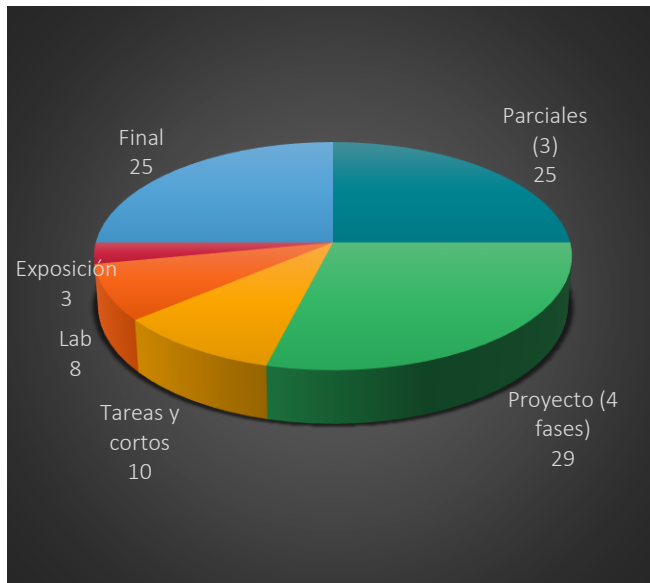
LABORATORIO

El auxiliar del curso llevará a cabo actividades de apoyo al desarrollo de las diferentes fases del proyecto y aplicación de las prácticas de las metodologías.

TAREAS Y CORTOS

En las clases durante el semestre se asignarán lecturas para resumen y discusión, así como exámenes cortos y quizzes.

DISTRIBUCIÓN DE LA EVALUACIÓN



POLÍTICAS

Las entregas fuera de fecha no son aceptadas.

No es permitido el uso de celulares durante la clase y exámenes parciales.

Debe existir respeto por las opiniones de los demás.

Como estudiantes universitarios, se espera que sepan y entiendan las guías de ética y plagio relacionadas con trabajos de otros autores.

Al final del semestre, no se asignarán trabajos extra para recuperar puntos de zona.

Cualquier conflicto entre los grupos del proyecto, debe ser resuelto por los integrantes de dicho grupo.

La atención de dudas es de preferencia durante las clases. Habrá atención por correo, pero no necesariamente una respuesta inmediata.

Se debe aprobar (61/100) el laboratorio del curso para tener derecho a examen final.

BIBLIOGRAFÍA

1. Beginning software engineering, Rod Stephens.
<https://www.amazon.com/Beginning-Software-Engineering-Rod-Stephens/dp/1118969146>
2. Software Requirements (Developer Best Practices) 3rd Edition, Karl Wiegers.
<https://www.amazon.com/Software-Requirements-Developer-Best-Practices/dp/0735679665>
3. Writing effective use cases. Alistair Cockburn.
<https://www.amazon.com/Writing-Effective-Cases-Alistair-Cockburn/dp/0201702258>
4. User stories applied: for agile software development. Mike Cohn.
<https://www.amazon.com/User-Stories-Applied-Software-Development/dp/0321205685>
5. The Scrum Guide,
<http://www.scrumguides.org/>
6. Scrum from the trenches – 2nd edition, Henrik Kniberg,
<http://www.infoq.com/minibooks/scrum-xp-from-the-trenches-2>
7. Extreme programming pocket guide. O'Really Media.
<https://www.amazon.com/Extreme-Programming-Pocket-Guide-Development-ebook/dp/BOOBIRRRX2>
8. Priming Kanban, Jesper Boeg,
<http://www.infoq.com/resource/minibooks/priming-kanban-jesper-boeg/en/pdf/PrimingKanban-JesperBoeg-Version2.pdf>

9. Agile project management with Kanban (Developer practices), Eric Brechner.
<https://www.amazon.com/Project-Management-Kanban-Developer-Practices/dp/0735698953>
10. Specification by example, Gojko Adzic.
<https://www.amazon.com/Specification-Example-Successful-Deliver-Software/dp/1617290084>
11. BDD in action, John Ferguson Smart.
<https://www.amazon.com/BDD-Action-Behavior-driven-development-lifecycle/dp/161729165X>