



Fecha: diciembre 2021

PROGRAMA ACADÉMICO: Ingeniería de Sistemas y Computación

SEMESTRE: Octavo

ASIGNATURA: Trabajo de campo

CÓDIGO: 8108272

NÚMERO DE CRÉDITOS: 4

PRESENTACIÓN

La asignatura Trabajo de Campo, es un curso teórico-práctico, que complementa las materias de ingeniería de software, bases de datos y telemática, ofreciendo un mayor dominio del enfoque ingenieril y sistemático para el desarrollo de software. El curso abarca las etapas de análisis y especificación de requisitos, diseño, programación, pruebas y evaluación de software, aunque el énfasis principal está en entender, hacer visible, controlar y mejorar el proceso de desarrollo de software mediante la utilización de métricas y modelos de evaluación de calidad, que permitan implementar buenas prácticas en el proceso de construir software.

De esta manera el curso aprovecha los conocimientos ya adquiridos por los estudiantes, proyectándolos a la producción de software con calidad; así se genera mayor responsabilidad en el desarrollo de software en su ejercicio profesional, a partir del planteamiento, ejecución y evaluación de un producto de base tecnológica para una problemática real.

JUSTIFICACIÓN

La gestión de proyectos de desarrollo de software es una tarea compleja, por el concepto de gestión como también por la naturaleza lógica del producto desarrollado. Actualmente existen técnicas de ingeniería de software y del área de gestión, aplicables a la administración de proyectos de desarrollo de software, y que ayudan a los jefes de proyectos y administradores a gestionar proyectos exitosos. Razón por la cual hoy en día en las instituciones del sector productivo se ha generado una dinámica de cambio que favorece la implementación de sistemas de información, con el fin de establecer instrumentos que conlleven a una mejora continua en planeación, evaluación y lograr una mayor efectividad y acierto en la toma de decisiones, es así que se hace necesaria una mejor coordinación y el establecimiento de estándares en conceptos, indicadores y variables para generar óptimos resultados y hacer más estables los procesos de planeación, evaluación e información, ya que las funciones siempre se realizarán a pesar de que la organización se modifique.

Es el interés del Ingeniero de sistemas y el enfoque del curso, aprovechar los avances tecnológicos disponibles en los campos de la informática y las telecomunicaciones para desarrollar un sistema de información que le permita ubicarse dentro del grupo de entidades que pueden ofrecer servicios con eficiencia, apoyar su toma de decisiones en cifras veraces y oportunas, para obtener así otros beneficios que se van observando durante la utilización del sistema, sin perder de vista el crecimiento acelerado que está sufriendo en cuanto a su estructura física, a la cantidad de personal que en ella labora y al sector productivo al que pertenece.



COMPETENCIAS

El desarrollo de las actividades de la asignatura involucra competencias generales como son:

- Interpretativa, mediante la cual el estudiante comprende e interpreta los conceptos fundamentales del proceso de desarrollo de software, calidad de software, evaluación de software y aplicación de las CASE en el proceso.
- Argumentativa, ya que el estudiante está en la capacidad de argumentar qué metodología de desarrollo de software utilizar en una determinada situación y por qué, elegir las herramientas apropiadas y ajustar el desarrollo a modelos, estándares o normas de calidad, que permitan obtener un proceso y producto software de alta calidad.
- Propositiva, mediante la cual el estudiante propone la solución informática a un problema real.

RESULTADOS DE APRENDIZAJE

Soluciono problemas reales que implican los procesos de desarrollo, calidad y evaluación de software, así como la aplicación de las herramientas CASE, para dar soluciones informáticas de calidad en el ámbito empresarial.

Gestiono y construyo proyectos de ingeniería de software específico, utilizando métricas y modelos de evaluación de calidad, que permitan mejorar el proceso de desarrollo de software y la implementación de buenas prácticas en el proceso de construcción de software, mediante el trabajo en equipo y aplicados en contextos empresariales.

METODOLOGÍA

El trabajo en esta asignatura se llevará a cabo básicamente utilizando la “Metodología de desarrollo de software” que determine el docente, la cual está orientada al desarrollo del proceso de enseñanza aprendizaje alrededor de la gestión, planeación, análisis, diseño y construcción de un proyecto de ingeniería de software específico, aplicado a una empresa y que el estudiante adelantará dentro de un proceso de acompañamiento efectuado por el docente, a través de asesorías y clases magistrales.

El docente cumple adicionalmente la labor de participar en el proceso de dimensión de los proyectos, debido a que es recurrente el que los estudiantes al no percibir suficientemente la complejidad de un tema eligen un proyecto que no todas las veces está en el justo medio de los propósitos del conocimiento que tiene la asignatura.

Este trabajo de desarrollar una aplicación real en una empresa se hará a través de la conformación de grupos de trabajo, según el análisis individual que se haya efectuado por parte del docente, con el fin de reunir diferentes tipos de fortalezas en un mismo proyecto y buscar así un trabajo interdisciplinario en equipo. Un valor agregado a las asistencias del docente en la asignatura es la generación de talleres que profundizarán en el conocimiento vago que tenga el estudiante en algunos temas tratados en asignaturas anteriores.

La entrega del proyecto se efectuará antes de concluir el semestre con el fin de ser evaluado con objetividad y poder emitir juicios para la posterior sustentación y socialización de los proyectos.

INVESTIGACIÓN

La investigación que aquí se pretende fomentar está relacionada al desarrollo de software (que se ajuste a los requerimientos del cliente y los usuarios), con aplicación de bases de datos, ingeniería para la WEB, arquitecturas y otros aspectos, así como también la implementación en lenguajes de cuarta y quinta generación.



Las herramientas de desarrollo utilizadas para la implementación de cada proyecto y por los diferentes grupos es tema de consulta y profundización propia de cada estudiante, además se tendrá que dar uso máximo de herramientas CASE, para cada una de las etapas de desarrollo del proyecto.

Aspecto fundamental en el proceso investigativo, se fundamenta en que el trabajo propuesto se respalda en la investigación profunda del estado del arte del proyecto seleccionado, de tal manera que el estudiante se actualice frente a los adelantos tecnológicos y proponga ideas que estén acordes con las innovaciones en el área.

MEDIOS AUDIOVISUALES

Ambiente Aula virtual moodle.
Salas de Informática.
Video Beam

EVALUACIÓN

EVALUACIÓN COLECTIVA

Se llevarán a cabo talleres y sustentaciones, los cuales se desarrollan de manera grupal, con uso de diferentes herramientas que permitan el desarrollo de habilidades en la aplicación de elementos teóricos y técnicos generales de la asignatura. Se desarrollará un proyecto de asignatura en grupos, la evaluación del seguimiento será de manera colectiva.

EVALUACIÓN INDIVIDUAL

Las evaluaciones teóricas y la sustentación del proyecto final se llevarán a cabo de forma individual.

CONTENIDOS TEMÁTICOS CENTRALES

UNIDAD I: PLANEACIÓN, ANÁLISIS Y DISEÑO

- 1.1. Conformación/Constitución del proyecto
- 1.2. Elicitación /Especificación de requisitos funcionales y no funcionales (Story user)
- 1.3. Estimación de actividades del proyecto (Sprints planning)
- 1.4. Diseño detallado y arquitectónico del Sistema de Información (SI)

UNIDAD II: SPRINT 1

- 2.1. Desarrollo requisitos funcionales/ No funcionales Backlog Sprint 1
- 2.2. Diseño y ejecución Test cases Sprint 1
 - 2.2.1. Pruebas unitarias
 - 2.2.2. Pruebas funcionales
 - 2.2.3. Pruebas de aceptación (UAT)
 - 2.2.4. Pruebas de stress / performance
- 2.3. Sprint 1 review
- 2.4. Sprint 1 retrospective
- 2.5. Sprint 2 planning

UNIDAD II: SPRINT 2

- 2.1. Desarrollo requisitos funcionales/ No funcionales Backlog Sprint 2
- 2.2. Diseño y ejecución Test cases Sprint 2



- 2.2.1. Pruebas unitarias
- 2.2.2. Pruebas funcionales
- 2.2.3. Pruebas de aceptación (UAT)
- 2.2.4. Pruebas de stress / performance
- 2.3. Sprint 2 review
- 2.4. Sprint 2 retrospective
- 2.5. Sprint 3 planning

UNIDAD III: SPRINT 3

- 3.1. Desarrollo requisitos funcionales/ No funcionales Backlog Sprint 3
- 3.2. Diseño y ejecución Test cases Sprint 3
 - 3.2.1. Pruebas unitarias
 - 3.2.2. Pruebas funcionales
 - 3.2.3. Pruebas de aceptación (UAT)
 - 3.2.4. Pruebas de stress / performance
- 3.3. Sprint 3 review
- 3.4. Sprint 3 retrospective
- 3.5. Sprint 4 planning

UNIDAD IV: SPRINT 4

- 4.1. Desarrollo requisitos funcionales/ No funcionales Backlog Sprint 4
- 4.2. Diseño y ejecución Test cases Sprint 4
 - 4.2.1. Pruebas unitarias
 - 4.2.2. Pruebas funcionales
 - 4.2.3. Pruebas de aceptación (UAT)
 - 4.2.4. Pruebas de stress / performance
- 4.3. Sprint 4 review
- 4.4. Sprint 4 retrospective

UNIDAD V: SUSTENTACIÓN FINAL DEL PROYECTO

- 4.1. Ajustes finales al sistema de información / Close down del proyecto
- 4.2. Sustentación final del proyecto

LECTURAS MÍNIMAS

Artículos y apartados bibliográficos de los diferentes temas a tratar, proporcionados por el Docente.

BIBLIOGRAFÍA

1. Agarwal, B. B.; Tayal, S. P.; Gupta, M. Software engineering and testing: an introduction. Jones and Bartlett Publishers, 2010
2. Alonso Amo, Fernando. Introducción a la ingeniería del software: modelos de desarrollo de programas. Delta Publicaciones Universitarias, 2014.
3. Bourque, Pierre; Fairley, Richard E. Guide to the software engineering body of knowledge: versión 3.0 : swebok. IEEE Computer Society, 2014.
4. Calero, Coral. et. al. Calidad del producto y proceso software. Ra-Ma Editorial, 2012.
5. Cervantes Maceda, Humberto. Arquitectura de software: conceptos y ciclo de desarrollo. Cengage Learning Editores, S.A. de C.V., 2016.
6. Debrauwer, Laurent; Piqueres Juan, Francisco Javier. Patrones de diseño en Java: los 23 modelos de diseño descripciones y soluciones ilustradas en UML 2 y Java. Ediciones ENI. 2ª. Edición. 2018.
7. Fernández Ledesma, Javier Darío. Ingeniería de software aplicada y orientada a objetos. Universidad

MACROPROCESO: DOCENCIA
PROCESO: GESTIÓN DE PROGRAMAS ACADÉMICOS
PROCEDIMIENTO: FORMULACION O ACTUALIZACION DEL PROYECTO ACADEMICO EDUCATIVO-PAE PARA PROGRAMAS DE PREGRADO
CONTENIDOS PROGRAMATICOS PROGRAMAS DE PREGRADO



Código: D-GPA-P01-F02

Versión: 02

Página 5 de 3

- Pontificia Bolivariana, 2015.
8. Goericke, S. The Future of Software Quality Assurance. Springer Verlag A.G., Tomado de: <https://biblio.uptc.edu.co:2625/content/pdf/10.1007%2F978-3-030-29509-7.pdf>
 9. Gruhn, V. Striemer, R. The Essence of Software Engineering. Springer Verlag A.G. 2018. Tomado de: <https://biblio.uptc.edu.co:2625/content/pdf/10.1007%2F978-3-319-73897-0.pdf>
 10. Kneuper, R. Software Processes and Life Cycle Models. Springer Verlag A.G. 2018. Tomado de: <https://biblio.uptc.edu.co:2625/content/pdf/10.1007%2F978-3-319-98845-0.pdf>
 11. Martinez, J. Desarrollo de interfaces. Editorial Ra-Ma. 2015. Tomado de: <https://biblio.uptc.edu.co:2566/?il=7944>
 12. Müller, P. Schaefer. I. Principled Software Development. Springer Verlag A.G. 2018. Tomado de: <https://biblio.uptc.edu.co:2625/content/pdf/10.1007%2F978-3-319-98047-8.pdf>
 13. Pantaleo, Guillermo. Calidad en el desarrollo de software. Alfaomega. 2ª. Edición. 2016.
 14. Pantaleo, Guillermo; Rinaudo, Ludmila Lis. Ingeniería de Software. Alfaomega. 1ª. Edición. 2015.
 15. Pfleeger, Shari Lawrence. Ingeniería del Software, Teoría y Práctica. Prentice Hall 2001, 3ª. Edición.
 16. Piattini Velthuis, Mario G. Análisis y diseño de aplicaciones informáticas de gestión : una perspectiva de ingeniería del software. Alfaomega Ra-Ma, 2004.
 17. Piattini , M, García, F. Medición de Software. Editorial Ra-Ma. 2015. Tomado de: <https://biblio.uptc.edu.co:3040/reader/medicion-de-software-1592585484>
 18. Pressman, Roger S. Ingeniería del software: Un enfoque práctico. McGraw-Hill, 7ª. Edición. 2010
 19. Reussner, R. et. al. Managed Software Evolution. Springer Verlag A.G. 2019. Tomado de: <https://biblio.uptc.edu.co:2625/content/pdf/10.1007%2F978-3-030-13499-0.pdf>
 20. Rumbaugh, James. et. al. El lenguaje unificado de modelado manual de referencia. Addison-Wesley, 2007.
 21. Sánchez Alonso, Salvador; Sicilia Urbán, Miguel Ángel; Rodríguez García, Daniel. Ingeniería del software: un enfoque desde la guía SWEBOK. Alfaomega. 2012.
 22. Sangwan, Raghvindeer S. Software and Systems Architecture in Action. Taylor & Francis Group, 2015
 23. Sanjuan Martínez, Oscar. et. al. Utilización de UML en ingeniería del software con objetos y componentes. Addison-Wesley, 2002
 24. Schach, Stephen R. et. al. Ingeniería de software clásica y orientada a objetos. McGraw-Hil. 6a ed. 2006.
 25. Senn, James A. Análisis y diseño de sistemas de información. McGraw-Hill.
 26. Sommerville, Ian. Ingeniería del Software. Prentice Education. 10ª Edición. 2016. Tomado de: <https://biblio.uptc.edu.co:2566/?il=7500>
 27. Whitten, Jeffrey L. Análisis y diseño de sistemas de información. McGraw-Hill.
 28. Tsui, Frank. Essentials of software engineering. Jones & Bartlett Learning. 3ª. Edición. 2014.
 29. Vizcaino, A. Desarrollo global de Software. Editorial Ra-Ma. 2015. Tomado de: <https://biblio.uptc.edu.co:2566/?il=7928>
 30. Zykov, S. Managing Software Crisis: A Smart Way to Enterprise Agility. Springer Verlag A.G. 2018. Tomado de: <https://biblio.uptc.edu.co:2625/content/pdf/10.1007%2F978-3-319-77917-1.pdf>