

UNIVERSIDAD NACIONAL DE ASUNCIÓN
FACULTAD POLITÉCNICA
LICENCIATURA EN CIENCIAS INFORMÁTICAS
PLAN 2009
PROGRAMA DE ESTUDIOS

Resolución N° 18/26/16-00 Acta N° 1041/17/12/2018 - ANEXO 04

I. - IDENTIFICACIÓN

- | | | |
|------|---------------------------------|-----------------------------|
| 1. | Asignatura | : Ingeniería de Software II |
| 2. | Grupo | : Siete |
| 3. | Horas semanales | : 5 horas |
| 3.1. | Clases teóricas | : 2 horas |
| 3.2. | Clases prácticas | : 3 horas |
| 4. | Total real de horas disponibles | : 90 horas |
| 4.1. | Total de Clases teóricas | : 36 horas |
| 4.2. | Total de clases prácticas | : 54 horas |

II. - JUSTIFICACIÓN

Esta asignatura tiene como finalidad el establecimiento y aplicación de los principios de la Ingeniería en el desarrollo de proyectos de software, considerando factores económicos, la fiabilidad del sistema, rendimiento y funcionamiento eficiente de acuerdo a las necesidades del usuario, reduciendo costos y complejidad.

III. - OBJETIVOS GENERALES

1. Describir las principales problemáticas que caracterizan el proceso de software en las distintas etapas de su ciclo de vida
2. Utilizar patrones de diseño en el diseño e implementación de software
3. Identificar las ventajas y desventajas en un proceso de selección de plataformas de diseño y desarrollo de software
4. Aplicar adecuadamente métodos y técnicas para la gestión de la configuración y el desarrollo e implementación del software

IV. - OBJETIVOS ESPECIFICOS

A. Conocimientos

1. Distinguir entre procesos del ciclo de vida, modelos del ciclo de vida y metodologías de ingeniería del software
2. Describir los atributos de calidad de software
3. Identificar los artefactos y sus características que constituyen la configuración de un sistema

B. HABILIDADES

1. Aplicar el modelo de ciclo de vida adecuado al desarrollo de cada producto de software
2. Comprender la utilidad y la responsabilidad de los ingenieros de software, asumiendo su código ético
3. Aplicar los niveles del diseño de software: diseño detallado de componentes, de interfaz de usuario, de datos y arquitectónico
4. Evaluar y seleccionar plataformas para el desarrollo y ejecución de aplicaciones
5. Integrar la arquitectura propuesta en un proceso de desarrollo de software
6. Presentar y negociar una arquitectura de software con los stakeholders (involucrados)
7. Aplicar patrones de diseño detallado a la solución
8. Gestionar la configuración como un elemento básico en la creación de software'

C. COMPETENCIAS

1. Aplicar destrezas en el manejo de productos y herramientas para el diseño y desarrollo de sistemas de información
2. Integrar las habilidades necesarias para realizar el modelado y resolución de problemas en el ámbito de los sistemas de información
3. Aplicar metodologías para el desarrollo rápido de sistemas de información
4. Desarrollar habilidades de comunicación personal efectiva, trabajo en grupo y relación con usuarios utilizando técnicas clásicas y herramientas informática

V. - PRE - REQUISITO

Ingeniería de Software I

VI. - CONTENIDO

6.1. Unidades programáticas

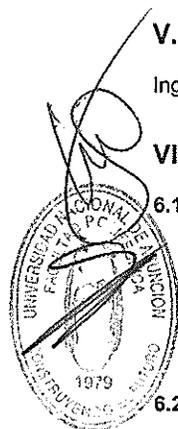
1. Diseño de software
2. El desarrollo de software
3. Pruebas de software
4. Gestión de la configuración del software (scm)
5. Gestión del proceso de ingeniería de software

6.2. Desarrollo de las unidades programáticas

1. DISEÑO DE SOFTWARE

Aprobado por Resolución N° 18/26/16-00 Acta N° 1041/17/12/2018 del Consejo Directivo de la FP-UNA

Página 1 de 4



- 1.1. Estructura y arquitectura del software
 - 1.1.1. Estructuras arquitectónicas y Viewpoints
 - 1.1.2. Patrones de diseño (patrones de microarquitectura)
 - 1.1.3. Familias de programas y frameworks
2. EL DESARROLLO DE SOFTWARE
 - 2.1. Fundamentos del desarrollo de Software
 - 2.1.1. Minimizar la complejidad
 - 2.1.2. Anticipar el cambio
 - 2.1.3. Desarrollar para la verificación
 - 2.1.4. Normas en el desarrollo
 - 2.2. Gestión del desarrollo
 - 2.2.1. Los modelos del desarrollo
 - 2.2.2. Planificación del desarrollo
 - 2.2.3. Medición del desarrollo
 - 2.3. Consideraciones prácticas
 - 2.3.1. Diseño del desarrollo
 - 2.3.2. Lenguajes del desarrollo
 - 2.3.3. Codificación
 - 2.3.4. Pruebas del desarrollo
 - 2.3.5. Reutilización
 - 2.3.6. Calidad del desarrollo
 - 2.3.7. Integración
3. PRUEBAS DE SOFTWARE Y CALIDAD
 - 3.1. Diferencias entre Pruebas y QA
 - 3.2. Fundamentos de las pruebas del software
 - 3.3. Los niveles de pruebas
 - 3.3.1. El objetivo de la pruebas
 - 3.3.2. Pruebas unitarias
 - 3.3.3. Pruebas funcionales
 - 3.4. Proceso de Prueba
 - 3.4.1. Modelo, Cobertura, Evaluación
 - 3.4.2. Actividades y Productos principales
 - 3.4.2.1. Diseño
 - 3.4.2.2. Ejecución
 - 3.4.2.3. Reporte
 - 3.4.3. Estandarización
 - 3.4.3.1. ISO/IEC 29119: Standard for Software Testing
 - 3.5. Calidad del Software
 - 3.5.1. Fundamentos
 - 3.5.2. Calidad de Producto, Proceso y Calidad en Uso
 - 3.5.3. Métricas
 - 3.5.4. Estándares de Calidad
 - 3.6. Testing Ágil
 - 3.6.1. Principios del testing ágil
 - 3.6.2. Prácticas de testing ágil
 - 3.6.2.1. Test Driven Development (TDD)
 - 3.6.2.2. Acceptance Test Driven Development (ATDD)
 - 3.6.2.3. Behaviour Driven Development (BDD)
 - 3.6.2.4. Automatización de pruebas
4. GESTION DE LA CONFIGURACION DEL SOFTWARE (SCM)
 - 4.1. Gestión del Proceso de SCM
 - 4.1.1. Contexto Organizacional de SCM
 - 4.1.2. Limitaciones y orientación para el proceso del SCM
 - 4.1.3. La planificación de SCM
 - 4.1.3.1. Organización y responsabilidades en el SCM
 - 4.1.3.2. Recursos y programación del SCM
 - 4.1.3.3. Selección de herramientas e implementaciones
 - 4.1.4. Plan del SCM
 - 4.1.5. Supervisión de la Gestión de Configuración de Software
 - 4.1.5.1. Auditorías internas al proceso del SCM
 - 4.2. Identificación de la Configuración del Software
 - 4.2.1. Identificación de elementos que hay que controlar
 - 4.2.1.1. Configuración del software
 - 4.2.1.2. Elemento de configuración de software
 - 4.2.1.3. Relaciones entre los elementos de configuración de software
 - 4.2.1.4. Versión de software
 - 4.2.1.5. Línea Base
 - 4.2.1.6. Librería software
 - 4.3. Control de la Configuración del Software
 - 4.3.1. Solicitar, evaluar y aprobar los cambios de software
 - 4.3.1.1. Cuadro de Control de la Configuración del software
 - 4.3.1.2. Proceso de solicitud de Cambio de software
 - 4.4. Gestión de versiones y entregas del Software
 - 4.4.1. La construcción de software
 - 4.4.2. Gestión de la versión del Software
- GESTIÓN DEL PROCESO DE INGENIERIA DE SOFTWARE
 - 5.1. Iniciación y Definición del Alcance
 - 5.1.1. La determinación y la negociación de los requisitos
 - 5.2. Planificación del proyecto Software
 - 5.2.1. Proceso de Planificación
 - 5.2.2. Determinar Entregables
 - 5.2.3. Esfuerzo, Programación y Estimación de costos
 - 5.2.4. Asignación de recursos



- 5.3. Gestión de Riesgos
- 5.4. Gestión del plan
- 5.5. Gestión de contratos con Proveedores
 - 5.5.1. Revisión y Evaluación del Desempeño
- 5.6. Planificación del proceso de medición
 - 5.6.1. Realizar el proceso de medición
 - 5.6.2. Evaluar Medición

VII. - ESTRATEGIAS METODOLÓGICAS

1. Clases teóricas: clases magistrales, exposición y discusión de los conocimientos básicos de la asignatura
2. Planteamiento y resolución de ejercicios y problemas en aula
3. Lecturas, discusiones, estudio de casos
4. Prácticas de laboratorio en las que se llevará a cabo el desarrollo de un Proyecto de Software utilizando herramientas, técnicas y métodos objeto de estudio de la asignatura, contribuyendo al desarrollo de la capacidad de análisis, razonamiento crítico y comprensión de las prácticas utilizadas.
5. Tutoría a distancia a través de la plataforma Educa

VIII. - MEDIOS AUXILIARES

1. Equipo multimedia
2. Pizarra.
3. Marcadores.
4. Borrador de pizarra.
5. Para las clases prácticas: laboratorio de PC.
6. Computadoras personales.
7. Plataforma virtual Educa

IX. - EVALUACIÓN

1. Exámenes Parciales

Para cada examen la distribución del puntaje de la evaluación será la siguiente:

- Proyecto de Software: 50%
Se hará una revisión del proyecto de software en base al avance previsto para la fecha del examen
- Examen Teórico: 50%
Abarcará lo desarrollado en el periodo correspondiente

2. Exámenes Finales

Para cada examen la distribución del puntaje de la evaluación será la siguiente:

- Proyecto de Software: 50%
Se evaluará la culminación en tiempo y forma del proyecto de software, cumpliendo los requerimientos establecidos en la rúbrica y en las especificaciones de requerimiento
- Examen Teórico: 50%
Abarcará todo el contenido desarrollado durante el semestre

Las calificaciones se rigen por el reglamento de cátedra de la FP-UNA.

X. - BIBLIOGRAFÍA

Complementaria

- Bass, L., Clements, P. & Kazman, R. (2003). *Software Architecture in Practice* (2° Ed.) Addison-Wesley Professional.
- Clements, P., Bachmann, F. & Bass L. (2008). *Documenting Software Architecture: Views and Beyond*. Ed. Addison-Wesley Professional.
- Piattini, M. (2007). *Análisis y diseño detallado de aplicaciones informáticas de gestión*. Ra-Ma.
- Shaw M. & Garlan, D. (1996). *Software Architecture: Perspectives on an Emerging Discipline*. Prentice Hall.
- Taylor, R. N., Medvidovic N. & Dashofy, E.M. (1996). *Software Architecture: Foundations, Theory and Practice*. Wiley.

DISPONIBLES EN LA COLECCIÓN DE LA BIBLIOTECA DE LA FP-UNA

- Braude, E. J. (2003). *Ingeniería de software: una perspectiva orientada a objetos*. México: Alfaomega
- Calero, C. (Ed.) (2015). *Green in software engineering*. Springer.
- Çengel, Y. A. & Palm, W. J. (2014). *Ecuaciones diferenciales para ingeniería y ciencias*. México: McGraw-Hill.
- Gamma, E., Helm, R., Johnson, R. & Vlissides, J. (2003). *Patrones de diseño: elementos de software orientado a objetos reusable*. Madrid: Pearson Educación.
- Jacobson, I., Grady Boock & Rumbaugh, J. (2000). *El proceso unificado de desarrollo de software*. Madrid: Pearson Educación
- Jones, C. (2007). *Estimación de costos y Administración de proyectos de software*. (2° Ed.). México: McGraw-Hill
- Larman, C. (2002). *UML y Patrones: una introducción al análisis y diseño orientado a objetos y al proceso unificado*. (2° Ed.). Madrid: Prentice Hall.

- Mortier, G. (2005). Técnicas de programación: guía fundamental de desarrollo de software. Buenos Aires: MP Ediciones.
- Pressman, R. S. (2002). *Ingeniería del software: un enfoque práctico*. (5° ed.) Madrid: McGraw-Hill.
- Pressman, R. S. (2010). *Ingeniería del software: un enfoque práctico*. (7° ed.). México: McGraw-Hill.
- Sánchez Alonso, S., Sicilia Urbán, M. Á. & Rodríguez García (2011). *Ingeniería del software: un enfoque desde la guía SWEBOK*. Madrid: Ibergaceta.
- Sommerville, I. (2011). *Ingeniería de software*. (9° ed.). México: Addison Wesley.
- Weitzenfeld, A. (2005). *Ingeniería del software orientada a objetos con UML, java e internet*. México: Thomson.

LIBROS ELECTRONICOS DISPONIBLES EN LA COLECCIÓN MGH

Pressman, R. (2010). *Ingeniería del software: un enfoque práctico (7a. ed.)*. Retrieved from <http://ebookcentral.proquest.com>

