



Provincia de Córdoba



Ministerio de Educación - D.G.E. T. Y F .P

INSTITUTO TÉCNICO SUPERIOR CORDOBA

Extensión Áulica Villa el Libertador

Defensa y Cacheuta S/N - Villa el Libertador - Córdoba

Planificación de:

Ingeniería de Software

Carrera: TECNICATURA SUPERIOR EN DESARROLLO DE SOFTWARE

Resolución N° 462/2012

Espacio Curricular Ingeniería de Software

Docente Juan Manuel Scavuzzo

Ciclo Lectivo **2019**

Modalidad Anual
Año Tercero
División A
Área de formación Específica

Horas reloj Anuales 107 hs
Horas Cátedra Anuales 160 hs C
Horas Cátedra Semanales 5 hs C

CONTENIDOS DESTACADOS:

Programas de trabajo y resolución conjunta de problemas.
 Modelos de ciclo de vida del software y de procesos de desarrollo.
 El problema del mantenimiento y las migraciones de plataforma.
 Metodologías tradicionales y ágiles. Metodologías ágiles, concepto de "sprint", fraccionamiento del producto en unidades realizables en un "sprint", cola de pendientes, mejora de productos provisorios (refactoring), variación de los roles y la documentación en el marco de un proceso en el que se aplican metodologías ágiles.
 Gestión de los cambios, conceptos de versión, "Guild", producto de la asignación.
 El problema de asegurar la calidad y elementos de métricas.
 Errores, Defectos, Fallas. Calidad de Software y Aseguramiento de la calidad. Pruebas
 Programacion Orientada a Objetos: Clases, objetos, Herencia , abstraccion, encapsulamiento, polimorfismo.
 Patrones de Programacion.

Programa Analítico (páginas): 2 a 5 de 5

Bibliografía (páginas): 5 de 5

Correlativas Obligatorias: Modelado de Sistemas

Correlativas Aconsejadas: -

El presente carece de validez sin la certificación de la Secretaría

Sello

Firma

Lugar y Fecha

1. FUNDAMENTACIÓN

La Ingeniería en Software está enfocada en las inconveniencias y situaciones que derivan de la construcción de aplicaciones/programas de gran porte ó pequeños también (software). En estos desarrollos la magnitud de la situación exige cierto orden y metodología de trabajo.

La ingeniería que nos atañe incluye las técnicas y métodos necesarios para cumplimentar con los objetivos, el término Ingeniería se presenta como canal para poner especial énfasis en indicar un método, una metodología SISTEMICA, dentro de un contexto intencional y organización cada vez más demandante y cambiante (globalizado). En la evolución del desarrollo de software se observan varios cambios de paradigmas, actualmente los elementos que más destacan son:

Tamaño y Complejidad (Software muy “grande”, cada vez con más capas de desarrollo y con mayor complejidad).

Equipos de desarrollo distribuidos (ya sea localmente o entre países)

Herramientas de colaboración que permiten comunicaciones antes no posibles (emails, mensaje de texto, videoconferencias).

Mayor nivel de integración, nunca antes visto, entre redes, sistemas, aplicaciones y entre personas y tecnología (la propia Internet es un claro ejemplo).

Mayor avance de nuevas tecnologías (en los últimos años hemos visto infinidad de “frameworks” y nuevos modelos de trabajo).

Creciente orientación de las tecnologías al negocio.

Los elementos mencionados no estaban presentes en la década del 60, en donde se desarrollaron la mayoría de las metodologías de desarrollo tradicionales, estos cambios de paradigmas dan origen a la Ingeniería en Software como tal.

Tal vez la primera definición sobre el tema se dio en la conferencia de NATO (NAUR AND RANDELL 1968), usando el término “Software engineering”

2. OBJETIVOS

Introducción a la Ingeniería del Software como una práctica enfocada en solucionar el caos que enfrenta la construcción de software, hoy a cualquier escala.

La materia se propone proveer al alumno de conocimientos teóricos sobre temas de avanzada en la Ingeniería del software, que serán desarrollados en profundidad durante el año, generando una síntesis de los conocimientos que el alumno ha recibido a lo largo de la carrera.

Asimismo, se propone el desarrollo de trabajos prácticos de aplicación profesional de modo tal de marcar una fuerte aplicación práctica de los conocimientos y el desarrollo de trabajo en equipo.

Desde esta perspectiva, los objetivos de la asignatura son que el alumno logre:

- Analizar y discutir los fundamentos de la Ingeniería de Software como una disciplina científica.
- Definir sus alcances, limitaciones y ejes problemáticos.
- Conocer los diferentes modelos de procesos software y los diferentes modelos de ciclo de vida del producto. Metodologías clásicas y metodologías ágiles.
- Analizar el marco de la Gestión de Proyectos de desarrollo de software
- Aplicar las diferentes metodologías, técnicas y herramientas en la gestión de Proyectos.
- Conocer y aplicar políticas de Gestión de configuración del software.
- Conocer y aplicar políticas de Gestión de Calidad de software
- Conocer y aplicar métricas de calidad de software.
- Conocer las nuevas tendencias sobre la ingeniería de software.

3. APRENDIZAJES Y CONTENIDOS

Unidad Didáctica N° 1 – Ingeniería del software, Metodologías

Que es la Ingeniería en software y para que nos sirve

Etapas del desarrollo de software: Relevamiento, Análisis, Diseño, Desarrollo, Testing.

Programas de trabajo y resolución conjunta de problemas.

Modelos de ciclo de vida del software y de procesos de desarrollo. El problema del mantenimiento y las migraciones de plataforma.

Metodologías tradicionales y ágiles. Metodologías ágiles. SCRUM. Concepto de sprint, fraccionamiento del producto en unidades realizables en un sprint. cola de pendientes, mejoras de productos provisionales (refactoring), variación de los roles y la documentación, marco de un proceso en el que se aplican metodologías ágiles.

Unidad Didáctica N° 2 – Calidad y Pruebas de Software

Definiciones Errores, Defectos, Fallas. Calidad de Software y Aseguramiento de la calidad. Problema de asegurar la calidad y elementos de métricas.

Que son las Pruebas de Software. Pruebas según nivel de integración. Pruebas de Caja Blanca y Caja Negra, Metodologías. Cobertura. Test Unitario. Implementación de UT en Python. Mocks. Mocks

en Python.

Unidad Didáctica N° 3 – Gestión de la Configuración

Control de Versiones. Como nos ayuda el control de Versiones. Comparativa de los diferentes CV mas usados.

Fundamentos GIT. Ramificaciones en GIT. Uso y configuracion de Proyecto en Github.

Conceptos de Build. Integracion Continua. Objetivo. Metodologia. Ventajas. Desventajas. Herramientas. Implementacion Practica. Metricas

Unidad Didáctica N° 5 – Objetos en python

Programacion Orientada a Objetos: Clases, objetos, Herencia, abstraccion, encapsulamiento, polimorfismo. Implementecion con Python 3. Metodos inherentes a las clases en Pyhton 3.

Conceptos de Coherencia y Cohecion en Clases. Patrones generales de software para asignación de responsabilidades GRASP.

Unidad Didáctica N° 6 – Conceptos avanzados

Exepciones. como se usan y cuando. Decoradores. Unicode.

API y Frameworks.

Patrones. Para que se usan. Implementacion. Analisis en Django y Qt.

4. METODOLOGÍA

Clases teoricas con filminas y ejemplos.

Clases practicas con dinamicas en clases y uso del laboratorio de computación.

5. INSTANCIAS Y CRITERIOS DE EVALUACIÓN

Se realizarán instancias de exámenes parciales orales abordando los contenidos desarrollados en clase. Además del nivel conceptual se evalúa capacidad de expresión.

CONDICIÓN DE REGULARIDAD:

- Asistir al 75% de las clases
- Aprobar todas las evaluaciones parciales orales o sus respectivos recuperatorios con un mínimo de 4 puntos equivalente al (50%).

CONDICIÓN DE PROMOCIÓN:

- Asistir al 75% de las clases
- Aprobar todas las evaluaciones parciales orales con un mínimo de 6 puntos y promedio de 7 puntos.
- Entregar y defender trabajo práctico final con un mínimo de 7 puntos.

6. BIBLIOGRAFÍA

BIBLIOGRAFÍA BÁSICA

- Ingeniería de Software - Ian Somerville.
- <https://git-scm.com/book/es/v2/Inicio---Sobre-el-Control-de-Versiones-Acerca-del-Control-de-Versiones>
- Object Oriented Software Construction - Bertrand Meyer

BIBLIOGRAFÍA COMPLEMENTARIA

- Software Quality Assurance from Theory to Implementation - Daniel Galin.
- Python 3 Patterns, Recipes and Idioms - Bruce Eckel

SITIOS WEB SUGERIDOS

- <http://docs.python.org.ar/tutorial/>
- <https://docs.python.org/3/>
- <http://agilemethodology.org/>
- <http://www.agilemanifesto.org/iso/es/>
-