

**ESCUELA POLITÉCNICA NACIONAL****PROGRAMA DE ESTUDIOS POR ASIGNATURA****SECCIÓN 1. DATOS GENERALES DE LA ASIGNATURA**

|                       |   |                                       |             |
|-----------------------|---|---------------------------------------|-------------|
| UNIDAD ACADÉMICA:     | <b>FACULTAD DE INGENIERÍA DE SISTEMAS</b>                   |                                       |             |
| CARRERA:              | <b>INGENIERÍA EN SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN</b> |                                       |             |
| EJE DE FORMACIÓN:     | <b>PROFESIONAL</b>  |                                       |             |
| ASIGNATURA:           | <b>INGENIERIA DE SOFTWARE II</b>                            |                                       |             |
| CÓDIGO:               | <b>SIC634</b>   | PENSUM:                               | <b>2009</b> |
| SEMESTRE REFERENCIAL: | <b>6</b>  | NRO. CRÉDITOS:                        | <b>4</b>    |
| TIPO:                 | Obligatoria: <b>X</b>                                       | Optativa:                             |             |
| HORAS SEMANALES:      | Teóricas: <b>4</b>  | Prácticas de Laboratorio /Ejercicios: | <b>0</b>    |
| TOTAL DE HORAS:       | Teóricas: <b>56</b>   | Prácticas de Laboratorio /Ejercicios: | <b>0</b>    |
|                       |   | Actividades de Evaluación:            | <b>8</b>    |

**ASIGNATURAS PRE-REQUISITOS:**

SIC544 Ingeniería de Software I

**ASIGNATURAS CO-REQUISITOS:**

Ninguna

**SECCIÓN 2. RESULTADOS DEL APRENDIZAJE DE LA ASIGNATURA****De conocimientos:**

Aplicar los fundamentos del Desarrollo de Software basado en procesos ágiles utilizando elementos métodos, técnicas, notaciones, herramientas y procedimientos teóricos, prácticos y de investigación. Explicar por qué la creación de componentes de programas correctos es importante en la producción de alta calidad del software.

Describir el comportamiento de un componente del programa.

**De destrezas:**

Adaptar los elementos pertinentes en el uso de procesos ágiles para el desarrollo de software.

Evaluar la ejecución de una variedad de segmentos de código.

Identificar errores de codificación comunes que conducen a programas inseguros (por ejemplo, desbordamientos de buffer, pérdidas de memoria, código malicioso) y aplicar estrategias para evitar este tipo de errores. [Uso]

Llevar a cabo una revisión de código personal (centrado en los errores comunes de codificación) en un componente del programa usando un siempre lista. [Uso]

Contribuir a una revisión de código centrarse en la corrección de los componentes. [Uso]

Refactorizar un programa mediante la identificación de oportunidades para aplicar la abstracción procedimental. [Uso]

Aplicar una variedad de estrategias para la prueba y depuración de programas sencillos. [Uso]

Construir , ejecutar y depurar programas utilizando un IDE moderno y herramientas asociadas, tales como herramientas de pruebas unitarias y depuradores visuales. [Uso]

Construir programas, depurarlos utilizando las bibliotecas estándar disponibles con un lenguaje de programación elegido. [Uso]

Analizar el grado en que el código de otro programador reúne documentación y estilo de normas de programación.

Aplicar estándares de documentación y estilos del programa coherentes que contribuyan a la legibilidad y de mantenimiento de software. [Uso]

De valores y actitudes:

Fomentar la responsabilidad, la ética profesional y el trabajo en equipo en el desarrollo de software.

## SECCIÓN 3. DETALLE DE LA ASIGNATURA

### Capítulo 1: EVOLUCIÓN DE SOFTWARE

- 1.1 Desarrollo de software en contextos industriales, códigos de bases pre-existentes
- 1.2 Evolución de software. Paradigmas tradicionales vs Paradigmas Agiles
- 1.3 Características del software mantenible
- 1.4 Reingeniería de sistemas
- 1.5 Reusabilidad de Software

### Capítulo 2: ETAPAS DEL PROCESO DE DESARROLLO DE SOFTWARE

- 2.1 Análisis de Requisitos o Elicitación de Requerimientos
- 2.2 Especificación de Requerimientos
- 2.3 Diseño y Arquitectura
- 2.4 Implementación
- 2.5 Pruebas
- 2.6 Implantación
- 2.7 Mantenimiento
- 2.8 Gestión de la configuración. Gestión de cambios y gestión de versión
- 2.9 Aseguramiento de la calidad de software

### Capítulo 3: FILOSOFÍA Y ENFOQUES ÁGILES

- 3.1 Desarrollo ágil de software
- 3.2 Proceso Unificado Ágil (AUP)
- 3.3 Big Design Up Front (BDUF)
- 3.4 Metodología de diseño constructorista (CDM)
- 3.5 Desarrollo de diseño-dirigido (D3)
- 3.6 No te repitas (DRY) o Una y sólo una vez (O3)
- 3.7 Método de desarrollo de sistemas dinámicos (DSDM)
- 3.8 Principio KISS (Keep It Simple, Stupid) "(Déjelo simple)"
- 3.9 Enfoque MIT
- 3.10 Model Driven Architecture (MDA)

### Capítulo 4: METODOLOGÍAS ÁGILES MÁS UTILIZADAS

- 4.1 XP o Programación Extrema.
- 4.2 Scrum
- 4.3 Crystal
- 4.4 Evolutionary Project Management (Evo)
- 4.5 Feature Driven Development (FDD)
- 4.6 Adaptive Software Development (ASD)
- 4.7 Lean Development (LD) y Lean Software Development (LSD)
- 4.8 RUP Ágil, existiendo dos variantes AUP y EUP
- 4.9 AM (Agile Modeling)

### Capítulo 5: XP, PROGRAMACIÓN EXTREMA

- 5.1 Qué es XP?
- 5.2 Cómo utilizar XP?
- 5.3 Beneficios de XP
- 5.4 Implementación de XP en el desarrollo de un producto
- 5.5 Herramientas que soportan XP

### Capítulo 6: SCRUM

- 6.1 Qué es SCRUM?
- 6.2 Cómo utilizar SCRUM?
- 6.3 Beneficios de SCRUM
- 6.4 Implementación de SCRUM en el desarrollo del producto

## 6.5 Herramientas que soportan SCRUM

## Capítulo 7: MSF ÁGIL

- 7.1 Qué es MSF AGIL?
- 7.2 Cómo utilizar MSF AGIL?
- 7.3 Beneficios de MSF AGIL
- 7.4 Implementación de MSF AGIL en el desarrollo del producto
- 7.5 Herramientas que soportan MSF AGIL

## Capítulo 8: RUP ÁGIL

- 8.1 Qué es RUP AGIL?
- 8.2 Cómo utilizar RUP AGIL?
- 8.3 Beneficios de RUP AGIL
- 8.4 Implementación de RUP AGIL en el desarrollo del producto
- 8.5 Herramientas que soportan RUP AGIL

## Capítulo 9: INTEGRACIÓN DE MEJORES PRÁCTICAS Y METODOLOGÍAS ÁGILES

- 9.1 Integración de diversas metodologías ágiles de desarrollo de software.
- 9.2 Metodologías ágiles contrapuestas.
- 9.3 Metodologías ágiles complementarias.
- 9.4 Como escalar o maximizar el alcance de las metodologías ágiles.

## BIBLIOGRAFÍA BÁSICA:

|   |   |
|---|---|
| 1 | Pressman, Poger. S Ingeniería del Software - Un Enfoque Práctico (7taEd). Ms Graw Hill España 2010. |
|---|---|

## BIBLIOGRAFÍA COMPLEMENTARIA:

|   |   |
|---|---|
| 1 | Fuente Wikipedia(2013), Ingeniería de Software: Herramientas de Diseño Asistido, Escalabilidad, Systems Development Life Cycle, Idef, Prototype, Pruebas de Rendimiento Del So, books LLC, ISBN:123070874X, 9781230708744 |
| 2 | Sommerville I(2005) Pearson Education, ISBN: 84-7829-074-5, septima edición   |
| 3 | Mike Cohn(2005). Agile Estimating and Planning Robert CMartin Series  |
| 4 | Shore J, Warden Sh(2007). The Art of Agile Development  |

**SECCIÓN 4. SUGERENCIAS DIDÁCTICAS**

|                                     |                                     |                           |                                     |
|-------------------------------------|-------------------------------------|---------------------------|-------------------------------------|
| Exposición oral (clase magistral)   | <input checked="" type="checkbox"/> | Exposición audiovisual    | <input checked="" type="checkbox"/> |
| Ejercicios dentro de clase          | <input checked="" type="checkbox"/> | Ejercicios fuera del aula | <input checked="" type="checkbox"/> |
| Conferencias (profesores invitados) | <input type="checkbox"/>            | Lecturas obligatorias     | <input checked="" type="checkbox"/> |
| Prácticas de laboratorio            | <input type="checkbox"/>            | Prácticas de campo        | <input type="checkbox"/>            |
| Trabajos de investigación           | <input checked="" type="checkbox"/> | Desarrollo de un proyecto | <input type="checkbox"/>            |
| Otras                               | <input type="checkbox"/>            |                           |                                     |

**SECCIÓN 5. FORMAS DE EVALUAR**

|                                  |                                     |                        |                                     |
|----------------------------------|-------------------------------------|------------------------|-------------------------------------|
| Pruebas parciales                | <input checked="" type="checkbox"/> | Examen final           | <input checked="" type="checkbox"/> |
| Trabajos y tareas fuera del aula | <input checked="" type="checkbox"/> | Asistencia a prácticas | <input type="checkbox"/>            |
| Participación en clase           | <input type="checkbox"/>            | Otras                  | <input type="checkbox"/>            |

**SECCIÓN 6. REQUISITOS DE EXPERIENCIA Y CONOCIMIENTOS DEL PROFESOR**

Formación como ingeniero de sistemas.  
Experiencia en proyectos de desarrollo de software con procesos ágiles.  
Capacitación o experiencia en docencia superior

## SECCIÓN 7. REQUERIMIENTOS DE INFRAESTRUCTURA/RECURSOS

Software de aplicación para metodos agiles. Pruebas metodológicas

FECHA DE ELABORACIÓN DEL PEA:  
RESPONSABLE DE ELABORACIÓN DEL PEA:

17 - Septiembre - 2014  
Comisión Reforma Curricular, FIS

FECHA ÚLTIMA REVISIÓN DEL PEA:  
RESPONSABLE DE ÚLTIMA REVISIÓN DEL PEA:

19 - Septiembre - 2014  
PhD. Jenny Torres