



*Universidad Autónoma “Gabriel René Moreno”*  
**FACULTAD DE INGENIERIA EN CIENCIAS  
DE LA COMPUTACIÓN Y TELECOMUNICACIONES**



**PROGRAMA ANALÍTICO DE ASIGNATURA**

**DATOS GENERALES**

<b>ASIGNATURA</b>	: Tecnologías Emergentes II
<b>SIGLA Y CODIGO</b>	: ELC - 010
<b>PERIODO</b>	: Sexto Semestre
<b>REQUISITOS</b>	: Ninguno
<b>HORAS</b>	: 6
<b>CREDITOS</b>	: 3
<b>PROFESOR</b>	: Ing.
<b>SISTEMA</b>	: Semestralizado
<b>PROGRAMA VIGENTE</b>	: 2014
<b>REVISADO EN</b>	: 5 de Marzo 2014

**FUNDAMENTACION:**

La aparición continua de nuevos productos y servicios, cuyo ciclo de vida se acorta, obliga a incrementar la productividad y disminuir el tiempo de reacción, adaptarse rápidamente a las variantes necesidades de los clientes y, en definitiva aumentar la capacidad de competir en un mercado mucho más amplio. Esta situación afecta muy especialmente a la industria software pues está llamada a ser, y de hecho ya es el pilar que sustenta y aporta valor al resto de sectores industriales.

Tomando en cuenta que un profesional formado en Ing. de Sistemas, necesita un marco de trabajo basado principios y buenas prácticas de ingeniería, para la producción de software orientado a satisfacer la demanda tecnológica en la pequeña y mediana empresa, utilizando el Framework SCRUM como marco de trabajo, en el desarrollo del



*Universidad Autónoma “Gabriel René Moreno”*  
**FACULTAD DE INGENIERIA EN CIENCIAS  
DE LA COMPUTACIÓN Y TELECOMUNICACIONES**



producto y Programación Extrema (XP) como metodología de gestión del proyecto dentro de un contexto ágil de desarrollo de software.

**OBJETIVO GENERAL:**

El objetivo principal de la materia es proponer el usos de metodologías y framework para la gestión de proyectos de ingeniería de software, basado en entornos ágiles y enfoques que calidad orientado a la pequeña y mediana empresa.

**OBJETIVOS ESPECÍFICOS:**

Al finalizar la asignatura el alumno será capaz de:

- Identificar los requerimientos de información y procesos existentes para la gestión y seguimientos de proyectos de ingeniería de software.
- Analizar los requerimientos, procesos y manuales revisando la alineación con las buenas prácticas de gestión de proyectos.
- Diseñar los instrumentos, artefactos del software y usar las herramientas que sustentan el proceso de ingeniería siguiendo un enfoque ágil.
- Desarrollar y definir el enfoque de calidad en base a los requerimientos propios de un proceso de desarrollo ágil.
- Aplicar la metodológica y framework en un caso de estudio como evidencia de la viabilidad de la propuesta.



*Universidad Autónoma “Gabriel René Moreno”*  
**FACULTAD DE INGENIERIA EN CIENCIAS  
DE LA COMPUTACIÓN Y TELECOMUNICACIONES**



## **CONTENIDO MÍNIMO**

UNIDAD I: INTRODUCCION A LAS METODOLOGIAS, FRAMEWORK Y ENTORNOS AGILES.

UNIDAD II: EL FRAMEWORK SCRUM.

UNIDAD III: METODOLOGIA PROGRAMACION EXTREMA (XP).

UNIDAD IV: INTEGRACION CONTINUA EN UN ENTORNO AGIL DE DESARROLLO DE SOFTWARE.

UNIDAD V: ENFOQUES DE CALIDAD Y SISTEMAS DE DOCUMENTACION.

UNIDAD VI: IMPLEMENTACION DE UN CASO SE ESTUDIO USANDO ENTORNOS AGILES.

## **MATERIALES Y MÉTODOS DE ENSEÑANZA**

La metodología básica a utilizar para lograr los objetivos propuestos constituirá la enseñanza o problema que conduzca hacia la independencia cognoscitiva del alumnado en el proceso de Enseñanza-Aprendizaje.

LOS METODOS A UTILIZAR EN LAS DIFERENTES UNIDADES SON LOS SIGUIENTES:

- Activos.
- Expositivos.
- Elaboración conjunta.

LAS FORMAS ORGANIZATIVAS QUE ALTERNATIVAMENTE SE APLICARAN A LOS DIFERENTE CAPITULOS SE DETALLARAN A CONTINUACION:

- Estudios de casos.
- Debates.
- Investigación individuales o en grupo.



*Universidad Autónoma “Gabriel René Moreno”*  
**FACULTAD DE INGENIERIA EN CIENCIAS  
DE LA COMPUTACIÓN Y TELECOMUNICACIONES**



- Exposiciones individuales o en grupo.

LOS MEDIOS DE ENSEÑANZAS A UTILIZAR SERAN:

- Materiales informativos.
- Materiales ilustrativos.
- Pizarrón.
- Multimedia.

Simultáneamente al desarrollo de cada tema pendiente se efectuara el análisis de casos sobre las problemáticas Nacionales, Regionales y Local.

**SISTEMA DE EVALUACION DEL APRENDIZAJE**

Se aplicaran evaluación de carácter frecuente, parcial y final

DESCRIPCION	TIPO DE EVALUACION	%	UNIDADES A EVALUAR
- TRABAJOS PRÁCTICOS - EXPOSICIÓN GRUPAL - PARTICIPACIÓN EN CLASE	EVALUACION DEL PROCESO	20%	TODAS LAS UNIDADES
- EXAMEN PARCIAL I	EVALUACION DE RESULTADOS	25%	UNIDAD I A III
- EXAMEN PARCIAL II	EVALUACION DE RESULTADOS	25%	UNIDAD IV a VI
- EXAMEN FINAL	EVALUACION DE RESULTADOS	30%	TODAS LAS UNIDADES
TOTAL		100%	



*Universidad Autónoma “Gabriel René Moreno”*  
**FACULTAD DE INGENIERIA EN CIENCIAS  
DE LA COMPUTACIÓN Y TELECOMUNICACIONES**



## **UNIDADES PROGRAMATICAS**

### **UNIDAD I: INTRODUCCION A LAS METODOLOGIAS, FRAMEWORK Y ENTORNOS AGILES**

1. Introducción a la metodología ágiles
2. Que es un proyecto exitoso, enfoque ágil
3. Manifiesto Ágil
  - 3.1. Valores y Principios
4. Metodologías Ágiles versus Metodologías Tradicionales
5. El mundo Ágil
6. El Camino hacia lo ágil.
  - 6.1. Planificación, Historias de usuario
  - 6.2. Programación en pareja (Pair Programming)
  - 6.3. Integración continua (Countinuous integration)
  - 6.4. Refactorización
  - 6.5. Desarrollo dirigido por pruebas (Test Driven Development)

#### **OBJETIVO:**

Introducir al estudiante sobre los objetivos principales de esta nueva forma de desarrollo de software y por qué su surgimiento.

### **UNIDAD II: EL FRAMEWORK SCRUM**

1. Ciclo de vida.
2. Elementos y artefactos necesarios para el desarrollo de aplicaciones.
3. Roles y responsabilidades.
4. Ejemplos de utilización.
5. Alcance y ámbito de uso.
6. Características adicionales.
7. Estado actual de investigación

#### **OBJETIVOS:**

Aplicar el Framework SCRUM para gestionar el ciclo de vida del producto software en un entorno real de desarrollo.



*Universidad Autónoma “Gabriel René Moreno”*  
**FACULTAD DE INGENIERIA EN CIENCIAS**  
**DE LA COMPUTACIÓN Y TELECOMUNICACIONES**



**UNIDAD III: METODOLOGIA PROGRAMACION EXTREMA (XP)**

1. Las Historias de Usuario
2. Roles XP
  - 2.1. Programador.
  - 2.2. Cliente.
  - 2.3. Encargado de pruebas (tester).
  - 2.4. Encargado de seguimiento (Tracker).
  - 2.5. Entrenador (coach).
  - 2.6. Consultor.
  - 2.7. Gestor (big boss).
3. Proceso XP
  - 3.1. FASE I: Exploración.
  - 3.2. FASE II: Planificación de la entrega.
  - 3.3. FASE III: Iteraciones.
  - 3.4. FASE IV: Producción.
  - 3.5. FASE V: Mantenimiento.
  - 3.6. FASE VI: Muerte del proyecto.
4. Prácticas XP
  - 4.1. El juego de la planificación.
  - 4.2. Entrega pequeñas.
  - 4.3. Metáfora.
  - 4.4. Diseño simple.
  - 4.5. Pruebas.
  - 4.6. Refactorización (refactoring).
  - 4.7. Programación en parejas.
  - 4.8. Propiedad colectiva del código.
  - 4.9. Integración continua.
  - 4.10. 40 horas por semana.
  - 4.11. Cliente in-situ
  - 4.12. Estándares de programación.
  - 4.13. Comentarios respecto de las prácticas.

**OBJETIVO UNIDADES III**

Aplicar el la metodología de Programación Extrema (XP) para gestionar el ciclo de vida del proyecto en un entorno real de desarrollo de software.



*Universidad Autónoma “Gabriel René Moreno”*  
**FACULTAD DE INGENIERIA EN CIENCIAS  
DE LA COMPUTACIÓN Y TELECOMUNICACIONES**



**UNIDAD IV: GESTIÓN DE LA CONFIGURACIÓN E INTEGRACIÓN CONTINÚA EN UN ENTORNO AGIL DE DESARROLLO DE SOFTWARE.**

1. Introducción y Ciclo de vida del producto.
2. Modelo de despliegue de integración continua.
3. Gestión de configuración de software (GCS).
4. Lineamientos para control de código fuente
5. Herramientas utilizadas
6. Con licenciamiento
7. Software libre
8. Licenciamiento gratuito.
9. Creación del repositorio
10. Estructura del repositorio y Configuración de permisos
11. Accesos a los Contenido s.
12. Los pasos para subir el contenido del proyecto al repositorio recién creado y vacío, Sincronización de cambios.
13. Traer últimos cambios del servidor, Subir al servidor los cambios locales.

**OBJETIVO UNIDADES IV**

Gestionar la Configuración del Software (GCS/SCM) definiendo las actividades para identificar y definir los elementos en el sistema, gestionar los cambios en el software y sus artefactos a lo largo de su ciclo de vida, estableciendo relaciones entre ellos, definiendo mecanismos para gestionar distintas versiones de estos elementos, y auditando e informando de los cambios.

**UNIDAD V: ENFOQUES DE CALIDAD Y SISTEMAS DE DOCUMENTACION.**

1. Introducción
2. Plan de calidad.
3. Definición de calidad
4. Artefactos del plan de calidad
5. Sistema de documentación.
  - a. Tabla de documentos del sistema de documentación.



*Universidad Autónoma “Gabriel René Moreno”*  
**FACULTAD DE INGENIERIA EN CIENCIAS**  
**DE LA COMPUTACIÓN Y TELECOMUNICACIONES**



- b. El Product Backlog, El Sprint Backlog.
  - c. Priorización de los ítems del Backlog de producto
  - d. Historia de usuario
  - e. El Dashboard
  - f. El BurnDown chart History
  - g. Formulario para pruebas
6. Definición de un caso de prueba (test case)
  7. Solicitud de cambios o nuevos requerimientos
  8. Solicitud de cambios o nuevos requerimientos.
  9. Lista de chequeo para revisión de código.
  10. Documentos para definir la arquitectura de la solución.

#### **OBJETIVO UNIDADES V**

Diseñar el plan de calidad donde se define el sistema de documentación, sistema de gestión de configuración, estándares de codificación, y modelo de evaluación de grado de madures del proceso de desarrollo también se orienta en la aplicación de técnicas y los planes de planes de calidad para el mejoramiento del producto, basado en la norma ISO / IEC 29119 aplicado a un entorno ágil de desarrollo de software.

#### **UNIDAD V: IMPLEMENTACION DE UN CASO SE ESTUDIO USANDO ENTORNOS AGILES.**

1. El Equipos SCRUM
2. Herramientas, Medición de calidad del proceso, Infraestructura
3. Estimaciones, Tiempo de desarrollo, Esfuerzo del proyecto
4. REQUERIMIENTOS
  - 4.1. Pila de producto expresado según sistema de documentación.
  - 4.2. La pila del producto (El Product Backlog).
  - 4.3. Requerimientos no funcionales.
5. ANALISIS.
  - 5.1. Especificación de requerimientos de software.
  - 5.2. Historia de usuario para el retiro de materias
6. DISEÑO.
  - 6.1. Arquitectura de la solución.



*Universidad Autónoma “Gabriel René Moreno”*  
**FACULTAD DE INGENIERIA EN CIENCIAS**  
**DE LA COMPUTACIÓN Y TELECOMUNICACIONES**



- 6.2. Diseño de la base de datos, Diagrama de clases conceptuales.
- 7. IMPLEMENTACION.
  - 7.1. Implementación de las clase de software
  - 7.2. Repositorio de código
  - 7.3. Modelo de despliegue para la solución
- 8. GESTIÓN DE LA CONFIGURACIÓN E INTEGRACIÓN CONTINÚA.
  - 8.1. Modelo de despliegue de la infraestructura de desarrollo
  - 8.2. Pruebas de software

### **OBJETIVO UNIDADES V**

Aplicar la metodológica y framework en un caso de estudio como evidencia de la viabilidad de la propuesta.

### **BIBLIOGRAFIA**

- Rumbaugh, J., Jacobson, I., & Boock, G. (1998). *El proceso Unificado de desarrollo de software*.
- Booch, G., Jacobson, I., & James, R. (2000.). *El Lenguaje Unificado de Modelado*. español año 2000.
- Fowler, M. (1998). *Continuous integration – Improving software quality and reducing risk*. Madrid.
- Henrik Kniberg, & Prólogos de Jeff Sutherland y Mike Cohn. (2007 C4Media Inc). *SCRUM Y XP DESDE LAS TRINCHERAS*. Estados Unidos de América: Projectalis.
- IEEE-STD. (1998). *IEEE-STD-830-1998 : ESPECIFICACIONES DE LOS REQUISITOS DEL*. EEUU.
- Internacional, F. (17 de Octubre de 2013). *Incorporación de la Responsabilidad Social Empresarial y Gobiernos Corporativos en la gestión PYMES de Bolivia*. Recuperado el 1 de 10 de 2013, de FUNDES : <http://pac.caf.com/upload/pdfs/FUNDES%20MEMORIA%20FINAL%20IMPRESI%20C3%93N.PDF>
- Joskowicz, J. (2008). *Metodología de desarrollo de software: Reglas y Prácticas en eXtreme Programming*. EEUU: Impreso en NOVATICA-EEUU.
- Jun. Garzas, & Piattini, M. (2007). *Factorias de software: Experiencias, tecnologia y organizacion Ra-ma*. EEUU: Creative Commons.
- Kniberg Pr, H., & SUTHERLAND-M. (s.f.). *SCRUM y XP desde las trincheras*.
- Pressman, R. S. (2002). *INGENIERÍA DEL SOFTWARE UN ENFOQUE PRACTICO*. Universidad Pontificia de Salamanca campus Madrid (España): Mc Graw Hill.
- Schwaber, k., & Sutherland, j. (2012). *SCRUM\_GUIDE*. UUEE: Scrum Aliance.