

**UNIVERSIDAD AUTÓNOMA DE CHIAPAS  
LICENCIATURA EN SISTEMAS COMPUTACIONALES**

**Área de formación:** Disciplinaria.  
**Unidad académica:** Ingeniería de Software II.  
**Ubicación:** Noveno Semestre.  
**Clave:** 2101  
**Horas semana-mes:** 4  
**Horas teoría:** 2  
**Horas práctica:** 2  
**Unidades CONAIC:** 42.67  
**Prerrequisitos:** Ninguno  
**Horas de infraestructura:** 2  
**Créditos:** 6

### **PRESENTACIÓN**

Actualmente el impulso a la ingeniería de software demanda el desarrollo de productos con calidad que garanticen el cumplimiento de los estándares internacionales.

Este curso está estructurado en tres unidades que presentan las tendencias en el desarrollo de software con calidad.

### **OBJETIVO GENERAL**

Capacitar al alumno para desarrollar, en forma efectiva y con calidad, proyectos de desarrollo de software aplicando las nuevas tendencias.

### **UNIDAD I: CONTROL DE CALIDAD DEL SOFTWARE**

**TIEMPO APROXIMADO:** 22 Horas.

**OBJETIVO DE LA UNIDAD:** Proporcionar al estudiante las generalidades, definiciones, estándares y los elementos que intervienen para alcanzar la calidad total del software.

### **CONTENIDO**

- 1.1 Calidad total por medio de un enfoque modular de arriba hacia abajo
- 1.2 Documentar el software con las herramientas adecuadas
- 1.3 Probar, mantener y Auditar el software
  - 1.3.1 El proceso de prueba
    - 1.3.1.1 Prueba del programa con datos de prueba
    - 1.3.1.2 Prueba de enlace con datos de prueba
    - 1.3.1.3 Prueba completa del sistema con datos de prueba
    - 1.3.1.4 Prueba completa del sistema con datos reales
  - 1.3.2 Mantenimiento
  - 1.3.3 Auditoria
- 1.4 Calidad de los sistemas
  - 1.4.1 Como alcanzar la calidad
  - 1.4.2 Ciclo de la calidad
  - 1.4.3 Aplicación del control de calidad y su influencia en el desarrollo de sistema
  - 1.4.4 Factores que determinan la calidad del software
    - 1.4.4.1 Correcto

- 1.4.4.2 Confiable
- 1.4.4.3 Robusto
- 1.4.4.4 Amigable
- 1.4.4.5 Verificable
- 1.4.4.6 Mantenible
- 1.4.4.7 Reparable
- 1.4.4.8 Reutilizable
- 1.4.4.9 Portable
- 1.4.4.10 Interoperabilidad
- 1.4.4.11 Productividad
- 1.4.4.12 Oportunidad
- 1.4.4.13 Transparencia
- 1.5 El estándar de calidad ISO 9000
- 1.5.1 El estándar ISO 9001
- 1.5.2 Factores de calidad ISO 9126
- 1.5.2.1 Funcionalidad
- 1.5.2.2 Fiabilidad
- 1.5.2.3 Usabilidad
- 1.5.2.4 Eficiencia
- 1.5.2.5 Mantenibilidad
- 1.5.2.6 Portabilidad
- 1.6 Métricas técnicas del software
- 1.6.1 Complección
- 1.6.2 Concisión
- 1.6.3 Consistencia
- 1.6.4 Estandarización de datos
- 1.6.5 Tolerancia al error
- 1.6.6 Eficiencia de la ejecución
- 1.6.7 Independencia del hardware
- 1.6.8 Facilidad de Auditoria
- 1.6.9 Exactitud
- 1.6.10 Estandarización de comunicaciones
- 1.7 Gestión de la configuración del software
- 1.7.1 Líneas base
- 1.7.2 El proceso de GCS
- 1.7.3 Tareas de la GCS
- 1.8 CMM

## **UNIDAD II.- INGENIERÍA DEL SOFTWARE CLIENTE/SERVIDOR**

**TIEMPO APROXIMADO:** 21 Horas.

**OBJETIVO DE LA UNIDAD:** Aplicar el desarrollo de software basado en arquitectura cliente – servidor.

- 2.1 Definición de Sistema Cliente/Servidor
- 2.2 Ventajas de los sistemas distribuidos
- 2.3 Componentes de los sistemas distribuidos
- 2.3.1 Categoría de los servidores
- 2.3.1.1 Servidores de archivos
- 2.3.1.2 Servidores de bases de datos

- 2.3.1.3 Servidores de software de grupo
- 2.3.1.4 Servidores Web
- 2.3.1.5 Servidores de correo
- 2.3.1.6 Servidores de objetos
- 2.3.1.7 Servidores de impresión
- 2.4 Software Intermedio (Middleware)
- 2.4.1 Software Intermedio General
- 2.4.2 Software Intermedio de Servicios
- 2.4.3 Middleware de objetos
- 2.5 Arquitecturas Estratificadas
- 2.5.1 Arquitectura de dos capas
- 2.5.2 Arquitectura de tres capas
- 2.6 Comercio electrónico
- 2.6.1 Categoría de sistemas de comercio electrónico
- 2.6.2 Diseño de sistemas distribuidos
- 2.6.2.1 Principios de diseño
- 2.6.2.1.1 Correspondencia del volumen de transmisión con los medios de transmisión
- 2.6.2.1.2 Mantenimiento de los datos más usados en un almacenamiento rápido
- 2.6.2.1.3 Mantenimiento de los datos cerca de donde se utilizan
- 2.6.2.1.4 Utilización de la duplicación de datos todo lo posible
- 2.6.2.1.5 Diseño para el fallo
- 2.7 Ingeniería de Seguridad
- 2.7.1 Tipos de Intrusiones
- 2.7.2 Encriptación
- 2.7.3 Firmas Digitales
- 2.7.4 Certificaciones digitales

### **UNIDAD III: INGENIERIA WEB**

**TIEMPO APROXIMADO:** 21 Horas.

**OBJETIVO DE LA UNIDAD:** Conocer el proceso, atributos, análisis y diseño de sistemas basados en web.

- 3.1 Atributos de aplicaciones basadas en web
- 3.1.1 Intensivas de red
- 3.1.2 Controlada por el contenido
- 3.1.3 Evolución continua
- 3.1.4 Seguridad
- 3.1.5 Estética
- 3.1.6 Categorías de las webApps
- 3.1.7 Atributos de calidad de las webApps
- 3.2 El proceso de la IWEB
- 3.3 Formulación y análisis de sistemas basados en web
- 3.4 Diseño para aplicaciones basadas en web
- 3.5 Prueba de las aplicaciones basadas en web

### **EXPERIENCIAS DE APRENDIZAJE**

- Temas de investigación.
- Proyectos de Investigación.
- Plenarias de Grupo.
- Casos Prácticos.
- Mesas Redondas.

- Lecturas de Reflexión.

## CRITERIOS DE EVALUACIÓN

3 Exámenes parciales	50%
Tareas	10%
Proyecto de Investigación	20%
Examen Final	<u>20%</u>
TOTAL	100%

## RECURSOS NECESARIOS

Recursos tecnológicos:

Microsoft visio, Proyecto Dia, Open Office.

## PRÁCTICAS SUGERIDAS

- Realizar un proceso de diseño de software utilizando el modelo cliente - servidor.
- Realizar un proceso de diseño de software para evaluar la calidad del software.
- Realizar un programa en Ruby on Rails que realice las operaciones de conexión a base de datos.
- Realizar un proyecto que integre los funciones de altas, bajas, consultas, modificación y reportes en Ruby on Rails.

## BIBLIOGRAFÍA BÁSICA

SOMMERVILLE, I. (2010). Ingeniería del software. México: Pearson educación.

BRAUDE, E. (2003). Ingeniería de software: una perspectiva orientada a objetos. México: Alfaomega.

PRESSMAN, R. (2010). Ingeniería de Software. México: McGraw Hill.

BOOCH, G. (1998). Análisis y Diseño Orientado a Objetos con Aplicaciones. México: Pearson.

BOOCH, G., et al. (2006). El lenguaje Unificado de Modelado. Madrid: Addison Wesley.

## BIBLIOGRAFÍA COMPLEMENTARIA

KENDALL, K., et al. (2005). Análisis y Diseño de Sistemas de Información. México: Pearson.

SCHACH, S. (2007). Ingeniería de software clásica y orientada a objetos. México: McGraw Hill.

GAMMA, E., et al. (1994). Design Patterns: Elements of Reusable Object-Oriented Software. USA: Addison-Wesley.

COHN, M. (2009). Succeeding with Agile: Software Development Using Scrum. USA: Addison-Wesley.

MATHUR, A. (2008). Foundations of Software Testing. USA: Addison-Wesley.