

UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA  
COORDINACIÓN DE FORMACIÓN BÁSICA  
COORDINACIÓN DE FORMACIÓN PROFESIONAL Y VINCULACIÓN UNIVERSITARIA

PROGRAMA DE UNIDAD DE APRENDIZAJE

I. DATOS DE IDENTIFICACIÓN

1. Unidad Académica (s): Facultad de Ciencias Administrativas, Mexicali  
Facultad de Contaduría y Administración, Tijuana  
Facultad de Ciencias Administrativas y Sociales, Ensenada

2. Programa (s) de estudio: (Técnico, Licenciatura (s)) Licenciatura en Informática 3. Vigencia del plan: 2009-2

4. Nombre de la Unidad de Aprendizaje Ingeniería de Software 5. Clave 11864

6. HC: 2 HL: 2 HT:        HPC:        HCL:        HE: 3 CR: 6

7. Ciclo Escolar: 2012-1 8. Etapa de formación a la que pertenece: Disciplinaria

9. Carácter de la Unidad de aprendizaje: Obligatoria X Optativa       

10. Requisitos para cursar la unidad de aprendizaje:

Firmas homologadas

Formuló:

## **II. PROPÓSITO GENERAL DE LA UNIDAD DE APRENDIZAJE**

Este curso se ubica dentro de la etapa disciplinaria es de carácter obligatorio y es la integración de diferentes disciplinas para el desarrollo de software. El propósito de la unidad de aprendizaje es el estudio, comprensión, aplicación y evaluación de técnicas y herramientas de modelado de ingeniería de software, para el desarrollo y documentación técnica de un producto de software con calidad. Con el apoyo de la gestión de proyectos, gestión de la calidad y modelos de referencia para el desarrollo de software.

## **III. COMPETENCIA DE LA UNIDAD DE APRENDIZAJE**

Programar aplicaciones de sistemas de información utilizando lenguajes y herramientas para desarrollo de software, que satisfagan necesidades de información específicas de la organización. Con organización y disciplina.

## **IV. EVIDENCIA (S) DE DESEMPEÑO**

- Documento técnico para la planificación de proyecto de informática.
- Documento técnico para el diseño de proyecto de informática
- Manual técnico y manual del usuario de proyecto de informática
- Desarrollo de software utilizando estandarizaciones de calidad siguiendo modelos de actualidad.
- Exposición del proyecto desarrollado.

## V. DESARROLLO POR UNIDADES

### Competencia

Comparar los diferentes procesos y modelos de referencia para la implementación de sistemas de software en las organizaciones mediante la utilizacion de técnicas, casos y herramientas aplicables que ayuden a determinar cuándo deben de utilizarse, con actitud proactiva y responsable.

### Contenido

Duración 8 hrs.

#### Unidad I Ingeniería de software

##### 1. Ingeniería de software

- 1.1. Definiciones
- 1.2. Principios de la Ingeniería de Software
- 1.3. Paradigmas de desarrollo de software
- 1.4. Industria del software
- 1.5. Procesos de software
- 1.6. Modelado de software

## V. DESARROLLO POR UNIDADES

### Competencia

Aplicar la administración de proyectos que surgen en los proyectos de software a través herramientas y técnicas de calendarización y planeación para garantizar la finalización de un proyecto de software cumpliendo con los requerimientos de los usuarios con calidad, responsabilidad y compromiso.

### Contenido

Duración 16 hrs.

#### Unidad II Gestión de proyectos

##### 2. Gestión de proyectos

- 2.1. Actividades de gestión
- 2.2. Planificación de proyectos
- 2.3. Gestión de riesgos
- 2.4. Gestión de recursos humanos
  - 2.4.1. Límites del pensamiento
  - 2.4.2. Liderazgo y trabajo en equipo
  - 2.4.3. Selección de personal
  - 2.4.4. Modelos de calidad aplicados al personal
- 2.5. Estimación de costos
  - 2.5.1. Productividad
  - 2.5.2. Técnicas de estimación
    - 2.5.2.1. Juicio experto
    - 2.5.2.2. Ley de Parkinson
    - 2.5.2.3. COCOMO 2
    - 2.5.2.4. Modelo algorítmico de costos

## V. DESARROLLO POR UNIDADES

### Competencia

Controlar el proceso de desarrollo de software a través de la aplicación de métricas de aseguramiento de calidad de software, así como de la implementación de estándares de procesos aplicables al desarrollo del software, con la finalidad de lograr la construcción de un sistema que cumpla con los requerimientos definidos por el usuario de forma proactiva, eficiente y responsable.

### Contenido

Duración **12 hrs.**

#### Unidad III Gestión de la calidad

##### 3. Gestión de la calidad

- 3.1. Introducción
- 3.2. Garantía de calidad y estándares
- 3.3. Planificación de la calidad
- 3.4. Control de la calidad del software
  - 3.4.1. Mediciones y métricas de software
  - 3.4.2. Análisis y modelado de procesos
  - 3.4.3. Mediciones del proceso
  - 3.4.4. Calidad del proceso
  - 3.4.5. El modelo de madurez del proceso (CMM)
  - 3.4.6. El modelo MOPROSOFT
- 3.5. Evaluación de la calidad
  - 3.5.1. Evaluación de la madurez de los procesos en el software
  - 3.5.2. Pruebas de software
  - 3.5.3. Control de la configuración del software

## V. DESARROLLO POR UNIDADES

### Competencia

Comparar y evaluar los diferentes procesos y modelos de referencia para la implementación de sistemas de software a través de las mejores prácticas de la ingeniería de software, para de esta manera determinar cuáles son los elementos necesarios en la construcción de un producto de software de calidad con eficiencia, responsabilidad y honestidad.

### Contenido

Duración 28 hrs.

#### Unidad IV Modelado y Diseño de software

##### 4. Modelado y Diseño de software

- 4.1. Modelado de Procesos de Software
  - 4.1.1. Modelado Orientado a Objetos
  - 4.1.2. Modelado de Software con UML
  - 4.1.3. Patrones de Diseño
  - 4.1.4. Modelos de Referencia
- 4.2. Diseño de Procesos de Software
  - 4.2.1. Principios y conceptos
  - 4.2.2. Tipos de diseño
    - 4.2.2.1. Modular Efectivo
    - 4.2.2.2. Arquitectónico
    - 4.2.2.3. Datos
    - 4.2.2.4. Interfaz de Usuario
    - 4.2.2.5. Componentes
  - 4.2.3. Documentación

## VI. ESTRUCTURA DE LAS PRÁCTICAS

No. de Práctica	Competencia(s)	Descripción	Material de Apoyo	Duración
1	Comparar los diferentes procesos y modelos de referencia para la implementación de sistemas de software en las organizaciones mediante la utilización de técnicas, casos y herramientas aplicables que ayuden a determinar cuándo deben de utilizarse, con actitud proactiva y responsable.	Describa las principales actividades en el proceso de diseño de software y las salidas de estas actividades. Utilizando un diagrama entidad-relación, muestre las posibles relaciones entre las salidas.	Computadora	1 hora
2	Aplicar la administración de proyectos que surgen en los proyectos de software a través herramientas y técnicas de calendarización y planeación para garantizar la finalización de un proyecto de software cumpliendo con los requerimientos de los usuarios con calidad, responsabilidad y compromiso.	Utilizando un plan de proyecto, realice un plan de proyecto. Realizando las siguientes actividades de gestión: 1. Duración de proyecto 2. Camino crítico 3. Agenda 4. Evaluación de casos de retraso y adelanto de actividades 5. Costos 6. Otros indicadores propuestos por el docente.	Computadora, MSProject o algun otro software de aplicación para administración de proyecto	2 horas
3		Determinación de costos de software mediante el uso de	Computadora, sitio web de apoyo (Ej.)	2 horas

		la metodología COCOMO 2	<a href="http://sunset.usc.edu/cse/pub/research/AgileCOCOMO/AgileCOCOMOII/Main.html">http://sunset.usc.edu/cse/pub/research/AgileCOCOMO/AgileCOCOMOII/Main.html</a> , software de aplicación (Costar)	
4	Controlar el proceso de desarrollo de software a través de la aplicación de métricas de aseguramiento de calidad de software, así como de la implementación de estándares de procesos aplicables al desarrollo del software, con la finalidad de lograr la construcción de un sistema que cumpla con los requerimientos definidos por el usuario de forma proactiva, eficiente y responsable.	Desarrolle un documento donde aplique el Modelo de Procesos de Software (MOPROSOFT) al desarrollo de su proyecto de software. Diferencie los diferentes niveles del modelo y adecúelos correctamente. 1. Alta dirección 2. Gestión 3. Operación	Computadora Norma NMX-059/01-NYCE-2005	2 horas
5	Comparar y evaluar los diferentes procesos y modelos de referencia para la implementación de sistemas de software a través de las mejores prácticas de la ingeniería de software, para de esta manera determinar cuáles son los elementos necesarios en la construcción de un producto de software de calidad con eficiencia, responsabilidad y honestidad.	Desarrollar UML para modelar los procesos de software de su proyecto. Elaborar una funcionalidad completa: 1. Descripción de la funcionalidad 2. Casos de uso 3. Diagrama de clase 4. Diagrama de pantalla 5. Escenario 5.1. Diagrama de Actividades 5.2. Diagrama de Estado 5.3. Diagrama de Secuencia	Computadora. Software para el Modelado UML	4 horas

## VII. METODOLOGÍA DE TRABAJO

Exposiciones por parte del maestro y del alumno, quien trabajará en equipo para realizar prácticas de laboratorio y en el desarrollo de un trabajo final en tres etapas. En la primera etapa se define el plan del proyecto, la segunda parte se muestra un avance que determine la funcionalidad parcial del sistema, y en la última parte se entrega el proyecto terminado para su posterior implantación. Se elabora también la documentación pertinente al avance del desarrollo del producto, el cual está determinado por el alcance del curso.

## VIII. CRITERIOS DE EVALUACIÓN

La evaluación de cada unidad de aprendizaje consiste en exámenes, exposición de un tema relacionado, tareas y prácticas, así como la exposición oral y escrita del desarrollo de un sistema de software.

### Rubros para evaluación

Rubro	Porcentaje
Exámenes	30
Tareas	20
Exposición sobre tema de actualidad de Ingeniería de Software	10
Proyecto (Producto de software, documentación y exposición final)	40

### Criterio de evaluación

#### *Exámenes*

- Tres exámenes parciales
  - Un examen teórico.
  - Dos exámenes prácticos.

#### *Tareas*

Las tareas que consistan en investigación de un tema en particular deberán incluir lo siguiente:

- Referencias bibliográficas

- Conclusión
- Ejemplos.

*Exposición sobre tema de actualidad de Ingeniería de Software*

- Será determinada por el profesor
- Referente a casos empresariales actuales.
- Referencias bibliográficas

*Proyecto*

- Se compone de tres fases.
- En cada fase se entregará los documentos solicitados y se hará una presentación del proyecto ante el grupo.

## IX. BIBLIOGRAFÍA

Básica	Complementaria
<p>Pressman, Roger S. <b>Ingeniería del software : un enfoque práctico</b> 6ta. Edición , 2010 México: McGraw-Hill Interamericana Editores S.A. de C.V. ISBN-10: 6071503140 ISBN-13: 978-6071503145</p> <p>Sommerville, Ian. <b>Ingenieria de Software,</b> 9na Edición, Pearson Addison Weasley, 2011. Pág: 792 p. : ISBN: 9786073206037</p> <p>Piattini Velthuis, Mario Gerardo. <b>Medicion y estimacion del software: Técnicas y métodos para mejorar la calidad y la productividad</b> Editor: Alfaomega ; 2008. Páginas: 308 p. : ISBN: 9789701514139</p>	<p>Braude, Eric J. <b>Ingeniería de software : una perspectiva orientada a objetos</b> Editor: Alfaomega, 2003. Páginas: 539 p. : ISBN: 9701508513</p> <p>Bruegge, Bernd. <b>Ingeniería de software orientada a objetos</b> Editor: Pearson Educación, Fecha de pub: 2002. Páginas: 553 p. : ISBN: 9702600103</p> <p>Piattini Velthuis, Mario Gerardo. <b>Análisis y diseño de aplicaciones informáticas de gestión : una perspectiva de ingeniería del software</b> Editor: Alfaomega ; 2004. Páginas: 710 p. : ISBN: 9701509870</p>