

1. Datos Generales de la asignatura

Nombre de la asignatura:	Desarrollo de Proyectos de Software II
Clave de la asignatura:	INC-1702
SATCA:	2-2-4
Carrera:	Ingeniería en Sistemas Computacionales

2. Presentación

Caracterización de la asignatura

La asignatura de Desarrollo de Proyectos de SW II, permite que el alumno termine el ciclo de desarrollo de un proyecto de software, aplicando los conocimientos sobre cómo debe de construirse, probarse y dar mantenimiento a un Software.

La aportación que esta asignatura le da al perfil profesional es la siguiente:

- La capacidad de codificar software utilizando buenas prácticas de programación, estándares de construcción, reutilización de código, etc.
- La capacidad de codificar un producto de software que se ha analizado y diseñado previamente.
- La capacidad para poder probar un software, a través del diseño de casos de prueba, pruebas de caja negra, caja blanca y pruebas de unidad.
- La capacidad de conocer y utilizar herramientas de pruebas automatizadas.
- La capacidad de complementar la construcción de la arquitectura 4 + 1 Vistas, creando la vista de desarrollo y la vista física a través de los diagramas de componentes y despliegue de UML.
- La Capacidad de conocer los principales elementos a considerar a la hora de implementar y dar mantenimiento al software.

La asignatura de Desarrollo de Proyectos de Software II se relaciona con materias precedentes como: Programación Orientada A Objetos, Estructura De Datos, Tópicos Avanzados De Programación, Fundamentos de Base de Datos, Taller de Base de Datos, Ingeniería De Software y Fundamentos De Ingeniería de Software, Programación Web, Desarrollo de Proyectos de Software I.

Posteriores: Gestión de Proyectos de Software, Metodologías Ágiles.

Requiere de competencias previas como: Manejo del lenguaje de modelado UML, dominio en el uso de herramientas CASE, uso de algún Sistema Manejador de Bases de Datos, dominio de algún lenguaje de programación orientado a objetos, dominio de algún entorno de desarrollo integrado (IDE) ,identificación de las etapas del ciclo de desarrollo de sistemas y de las diferentes plataformas operativas.

Intención didáctica

En la primera unidad, tratan los temas referentes a estandarización y buenas prácticas a la hora de construir un software, el alumno deberá de concluir con la codificación del sistema que inicio en la asignatura anterior Desarrollo de Proyectos de Software I.

En la segunda unidad, se le enseña al alumno como crear casos de prueba, diseñar pruebas de caja negra, caja blanca y aplicarlas al proyecto desarrollado. Además se utilizará una herramienta automatizada para aplicar pruebas de unidad al mismo proyecto.

En la tercera unidad, se verán los diagramas de componentes y despliegue de UML para que el alumno pueda construir las últimas dos vistas de la Arquitectura 4+1 vistas del sistema desarrollado.

En la cuarta Unidad, se verán los posibles escenarios para la implementación de sistema en un entorno real y se explicará cómo debe llevarse a cabo el manteniendo al Software construido para así cerrar el ciclo de desarrollo.

3. Participantes en el diseño y seguimiento curricular del programa

Lugar y fecha de elaboración o revisión	Participantes	Evento
Instituto Tecnológico de Hermosillo 12 Septiembre – Diciembre 2016	Academia de Sistemas y Computación	Jornadas Curriculares para definir las especialidades de las carreras de ISC e II.

4. Competencia(s) a desarrollar

Competencia(s) específica(s) de la asignatura

Construcción e Implementación de software, que sea codificado utilizando estándares y buenas prácticas de codificación, y que se base en un análisis y diseño orientado a objetos utilizando la Arquitectura 4 +1 Vistas y UML y que pueda ser probado y verificado para su correcta implementación.

5. Competencias previas

- Dominio de algún lenguaje de programación orientado a objetos.
- Dominio de un IDE (Entorno de Desarrollo Integrado)
- Dominio de diseño e implementación de Bases de Datos.
- Manejo del Lenguaje Unificado de Modelado (UML).
- Conocimientos de Arquitectura de Software.

6. Temario

No.	Temas	Subtemas
1	Construcción	<p>1.1 No da igual cómo esté construido</p> <p>1.2 Objetivos</p> <p>1.3 Introducción</p> <p>1.4 Lenguajes de construcción</p> <p>1.5 Reutilización del código</p> <p>1.6 Principios fundamentales de la construcción de software</p> <p> 1.6.1 Minimizar la complejidad</p> <p> 1.6.2 Anticipar los cambios</p> <p> 1.6.3 Construir para verificar</p> <p> 1.6.4 Utilización de estándares</p> <p>(terminar de codificar el sistema la parte de procesos y salidas)</p>
2	Pruebas de SW	<p>2.1 El porqué de las pruebas</p> <p>2.2 Objetivos</p> <p>2.3 Introducción</p> <p> 2.3.1 Conceptos fundamentales</p> <p> 2.3.2 Limitaciones en la realización de pruebas</p> <p> 2.3.3 Las pruebas y el riesgo</p> <p>2.4 El proceso de prueba</p> <p>2.5 Técnicas de prueba</p> <p>2.6 Pruebas de caja blanca y de caja negra</p> <p>2.7 Pruebas unitarias</p> <p>2.8 Evaluación de las pruebas realizadas</p> <p>(El alumno debe poder diseñar casos de prueba mediante las técnicas de pruebas de caja negra y caja blanca y utilizar alguna herramienta automatizada para realizar pruebas de unidad)</p>
3	Arquitectura de SW	<p>3.1 Diagramas de componentes (Para poder terminar la versión final el diagrama de componentes debe de terminar la codificación del sistema</p> <p> 3.2 Diagramas de despliegue (Determinar un esquema posible de implementación física)</p> <p>(terminar con la construcción de la arquitecturas 4 +1 Vistas, se elaboraran las dos últimas vistas de la arquitectura, la vista de desarrollo y la vista física a través de los diagramas de componentes y despliegue de UML)</p>

--	--	--

7. Actividades de aprendizaje de los temas

1. Construcción	
Competencias	Actividades de Aprendizaje
<p>Identifica y aplica buenas prácticas, reutilización de código y estándares de codificación para codificar un proyecto de software en un entorno de desarrollo integrado (IDE).</p> <p>Genéricas: Capacidad para aplicar los temas vistos a la construcción del proyecto de software. Capacidad de trabajo de equipo. Capacidad oral y escrita. Capacidad para recibir retroalimentación.</p>	<p>Analizar y aplicar patrones de diseño a su proyecto. Realizar la programación del proyecto de software mediante el paradigma Orientado a Objetos en el IDE seleccionado. Revisar que la codificación del proyecto debe ser acorde al diseño del diagrama de clases propuesto. Presentar Avances del proyecto cada 15 días para dar seguimiento con el profesor y recibir retroalimentación. Elaborar correcciones de acuerdo a la retroalimentación recibida en caso de ser necesarias.</p>
2. Pruebas de Software	
Competencias	Actividades de Aprendizaje
<p>Implementar una estrategia para realizar pruebas a su proyecto de software.</p> <p>Genéricas: Capacidad para aplicar los temas vistos a la elaboración del plan de pruebas. Capacidad de trabajo de equipo. Capacidad oral y escrita. Capacidad para recibir retroalimentación.</p>	<p>Diseñar casos de prueba para realizar pruebas de Caja Negra y Caja Blanca. Aplicar pruebas de Caja Negra y Caja Blanca por lo menos a una sección del programa. Utilizar una Herramienta de pruebas automatizadas para realizar pruebas de Unidad al Proyecto. Presentar los resultados de la aplicación de las pruebas para ser valoradas por el profesor y recibir retroalimentación. Elaborar correcciones de acuerdo a la retroalimentación recibida en caso de ser necesarias.</p>
3. Arquitectura de SW	
Competencias	Actividades de Aprendizaje
<p>Construirá los diagramas de componentes y despliegue de UML para terminar con la construcción de las vistas de desarrollo y vista física de la arquitectura 4+ 1 vistas.</p>	<p>Aplicar el Lenguaje Unificado Modelado (UML) específicamente Diagramas de componentes y despliegue para terminar la arquitectura del sistema, debe utilizar alguna herramienta que permita utilizar</p>

<p>Capacidad para aplicar los temas vistos a la elaboración del plan de pruebas. Capacidad de trabajo de equipo. Capacidad oral y escrita. Capacidad para recibir retroalimentación.</p>	<p>UML. Presentar a revisión los diagramas y la arquitectura completa para revisar que sean acorde al software construido y recibir retroalimentación. Elaborar correcciones de acuerdo a la retroalimentación recibida en caso de ser necesarias.</p>
4. Implementación y Mantenimiento	
Competencias	Actividades de Aprendizaje
<p>Conocer las técnicas para la implantación y Mantenimiento del software.</p>	<p>Discutir, en equipo, las características que diferencian a cada uno de las técnicas de implementación. Determinar y aplicar la técnica seleccionada. Probar el correcto funcionamiento del Software. Realizar una síntesis sobre el mantenimiento del software, discutir, en equipo, las características del mantenimiento de software.</p>

7. Actividades de aprendizaje de los temas

<ul style="list-style-type: none"> ● Investigar en diversas fuentes de investigación acerca de los temas vistos en clase. ● Presentación de avances periódicos establecidos previamente por el profesor. ● Desarrollar actividades de aprendizaje que propicien la aplicación de los conceptos, modelos y metodologías que se van aprendiendo en el desarrollo de la asignatura. ● Desarrollar un proyecto real, que involucre el manejo de conocimientos de cada contenido temático desarrollado en el transcurso del curso, y considerando temas afines de otras asignaturas. ● Propiciar el trabajo en equipo. ● El profesor debe utilizar ejemplos reales para que el alumno pueda entender el concepto teórico y poder llevarlo a la práctica aplicándolo en su proyecto.
--

8. Prácticas

<ul style="list-style-type: none"> ● Codificación de la parte de procesos y salidas del sistema en un lenguaje orientado a objetos utilizando un IDE. ● Utilización de una herramienta de pruebas automatizadas para la realización de pruebas unitarias. ● Aplicación de pruebas de Caja Negra y Caja Blanca Utilizando el IDE.

- Elaboración de diagrama de componentes y despliegue en alguna Herramienta de Modelado con UML (ejemplo: StarUML).
- Prueba de Implementación del proyecto en un entorno real.

9. Proyecto de Asignatura

El alumno debe de terminar con el Desarrollo e Implementación del proyecto de software que se inicio en la asignatura de Desarrollo de Proyectos de Software I.

El proyecto debe llevarse a cabo en equipo de máximo tres personas.

10. Evaluación por competencias

Se recomienda la siguiente evaluación:

Evaluación parciales (4 Unidades):

- Tareas y trabajos de Investigación 10%
- Avances de Proyecto 30%
- Exámenes Parciales 60%

Evaluación Global

- Evaluaciones parciales 60%
- Proyecto Final 40%

11.- Fuentes de Información

1. Fowler, Martin, (1999). UML Gota a Gota Ed. Addison Wesley.
2. Larman, Craig (1999). UML y patrones. Ed. Pearson.
3. Meyer, Bertrand (1999). Construcción de Software Orientada a Objetos. Ed. Prentice Hall.
4. Jacobson, Ivar. (2000). El Proceso unificado de desarrollo de Software. Ed. Addison Wesley.
5. Sommerville, Ian (2001). Ingeniería de Software. Ed. Prentice Hall.
6. Pressman Roger S (2001). Ingeniería del Software, 5/E. Ed. Mc.Graw-Hill.