

ASIGNATURA DE METODOLOGÍAS Y MODELADO DE DESARROLLO DE SOFTWARE

1. Competencias	Desarrollar soluciones tecnológicas para entornos Web mediante fundamentos de programación orientada a objetos, base de datos y redes de área local que atiendan las necesidades de las organizaciones.
2. Cuatrimestre	Segundo
3. Horas Teóricas	23
4. Horas Prácticas	52
5. Horas Totales	75
6. Horas Totales por Semana Cuatrimestre	5
7. Objetivo de aprendizaje	El alumno implementará modelos de desarrollo de software mediante el análisis y diseño para dar solución a problemáticas planteadas.

Unidades de Aprendizaje	Horas		
	Teóricas	Prácticas	Totales
I. Ingeniería de requerimientos	3	6	9
II. Modelado de software	11	18	29
III. Procesos de negocios	2	10	12
IV. Metodologías de desarrollo de software	7	18	25
Totales	23	52	75

ELABORÓ:	Comité técnico académico de diseño curricular del subsistema de CGUTyP de la familia de carreras de TSU en Tecnologías de la Información	REVISÓ:	Dirección Académica	
APROBÓ:	C. G. U. T. y P.	FECHA DE ENTRADA EN VIGOR:	Septiembre de 2018	

METODOLOGÍAS Y MODELADO DE DESARROLLO DE SOFTWARE

UNIDADES DE APRENDIZAJE

1. Unidad de aprendizaje	I. Ingeniería de requerimientos
2. Horas Teóricas	3
3. Horas Prácticas	6
4. Horas Totales	9
5. Objetivo de la Unidad de Aprendizaje	El alumno realizará el análisis de problemas mediante técnicas de recolección de información para generar el documento de especificación de requerimientos de un proyecto de software.

Temas	Saber	Saber hacer	Ser
Técnicas de recolección de requerimientos: Entrevistas, encuestas, observación y listas de verificación.	Identificar las técnicas de recolección de requerimientos para un proyecto de desarrollo de software (Entrevistas, encuestas, observación y listas de verificación).	Diseñar las herramientas para la recolección de datos como: guía de entrevista, encuesta, guía de observación y lista de verificación.	Sistemático. Analítico. Observador. Crítico. Colaborativo. Ético.
Análisis y síntesis de información.	Definir los tipos de requerimientos para un proyecto de software de acuerdo al dominio de la aplicación.	Clasificar los requerimientos para un proyecto de software.	Sistemático. Analítico. Observador. Crítico. Colaborativo. Ético.
Especificación y validación de requerimientos. IEEE-830 y plantillas SRS.	Distinguir los requerimientos de software de acuerdo al estándar IEEE 830-1998, utilizando técnicas de validación de requerimientos.	Proponer la plantilla adecuada para el tipo de proyecto de acuerdo al estándar IEEE830.	Sistemático. Analítico. Observador. Crítico. Colaborativo. Ético.

ELABORÓ:	Comité técnico académico de diseño curricular del subsistema de CGUTyP de la familia de carreras de TSU en Tecnologías de la Información	REVISÓ:	Dirección Académica	
APROBÓ:	C. G. U. T. y P.	FECHA DE ENTRADA EN VIGOR:	Septiembre de 2018	

METODOLOGÍAS Y MODELADO DE DESARROLLO DE SOFTWARE

PROCESO DE EVALUACIÓN

Resultado de aprendizaje	Secuencia de aprendizaje	Instrumentos y tipos de reactivos
<p>Entrega el documento de "Especificación de Requerimientos de Software" (ERS) que incluya:</p> <ul style="list-style-type: none"> • Fecha • Nombre del Proyecto • Objetivo • Alcance • Descripción funcional • Requerimientos: <ul style="list-style-type: none"> - Software - Hardware 	<ol style="list-style-type: none"> 1. Comprender las técnicas de recolección. 2. Comprender el proceso de análisis de datos con base a requerimientos. 3. Identificar los requerimientos funcionales y no funcionales de acuerdo al estándar IEEE 830. 	<ul style="list-style-type: none"> - Ejercicios prácticos. - Listas de cotejo.

ELABORÓ:	Comité técnico académico de diseño curricular del subsistema de CGUTyP de la familia de carreras de TSU en Tecnologías de la Información	REVISÓ:	Dirección Académica	
APROBÓ:	C. G. U. T. y P.	FECHA DE ENTRADA EN VIGOR:	Septiembre de 2018	

METODOLOGÍAS Y MODELADO DE DESARROLLO DE SOFTWARE

PROCESO ENSEÑANZA APRENDIZAJE

Métodos y técnicas de enseñanza	Medios y materiales didácticos
<ul style="list-style-type: none">- Equipos colaborativos.- Práctica demostrativa.- Solución de problemas.	<ul style="list-style-type: none">Pizarrón.Plumones.Computadora.Internet.Equipo multimedia.Ejercicios prácticos.Plataformas virtuales.

ESPACIO FORMATIVO

Aula	Laboratorio / Taller	Empresa
X		

ELABORÓ:	Comité técnico académico de diseño curricular del subsistema de CGUTyP de la familia de carreras de TSU en Tecnologías de la Información	REVISÓ:	Dirección Académica	
APROBÓ:	C. G. U. T. y P.	FECHA DE ENTRADA EN VIGOR:	Septiembre de 2018	

METODOLOGÍAS Y MODELADO DE DESARROLLO DE SOFTWARE

UNIDADES DE APRENDIZAJE

1. Unidad de aprendizaje	II. Modelado de software
2. Horas Teóricas	11
3. Horas Prácticas	18
4. Horas Totales	29
5. Objetivo de la Unidad de Aprendizaje	El alumno construirá los modelos de proyecto de software con base a un tipo de arquitectura definida para dar solución a casos establecidos.

Temas	Saber	Saber hacer	Ser
Tipos de arquitecturas: SOA, Micro servicios, cliente - servidor, monolítica, distribuido, capas.	Definir los tipos de arquitectura: SOA, Micro servicios, cliente - servidor, monolítica, distribuido, capas.	Esquematizar la perspectiva del proyecto empleando vistas de la arquitectura.	Sistemático. Analítico. Observador. Crítico. Colaborativo. Ético.
Modelado UML.	Identificar la estructura del lenguaje UML de acuerdo a las áreas estructural, dinámica, gestión del modelo y extensiones (Casos de uso, diagrama de clases, Diagramas de actividades, secuencia, componentes, despliegue).	Diseñar el modelado de software mediante la estructura estática y dinámica de UML (Casos de uso, clases, secuencia, componentes, despliegue, estado).	Sistemático. Analítico. Crítico. Coherente. Colaborativo. Asertivo. Organizado.

ELABORÓ:	Comité técnico académico de diseño curricular del subsistema de CGUTyP de la familia de carreras de TSU en Tecnologías de la Información	REVISÓ:	Dirección Académica	
APROBÓ:	C. G. U. T. y P.	FECHA DE ENTRADA EN VIGOR:	Septiembre de 2018	

METODOLOGÍAS Y MODELADO DE DESARROLLO DE SOFTWARE

PROCESO DE EVALUACIÓN

Resultado de aprendizaje	Secuencia de aprendizaje	Instrumentos y tipos de reactivos
<p>Entrega el documento de "Modelado de Software" que incluya los diagramas UML:</p> <ul style="list-style-type: none">• Casos de uso.• Clases.• Secuencia.• Colaboración.• Estado.	<ol style="list-style-type: none">1. Comprender las diversas formas de representación de requerimientos.2. Identificar la arquitectura de software correspondiente para la solución.3. Identificar el prototipo conceptual del software mediante diagramación UML.	<ul style="list-style-type: none">- Ejercicios prácticos.- Listas de cotejo.

ELABORÓ:	Comité técnico académico de diseño curricular del subsistema de CGUTyP de la familia de carreras de TSU en Tecnologías de la Información	REVISÓ:	Dirección Académica	
APROBÓ:	C. G. U. T. y P.	FECHA DE ENTRADA EN VIGOR:	Septiembre de 2018	

METODOLOGÍAS Y MODELADO DE DESARROLLO DE SOFTWARE

PROCESO ENSEÑANZA APRENDIZAJE

Métodos y técnicas de enseñanza	Medios y materiales didácticos
<ul style="list-style-type: none"> - Equipos colaborativos. - Práctica demostrativa. - Solución de problemas. 	<ul style="list-style-type: none"> Pizarrón. Plumones. Computadora. Internet. Equipo multimedia. Ejercicios prácticos. Plataformas virtuales. Diagramadores.

ESPACIO FORMATIVO

Aula	Laboratorio / Taller	Empresa
X	X	

ELABORÓ:	Comité técnico académico de diseño curricular del subsistema de CGUTyP de la familia de carreras de TSU en Tecnologías de la Información	REVISÓ:	Dirección Académica	
APROBÓ:	C. G. U. T. y P.	FECHA DE ENTRADA EN VIGOR:	Septiembre de 2018	

METODOLOGÍAS Y MODELADO DE DESARROLLO DE SOFTWARE

UNIDADES DE APRENDIZAJE

1. Unidad de aprendizaje	III. Procesos de negocios
2. Horas Teóricas	2
3. Horas Prácticas	10
4. Horas Totales	12
5. Objetivo de la Unidad de Aprendizaje	El alumno representará los procesos de las organizaciones a través de los diagramas UML para comprender el modelo de negocio.

Temas	Saber	Saber hacer	Ser
Modelado de negocios.	Describir los conceptos de representación de los procesos de negocios.	Esquematizar los procesos de negocio actuales a través de diagramas UML (roles y actividades).	Sistemático. Analítico. Crítico. Coherente. Colaborativo. Asertivo. Organizado.
Diagrama de contexto.	Describir las relaciones del sistema con su entorno dentro de la organización mediante diagramas UML (distribución).	Elaborar diagramas UML (distribución) que describan la interacción de los componentes de software con respecto a la organización.	Sistemático. Analítico. Crítico. Coherente. Colaborativo. Asertivo. Organizado.

ELABORÓ:	Comité técnico académico de diseño curricular del subsistema de CGUTyP de la familia de carreras de TSU en Tecnologías de la Información	REVISÓ:	Dirección Académica	
APROBÓ:	C. G. U. T. y P.	FECHA DE ENTRADA EN VIGOR:	Septiembre de 2018	

METODOLOGÍAS Y MODELADO DE DESARROLLO DE SOFTWARE

PROCESO DE EVALUACIÓN

Resultado de aprendizaje	Secuencia de aprendizaje	Instrumentos y tipos de reactivos
<p>Entrega el documento de "Modelado y Arquitectura de Software" que incluya los diagramas UML:</p> <ul style="list-style-type: none"> • Casos de uso • Clases • Secuencia • Estado • Componentes • Actividades • Despliegue 	<ol style="list-style-type: none"> 1. Identificar la relación de los stakeholders con la aplicación de software. 2. Comprender la funcionalidad del software con el propósito del cliente utilizando el mapeo de procesos. 3. Comprender la funcionalidad del software de acuerdo a su entorno. 	<ul style="list-style-type: none"> - Ejercicios prácticos. - Listas de cotejo.

ELABORÓ:	Comité técnico académico de diseño curricular del subsistema de CGUTyP de la familia de carreras de TSU en Tecnologías de la Información	REVISÓ:	Dirección Académica	
APROBÓ:	C. G. U. T. y P.	FECHA DE ENTRADA EN VIGOR:	Septiembre de 2018	

METODOLOGÍAS Y MODELADO DE DESARROLLO DE SOFTWARE

PROCESO ENSEÑANZA APRENDIZAJE

Métodos y técnicas de enseñanza	Medios y materiales didácticos
<ul style="list-style-type: none"> - Equipos colaborativos. - Práctica demostrativa. - Solución de problemas. 	<ul style="list-style-type: none"> Pizarrón. Plumones. Computadora. Internet. Equipo multimedia. Ejercicios prácticos. Plataformas virtuales. Diagramadores.

ESPACIO FORMATIVO

Aula	Laboratorio / Taller	Empresa
X	X	

ELABORÓ:	Comité técnico académico de diseño curricular del subsistema de CGUTyP de la familia de carreras de TSU en Tecnologías de la Información	REVISÓ:	Dirección Académica	
APROBÓ:	C. G. U. T. y P.	FECHA DE ENTRADA EN VIGOR:	Septiembre de 2018	

METODOLOGÍAS Y MODELADO DE DESARROLLO DE SOFTWARE

UNIDADES DE APRENDIZAJE

1. Unidad de aprendizaje	IV. Metodologías de desarrollo de software
2. Horas Teóricas	7
3. Horas Prácticas	18
4. Horas Totales	25
5. Objetivo de la Unidad de Aprendizaje	El alumno identificará las metodologías más comunes para el desarrollo de software.

Temas	Saber	Saber hacer	Ser
Metodologías de desarrollo tradicionales: cascada, modelo en V y espiral.	Definir los conceptos de las metodologías tradicionales de desarrollo (cascada, modelo en V y espiral). Distinguir las ventajas y desventajas de las metodologías tradicionales de desarrollo.	Seleccionar la metodología apropiada de acuerdo al tipo de proyecto.	Sistemático. Analítico. Crítico. Coherente. Colaborativo. Asertivo. Organizado.
Metodologías de desarrollo ágiles: Scrum y XP.	Definir los conceptos de las metodologías de desarrollo ágiles (Scrum y XP). Distinguir las ventajas y desventajas de las metodologías de desarrollo ágiles.	Seleccionar la metodología ágil que se adapte a las condiciones de un proyecto de software.	Sistemático. Analítico. Crítico. Coherente. Colaborativo. Asertivo. Organizado.
Metodologías de desarrollo Web.	Explicar la extensión WAE (Web Application Extension). Identificar los estereotipos de UML para el desarrollo de una aplicación Web.	Elaborar el modelado de aplicaciones Web mediante la estructura estática y dinámica de UML (Casos de uso, clases, secuencia, componentes, despliegue, estado), aplicando la extensión WAE.	Sistemático. Analítico. Crítico. Coherente. Colaborativo. Asertivo. Organizado.

ELABORÓ:	Comité técnico académico de diseño curricular del subsistema de CGUTyP de la familia de carreras de TSU en Tecnologías de la Información	REVISÓ:	Dirección Académica	
APROBÓ:	C. G. U. T. y P.	FECHA DE ENTRADA EN VIGOR:	Septiembre de 2018	

METODOLOGÍAS Y MODELADO DE DESARROLLO DE SOFTWARE

PROCESO DE EVALUACIÓN

Resultado de aprendizaje	Secuencia de aprendizaje	Instrumentos y tipos de reactivos
<p>Entrega un portafolio de evidencias que incluya:</p> <ul style="list-style-type: none"> • Especificación de Requerimientos de Software (ERS). • Modelado de Software. • Arquitectura de Software. • Extensión WAE. 	<ol style="list-style-type: none"> 1. Conocer las distintas metodologías tradicionales y ágiles. 2. Identificar las ventajas y desventajas de las metodologías de desarrollo tradicionales y ágiles. 3. Utilizar las metodologías apropiadas de acuerdo a los proyectos de desarrollo de software. 	<ul style="list-style-type: none"> - Estudio de casos. - Lista de cotejo.

ELABORÓ:	Comité técnico académico de diseño curricular del subsistema de CGUTyP de la familia de carreras de TSU en Tecnologías de la Información	REVISÓ:	Dirección Académica	
APROBÓ:	C. G. U. T. y P.	FECHA DE ENTRADA EN VIGOR:	Septiembre de 2018	

METODOLOGÍAS Y MODELADO DE DESARROLLO DE SOFTWARE

PROCESO ENSEÑANZA APRENDIZAJE

Métodos y técnicas de enseñanza	Medios y materiales didácticos
<ul style="list-style-type: none">- Equipos colaborativos.- Análisis de casos.- Solución de problemas.	<p>Pizarrón. Plumones. Computadora. Internet. Equipo multimedia. Ejercicios prácticos. Plataformas virtuales. Diagramadores.</p>

ESPACIO FORMATIVO

Aula	Laboratorio / Taller	Empresa
X	X	

ELABORÓ:	Comité técnico académico de diseño curricular del subsistema de CGUTyP de la familia de carreras de TSU en Tecnologías de la Información	REVISÓ:	Dirección Académica	
APROBÓ:	C. G. U. T. y P.	FECHA DE ENTRADA EN VIGOR:	Septiembre de 2018	

METODOLOGÍAS Y MODELADO DE DESARROLLO DE SOFTWARE

CAPACIDADES DERIVADAS DE LAS COMPETENCIAS PROFESIONALES A LAS QUE CONTRIBUYE LA ASIGNATURA

Capacidad	Criterios de Desempeño
Establecer requerimientos funcionales y no funcionales mediante técnicas y metodologías de análisis de requerimientos para atender la necesidad planteada.	Entrega un documento de levantamiento de requerimientos que incluya: <ul style="list-style-type: none"> • Fecha • Nombre del Proyecto • Objetivo • Alcance • Descripción funcional • Requerimientos: <ul style="list-style-type: none"> - Software - Hardware
Diseñar propuestas de interfaces web considerando las especificaciones del cliente y técnicas de diseño web para mejorar el entorno visual.	Entrega diseño de las interfaces del sitio WEB integrando lo siguiente: <ul style="list-style-type: none"> • Mockups con componentes de diseño (Imágenes, logo corporativo, galerías, calendarios, redes sociales, banners, paletas de colores). • Componentes de control (menús, combos, carrito de compras). • Mapa de sitio: navegación. • Justificación técnica del diseño.
Codificar aplicaciones web a través de los fundamentos de programación orientada a objetos y conexión a base de datos para desarrollarla.	Entrega el código fuente documentado de la aplicación web: <ul style="list-style-type: none"> • Métodos. • Atributos. • Variables. • Conexión a la base de datos. • Componentes.

ELABORÓ:	Comité técnico académico de diseño curricular del subsistema de CGUTyP de la familia de carreras de TSU en Tecnologías de la Información	REVISÓ:	Dirección Académica	
APROBÓ:	C. G. U. T. y P.	FECHA DE ENTRADA EN VIGOR:	Septiembre de 2018	

METODOLOGÍAS Y MODELADO DE DESARROLLO DE SOFTWARE

FUENTES BIBLIOGRÁFICAS

Autor	Año	Título del Documento	Ciudad	País	Editorial
Damien André, Jean-Luc Charles, Ivan Iordanoff	2015 ISBN 9781848217720	<i>Object Oriented Approach and UML</i>	EEUU	EEUU	Wiley
Karoly Nyisztor	2018 ISBN 1980818495	<i>UML and Object-Oriented Design Foundations: Understanding Object-Oriented Programming and the Unified Modeling Language (Professional Skills Book 1)</i>	EEUU	EEUU	Amazon
Barbara Haley Wixom	2015 9781118804674	<i>Systems Analysis and Design: An Object-Oriented Approach with UML</i>	EEUU	EEUU	Wiley
Phillip A. Laplante	2017 ISBN 9781138196117	<i>Requirements Engineering for Software and Systems, Third Edition</i>	Boca Ratón, FL	Estados Unidos	Auerbach Publications
Humberto Cervantes Maceda, Perla Velasco-Elizondo, Luis Fernando Castro Careaga	2016 ISBN 9786075224565	<i>Arquitectura de Software: Conceptos y Ciclo de Desarrollo.</i>	Ciudad de México	México	Cengage Learning

ELABORÓ:	Comité técnico académico de diseño curricular del subsistema de CGUTyP de la familia de carreras de TSU en Tecnologías de la Información	REVISÓ:	Dirección Académica	
APROBÓ:	C. G. U. T. y P.	FECHA DE ENTRADA EN VIGOR:	Septiembre de 2018	

METODOLOGÍAS Y MODELADO DE DESARROLLO DE SOFTWARE

FUENTES BIBLIOGRÁFICAS

Autor	Año	Título del Documento	Ciudad	País	Editorial
Jeff Sutherland	2015 ISBN 9788408135326	<i>Scrum: el nuevo y revolucionario modelo organizativo que cambiará tu vida</i>	Barcelona	España	Planeta
Tridibesh Satpathy	2017 ISBN 9780989925204	<i>Una guía para el Cuerpo de Conocimiento de Scrum (Guía SBOOK™) – 3ra Edición</i>	Arizona	Estados Unidos	ScrumStudy™

ELABORÓ:	Comité técnico académico de diseño curricular del subsistema de CGUTyP de la familia de carreras de TSU en Tecnologías de la Información	REVISÓ:	Dirección Académica	
APROBÓ:	C. G. U. T. y P.	FECHA DE ENTRADA EN VIGOR:	Septiembre de 2018	