

ASIGNATURA DE ARQUITECTURAS DE SOFTWARE

1. Competencias	Construir soluciones de software seguro y sistemas inteligentes mediante la dirección y el liderazgo en la gestión de proyectos, integrando metodologías y arquitecturas de desarrollo para la optimización de proyectos de investigación, innovación, desarrollo tecnológico y emprendimiento, bajo la normatividad aplicable.
2. Cuatrimestre	Séptimo
3. Horas Teóricas	21
4. Horas Prácticas	54
5. Horas Totales	75
6. Horas Totales por Semana Cuatrimestre	5
7. Objetivo de aprendizaje	El alumno establecerá arquitecturas de software con base en el análisis de requerimientos para satisfacer los atributos de calidad del software y servir como guía en el desarrollo

Unidades de Aprendizaje	Horas		
	Teóricas	Prácticas	Totales
1. Fundamentos de arquitecturas de software	10	5	15
2. Modelado de arquitecturas de software	5	25	30
3. Patrones de diseño	6	24	30
Totales			75

ELABORÓ:	Comité de Directores de la Carrera de Ingeniería en Desarrollo y Gestión de Software	REVISÓ:	Dirección Académica	
APROBÓ:	C. G. U. T. y P.	FECHA DE ENTRADA EN VIGOR:	Septiembre de 2020	

ARQUITECTURAS DE SOFTWARE

UNIDADES DE APRENDIZAJE

1. Unidad de aprendizaje	1. Fundamentos de arquitecturas de software
2. Horas Teóricas	10
3. Horas Prácticas	5
4. Horas Totales	15
5. Objetivo de la Unidad de Aprendizaje	El alumno determinará la arquitectura de software de acuerdo a requerimientos para guiar la construcción de los componentes de software.

Temas	Saber	Saber hacer	Ser
Introducción a las arquitecturas de software	Definir el concepto de arquitectura de software Describir las características las arquitecturas de software		Analítico razonamiento Deductivo uso De Procesos Cognitivos razonamiento Lógico
Frameworks de arquitecturas	Describir los frameworks de arquitectura de software.		Analítico razonamiento Deductivo uso De Procesos Cognitivos razonamiento Lógico
Estilos de arquitectura de software	Identificar requerimientos Funcionales y No funcionales Distinguir los estilos de arquitecturas de software - Cliente-Servidor - Microservicios - N capas - Orientada a servicios - Dirigida a eventos - Basada en espacio - Microkernel - Serverless	Establecer la arquitectura de software cumpliendo con los requerimientos funcionales y no funcionales.	Analítico razonamiento Deductivo uso De Procesos Cognitivos razonamiento Lógico

ELABORÓ:	Comité de Directores de la Carrera de Ingeniería en Desarrollo y Gestión de Software	REVISÓ:	Dirección Académica	
APROBÓ:	C. G. U. T. y P.	FECHA DE ENTRADA EN VIGOR:	Septiembre de 2020	

ARQUITECTURAS DE SOFTWARE

PROCESO DE EVALUACIÓN

Resultado de aprendizaje	Secuencia de aprendizaje	Instrumentos y tipos de reactivos
<p>Elabora un documento que incluya:</p> <ul style="list-style-type: none">- Estilos de arquitectura de software- Características de las arquitecturas de software <p>Elaborar un documento a partir de casos prácticos que incluya:</p> <ul style="list-style-type: none">- Requerimientos Funcionales y No Funcionales- Justificación de la arquitectura seleccionada	<ol style="list-style-type: none">1. Identificar los estilos de arquitecturas de software2. Comprender las características de las arquitecturas de software3. Analizar los requerimientos Funcionales y No funcionales4. Explicar el uso de las arquitecturas de software	<ol style="list-style-type: none">1. Ejercicios Prácticos2. Lista de cotejo

ELABORÓ:	Comité de Directores de la Carrera de Ingeniería en Desarrollo y Gestión de Software	REVISÓ:	Dirección Académica	
APROBÓ:	C. G. U. T. y P.	FECHA DE ENTRADA EN VIGOR:	Septiembre de 2020	

ARQUITECTURAS DE SOFTWARE

PROCESO ENSEÑANZA APRENDIZAJE

Métodos y técnicas de enseñanza	Medios y materiales didácticos
<ul style="list-style-type: none">- Prácticas en laboratorios dirigidas y no dirigidas- Solución de problemas- Tareas de investigación	<ul style="list-style-type: none">- Equipos de cómputo- Proyector- Internet- Pizarrón y marcadores- Plataformas virtuales

ESPACIO FORMATIVO

Aula	Laboratorio / Taller	Empresa
	X	

ELABORÓ:	Comité de Directores de la Carrera de Ingeniería en Desarrollo y Gestión de Software	REVISÓ:	Dirección Académica	
APROBÓ:	C. G. U. T. y P.	FECHA DE ENTRADA EN VIGOR:	Septiembre de 2020	

ARQUITECTURAS DE SOFTWARE

UNIDADES DE APRENDIZAJE

6. Unidad de aprendizaje	2. Modelado de arquitecturas de software
7. Horas Teóricas	5
8. Horas Prácticas	25
9. Horas Totales	30
10. Objetivo de la Unidad de Aprendizaje	El alumno elaborará el modelo de arquitecturas de software para establecer el marco de referencia y guiar el proceso de desarrollo de software.

Temas	Saber	Saber hacer	Ser
Modelos y lenguajes de descripción de arquitectura	Identificar modelos de arquitectura. Identificar lenguajes de descripción de arquitectura.	Seleccionar modelos y lenguajes de descripción de arquitectura.	Analítico razonamiento Deductivo uso De Procesos Cognitivos razonamiento Lógico trabajo En Equipo
Diagramas y herramientas de diseño arquitectónico de software	Describir diagramas de diseño arquitectónico de software. Identificar herramientas de modelado de arquitecturas de software.	Diseñar arquitecturas de software con herramientas de modelado. Implementar arquitecturas en el desarrollo de software.	Analítico razonamiento Deductivo uso De Procesos Cognitivos razonamiento Lógico trabajo En Equipo

ELABORÓ:	Comité de Directores de la Carrera de Ingeniería en Desarrollo y Gestión de Software	REVISÓ:	Dirección Académica	
APROBÓ:	C. G. U. T. y P.	FECHA DE ENTRADA EN VIGOR:	Septiembre de 2020	

ARQUITECTURAS DE SOFTWARE

PROCESO DE EVALUACIÓN

Resultado de aprendizaje	Secuencia de aprendizaje	Instrumentos y tipos de reactivos
<p>Elabora un documento a partir de un caso de estudio que incluya:</p> <ul style="list-style-type: none">- Modelo y lenguaje seleccionado- Diagramas de la arquitectura de software <p>Elabora prototipo funcional de software en el que aplique una arquitectura</p>	<ol style="list-style-type: none">1. Identificar los modelos2. Identificar los lenguajes de descripción3. Comprender las primitivas de los diagramas de arquitectura4. Explicar el uso de las herramientas de modelado5 Relacionar los diagramas de arquitectura con el prototipo funcional de software.	<ol style="list-style-type: none">1. Estudio de casos2. Lista de cotejo

ELABORÓ:	Comité de Directores de la Carrera de Ingeniería en Desarrollo y Gestión de Software	REVISÓ:	Dirección Académica	
APROBÓ:	C. G. U. T. y P.	FECHA DE ENTRADA EN VIGOR:	Septiembre de 2020	

ARQUITECTURAS DE SOFTWARE

PROCESO ENSEÑANZA APRENDIZAJE

Métodos y técnicas de enseñanza	Medios y materiales didácticos
<ul style="list-style-type: none">- Prácticas en laboratorios dirigidas y no dirigidas- Estudio de casos- Solución de problemas	<ul style="list-style-type: none">- Equipos de cómputo- Proyector- Internet- Pizarrón y marcadores- Plataformas virtuales- herramientas de modelado- IDE de desarrollo- Manejadores de base de datos

ESPACIO FORMATIVO

Aula	Laboratorio / Taller	Empresa
	X	

ELABORÓ:	Comité de Directores de la Carrera de Ingeniería en Desarrollo y Gestión de Software	REVISÓ:	Dirección Académica	
APROBÓ:	C. G. U. T. y P.	FECHA DE ENTRADA EN VIGOR:	Septiembre de 2020	

ARQUITECTURAS DE SOFTWARE

UNIDADES DE APRENDIZAJE

11. Unidad de aprendizaje	3. Patrones de diseño
12. Horas Teóricas	6
13. Horas Prácticas	24
14. Horas Totales	30
15. Objetivo de la Unidad de Aprendizaje	El alumno determinará patrones de diseño para resolver problemas específicos en el desarrollo de software.

Temas	Saber	Saber hacer	Ser
Introducción a los patrones de diseño	<p>Definir el concepto de patrón de diseño.</p> <p>Enlistar los patrones de diseño:</p> <ul style="list-style-type: none"> - Creacionales - Estructurales - De comportamientos - Emergentes <p>Definir los antipatrones y los patrones GRASP</p>		<p>Analítico</p> <p>Razonamiento Deductivo</p> <p>Uso de procesos cognitivos</p> <p>Razonamiento lógico</p> <p>Razonamiento deductivo</p> <p>Razonamiento inductivo</p> <p>Trabajo en equipo</p>
Patrones de diseño GoF	<p>Describir las características de los patrones de diseño:</p> <ul style="list-style-type: none"> - creacionales - estructurales - de comportamiento 	<p>Realizar diagramas de los patrones de diseño:</p> <ul style="list-style-type: none"> - Creacionales - Estructurales - De comportamiento <p>Implementar patrones de diseño en el desarrollo de software:</p> <ul style="list-style-type: none"> - Creacionales - Estructurales - De comportamiento 	<p>Analítico</p> <p>Razonamiento Deductivo</p> <p>Uso de procesos cognitivos</p> <p>Razonamiento lógico</p> <p>Razonamiento deductivo</p> <p>Razonamiento inductivo</p> <p>Trabajo en equipo</p>

ELABORÓ:	Comité de Directores de la Carrera de Ingeniería en Desarrollo y Gestión de Software	REVISÓ:	Dirección Académica	
APROBÓ:	C. G. U. T. y P.	FECHA DE ENTRADA EN VIGOR:	Septiembre de 2020	

Temas	Saber	Saber hacer	Ser
Patrones de diseño emergentes	Describir las características de los patrones de diseño emergentes - MVC - DAO - CQRS - DDD - MVVM - MVP	Realizar diagramas de los patrones de diseño: - MVC - DAO - CQRS - DDD - MVVM - MVP Implementar patrones de diseño en el desarrollo de software: - MVC - DAO - CQRS - DDD - MVVM - MVP	Analítico Razonamiento Deductivo Uso de procesos cognitivos Razonamiento lógico Razonamiento deductivo Razonamiento inductivo Trabajo en equipo

ELABORÓ:	Comité de Directores de la Carrera de Ingeniería en Desarrollo y Gestión de Software	REVISÓ:	Dirección Académica	
APROBÓ:	C. G. U. T. y P.	FECHA DE ENTRADA EN VIGOR:	Septiembre de 2020	

ARQUITECTURAS DE SOFTWARE

PROCESO DE EVALUACIÓN

Resultado de aprendizaje	Secuencia de aprendizaje	Instrumentos y tipos de reactivos
<p>Elabora un documento a partir de casos prácticos que incluya:</p> <ul style="list-style-type: none"> - Justificación de Patrones de diseño seleccionados - Diagramas de los patrones de diseño seleccionados <p>Elabora prototipo funcional de software utilizando al menos un patrón de diseño:</p> <ul style="list-style-type: none"> - Creacional. - Estructural. - De comportamiento. 	<ol style="list-style-type: none"> 1. Identificar los patrones de diseño 2. Comprender las características de los patrones de diseño 4. Explicar el uso de las herramientas de modelado 5 Relacionar los patrones de diseño con el prototipo funcional de software. 	<ol style="list-style-type: none"> 1. Ejercicios Prácticos 2. Lista de cotejo

ELABORÓ:	Comité de Directores de la Carrera de Ingeniería en Desarrollo y Gestión de Software	REVISÓ:	Dirección Académica	
APROBÓ:	C. G. U. T. y P.	FECHA DE ENTRADA EN VIGOR:	Septiembre de 2020	

ARQUITECTURAS DE SOFTWARE

PROCESO ENSEÑANZA APRENDIZAJE

Métodos y técnicas de enseñanza	Medios y materiales didácticos
<ul style="list-style-type: none">- Prácticas en laboratorios dirigidas y no dirigidas- Solución de problemas- Tareas de investigación	<ul style="list-style-type: none">- Equipos de cómputo- Proyector- Internet- Pizarrón y marcadores- Plataformas virtuales- herramientas de modelado- IDE de desarrollo- Manejadores de base de datos- Frameworks

ESPACIO FORMATIVO

Aula	Laboratorio / Taller	Empresa
	X	

ELABORÓ:	Comité de Directores de la Carrera de Ingeniería en Desarrollo y Gestión de Software	REVISÓ:	Dirección Académica	
APROBÓ:	C. G. U. T. y P.	FECHA DE ENTRADA EN VIGOR:	Septiembre de 2020	

ARQUITECTURAS DE SOFTWARE

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

Capacidad	Criterios de Desempeño
Determinar procesos y herramientas de machine learning, data mining y big data mediante el análisis del problema de acuerdo a las características, ubicación de los datos y normativa aplicable para establecer un plan de desarrollo e integración que cubra las necesidades de información.	Entrega un plan de desarrollo e integración que incluya: <ul style="list-style-type: none"> - Planteamiento del problema. - Definición necesidades de información. - Descripción de la ubicación y características de los datos. - Justificación de la normativa aplicable. - Justificación las herramientas y procesos a utilizar.
Implementar modelos de procesamiento mediante la incorporación de API's y frameworks de desarrollo tomando en cuenta la normatividad aplicable para la integración de sistemas inteligentes .	Entrega productos generados que incluyan: <ul style="list-style-type: none"> - Diagrama de arquitectura software. - Requerimientos técnicos para integrar el modelo a la aplicación. - Descripción de los datos de entrada y salida.
Determinar la arquitectura de software mediante el análisis de los requerimientos, la definición de los componentes y la relación entre ellos para satisfacer los atributos de calidad del software y servir como guía en el desarrollo	Entrega un documento de arquitectura de software que incluya: <ul style="list-style-type: none"> - Requerimientos funcionales y no funcionales de desarrollo de software. - Justificación de la arquitectura seleccionada. - Diagrama de arquitectura de software. - Características y funciones de los componentes. - Diagramas necesarios de acuerdo a la arquitectura seleccionada.
Implementar modelos de datos mediante herramientas y sistemas gestores para garantizar la disponibilidad e integridad de la información	1. Entrega un documento que incluya: <ul style="list-style-type: none"> - Justificación de las herramientas a utilizar para los modelos de datos. - Modelos de datos. - Descripción de metadatos. 2. Entrega archivos, credenciales de registro y secuencia de configuración para la creación de los modelos de datos.

ELABORÓ:	Comité de Directores de la Carrera de Ingeniería en Desarrollo y Gestión de Software	REVISÓ:	Dirección Académica	
APROBÓ:	C. G. U. T. y P.	FECHA DE ENTRADA EN VIGOR:	Septiembre de 2020	

Capacidad	Criterios de Desempeño
<p>Desarrollar componentes mediante el uso de patrones de diseño, APIs y frameworks de desarrollo conforme a la arquitectura establecida para la integración continua de unidades del software</p>	<p>1. Entrega un documento que incluya:</p> <ul style="list-style-type: none"> - Justificación de las herramientas para el desarrollo de componentes. - Justificación de los patrones de diseño a utilizar. - Justificación de los lenguajes de programación a utilizar. <p>2. Entrega archivos de código fuente y configuración de los componentes desarrollados.</p> <p>3. Entrega evidencia de almacenamiento de archivos de código fuente y actualizaciones en la documentación del proyecto en la plataforma de versionamiento seleccionada.</p>
<p>Implementar soluciones de software mediante la integración de componentes de acuerdo a la arquitectura definida para su liberación en un ambiente de producción</p>	<p>1. Entrega manuales del software establecidos en el plan de desarrollo.</p> <p>2. Entrega la versión estable del software integrando los componentes de acuerdo al plan de entregas continuas.</p> <p>3. Entrega evidencia de almacenamiento de archivos de código fuente y actualizaciones de la documentación del proyecto en la plataforma de versionamiento establecida.</p>

ELABORÓ:	Comité de Directores de la Carrera de Ingeniería en Desarrollo y Gestión de Software	REVISÓ:	Dirección Académica	
APROBÓ:	C. G. U. T. y P.	FECHA DE ENTRADA EN VIGOR:	Septiembre de 2020	

ARQUITECTURAS DE SOFTWARE

FUENTES BIBLIOGRÁFICAS

Autor	Año	Título del Documento	Ciudad	País	Editorial
Mark Richards, Neal Ford	2020 ISBN:97:81-492043-454	<i>Fundamentals of Software Architecture</i>	New York	United States	O'Reilly Media
Robert C. Martin	2018 ISBN:013-449416-4-860	<i>Clean Architecture</i>	Boston	United States	Addison - Wesley - Pearson
Angel Arias, Alicia Durango	2016 ISBN:97:81-52336-5-487	<i>Ingeniería y Arquitectura del Software</i>	Vigo	España	IT Campus Academy
Humberto Cervantes, Rick Kazman	2016 ISBN:97:80-13439-0-789	<i>Designing Software Architectures: A Practical Approach</i>	Boston	United States	Addison - Wesley - Pearson
Laurent Debrauwer	2018 ISBN:97:82-40901-6-349	<i>Patrones de Diseño en Java</i>	Barcelona	España	Eni Ediciones
David Roldan Martinez, Pedro J. Valderas	2018 ISBN:97:88-49964-7-654	<i>Microservicios, un enfoque integrado</i>	Jarama	España	Ra-Ma
Fien Van Der Heyde	2019 ISBN:97:82-40901-8-046	<i>UML 2.5 Domine el diseño con los Patrones de diseño</i>	Barcelona	España	Eni Ediciones

ELABORÓ:	Comité de Directores de la Carrera de Ingeniería en Desarrollo y Gestión de Software	REVISÓ:	Dirección Académica	
APROBÓ:	C. G. U. T. y P.	FECHA DE ENTRADA EN VIGOR:	Septiembre de 2020	

Autor	Año	Título del Documento	Ciudad	País	Editorial
IEEE/ISO/IEC	2011 ISBN(s):9781504413671, 9780738171678, 9780738171425	<i>ISO/IEC/IEEE Std 42010:2011(E), Systems and software engineering -- Architecture description</i>	New York	United States	ISO/IEC/IEEE

ELABORÓ:	Comité de Directores de la Carrera de Ingeniería en Desarrollo y Gestión de Software	REVISÓ:	Dirección Académica	
APROBÓ:	C. G. U. T. y P.	FECHA DE ENTRADA EN VIGOR:	Septiembre de 2020	