

PLANEACIÓN DIDÁCTICA DESDE LA ENSEÑANZA  
BASADA EN COMPETENCIAS

Programa Educativo: INGENIERÍA EN DESARROLLO Y GESTIÓN DE SOFTWARE	Facilitador: LIC. DULCE SUGEY RODRÍGUEZ GONZÁLEZ
Cuatrimestre: 7 "M"	Periodo Escolar: SEPTIEMBRE-DICIEMBRE-2020

1. DATOS GENERALES DE LA ASIGNATURA					
Nombre de la asignatura:	Arquitectura del Software				
Competencia(s) que desarrolla:	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.				
Horas prácticas:	54	Horas teóricas:	21	Horas totales:	75
Objetivo:	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.				
Nombre de las unidades temáticas:	1. Fundamentos de arquitecturas de software 2. Modelado de arquitecturas de software 3. Patrones de diseño				
2. DATOS DE LAS UNIDADES TEMÁTICAS					
Número y nombre de la unidad temática	Objetivo general por unidad temática	Temas de cada unidad temática			
1. Fundamentos de arquitecturas de software	El alumno determinará la arquitectura de software de acuerdo a requerimientos para guiar la construcción de los componentes de software.	- Introducción a las arquitecturas de software. - Frameworks de arquitecturas - Estilos de arquitectura de software.			
2. Modelado de arquitecturas de software	El alumno elaborará el modelo de arquitecturas de software para establecer el marco de referencia y guiar el proceso de desarrollo de software.	- Modelos y lenguajes de descripción de arquitectura. - Diagramas y herramientas de diseño arquitectónico de software			
3. Patrones de diseño	El alumno determinará patrones de diseño para resolver problemas específicos en el desarrollo de software.	- Introducción a los patrones de diseño. - Patrones de diseño GoF. - Patrones de diseño emergentes			

**3. SECUENCIA DIDÁCTICA POR UNIDAD TEMÁTICA(UNA TABLA POR UNIDAD DE CURSO)**

<b>Unidad:</b>	Fundamentos de arquitecturas de software	<b>Duración (Horas)*:</b>	15
<b>Objetivo de unidad:</b>	El alumno determinará la arquitectura de software de acuerdo a requerimientos para guiar la construcción de los componentes de software.		
Tipos de Saberes			
Saber	Saber Hacer	Ser	
<p>* Definir el concepto de arquitectura de software</p> <p>Describir las características las arquitecturas de software</p> <p>* Describir los frameworks de arquitectura de software.</p> <p>* Identificar requerimientos Funcionales y No funcionales</p> <p>Distinguir los estilos de arquitecturas de software</p> <ul style="list-style-type: none"> <li>- Cliente-Servidor</li> <li>- Microservicios</li> <li>- N capas</li> <li>- Orientada a servicios</li> <li>- Dirigida a eventos</li> <li>- Basada en espacio</li> <li>- Microkernel</li> <li>- Serverless</li> </ul>	<p>Establecer la arquitectura de software cumpliendo con los requerimientos funcionales y no funcionales.</p>	<p>Analítico</p> <p>razonamiento Deductivo</p> <p>uso De Procesos Cognitivos</p> <p>razonamiento Lógico</p>	
Resultado de la unidad de aprendizaje			
<p>Elabora un documento que incluya:</p> <ul style="list-style-type: none"> <li>- Estilos de arquitectura de software</li> <li>- Características de las arquitecturas de software</li> </ul> <p>Elaborar un documento a partir de casos prácticos que incluya:</p> <ul style="list-style-type: none"> <li>- Requerimientos Funcionales y No Funcionales</li> <li>- Justificación de la arquitectura seleccionada</li> </ul>			

Secuencia didáctica		
Actividades iniciales	Actividades de desarrollo	Actividades finales
1. Evaluación diagnóstica 2. Identificar los estilos de arquitectura de software	1. Comprender las características de las arquitecturas de software 2. Analizar los requerimientos Funcionales y No funcionales 3. Realizar informe sobre arquitectura y características	1. Explicar el uso de las arquitecturas de software 2. Realizar informe que contenga requerimientos funcionales y no funcionales. 3. Evaluación sumativa
<b>Medios y materiales didácticos:</b>	Computadora, Pizarrón / Plumones , Internet	
<b>Estrategias de enseñanza:</b>	Aprendizaje orientado a proyectos	
<b>Técnicas de enseñanza:</b>	Lluvia de ideas, Equipos	
<b>Estrategias de aprendizaje:</b>	Mapas conceptuales , Resumen	
<b>Evidencias de aprendizaje:</b>	Documentos digitales que contengan estilos de arquitectura y sus características y casos prácticos con definición de sus requerimientos funcionales y requerimientos no funcionales.	

4. DESCRIPCIÓN DEL SISTEMA DE EVALUACIÓN DE LA UNIDAD DE APRENDIZAJE			
Tipo de Evaluación	Estrategia de Evaluación	Instrumento de Evaluación	
Evaluación Diagnóstica:	Informes	Tipo de Instrumento	
		Lista de Cotejo o verificación	
Evaluación Formativa:		Tipo de instrumento	Valor del instrumento (%)
	Informes	Lista de Cotejo o verificación	30 %
	Portafolio de evidencias	Diario de clase	30 %
	Otro	Examen	40 %
			100 %
Evaluación Sumativa (Fecha de asignación de la calificación)	09/10/2020		

3. SECUENCIA DIDÁCTICA POR UNIDAD TEMÁTICA(UNA TABLA POR UNIDAD DE CURSO)			
<b>Unidad:</b>	Modelado de arquitecturas de software	<b>Duración (Horas)*:</b>	30
<b>Objetivo de unidad:</b>	El alumno elaborará el modelo de arquitecturas de software para establecer el marco de referencia y guiar el proceso de desarrollo de software.		
Tipos de Saberes			
Saber	Saber Hacer	Ser	
* Identificar modelos de arquitectura.  Identificar lenguajes de descripción de arquitectura.  * Describir diagramas de diseño arquitectónico de software.  Identificar herramientas de modelado de arquitecturas de software.	- Seleccionar modelos y lenguajes de descripción de arquitectura.  - 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	
Resultado de la unidad de aprendizaje			
Elabora un documento a partir de un caso de estudio que incluya: - Modelo y lenguaje seleccionado - Diagramas de la arquitectura de software  Elabora prototipo funcional de software en el que aplique una arquitectura			

Secuencia didáctica		
Actividades iniciales	Actividades de desarrollo	Actividades finales
1. Evaluación diagnóstica 2. Identificar modelos	1. Identificar lenguajes de descripción 2. Comprender las primitivas de los diagramas de arquitectura 3. Ejercicio práctico	1. Explicar el uso de las herramientas de modelado 2. Relacionar los diagramas de arquitectura con el prototipo funcional de software 3. Evaluación sumativa
<b>Medios y materiales didácticos:</b>	Computadora, Pizarrón / Plumones, Internet	
<b>Estrategias de enseñanza:</b>	Aprendizaje orientado a proyectos, Proyecto colaborativo	
<b>Técnicas de enseñanza:</b>	Lluvia de ideas, Equipos	
<b>Estrategias de aprendizaje:</b>	Mapas conceptuales, Cuadros sinópticos	
<b>Evidencias de aprendizaje:</b>	Documentación que incluya Modelado y lenguaje seleccionado, diagramas de arquitectura de software y un prototipo funcional de software en el que se aplique una arquitectura.	

4. DESCRIPCIÓN DEL SISTEMA DE EVALUACIÓN DE LA UNIDAD DE APRENDIZAJE			
Tipo de Evaluación	Estrategia de Evaluación	Instrumento de Evaluación	
Evaluación Diagnóstica:	Contrato de aprendizaje	Tipo de Instrumento	
		Cuestionarios de autoevaluación	
Evaluación Formativa:		Tipo de instrumento	Valor del instrumento (%)
	Informes	Lista de Cotejo o verificación	20 %
	Proyectos	Lista de Cotejo o verificación	40 %
	Otro	Lista de Cotejo o verificación	40 %
			100 %
Evaluación Sumativa (Fecha de asignación de la calificación)	23/11/2020		

**3. SECUENCIA DIDÁCTICA POR UNIDAD TEMÁTICA(UNA TABLA POR UNIDAD DE CURSO)**

<b>Unidad:</b>	Patrones de diseño	<b>Duración (Horas)*:</b>	30
<b>Objetivo de unidad:</b>	El alumno determinará patrones de diseño para resolver problemas específicos en el desarrollo de software.		

**Tipos de Saberes**

Saber	Saber Hacer	Ser
<p>* Definir el concepto de patrón de diseño.</p> <p>Enlistar los patrones de diseño:</p> <ul style="list-style-type: none"> <li>- Creacionales</li> <li>- Estructurales</li> <li>- De comportamientos</li> <li>- Emergentes</li> </ul> <p>Definir los antipatrones y los patrones GRASP</p> <p>* Describir las características de los patrones de diseño:</p> <ul style="list-style-type: none"> <li>- creacionales</li> <li>- estructurales</li> <li>- de comportamiento</li> </ul> <p>*Describir las características de los patrones de diseño emergentes</p> <ul style="list-style-type: none"> <li>- MVC</li> <li>- DAO</li> <li>- CQRS</li> <li>- DDD</li> <li>- MVVM</li> <li>- MVP</li> </ul>	<p>* Realizar diagramas de los patrones de diseño:</p> <ul style="list-style-type: none"> <li>- Creacionales</li> <li>- Estructurales</li> <li>- De comportamiento</li> </ul> <p>Implementar patrones de diseño en el desarrollo de software:</p> <ul style="list-style-type: none"> <li>- Creacionales</li> <li>- Estructurales</li> <li>- De comportamiento</li> </ul> <p>*Realizar diagramas de los patrones de diseño:</p> <ul style="list-style-type: none"> <li>- MVC</li> <li>- DAO</li> <li>- CQRS</li> <li>- DDD</li> <li>- MVVM</li> <li>- MVP</li> </ul> <p>Implementar patrones de diseño en el desarrollo de software:</p> <ul style="list-style-type: none"> <li>- MVC</li> <li>- DAO</li> <li>- CQRS</li> <li>- DDD</li> <li>- MVVM</li> <li>- MVP</li> </ul>	<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>

**Resultado de la unidad de aprendizaje**

<p>Elabora un documento a partir de casos prácticos que incluya:</p> <ul style="list-style-type: none"> <li>- Justificación de Patrones de diseño seleccionados</li> <li>- Diagramas de los patrones de diseño seleccionados</li> </ul> <p>Elabora prototipo funcional de software utilizando al menos un patrón de diseño:</p> <ul style="list-style-type: none"> <li>- Creacional.</li> <li>- Estructural.</li> <li>- De comportamiento.</li> </ul>
---



Secuencia didáctica		
Actividades iniciales	Actividades de desarrollo	Actividades finales
1. Evaluación diagnóstica 2. Identificar patrones de diseño 3. Comprender las características de los patrones de diseño	1. Explicar el uso de las herramientas de modelado 2. Realizar informe con casos prácticos, justificando los patrones y diagramas de patrones de diseño	1. Relacionar los patrones de diseño con el prototipo funcional de software 2. Evaluación diagnóstica
<b>Medios y materiales didácticos:</b>	Computadora, Pizarrón / Plumones , Internet	
<b>Estrategias de enseñanza:</b>	Aprendizaje orientado a proyectos, Proyecto colaborativo	
<b>Técnicas de enseñanza:</b>	Lluvia de ideas, Equipos	
<b>Estrategias de aprendizaje:</b>	Mapas conceptuales , Mapas mentales, Cuadros comparativos	
<b>Evidencias de aprendizaje:</b>	Elabora un documento a partir de casos prácticos que incluya: - Justificación de Patrones de diseño seleccionados - Diagramas de los patrones de diseño seleccionados Elabora prototipo funcional de software utilizando al menos un patrón de diseño: - C	

4. DESCRIPCIÓN DEL SISTEMA DE EVALUACIÓN DE LA UNIDAD DE APRENDIZAJE			
Tipo de Evaluación	Estrategia de Evaluación	Instrumento de Evaluación	
Evaluación Diagnóstica:	Otro	Tipo de Instrumento	
		Cuestionarios de autoevaluación	
Evaluación Formativa:		Tipo de instrumento	Valor del instrumento (%)
	Exposiciones orales	Lista de Cotejo o verificación	15 %
	Informes	Lista de Cotejo o verificación	15 %
	Proyectos	Lista de Cotejo o verificación	30 %
	Otro	Lista de Cotejo o verificación	40 %
			100 %
Evaluación Sumativa (Fecha de asignación de la calificación)	09/12/2020		

**5. DESCRIPCIÓN DEL PROYECTO INTEGRADOR (Requisitar únicamente para asignaturas integradoras)**

<b>Objetivo:</b>	
<b>Asignaturas que contribuyen a la competencia específica:</b>	
<b>Componentes del proyecto:</b>	

LIC. DULCE SUGEY RODRÍGUEZ GONZÁLEZ

**Elaboró**

El Nith, Ixmiquilpan, Hidalgo

**Lugar**

MTRO. ROMÁN BRAVO CADENA

**Vo. Bo. del Director del PE**

04/09/2020

**Fecha de elaboración**