



Programa de asignatura por competencias de educación superior

Sección I. Identificación del Curso

Tabla 1. Identificación de la Planificación del Curso.

Actualización:	Agosto 29, 2022				
Carrera:	Ingeniería en Desarrollo de Software	Asignatura:	Arquitectura de software		
Academia:	Ciencias Computacionales y Programación /	Clave:	19SDS38		
Módulo formativo:	Programación aplicada	Seriación:	- -		
Tipo de curso:	Presencial	Prerrequisito:	- -		
Semestre:	Quinto	Créditos:	5.63	Horas semestre:	90 horas
Teoría:	2 horas	Práctica:	2 horas	Trabajo indpt.:	1 hora
				Total x semana:	5 horas

Sección II. Objetivos educacionales

Tabla 2. Objetivos educacionales

Objetivos educacionales		Criterios de desempeño	Indicadores
1	Los egresados gestionarán recursos relacionados con el desarrollo de software en alguna organización.	Los egresados podrán aplicar metodologías en el desarrollo de proyectos en el contexto laboral.	20% de los egresados aplicarán metodologías en el desarrollo de software en su contexto laboral.
2	Los egresados diseñarán e implementarán soluciones innovadoras mediante el uso de tecnologías de la información.	Los egresados participarán activamente en el ciclo de desarrollo e integración continuos	25% de los egresados desempeñarán labores de desarrollo e integración continuos.
3	Los egresados desarrollarán conocimiento especializado que les permite enfocarse en un área del conocimiento específico del desarrollo de software.	Los egresados desempeñarán actividades orientadas al aseguramiento de los activos de información de manera resiliente, la gestión de la infraestructura de redes y comunicaciones, o integrando hardware y software para crear soluciones IoT; así como el uso de inteligencia artificial para gestionar datos y reconocer patrones que determinen oportunidades de negocio en las organizaciones.	5% de los egresados desempeñarán labores en desarrollo de soluciones IoT.
4	Los egresados serán capaces de emprender un negocio basado en el desarrollo de un producto o servicio de tecnologías de la información, aportando valor a la generación de empleos e incrementar el bienestar económico y social, de forma ecológica y sustentable.	Los egresados serán capaces de emprender un negocio basado en el desarrollo propio de un producto o servicio de tecnologías de la información.	2% de los egresados tendrán participación en el acta constitutiva de una empresa creada a partir del desarrollo de software para ofrecer un producto o servicio.



Atributos de egreso de plan de estudios		Criterios de desempeño	Componentes
1	Aplicar y analizar procesos de diseño de ingeniería para generar una experiencia de usuario que asegure cubrir las necesidades como las expectativas de clientes y partes interesadas, utilizando y gestionando la infraestructura de red necesaria.	- Realizar procesos adecuados de diseño de ingeniería para con ello desarrollar software altamente eficiente.	1.1 Patrones y antipatrones de arquitectura de software. 1.1.1 Microkernel. 1.1.2 Microservicios. 1.1.3 Arquitectura en capas. 1.1.4 Arquitectura basada en eventos. 1.1.5 Arquitectura basada en el espacio. 1.2 Elección de una arquitectura basada en las necesidades específicas requeridas para lograr la experiencia de usuario buscada en el desarrollo de software. 1.3 Variables de desempeño. 1.4 Elasticidad, escalabilidad, extensibilidad, adaptabilidad; como parte del análisis en la arquitectura de software. 2.1 Herramientas de monitoreo de indicadores. 2.2 Establecimiento de métricas en el ciclo de desarrollo e integración continuos. 2.3 Productividad del ciclo. 2.4 Costos por incremento. 2.5 Medición del impacto en la experiencia del usuario. 2.6 Estrategias de continuidad. 3.1 Modelado de despliegue. 3.2 Técnicas de despliegue. 3.2.1 Big bang. 3.2.2 Rolling.



Continuación: Tabla 2. Objetivos educacionales (continuación)

No.	Atributos de egreso de plan de estudios	Criterios de desempeño	Componentes
			3.2.3 Blue Green. 3.2.4 Canary deployment. 3.3 Entrega, despliegue e integración continuos. 3.4 Flujos de trabajo. 3.4.1 Control de versiones. 3.4.2 Revisión de código. 3.4.3 Integración continua. 3.4.4 Gestión de la configuración. 3.4.5 Automatización de publicaciones. 3.4.6 Supervisión de infraestructuras.
2	Identificar su responsabilidad ética y profesional con el entorno sociocultural y ambiental para aplicar estándares, así como fundamentos legales y normativos, aportando valor al contexto social y sustentable.	- Reconocer sus responsabilidades éticas y profesionales siendo con ello un mejor profesionista para la sociedad.	1.1 Patrones y antipatrones de arquitectura de software. 1.1.1 Microkernel. 1.1.2 Microservicios. 1.1.3 Arquitectura en capas. 1.1.4 Arquitectura basada en eventos. 1.1.5 Arquitectura basada en el espacio. 1.2 Elección de una arquitectura basada en las necesidades específicas requeridas para lograr la experiencia de usuario buscada en el desarrollo de software. 1.3 Variables de desempeño. 1.4 Elasticidad, escalabilidad, extensibilidad, adaptabilidad; como parte del análisis en la arquitectura de software.
3	Reconocer la mejora continua como parte de su desarrollo profesional para mantener un perfil actualizado en desarrollo de software para el diseño e implementación de productos y servicios basados en tecnologías con las tendencias emergentes.	- Actualizarse permanentemente en los patrones arquitectónicos de diseño de software con el propósito de mantenerse al día y desarrollar software con elasticidad, escalabilidad, extensibilidad y adaptabilidad.	1.1 Patrones y antipatrones de arquitectura de software. 1.1.1 Microkernel. 1.1.2 Microservicios. 1.1.3 Arquitectura en capas. 1.1.4 Arquitectura basada en eventos. 1.1.5 Arquitectura basada en el espacio.



Continuación: Tabla 2. Objetivos educacionales (continuación)

No.	Atributos de egreso de plan de estudios	Criterios de desempeño	Componentes
			1.2 Elección de una arquitectura basada en las necesidades específicas requeridas para lograr la experiencia de usuario buscada en el desarrollo de software. 1.3 Variables de desempeño. 1.4 Elasticidad, escalabilidad, extensibilidad, adaptabilidad; como parte del análisis en la arquitectura de software.

Sección III. Atributos de la asignatura

Tabla 3. Atributos de la asignatura

Problema a resolver		
Establecer la adecuada planeación sobre la plataforma tecnológica requerida para desarrollar, desplegar y mantener en producción un producto o servicio mediado por software; así como establecer con base en ello, los patrones de arquitectura adecuados para optimizar recursos, mejorar el desempeño, garantizar la mejora continua y responder de manera resiliente a los cambios en el ambiente de negocio y la manifestación de riesgos.		
Atributos (competencia específica) de la asignatura		
Realizar la elección de un patrón adecuado de arquitectura de software, diseñar e implementar los mecanismos de despliegue de las aplicaciones y establecer mecanismos para medir el desempeño y la continuidad de negocio.		
Aportación a la competencia específica		Aportación a las competencias transversales
Saber	Saber hacer	Saber Ser
<ul style="list-style-type: none"> - Conocer sobre las diferencias entre los procesos de soporte y los procesos sustantivos del desarrollo de software. - Reconocer la importancia de lograr la identificación adecuada del alcance del proyecto que busca como resultado el producto o servicio que satisface las necesidades y expectativas de sus clientes. 	<ul style="list-style-type: none"> - Determinar el patrón de arquitectura de software adecuado al propósito de desarrollo. - Realizar la planeación de las actividades, calcula los costos de acuerdo con la plataforma tecnológica necesaria de acuerdo con la estrategia. - Determinar mejores prácticas para la operación de las actividades de desarrollo y despliegue de productos y servicios de software. - Determinar la estrategia de desarrollo de software basado en un patrón de arquitectura y la plataforma tecnológica que permite lograr un desarrollo, despliegue y desempeño del producto o servicio de manera óptima. 	<ul style="list-style-type: none"> - Realiza las actividades de planeación, ejecución y control del despliegue de los productos o servicios a producción con base en los patrones de arquitectura adecuados para optimizar recursos, mejorar el desempeño, garantizar la mejora continua y responder de manera resiliente a los cambios en el ambiente de negocio y la manifestación de riesgos. - Trabajo en equipo. - Trabajo colaborativo.
Producto integrador de la asignatura, considerando los avances por unidad		
Propuesta, justificación y planeación de un proceso de despliegue, operación y desarrollo continuo.		

Sección IV. Desglose específico por cada unidad formativa

Tabla 4.1. Desglose específico de la unidad "Patrones de arquitectura"

Número y nombre de la unidad: 1. Patrones de arquitectura							
Tiempo y porcentaje para esta unidad:		Teoría:	11 horas	Práctica:	10 horas	Porcentaje del programa:	29.17%
Aprendizajes esperados:		Identificar los aspectos relevantes de la elección adecuada de una arquitectura de desarrollo de software a fin de tomar las mejores decisiones de diseño, adicionalmente podría identificar deficiencias de diseño, eficiencia y seguridad en software de legado.					
Temas y subtemas (secuencia)	Criterios de desempeño	Estrategias didácticas	Estrategias de evaluación	Producto Integrador de la unidad (Evidencia de aprendizaje de la unidad)			
1 Patrones y antipatrones de arquitectura de software. 1.1 Microkernel. 1.2 Microservicios. 1.3 Arquitectura en capas. 1.4 Arquitectura basada en eventos. 1.5 Arquitectura basada en el espacio. 2 Elección de una arquitectura basada en las necesidades específicas requeridas para lograr la experiencia de usuario buscada en el desarrollo de software. 3 Variables de desempeño. 4 Elasticidad, escalabilidad, extensibilidad, adaptabilidad; como parte del análisis en la arquitectura de software.	Saber: - Conocer patrones de arquitectura de software. - Conocer las variables y métricas relevantes a la arquitectura de software. Saber hacer: - Identificar los patrones, las variables y métricas relevantes a la arquitectura de software. Ser: - Realiza las actividades de planeación,	- Rescate de conocimientos previos. - Investigación de la información conceptual en fuentes diversas. - Análisis de casos.	Evaluación diagnóstica: - Identificar conocimiento previo. Evaluación formativa: - Trabajo individual conceptual, mediante la realización de ensayo. - Trabajo colaborativo en el que propongan soluciones al análisis de caso. Evaluación sumativa: - Trabajo integrador en equipo para iniciar con un probable proyecto final.	Realización de la identificación de la arquitectura de software de un proyecto.			



Continuación: Tabla 4.1. Desglose específico de la unidad "Patrones de arquitectura"

Temas y subtemas (secuencia)	Criterios de desempeño	Estrategias didácticas	Estrategias de evaluación	Producto Integrador de la unidad
	<p>ejecución y control del despliegue de los productos o servicios a producción con base en los patrones de arquitectura adecuados para optimizar recursos, mejorar el desempeño, garantizar la mejora continua y responder de manera resiliente a los cambios en el ambiente de negocio y la manifestación de riesgos.</p> <ul style="list-style-type: none"> - Trabajo en equipo. - Trabajo colaborativo. 			

Bibliografía

- Vance, W. (2020). Arquitectura limpia: Guía completa para principiantes Aprende todo sobre las estructuras de software utilizando los principios de la arquitectura limpia. Estados Unidos: Joiningthedotstv Limited.
- Martin, R. (2018). Arquitectura Limpia. España: Anaya Multimedia.

Sección IV. Desglose específico por cada unidad formativa

Tabla 4.2. Desglose específico de la unidad "Desarrollo e integración continuos."

Número y nombre de la unidad: 2. Desarrollo e integración continuos.							
Tiempo y porcentaje para esta unidad:		Teoría:	13 horas	Práctica:	13 horas	Porcentaje del programa:	36.11%
Aprendizajes esperados: Establecer procesos de continuidad y mejora para mantener el ciclo de desarrollo e integración continuos.							
Temas y subtemas (secuencia)	Criterios de desempeño	Estrategias didácticas	Estrategias de evaluación	Producto Integrador de la unidad (Evidencia de aprendizaje de la unidad)			
2.1 Herramientas de monitoreo de indicadores. 2.2 Establecimiento de métricas en el ciclo de desarrollo e integración continuos. 2.3 Productividad del ciclo. 2.4 Costos por incremento. 2.5 Medición del impacto en la experiencia del usuario. 2.6 Estrategias de continuidad.	Saber: - Establecer las métricas adecuadas para incrementar la productividad, disminuir costos y asegurar un impacto positivo en la experiencia del usuario. Saber hacer: - Incrementar la productividad y disminuir costos a través del establecimiento de métricas. Ser: - Realiza las actividades de planeación, ejecución y control del despliegue de los	- Investigación de la información conceptual en fuentes diversas. - Análisis de casos.	Evaluación formativa: - Trabajo individual conceptual, mediante la realización de ensayo. - Trabajo colaborativo en el que propongan soluciones al análisis de caso. Evaluación sumativa: - Trabajo integrador en equipo para iniciar con su proyecto final.	Propuesta de planes y estrategias de medición y control del ciclo de desarrollo e integración continuos del producto o servicio orientado al segmento de mercado considerado en su proyecto final.			



Continuación: Tabla 4.2. Desglose específico de la unidad "Desarrollo e integración continuos."

Temas y subtemas (secuencia)	Criterios de desempeño	Estrategias didácticas	Estrategias de evaluación	Producto Integrador de la unidad
	<p>productos o servicios a producción con base en los patrones de arquitectura adecuados para optimizar recursos, mejorar el desempeño, garantizar la mejora continua y responder de manera resiliente a los cambios en el ambiente de negocio y la manifestación de riesgos.</p> <ul style="list-style-type: none">- Trabajo en equipo.- Trabajo colaborativo.			
Bibliografía				
<p>- Vance, W. (2020). Arquitectura limpia: Guía completa para principiantes Aprende todo sobre las estructuras de software utilizando los principios de la arquitectura limpia. Estados Unidos: Joiningthedotstv Limited.</p> <p>- Martin, R. (2018). Arquitectura Limpia. España: Anaya Multimedia.</p>				

Sección IV. Desglose específico por cada unidad formativa

Tabla 4.3. Desglose específico de la unidad "Ambientes de producción y despliegue de producto o servicio."

Número y nombre de la unidad: 3. Ambientes de producción y despliegue de producto o servicio.							
Tiempo y porcentaje para esta unidad:		Teoría:	12 horas	Práctica:	13 horas	Porcentaje del programa:	34.72%
Aprendizajes esperados:		Utilizar metodologías y herramientas para determinar las necesidades y expectativas de un producto o servicio creado por desarrollo de software, que sea innovador y que se establezca la experiencia de usuario deseada.					
Temas y subtemas (secuencia)	Criterios de desempeño	Estrategias didácticas	Estrategias de evaluación	Producto Integrador de la unidad (Evidencia de aprendizaje de la unidad)			
3.1 Modelado de despliegue. 3.2 Técnicas de despliegue. 3.2.1 Big bang. 3.2.2 Rolling. 3.2.3 Blue Green. 3.2.4 Canary deployment. 3.3 Entrega, despliegue e integración continuos. 3.4 Flujos de trabajo. 3.4.1 Control de versiones. 3.4.2 Revisión de código. 3.4.3 Integración continua. 3.4.4 Gestión de la configuración. 3.4.5 Automatización de publicaciones. 3.4.6 Supervisión de infraestructuras.	Saber: - Conocer cómo realizar el despliegue de una aplicación. Saber hacer: - Realizar el despliegue de una aplicación. - Establecer una estrategia de despliegue e integración continuos.	- Investigación de la información conceptual en fuentes diversas. - Análisis de casos.	Evaluación formativa: - Realización de proyecto integrador de asignatura. Evaluación sumativa: - Aplicación de herramientas de despliegue e integración continuos.	Proyecto integrador de la asignatura.			



Continuación: Tabla 4.3. Desglose específico de la unidad "Ambientes de producción y despliegue de producto o servicio."

Temas y subtemas (secuencia)	Criterios de desempeño	Estrategias didácticas	Estrategias de evaluación	Producto Integrador de la unidad
	Ser: - Realiza las actividades de planeación, ejecución y control del despliegue de los productos o servicios a producción con base en los patrones de arquitectura adecuados para optimizar recursos, mejorar el desempeño, garantizar la mejora continua y responder de manera resiliente a los cambios en el ambiente de negocio y la manifestación de riesgos. - Trabajo en equipo. - Trabajo colaborativo.			
Bibliografía				
- Vance, W. (2020). Arquitectura limpia: Guía completa para principiantes Aprende todo sobre las estructuras de software utilizando los principios de la arquitectura limpia. Estados Unidos: Joiningthedotstv Limited. - Martin, R. (2018). Arquitectura Limpia. España: Anaya Multimedia.				



V. Perfil docente

Tabla 5. Descripción del perfil docente

Perfil deseable docente para impartir la asignatura
<p>Carrera(s): - Ingeniería en Desarrollo de Software.</p> <p>- Ingeniería en Electrónica o similar con experiencia en administración de proyectos</p> <p>o carrera afín</p> <ul style="list-style-type: none">- Con experiencia docente en el campo deseable de 2 años. Manejo de TIC's. Con habilidades pedagógicas y uso de metodologías alternativas de enseñanza.- Experiencia mínima de dos años- Mínimo Licenciatura o superior