



Programa de asignatura por competencias de educación superior

Sección I. Identificación del Curso

Tabla 1. Identificación de la Planificación del Curso.

Actualización:	Noviembre 23, 2022				
Carrera:	Ingeniería en Desarrollo de Software	Asignatura:	Introducción al desarrollo de software		
Academia:	Desarrollo de software /	Clave:	19SDS01		
Módulo formativo:	Gestión y operación de proyectos de TI	Seriación:	- -		
Tipo de curso:	Presencial	Prerrequisito:	- -		
Semestre:	Primero	Créditos:	6.75	Horas semestre:	108 horas
Teoría:	3 horas	Práctica:	1 hora	Trabajo indpt.:	2 horas
				Total x semana:	6 horas

Sección II. Objetivos educacionales

Tabla 2. Objetivos educacionales

Objetivos educacionales		Criterios de desempeño	Indicadores
OE1	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.
OE2	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.
OE5	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
AE1	Aplicar los conocimientos de ciencias básicas como física y matemáticas, así como las ciencias de la ingeniería para generar nuevos productos o servicios basándose en la innovación tecnológica.	- Conocer herramientas y metodologías innovadoras para aplicar en el desarrollo de software.	1.1 Herramientas de innovación. 1.1.1 El proceso de diseño de soluciones basado en Design thinking. 1.1.2 Herramientas asociadas con el pensamiento de diseño. 1.1.3 Técnicas de facilitación asociadas con el proceso de diseño de soluciones. 1.1.4 Técnica Delphi y el juicio de expertos.
AE7	Conducir equipos de trabajo interdisciplinarios con principios y valores para solventar problemáticas en la industria del software.	- Utilizar metodologías ágiles e innovadoras para aplicar en desarrollos en la industria del software.	2.1 Metodologías ágiles. 2.1.1 Scrum. 2.1.2 TDD. 2.1.3 BDD.



Continuación: Tabla 2. Objetivos educacionales (continuación)

No.	Atributos de egreso de plan de estudios	Criterios de desempeño	Componentes
			<p>2.1.4 Kanban.</p> <p>2.1.5 Planeación de sprint.</p> <p>2.1.6 Historias de usuario.</p> <p>2.1.7 Casos de prueba.</p> <p>2.1.8 Pruebas unitarias.</p> <p>2.1.9 Cierre de sprint.</p> <p>3.1 Desarrollo de habilidades fisiomotoras específicas en el uso adecuado de teclados para desarrollo de software.</p> <p>3.2 Riesgos en la inadecuada escritura y organización del código fuente.</p> <p>3.3 Buenas prácticas “clean code” para la escritura de programas fuente.</p> <p>3.4 Revisión de código en pares “code review”</p> <p>3.5 Herramientas de gestión de las versiones y ramificaciones del código fuente (herramientas Git).</p>

Sección III. Atributos de la asignatura

Tabla 3. Atributos de la asignatura

Problema a resolver		
Realizar la adecuada identificación de necesidades y expectativas, experiencia de usuario y cómo dirigirlos a los proyectos de desarrollo de software, así como conocer algunas formas adecuadas para realizarlo.		
Atributos (competencia específica) de la asignatura		
Conocer las mejores prácticas de desarrollo de software, metodologías ágiles y control de versiones que le permitan al estudiante realizar de manera adecuada sus desarrollos de software para mejorar la experiencia de usuario y clientes.		
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. - Reconocer la importancia de la innovación y la disrupción como componente del desarrollo de software. - Conocer las herramientas y mejores prácticas de gestión y operación de proyectos. 	<ul style="list-style-type: none"> - Realizar estudios basados en Design Thinking para determinar una experiencia de usuario. - Conocer y aplicar mejores prácticas para organizar, documentar y respaldar su trabajo. - Determinar mejores prácticas para la operación de las actividades de proyecto en equipos pequeños. 	<ul style="list-style-type: none"> - Realiza las actividades de desarrollo de software de manera ágil, con sustento metodológico y con herramientas adecuadas para el trabajo colaborativo en grupos pequeños.
Producto integrador de la asignatura, considerando los avances por unidad		
<ul style="list-style-type: none"> - Experiencia de usuario documentada. - Trabajo colaborativo basado en mejores prácticas, y utilizando herramientas que favorezcan la organización y seguimiento del trabajo en pequeños grupos. 		

Sección IV. Desglose específico por cada unidad formativa

Tabla 4.1. Desglose específico de la unidad "Experiencia de usuario."

Número y nombre de la unidad: 1. Experiencia de usuario.							
Tiempo y porcentaje para esta unidad:		Teoría:	12 horas	Práctica:	12 horas	Porcentaje del programa:	33.33%
Aprendizajes esperados:		Identificar los aspectos relevantes a la experiencia de usuario requerida en el producto o servicio mediante el uso de un marco metodológico adecuado.					
Temas y subtemas (secuencia)	Criterios de desempeño	Estrategias didácticas	Estrategias de evaluación	Producto Integrador de la unidad (Evidencia de aprendizaje de la unidad)			
1.1 Herramientas de innovación. 1.1.1 El proceso de diseño de soluciones basado en Design thinking. 1.1.2 Herramientas asociadas con el pensamiento de diseño. 1.1.3 Técnicas de facilitación asociadas con el proceso de diseño de soluciones. 1.1.4 Técnica Delphi y el juicio de expertos.	Saber: - Conocer las herramientas de innovación. Saber hacer: - Establecer un marco metodológico para recuperar la experiencia de usuario que debe ser implementada en un producto o servicio innovador. Ser: - Realiza las actividades de desarrollo de software de manera ágil, con sustento metodológico y con herramientas adecuadas para el trabajo colaborativo en grupos pequeños.	- Preguntas intercaladas para identificar conocimiento previo. - Investigación de la información conceptual en fuentes diversas. - Análisis de casos. - Proyecto. - Explicación de temas con apoyo de recursos didácticos.	Evaluación diagnóstica: - Rescatar 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 su proyecto final.	- Realización de la identificación de la base metodológica a seguir para obtener las características de un producto o servicio innovador.			
Bibliografía							
- Mootee, I.; Menezo, D. (2014). Design thinking para la innovación estratégica: Lo que no te pueden enseñar en las escuelas de negocios ni en las de diseño (Spanish Edition). España: Empresa Activa. - Lewrick, M.; Link, P.; Leifer L. (2020). The Design Thinking Toolbox: A Guide to Mastering the Most Popular and Valuable Innovation Methods. USA: editorial Wiley.							

Sección IV. Desglose específico por cada unidad formativa

Tabla 4.2. Desglose específico de la unidad "Operación de proyectos en grupos pequeños."

Número y nombre de la unidad:		2. Operación de proyectos en grupos pequeños.					
Tiempo y porcentaje para esta unidad:		Teoría:	12 horas	Práctica:	12 horas	Porcentaje del programa:	33.33%
Aprendizajes esperados:		Innovar mediante el desarrollo de software identificando el conjunto de características de producto que aportan valor a un producto o servicio demandado por el cliente y partes interesadas.					
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 Metodologías ágiles. 2.1.1 Scrum. 2.1.2 TDD. 2.1.3 BDD. 2.1.4 Kanban. 2.1.5 Planeación de sprint. 2.1.6 Historias de usuario. 2.1.7 Casos de prueba. 2.1.8 Pruebas unitarias. 2.1.9 Cierre de sprint.	Saber: - Conocer la planeación de sprints para lograr la realización de las características de la pila de producto. Saber hacer: - Determinar el paradigma ágil más conveniente para la realización de un proyecto. - Valorar las diferencias y similitudes de las diferentes opciones de desarrollo ágil. Ser: - Realiza las actividades de desarrollo de	- Investigación de la información conceptual en fuentes diversas. - Análisis de casos. - Proyecto. - Explicación de temas con apoyo de recursos didácticos.	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 distribución y promoción del producto o servicio orientado al segmento de mercado considerado en su proyecto de asignatura.			



Continuación: Tabla 4.2. Desglose específico de la unidad "Operación de proyectos en grupos pequeños."

Temas y subtemas (secuencia)	Criterios de desempeño	Estrategias didácticas	Estrategias de evaluación	Producto Integrador de la unidad
	software de manera ágil, con sustento metodológico y con herramientas adecuadas para el trabajo colaborativo en grupos pequeños.			
Bibliografía				
<p>- Project Management Institute. (2021). Guía de los fundamentos para la dirección de proyectos: Guía del Pmbok. 7ª edición. USA: Project Management Institute.</p> <p>- Alaimo, M.; Salias, M. (2015). Proyectos Ágiles con Scrum: Flexibilidad, aprendizaje, innovación y colaboración en contextos complejos (Spanish Edition). 2ª edición. México: Kleer.</p>				

Sección IV. Desglose específico por cada unidad formativa

Tabla 4.3. Desglose específico de la unidad "Buenas prácticas y herramientas de organización y seguimiento de los avances de desarrollo de software en un pr"

Número y nombre de la unidad: 3. Buenas prácticas y herramientas de organización y seguimiento de los avances de desarrollo de software en un pr							
Tiempo y porcentaje para esta unidad:		Teoría:	12 horas	Práctica:	12 horas	Porcentaje del programa:	33.33%
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)			
<p>3.1 Desarrollo de habilidades fisiomotoras específicas en el uso adecuado de teclados para desarrollo de software.</p> <p>3.2 Riesgos en la inadecuada escritura y organización del código fuente.</p> <p>3.3 Buenas prácticas "clean code" para la escritura de programas fuente.</p> <p>3.4 Revisión de código en pares "code review"</p> <p>3.5 Herramientas de gestión de las versiones y ramificaciones del código fuente (herramientas Git).</p>	<p>Saber:</p> <ul style="list-style-type: none"> - Conocer herramientas Git para gestionar los avances de desarrollo de software en pequeños grupos. <p>Saber hacer:</p> <ul style="list-style-type: none"> - Realizar la escritura y organización de código fuente de manera adecuada para garantizar el seguimiento, la claridad y la identificación de elementos - Utilizar herramientas Git para gestionar los avances de desarrollo de software en pequeños grupos. 	<ul style="list-style-type: none"> - Investigación de la información conceptual en fuentes diversas. - Análisis de casos. - Proyecto. - Explicación de temas con apoyo de recursos didácticos. - Prácticas de laboratorio orientadas a gestionar sus códigos. 	<p>Evaluación formativa:</p> <ul style="list-style-type: none"> - Elaboración de estrategias de organización y respaldo de archivos fuentes. - Prácticas de laboratorio. <p>Evaluación sumativa:</p> <ul style="list-style-type: none"> - Trabajo integrador en equipo para iniciar con su proyecto final. 	<ul style="list-style-type: none"> - Respaldo y organización de un pequeño proyecto de desarrollo de software, realizado en la asignatura de programación. 			



Continuación: Tabla 4.3. Desglose específico de la unidad "Buenas prácticas y herramientas de organización y seguimiento de los avances de desarrollo de software en un pr"

Temas y subtemas (secuencia)	Criterios de desempeño	Estrategias didácticas	Estrategias de evaluación	Producto Integrador de la unidad
	Ser: - Realiza las actividades de desarrollo de software de manera ágil, con sustento metodológico y con herramientas adecuadas para el trabajo colaborativo en grupos pequeños.			
Bibliografía				
- Lewrick, M.; Link, P.; Leifer, L. (2020). The Design Thinking Toolbox: A Guide to Mastering the Most Popular and Valuable Innovation Methods. USA: editorial Wiley.				



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. o carrera afín</p> <p>- Experiencia en administración de proyectos.</p> <p>- Experiencia mínima de dos años.</p> <p>- Ingeniero titulado o superior.</p>