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:**	Desarrollo de software seguro	
Academia:	Desarrollo de software /		**Clave:**	19SDSSI02	
Módulo formativo:	Internet de las cosas		**Seriación:**	- -	
Tipo de curso:	Presencial		**Prerrequisito:**	- -	
Semestre:	Séptimo	**Créditos:** 6.75	**Horas semestre:**	108 horas	
Teoría:	2 horas	**Práctica:** 2 horas	**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.
OE3	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.
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
AE2	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.	- Gestionarán recursos relacionados con el desarrollo de software en alguna organización.	1.1 Visión panorámica de las vulnerabilidades y sus costes. 1.2 Propiedades del software seguro y resiliente. 1.3 Errores de programación más peligrosos según el CWE/SANS Top 25. 1.4 Conceptos de seguridad.
AE3	Desarrollar una experimentación adecuada para recopilar, almacenar y analizar grandes cantidades de información basándose en el juicio ingenieril para crear productos o servicios innovadores mediados por software.	- Diseñarán e implementarán soluciones innovadoras mediante el uso de tecnologías de la información.	2.1 Seguridad y resiliencia a lo largo del ciclo de vida. 2.2 Puntos de ataque y seguridad perimetral. 2.3 Buenas prácticas según OWASP (Open Web Application Security Project). 3.1 Conceptos de diseño seguro. 3.2 Proceso de diseño. 3.3 Arquitectura. 3.4 Tecnologías.
AE5	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.	- Reconocerán sus responsabilidades éticas y profesionales para el desarrollo de software seguro.	4.1 Vulnerabilidades y controles comunes de software. 4.2 Prácticas de código defensivo. 4.3 Proceso de software seguro.

Sección III. Atributos de la asignatura

Tabla 3. Atributos de la asignatura

Problema a resolver		
Conocer buenas prácticas para la implementación de software seguro.		
Atributos (competencia específica) de la asignatura		
Codificar un software seguro aplicando buenas prácticas de desarrollo con base a un plan estratégico.		
Aportación a la competencia específica		**Aportación a las competencias transversales**
Saber	Saber hacer	Saber Ser
- Identificar las vulnerabilidades de un sistema, utilizando herramientas de auditoría informática, para implementar procesos de desarrollo de software seguro.	- Proteger los sistemas de las vulnerabilidades identificadas, utilizar para ello herramientas de auditoría informática, e implementar procesos de desarrollo de software seguro.	- Habilidades interpersonales. - Apreciación de la diversidad. - Compromiso ético - Habilidad para trabajar de forma autodidacta, de forma individual y por equipo.
Producto integrador de la asignatura, considerando los avances por unidad		
Propuesta de desarrollo de software aplicando metodología de software seguro.		

Sección IV. Desglose específico por cada unidad formativa

Tabla 4.1. Desglose específico de la unidad "Introducción."

Número y nombre de la unidad:	1. Introducción.					
Tiempo y porcentaje para esta unidad:	Teoría:	9 horas	Práctica:	9 horas	Porcentaje del programa:	25%
Aprendizajes esperados:	Identificar las buenas prácticas en temas de seguridad y su importancia dentro de los procesos de desarrollo de software y su ciclo de vida.					

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 Visión panorámica de las vulnerabilidades y sus costes. 1.2 Propiedades del software seguro y resiliente. 1.3 Errores de programación más peligrosos según el CWE/SANS Top 25. 1.4 Conceptos de seguridad.	Saber: - Conocer las vulnerabilidades de software, conceptos de seguridad y errores más comunes. Saber hacer: - Analizar y considerar las vulnerabilidades de software, conceptos de seguridad y errores más comunes. Ser: - Habilidades interpersonales. - Apreciación de la diversidad. - Compromiso ético - Habilidad para trabajar de forma autodidacta, de forma individual y por equipo.	- Preguntas intercaladas para identificar conocimiento previo. - Presentación de material teórico a través de diversos medios (diapositivas, proyector, videoconferencia, computadora, internet) - Tareas de investigación.	Evaluación diagnóstica: - Identificar conocimientos previos. Evaluación formativa: - Mapa mental, mapa conceptual, resumen. Evaluación sumativa: - Examen.	Portafolio de evidencias: - Actividades realizadas en clase.

Bibliografía

- Project Management Ins|titute. (2017). A Guide to the Project Management Body of Knowledge (PMBOK® Guide). 6a edition. USA: Project Management Institute.

Continuación: Tabla 4.1. Desglose específico de la unidad "Introducción."
Bibliografía
- Normas ISO. (s. f.). ISO 27001 - Seguridad de la información: norma ISO IEC 27001/27002. Recuperado octubre de 2022, de https://www.normas-iso.com/iso-27001 - OWASP Foundation, the Open Source Foundation for Application Security | OWASP Foundation. (s. f.). Recuperado octubre de 2022, dehttps://owasp.org - Invicti. (s. f.). Invicti | Web Application Security For Enterprise. Recuperado octubre de 2022, de https://www.netsparker.com - Offensive Security´s Exploit Database Archive. (s. f.). Recuperado octubre de 2022, de https://www.exploit-db.com

Sección IV. Desglose específico por cada unidad formativa

Tabla 4.2. Desglose específico de la unidad "Plan estratégico."

Número y nombre de la unidad:	2. Plan estratégico.					
Tiempo y porcentaje para esta unidad:	Teoría:	9 horas	Práctica:	9 horas	Porcentaje del programa:	25%
Aprendizajes esperados:	Desarrollar un plan estratégico que permita desarrollar software seguro y resiliente aplicando buenas prácticas de OWASP.					

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 Seguridad y resiliencia a lo largo del ciclo de vida. 2.2 Puntos de ataque y seguridad perimetral. 2.3 Buenas prácticas según OWASP (Open Web Application Security Project).	Saber: - Reconocer actividades que atiendan las buenas prácticas en seguridad en el desarrollo de software y su ciclo de vida. Saber hacer: - Implementar actividades que atiendan las buenas prácticas en seguridad en el desarrollo de software y su ciclo de vida. Ser: - Habilidades interpersonales. - Apreciación de la diversidad. - Compromiso ético - Habilidad para trabajar de forma autodidacta, de forma individual y por	- Presentación de material teórico a través de diversos medios (diapositivas, proyector, videoconferencia, computadora, internet) - Tareas de investigación. - Prácticas.	Evaluación formativa: - Mapa mental, mapa conceptual, resumen, prácticas de laboratorio. Evaluación sumativa: - Examen.	Desarrollo de la documentación y registros necesarios para la implementación de actividades de verificación de la seguridad en el proceso de desarrollo de software.

Continuación: Tabla 4.2. Desglose específico de la unidad "Plan estratégico."				
Temas y subtemas (secuencia)	Criterios de desempeño	Estrategias didácticas	Estrategias de evaluación	Producto Integrador de la unidad
	equipo.			
Bibliografía				
- Normas ISO. (s. f.). ISO 27001 - Seguridad de la información: norma ISO IEC 27001/27002. Recuperado octubre de 2022, de https://www.normas-iso.com/iso-27001 - OWASP Foundation, the Open Source Foundation for Application Security | OWASP Foundation. (s. f.). Recuperado octubre de 2022, de https://owasp.org				

Sección IV. Desglose específico por cada unidad formativa

Tabla 4.3. Desglose específico de la unidad "Buenas prácticas de desarrollo."

Número y nombre de la unidad:	3. Buenas prácticas de desarrollo.					
Tiempo y porcentaje para esta unidad:	Teoría:	9 horas	Práctica:	9 horas	Porcentaje del programa:	25%
Aprendizajes esperados:	Conocer e implementar buenas prácticas de desarrollo de software seguro desde sus etapas iniciales del diseño del mismo.					

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 Conceptos de diseño seguro. 3.2 Proceso de diseño. 3.3 Arquitectura. 3.4 Tecnologías.	Saber: - Identificar las buenas prácticas relacionadas con la seguridad desde las etapas iniciales del diseño de software. Saber hacer: - Implementar las buenas prácticas relacionadas con la seguridad desde las etapas iniciales del diseño de software. Ser: - Habilidades interpersonales. - Apreciación de la diversidad. - Compromiso ético - Habilidad para trabajar de forma autodidacta, de forma individual y por	- Presentación de material teórico a través de diversos medios (diapositivas, proyector, videoconferencia, computadora, internet). - Tareas de investigación. - Prácticas.	Evaluación formativa: - Mapa mental, mapa conceptual, resumen, prácticas de laboratorio. Evaluación sumativa: - Examen.	Desarrollo de una aplicación orientada a la atención de las principales vulnerabilidades de seguridad, con su propuesta de código defensivo para su implementación en procesos de desarrollo de software.

Continuación: Tabla 4.3. Desglose específico de la unidad "Buenas prácticas de desarrollo."				
Temas y subtemas (secuencia)	Criterios de desempeño	Estrategias didácticas	Estrategias de evaluación	Producto Integrador de la unidad
	equipo.			
Bibliografía				
- OWASP Foundation, the Open Source Foundation for Application Security | OWASP Foundation. (s. f.). Recuperado octubre de 2022, de https://owasp.org - Vega Vulnerability Scanner. (s. f.). Recuperado octubre de 2022, de https://subgraph.com/vega/index.en.html				

Sección IV. Desglose específico por cada unidad formativa

Tabla 4.4. Desglose específico de la unidad "Implementación/Codificación Software Seguro."

Número y nombre de la unidad:	4. Implementación/Codificación Software Seguro.					
Tiempo y porcentaje para esta unidad:	Teoría:	9 horas	Práctica:	9 horas	Porcentaje del programa:	25%
Aprendizajes esperados:	Conocer las principales vulnerabilidades de seguridad en la codificación para su prevención y programación de software seguro.					

Temas y subtemas (secuencia)	**Criterios de desempeño**	**Estrategias didácticas**	**Estrategias de evaluación**	**Producto Integrador de la unidad (Evidencia de aprendizaje de la unidad)**
4.1 Vulnerabilidades y controles comunes de software. 4.2 Prácticas de código defensivo. 4.3 Proceso de software seguro.	Saber: - Conocer los principales errores y vulnerabilidades que se presentan en el desarrollo de software en su etapa de codificación. Saber hacer: - Identificar los principales errores y vulnerabilidades que se presentan en el desarrollo de software en su etapa de codificación. Ser: - Habilidades interpersonales. - Apreciación de la diversidad. - Compromiso ético - Habilidad para trabajar de forma	- Presentación de material teórico a través de diversos medios (diapositivas,proyector, videoconferencia, computadora, internet) - Tareas de investigación. - Prácticas.	Evaluación formativa: - Mapa mental, mapa conceptual, resumen, prácticas de laboratorio. Evaluación sumativa: - Examen.	Desarrollo de una aplicación orientada a la atención de las principales vulnerabilidades de seguridad, con su propuesta de código defensivo para su implementación en procesos de desarrollo de software.

Continuación: Tabla 4.4. Desglose específico de la unidad "Implementación/Codificación Software Seguro."				
Temas y subtemas (secuencia)	Criterios de desempeño	Estrategias didácticas	Estrategias de evaluación	Producto Integrador de la unidad
	autodidacta, de forma individual y por equipo.			
Bibliografía				
- OWASP Foundation, the Open Source Foundation for Application Security | OWASP Foundation. (s. f.). Recuperado octubre de 2022, de https://owasp.org - Vega Vulnerability Scanner. (s. f.). Recuperado octubre de 2022, de https://subgraph.com/vega/index.en.html				

V. Perfil docente

Tabla 5. Descripción del perfil docente

Perfil deseable docente para impartir la asignatura
Carrera(s): - Ingeniería en Computación. - Licenciatura en Informática. - Licenciatura en Sistemas de Información o carreras afines. o carrera afín - Ingeniero en Computación, Licenciado en Informática, Licenciado en Sistemas de Información o carreras afines. - Experiencia mínima de dos años - Ingeniero en Computación, Licenciado en Informática, Licenciado en Sistemas de Información o carreras afines.