Programa de asignatura
por competencias de educación superior

Sección I. Identificación del Curso

Tabla 1. Identificación de la Planificación del Curso.

Actualización:	Marzo 13, 2024				
Carrera:	Ingeniería en Desarrollo de Software			**Asignatura:**	Ingeniería inversa
Academia:	Desarrollo de software /			**Clave:**	19SDSSI04
Módulo formativo:	Gestión y operación de proyectos de TI			**Seriación:**	- -
Tipo de curso:	Presencial			**Prerrequisito:**	19SDSSI01 - Hacking ético
Semestre:	Octavo	**Créditos:**	6.75	**Horas semestre:**	108 horas
Teoría:	2 horas	**Práctica:**	2 horas	**Trabajo indpt.:**	2 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 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.	- Comprenderá los fundamentos teóricos, herramientas y técnicas de la ingeniería inversa, así como sus aplicaciones en el desarrollo de software.	1.1 Definición y conceptos básicos sobre ingeniería inversa. 1.2 Ingeniería inversa de software. 1.3 Aplicaciones en la industria. 1.4 Software de bajo nivel. 1.4.1 Introducción a ensamblador. 1.4.2 Análisis de Código Binario. 1.5 Ética y legalidad en ingeniería inversa. 1.5.1Copyright. 1.5.2 Licenciamientos. 1.5.3 Leyes y normatividad.
2	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.	- Aplicará diversas herramientas y técnicas de ingeniería inversa para analizar y comprender el funcionamiento interno del software, manteniendo una postura ética y responsable en su uso, y desarrollando habilidades prácticas para identificar vulnerabilidades y establecer entornos de análisis seguros.	2.1 Análisis estático y dinámico. 2.2 Desensambladores. 2.3 Depuradores. 2.4 Descompiladores. 2.5 Reconstrucción de estructuras de datos y algoritmos. 2.6 Ingeniería Inversa de Malware y Aplicaciones. 2.6.1 Montando un laboratorio de análisis. 2.6.2 Ingeniería inversa en aplicaciones ejecutables y empaquetadas. 2.6.3 Identificación y explotación de vulnerabilidades.
3		- Comprenderá los fundamentos de la criptografía y su aplicación en sistemas de seguridad, habilidades prácticas para implementar y gestionar herramientas criptográficas, y una actitud ética y responsable al trabajar con información sensible protegida por estos sistemas.	3.1 Conceptos base de criptografía. 3.2 Sistemas criptográficos. 3.2.1 KMS. 3.2.2 HSM. 3.2.3 PKI.

Continuación: Tabla 2. Objetivos educacionales (continuación			
No.	Atributos de egreso de plan de estudios	Criterios de desempeño	Componentes
	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.		3.2.4 TPM. 3.2.5 Certificados digitales. 3.2.6 Firmas digitales. 3.2.7 Protocolos: SSL, TLS, ECDH. 3.2.8 Sistemas de identificación digital. 3.3 Algoritmos criptográficos modernos. 3.4 Liberarías de encriptación. 3.5 Criptografía en bases de datos. 3.6 Herramienta: Cryptex. 4.1 Técnicas antireversing. 4.1.1 Ofuscación de código. 4.1.2 Encriptación del código. 4.1.3 IsDebuggerPresent API. 4.1.4 Checksums. 4.2 Parchado de aplicaciones. 4.3 Detección y evasión de técnicas de análisis dinámico. 4.4 Protección de la información en memoria. 4.5 Integridad del código y firmado digital. 4.6 Monitoreo y detección de intrusiones.

Sección III. Atributos de la asignatura

Tabla 3. Atributos de la asignatura

Problema a resolver		
Reconstruir el código de un programa existente para de esta forma, se pueda optimizar, solventar errores, analizar los programas de la competencia y desarrollar nuevos productos, respetando la propiedad intelectual y las regulaciones legales vigentes.		
Atributos (competencia específica) de la asignatura		
Comprender y aplicar las diversas herramientas, conceptos y técnicas de ingeniería inversa para analizar y comprender el funcionamiento interno del software, respetando la propiedad intelectual y las regulaciones legales vigentes.		
Aportación a la competencia específica		**Aportación a las competencias transversales**
Saber	Saber hacer	Saber Ser
- Comprender los conceptos fundamentales y las técnicas de la ingeniería inversa para analizar y aplicar herramientas éticas en el análisis de sistemas y programas.	- Aplicar diversas herramientas y técnicas de ingeniería inversa para analizar y comprender el funcionamiento interno del software	Mantendrá una postura ética y responsable en el uso, y desarrollo de habilidades prácticas para identificar vulnerabilidades y establecer entornos de análisis seguros.
Producto integrador de la asignatura, considerando los avances por unidad		
Portafolio de evidencias que contenga la aplicación de técnicas de ingeniería inversa sobre software de código abierto populares, como navegadores, sistemas operativos para escritorio o embebidos.		

Sección IV. Desglose específico por cada unidad formativa

Tabla 4.1. Desglose específico de la unidad "Fundamentos de ingeniería inversa."

Número y nombre de la unidad:	1. Fundamentos de ingeniería inversa.					
Tiempo y porcentaje para esta unidad:	Teoría:	12 horas	Práctica:	8 horas	Porcentaje del programa:	27.78%
Aprendizajes esperados:	Comprender los conceptos fundamentales y las técnicas de la ingeniería inversa para analizar y aplicar herramientas éticas en el análisis de sistemas y programas, respetando la propiedad intelectual y las regulaciones legales vigentes.					

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 Definición y conceptos básicos sobre ingeniería inversa. 1.2 Ingeniería inversa de software. 1.3 Aplicaciones en la industria. 1.4 Software de bajo nivel. 1.4.1 Introducción a ensamblador. 1.4.2 Análisis de Código Binario. 1.5 Ética y legalidad en ingeniería inversa. 1.5.1Copyright. 1.5.2 Licenciamientos. 1.5.3 Leyes y normatividad.	Saber: - Comprender los fundamentos teóricos, herramientas y técnicas de la ingeniería inversa, así como sus aplicaciones en el desarrollo de software. Saber hacer: - Ser capaz de aplicar métodos de ingeniería inversa para analizar sistemas y programas, identificar oportunidades de aplicación y realizar análisis de código binario de manera ética y precisa.	-Investigación. -Resúmenes. -Mapas conceptuales. -Exposición por el docente. -Exposición por parte del alumno. -Ejercicios.	Evaluación diagnóstica: - Rescatar conocimiento previo. Evaluación formativa: -Retroalimentación de trabajos. -Autoevaluación. -Coevaluación. -Evaluaciones en base a TIC´s.: formativas con retroalimentación y automáticas. Evaluación sumativa: -Prueba escrita. -Pruebas orales. -Evaluación mediante procesos de investigación. -Evaluación basada en proyectos.	-Portafolio de evidencias con los trabajos realizados durante la unidad.

Continuación: Tabla 4.1. Desglose específico de la unidad "Fundamentos de ingeniería inversa."				
Temas y subtemas (secuencia)	Criterios de desempeño	Estrategias didácticas	Estrategias de evaluación	Producto Integrador de la unidad
	Ser: - Desarrollar una actitud reflexiva y ética hacia la ingeniería inversa, manteniendo la integridad, respetando los derechos de propiedad intelectual, y actuando con responsabilidad y transparencia en el manejo de la información y el cumplimiento de las normativas legales.			

Bibliografía

- Eilam, E. (2011). Reversing: Secrets of Reverse Engineering. Willey.
- Sikorski, M.; Honig, A. (2012). Practical Malware Analysis. No Starch Press.
- Dang, B.; Gazet, A.; Bachaalany, E. (2014). Practical Reverse Engineering: x86, x64, ARM, Windows Kernel, Reversing Tools, and Obfuscation. Wiley.

Sección IV. Desglose específico por cada unidad formativa

Tabla 4.2. Desglose específico de la unidad "Herramientas / taller para la ingeniería inversa."

Número y nombre de la unidad:	2. Herramientas / taller para la ingeniería inversa.					
Tiempo y porcentaje para esta unidad:	Teoría:	12 horas	Práctica:	6 horas	Porcentaje del programa:	25%
Aprendizajes esperados:	Aplicar diversas herramientas y técnicas de ingeniería inversa para analizar y comprender el funcionamiento interno del software, manteniendo una postura ética y responsable en su uso, y desarrollando habilidades prácticas para identificar vulnerabilidades y establecer entornos de análisis seguros.					

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 Análisis estático y dinámico. 2.2 Desensambladores. 2.3 Depuradores. 2.4 Descompiladores. 2.5 Reconstrucción de estructuras de datos y algoritmos. 2.6 Ingeniería Inversa de Malware y Aplicaciones. 2.6.1 Montando un laboratorio de análisis. 2.6.2 Ingeniería inversa en aplicaciones ejecutables y empaquetadas. 2.6.3 Identificación y explotación de vulnerabilidades.	Saber: - Comprender las herramientas, técnicas y métodos utilizados en la ingeniería inversa, incluyendo desensambladores, análisis estático/dinámico, reconstrucción de estructuras, ingeniería inversa de malware y aplicaciones, montaje de laboratorios de análisis y la identificación de vulnerabilidades en software. Saber hacer: - Aplicar activamente las técnicas y herramientas mencionadas para analizar y comprender el funcionamiento interno	-Investigación. -Resúmenes. -Mapas conceptuales. -Exposición por el docente. -Exposición por parte del alumno. -Ejercicios.	Evaluación formativa: -Retroalimentación de trabajos. -Autoevaluación. -Coevaluación. -Evaluaciones en base a TIC´s.: formativas con retroalimentación y automáticas. Evaluación sumativa: -Prueba escrita. -Pruebas orales. -Evaluación mediante procesos de investigación. -Evaluación basada en proyectos.	-Portafolio de evidencias con los trabajos realizados durante la unidad.

Continuación: Tabla 4.2. Desglose específico de la unidad "Herramientas / taller para la ingeniería inversa."				
Temas y subtemas (secuencia)	Criterios de desempeño	Estrategias didácticas	Estrategias de evaluación	Producto Integrador de la unidad
	de software, reconstruir estructuras de datos y algoritmos, identificar vulnerabilidades y establecer entornos de laboratorio para el análisis de software. Ser: - Reconocer la importancia ética de proteger la propiedad intelectual y la confidencialidad al implementar estrategias de ingeniería inversa, manteniendo integridad al usar herramientas de análisis y respetando las leyes y regulaciones relacionadas.			

Bibliografía

- Eilam, E. (2011). Reversing: Secrets of Reverse Engineering. Willey.
- Sikorski, M.; Honig, A. (2012). Practical Malware Analysis. No Starch Press.
- Dang, B.; Gazet, A.; Bachaalany, E. (2014). Practical Reverse Engineering: x86, x64, ARM, Windows Kernel, Reversing Tools, and Obfuscation. Wiley.
- Eagle, C. (2011). The IDA Pro Book: The Unofficial Guide to the World's Most Popular Disassembler. No Starch Press.
- Ligh, M.; Adair, S.; Hartstein, B.; Richard, M. (2012). Malware Analyst's Cookbook and DVD: Tools and Techniques for Fighting Malicious Code. Wiley

Sección IV. Desglose específico por cada unidad formativa

Tabla 4.3. Desglose específico de la unidad "Ingeniería inversa a sist Iemas criptográficos."

Número y nombre de la unidad:	3. Ingeniería inversa a sist Iemas criptográficos.					
Tiempo y porcentaje para esta unidad:	Teoría:	12 horas	Práctica:	6 horas	Porcentaje del programa:	25%
Aprendizajes esperados:	Comprender los fundamentos de la criptografía y su aplicación en sistemas de seguridad, habilidades prácticas para implementar y gestionar herramientas criptográficas, una actitud ética y responsable al trabajar con información sensible protegida por estos sistemas.					

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 base de criptografía. 3.2 Sistemas criptográficos. 3.2.1 KMS. 3.2.2 HSM. 3.2.3 PKI. 3.2.4 TPM. 3.2.5 Certificados digitales. 3.2.6 Firmas digitales. 3.2.7 Protocolos: SSL, TLS, ECDH. 3.2.8 Sistemas de identificación digital. 3.3 Algoritmos criptográficos modernos. 3.4 Liberarías de encriptación. 3.5 Criptografía en bases de datos. 3.6 Herramienta: Cryptex.	Saber: - Comprender los fundamentos teóricos de la criptografía, sistemas criptográficos, algoritmos modernos y protocolos de seguridad. Saber hacer: - Implementar sistemas de seguridad como certificados digitales, firmas digitales, y herramientas como Cryptex, además de aplicar algoritmos criptográficos modernos en sistemas de identificación digital y bases de datos.	-Investigación. -Resúmenes. -Mapas conceptuales. -Exposición por el docente. -Exposición por parte del alumno. -Ejercicios.	Evaluación formativa: -Retroalimentación de trabajos. -Autoevaluación. -Coevaluación. -Evaluaciones en base a TIC´s.: formativas con retroalimentación y automáticas. Evaluación sumativa: -Prueba escrita. -Pruebas orales. -Evaluación mediante procesos de investigación. -Evaluación basada en proyectos.	-Portafolio de evidencias con los trabajos realizados durante la unidad.

Continuación: Tabla 4.3. Desglose específico de la unidad "Ingeniería inversa a sist lemas criptográficos."				
Temas y subtemas (secuencia)	Criterios de desempeño	Estrategias didácticas	Estrategias de evaluación	Producto Integrador de la unidad
	Ser: - Mantener una actitud ética y responsable en el manejo de información protegida por sistemas criptográficos, respetando la privacidad, seguridad y confidencialidad de los datos, y actuando con integridad al aplicar conceptos de seguridad digital.			

Bibliografía

- Eilam, E. (2011). Reversing: Secrets of Reverse Engineering. Willey.
- Sikorski, M.; Honig, A. (2012). Practical Malware Analysis. No Starch Press.
- Schneier, B. (2015). Applied Cryptography: Protocols, Algorithms, and Source Code in C. Wiley.
- Stallings, W. (2020). Cryptography and Network Security: Principles and Practice. Pearson.

Sección IV. Desglose específico por cada unidad formativa

Tabla 4.4. Desglose específico de la unidad " Buenas prácticas para prevención de ingeniería inversa."

Número y nombre de la unidad:	4. Buenas prácticas para prevención de ingeniería inversa.					
Tiempo y porcentaje para esta unidad:	Teoría:	12 horas	Práctica:	4 horas	Porcentaje del programa:	22.22%
Aprendizajes esperados:	Ser capaces de aplicar estrategias avanzadas de protección en el desarrollo de software para prevenir la ingeniería inversa, manteniendo una postura ética y responsable en el manejo de técnicas y herramientas de seguridad digital.					

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 Técnicas antireversing 4.1.1 Ofuscación de código 4.1.2 Encriptación del código 4.1.3 IsDebuggerPresent API 4.1.4 Checksums 4.2 Parchado de aplicaciones 4.3 Detección y evasión de técnicas de análisis dinámico 4.4 Protección de la información en memoria 4.5 Integridad del código y firmado digital. 4.6 Monitoreo y detección de intrusiones.	Saber: - Comprender estrategias y herramientas para prevenir la ingeniería inversa en el desarrollo de software, incluyendo técnicas antireversing, ofuscación, encriptación, herramientas como IsDebuggerPresent y checksums, además de conceptos como detección de análisis dinámico, protección de datos en memoria, integridad del código, diseño seguro de sistemas y monitoreo de intrusiones. Saber hacer: - Aplicar técnicas y herramientas	-Investigación. -Resúmenes. -Mapas conceptuales. -Exposición por el docente. -Exposición por parte del alumno. -Ejercicios.	Evaluación formativa: -Retroalimentación de trabajos. -Autoevaluación. -Coevaluación. -Evaluaciones en base a TIC´s.: formativas con retroalimentación y automáticas. Evaluación sumativa: -Prueba escrita. -Pruebas orales. -Evaluación mediante procesos de investigación. -Evaluación basada en proyectos.	-Portafolio de evidencias con los trabajos realizados durante la unidad.

Continuación: Tabla 4.4. Desglose específico de la unidad " Buenas prácticas para prevención de ingeniería inversa."				
Temas y subtemas (secuencia)	Criterios de desempeño	Estrategias didácticas	Estrategias de evaluación	Producto Integrador de la unidad
	mencionadas para diseñar sistemas resistentes a la ingeniería inversa, implementar encriptación, firmado digital, detección de técnicas de análisis dinámico, monitoreo de intrusiones, realizar parches en aplicaciones y proteger información en memoria durante la ejecución del software. Ser: - Reconocer la importancia ética de proteger la propiedad intelectual, la confidencialidad y la integridad del software al aplicar estas estrategias, evitando su uso indebido o ilegal.			

Bibliografía

- Seitz, J. (2009). Gray Hat Python: Python Programming for Hackers and Reverse Engineers. No Starch Press.
- Sun, L.; Tian, Y.; Wang, Y.; Zhu, T. (2019). Deep reinforcement learning for automated vulnerability discovery in binary code. IEEE Transactions on Cybernetics, 49(5), 1943-1955.
- Cyber Grand Challenge (2019-2023). DARPA. https://www.darpa.mil/program/cyber-grand-challenge.
- Project PASTA. (2020-present). Google AI. https://www.youtube.com/watch?v=CoPlWxkKS-0.
- National Science Foundation (NSF) Cybersecurity program. https://www.nsf.gov/news/special_reports/cybersecurity/index.jsp.
- Zhang, Y.;Yan, J.; Xiao, X.; Zhou, Y. (2021). Towards automated reverse engineering of obfuscated Android applications. IEEE Transactions on Dependable and Secure Computing, 19(1), 231-244.
- Zhang, H.; Sun, L.; Su, Z. (2023). Deep learning for symbolic execution: A survey and new perspectives. arXiv preprint arXiv:2301.07773.

V. Perfil docente

Tabla 5. Descripción del perfil docente

Perfil deseable docente para impartir la asignatura
Carrera(s): Ingeniería en Desarrollo de Software, Ingeniería en Computación o carrera afín. o carrera afín - Experiencia profesional relacionada con la materia de probabilidad enfocada a la Inteligencia Artificial. Experiencia docente mínima de dos años. - Experiencia mínima de dos años - Grado académico, mínimo Maestría relacionada con el área de conocimiento.