1. Datos Generales de la asignatura

Nombre de la asignatura:	Ciclos de Vida de Software Seguro
Clave de la asignatura:	
SATCA:	3-2-5
Carrera:	Ingeniería en Sistemas Computacionales e Ingeniería en Tecnologías de la Información y Comunicaciones

2. Presentación

Caracterización de la asignatura

Esta asignatura pretende proveer conceptos, estándares y metodologías a las personas que participan en el ciclo de vida de desarrollo del software sobre la proliferación de vulnerabilidades de seguridad derivadas de procesos de desarrollo insuficientes estableciendo mejores prácticas y validando la competencia de un individuo en relación con la solución de problemas de seguridad a lo largo del ciclo de vida del software. Las amenazas emergentes en la actualidad incluyen varios riesgos de seguridad que se aprovechan de las fallas y limitaciones del código en muchos productos y servicios de tecnología que se han vuelto indispensables para individuos y negocios en la vida cotidiana. Los grupos de crimen organizado han agudizado su enfoque y han incrementado la frecuencia de sus ataques contra la capa de aplicación del modelo OSI, haciendo de la seguridad en las aplicaciones de software una prioridad principal para proteger información susceptible. Los temas que abarcan son; Conceptos de Software Seguro, Requerimientos de Software Seguro, Diseño de Software Seguro, Implementación/Codificación de Software Seguro. Los objetivos específicos que se pretenden cubrir son: • Comprender los conceptos y elementos que constituyen el software seguro. • Estar familiarizado con los principios de administración del riesgo y como éste concierne al desarrollo de software. • Conocer cómo se aplican los conceptos de seguridad de la información en el desarrollo de software. • Conocer diversos aspectos de diseño que necesitan ser tomados en consideración para diseñar software resistente a hackeo.
Intención didáctica
Por lo dicho anteriormente, el docente debe guiar el proceso de desarrollo emparejándolo con las unidades de aprendizaje sin perder de vista que cada equipo puede requerir un soporte distinto, es decir, que el docente maneje de forma dinámica el temario para que sea capaz de adaptarlo al contexto de cada equipo de trabajo.

3. Participantes en el diseño y seguimiento curricular del programa

Lugar y fecha de elaboración o revisión	Participantes	Observaciones

Instituto Tecnológico de Morelia. *Morelia, Michoacán, Agosto de 2018.*	*M.C. Abel Alberto Pintor Estrada* *M.A. Laura Nelly Alvarado Zamora*	*Definición de los programas de estudio para la especialidad.*

4. Competencia(s) a desarrollar

Competencia general de la asignatura
1. Presentar conceptos, normativas y estándares que cubran personas, procesos y elementos de tecnología involucrados en el desarrollo de software seguro a lo largo del ciclo de vida de un proyecto de desarrollo de software, comenzando con el análisis de requerimientos, el diseño y codificación-implementación del software.
Competencias especificas • Comprender, analizar y aplicar los principios, tendencias y técnicas para el desarrollo de software • Conocer las fases del ciclo de vida del desarrollo del software e identificar amenazas, vulnerabilidades y controles con el objetivo de desarrollar el software seguro.
Competencias genéricas **Competencias instrumentales:** • Utilizar de forma apropiada teorías, procedimientos y herramientas en el desarrollo profesional de la ingeniería informática en todos sus ámbitos (especificación, diseño, implementación, despliegue -implantación- y evaluación de productos) de manera que se demuestre la comprensión de los compromisos adoptados en las decisiones de diseño. **Competencias interpersonales:** • Capacidad de razonamiento crítico, lógico y matemático. Capacidad para resolver problemas dentro de su área de estudio. Capacidad de abstracción: capacidad de crear y utilizar modelos que reflejen situaciones reales. Capacidad de diseñar y realizar experimentos sencillos, y analizar e interpretar sus resultados. Capacidad de análisis, síntesis y evaluación. **Competencias sistémicas:** • Desarrollar, mantener y evaluar servicios y sistemas software que satisfagan todos los requisitos del usuario, que se comporten de forma fiable y eficiente, que tengan un desarrollo y mantenimiento asequible y que cumplan normas de desarrollo de software seguro.

5. Competencias previas

1. Conocimiento y aplicación de los principios, metodologías y ciclos de vida de la ingeniería de software.
2. Fundamentos de Programación, Programación Orientada a Objetos, Estructuras de Datos y Tópicos de Programación Avanzada.

6. Temario

No.	Temas	Subtemas
1	Conceptos de Software Seguro	1.1 Seguridad Holística 1.2 Conceptos de seguridad 1.3 Conceptos de diseño seguro 1.4 Administración de riesgos 1.5 Normativa aplicable
2	Requerimientos de Software Seguro	2.1 Recursos para requerimientos de seguridad 2.2 Separación de políticas 2.3 Clasificación de datos 2.4 Matriz de objetos/sujetos 2.5 Requirements of Traceability Matrix
3	Diseño de Software Seguro	3.1 Conceptos de diseño seguro 3.2 Proceso de diseño 3.3 Arquitectura 3.4 Tecnologías
4	Implementación/Codificación de Software Seguro	4.1 Vulnerabilidades y controles comunes de software 4.2 Prácticas de código defensivo 4.3 Proceso de software seguro

7. Actividades de aprendizaje de los temas

Unidad 1: Conceptos de Software Seguro	
Competencias	Actividades de aprendizaje
Específica(s): Comprender los conceptos y elementos que constituyen el software seguro. Genéricas: • Capacidad de análisis y síntesis. • Capacidad de organizar y planificar. • Comunicación oral y escrita. • Habilidad para buscar y analizar información proveniente de fuentes diversas. • Solución de problemas. • Toma de decisiones. • Capacidad crítica y autocrítica. • Capacidad de trabajar en equipo. • Capacidad de comunicar sus ideas. • Capacidad de liderazgo. • Capacidad de aplicar los conocimientos en la práctica. • Habilidades de investigación. • Capacidad de aprender. • Capacidad de adaptarse a nuevas situaciones. • Capacidad de generar nuevas ideas (creatividad). • Habilidad para trabajar en forma autónoma. • Preocupación por la calidad. • Búsqueda del logro.	Realizar una investigación sobre conceptos y elementos de software seguro, comentarlo en equipos y llegar a una conclusión. Realizar prácticas que permitan familiarizarse con los conceptos de software seguro y elaborar su correspondiente reporte. Realizar prácticas que permitan familiarizarse con la definición y obtención de requerimientos de seguridad y elaborar su correspondiente reporte. Realizar prácticas que permitan determinar y administrar los riesgos de seguridad de un proyecto de desarrollo de software y elaborar su correspondiente reporte. Partiendo de casos de estudio plantear soluciones e identificar problemas de seguridad en las diferentes fases del ciclo de vida de un proyecto de desarrollo de software.
Unidad 2: Requerimientos de Software Seguro	
Competencias	Actividades de aprendizaje

Específica(s): Diferenciar e Incorporar los tipos de requerimientos de seguridad en la obtención de requerimientos del software. Genéricas: • Capacidad de análisis y síntesis. • Capacidad de organizar y planificar. • Comunicación oral y escrita. • Habilidad para buscar y analizar información proveniente de fuentes diversas. • Solución de problemas. • Toma de decisiones. • Capacidad crítica y autocrítica. • Capacidad de trabajar en equipo. • Capacidad de comunicar sus ideas. • Capacidad de liderazgo. • Capacidad de aplicar los conocimientos en la práctica. • Habilidades de investigación. • Capacidad de aprender. • Capacidad de adaptarse a nuevas situaciones. • Capacidad de generar nuevas ideas (creatividad). • Habilidad para trabajar en forma autónoma. • Preocupación por la calidad. • Búsqueda del logro.	Realizar una investigación sobre requerimientos de seguridad, comentarlo en equipos y llegar a una conclusión. Partiendo de casos de estudio plantear soluciones e identificar problemas de obtención u omisión de requerimientos de seguridad de un proyecto de desarrollo de software. El alumno levanta requerimientos de seguridad de un proyecto integrador, enfocándose en la integridad, las confidencialidad, la disponibilidad, autenticación y autorización.
Unidad 3: Diseño de Software Seguro	
Competencias	Actividades de aprendizaje

Específica(s): Conocer las metodologías y mejores prácticas que emplea la industria del software para trasladar las especificaciones de seguridad del software durante la fase de codificación. Genéricas: • Capacidad de análisis y síntesis. • Capacidad de organizar y planificar. • Comunicación oral y escrita. • Habilidad para buscar y analizar información proveniente de fuentes diversas. • Solución de problemas. • Toma de decisiones. • Capacidad crítica y autocrítica. • Capacidad de trabajar en equipo. • Capacidad de comunicar sus ideas. • Capacidad de liderazgo. • Capacidad de aplicar los conocimientos en la práctica. • Habilidades de investigación. • Capacidad de aprender. • Capacidad de adaptarse a nuevas situaciones. • Capacidad de generar nuevas ideas (creatividad). • Habilidad para trabajar en forma autónoma. • Preocupación por la calidad. • Búsqueda del logro.	Realizar una investigación sobre el proceso de diseño, comentarlo en equipos y llegar a una conclusión. Realizar una investigación sobre las consideraciones de seguridad en el diseño del software, arquitectura y tecnologías, comentarlo en equipos y llegar a una conclusión. Partiendo de casos de estudio plantear problemáticas de diseño de software y proponer soluciones que apliquen principios de diseño seguro de software dentro de un proyecto de desarrollo de software.
Unidad 4: Implementación/Codificación de Software Seguro	
Competencias	Actividades de aprendizaje

Específica(s): Conocer e implementar técnicas de codificación defensiva y técnicas de protección de código que permitan fortalecer una aplicación ante ataques comunes del software. Genéricas: • Capacidad de análisis y síntesis. • Capacidad de organizar y planificar. • Comunicación oral y escrita. • Habilidad para buscar y analizar información proveniente de fuentes diversas. • Solución de problemas. • Toma de decisiones. • Capacidad crítica y autocrítica. • Capacidad de trabajar en equipo. • Capacidad de comunicar sus ideas. • Capacidad de liderazgo. • Capacidad de aplicar los conocimientos en la práctica. • Habilidades de investigación. • Capacidad de aprender. • Capacidad de adaptarse a nuevas situaciones. • Capacidad de generar nuevas ideas (creatividad). • Habilidad para trabajar en forma autónoma. • Preocupación por la calidad. • Búsqueda del logro.	Realizar una investigación sobre los ataques comunes al software y cómo esas vulnerabilidades pudieran ser explotadas, comentarlo en equipos y llegar a una conclusión. Realizar una investigación sobre las técnicas de codificación defensiva y técnicas de protección de código, comentarlo en equipos y llegar a una conclusión. Desarrollar e implementar un proyecto de desarrollo de software donde implemente medidas de seguridad y contramedidas utilizando principios de codificación defensiva.

8. Práctica(s)

1. Elija un escenario ya sea simulado u organizacional donde se pueda detectar alguna problemática para que realicen la implementación de una arquitectura adecuada del proyecto
2. Analiza y documenta la solución dada en la práctica anterior, utilizando modelos, arquitecturas y adapta el resultado obtenido enfocado en un entorno de calidad.
3. Lleva a cabo la gestión del proyecto de software elegido por los equipos de trabajo, considerando lo siguiente:
   3.1 Documenta adecuadamente cada fase
   3.2 Integra y justifica un equipo de desarrollo acorde a la metodología seleccionada para el desarrollo del proyecto de software.
   3.3 Presenta durante el semestre avances.
   3.4 Expone al final del semestre los resultados.

9. Proyecto de asignatura (Para fortalecer las competencias de la asignatura con otras asignaturas)

El proyecto integrador se realizará aplicando las competencias previas y vinculándolas con las competencias de las materias del semestre en curso; el proyecto integrador también debe tener un método de evaluación para acreditar la asignatura.

El proyecto integrador debe considerar las siguientes fases:

1. Contextualización o diagnóstico

2. Fundamentación

3. Planeación

4. Ejecución

5. Evaluación

6. Socialización

10. Evaluación por competencias

La evaluación debe ser continua y cotidiana por lo que se debe considerar el desempeño en cada una de las actividades de aprendizaje, haciendo especial énfasis en:

- Actividades que permitan la evaluación de conocimientos: cuestionarios, exámenes escritos, exámenes orales, entre otros.
- Actividades que permitan la evaluación de habilidades: Evaluar ejercicios, prácticas, proyectos de desarrollo tecnológico, proyectos de investigación, proyectos a través de la triple hélice, entre otras.
- Actividades que permitan la evaluación de actitudes: participación en clase, entrega puntual de sus asignaciones, puntualidad y asistencia, orden en el grupo, entre otras.
- Utilizar diferentes instrumentos de evaluación y sus respectivas rúbricas, para poder evaluar ampliamente y continuamente los aspectos conceptuales, procedimentales y actitudinales.

11. Fuentes de información

- Título: Official (ISC)2 Guide to the CSSLP CBK, Second Edition (ISC)2 Press. Autor: Mano Paul, Edición 2. Editor CRC Press, 2013. ISBN 1466571330, 9781466571334. N.º de páginas: 800.
- IEEE Std 610.12-1990, **"IEEE Standard Glossary of Software Engineering Terminology",**recuperado el 24 de Agosto de 2018 en: http://standards.ieee.org/reading/ieee/std_public/description/se/610.12-1990_desc.html
- Huiming Yu, N. Jones, G. Bullock and Xiaohong Yuan Yuan, "Teaching secure software engineering: Writing secure code," *2011 7th Central and Eastern European Software Engineering Conference (CEE-SECR)*, Moscow, 2011, pp. 1-5. doi: 10.1109/CEE-SECR.2011.6188473.
  URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6188473&isnumber=6188458
- N. H. Z. Zenah and N. A. Aziz, "Secure coding in software development," 2011 Malaysian Conference in Software Engineering, Johor Bahru, 2011, pp. 458-464. doi: 10.1109/MySEC.2011.6140716.
  URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6140716&isnumber=6140630