| UNIVERSIDAD AUTÓNOMA DE CHIHUAHUA<br>Clave: 08MSU0017H<br>FACULTAD DE INGENIERÍA<br>Clave: 08USU4053W<br>**PROGRAMA DEL CURSO: SEGURIDAD DE SOFTWARE** | | |
|---|---|---|
| | **DES:** | Ingeniería |
| | **Programa(s) Educativo(s):** | Ingeniería de Software |
| | **Tipo de materia:** | Obligatoria |
| | **Clave de la materia:** | IS0702 |
| | **Cuatrimestre:** | 7 |
| | **Área en plan de estudios:** | Específica |
| | **Créditos** | 5.4 |
| | **Total de horas por semana:** | 4 horas |
| | *Teoría: Virtual* | 4 horas |
| | *Práctica* | |
| | *Taller:* | |
| | *Laboratorio:* | |
| | *Prácticas complementarias:* | |
| | *Trabajo extra-clase:* | 4 horas |
| | **Total de horas por cuatrimestre:** | 96 horas |
| | Fecha de actualización: | Octubre de 2015 |
| | *Materia requisito:* | |

**PROPÓSITO DEL CURSO:**

"El estudiante valora los elementos para la construcción de software seguro en todas las etapas del ciclo de vida a través del análisis de las mejores prácticas y herramientas actuales para minimizar los riesgos que comprometen la seguridad del producto"

| COMPETENCIAS | DOMINIOS COGNITIVOS. | RESULTADOS DE APRENDIZAJE. |
|---|---|---|
| **Competencias profesionales:**<br><br>**Evaluación de proyectos de ingeniería,** Desarrolla las actividades propias de su profesión con base en procesos de calidad y mejora continua<br><br>**Competencias específicas:**<br>**Calidad de Software:** Construye software a través de la aplicación de técnicas, metodologías y estándares, que aplicados sistemáticamente garantizan la | **1. Fundamentos de Seguridad de Software.**<br>1.1. Fundamentos<br>1.1.1.Importancia<br>1.1.2.Terminología.<br>1.1.3. Actores<br>1.1.4.Impacto presupuestal<br>1.1.5.Identificación de amenazas<br>1.2. Buenas prácticas de Seguridad<br>1.2.1. Elicitacion segura de requerimientos<br>1.2.2.Arquitectura segura<br>1.2.3. Diseño detallado seguro.<br>1.2.4. Código seguro<br>1.2.5. Puesta en marcha y operación.<br>1.3. Ciclo de vida del software seguro. | Construye soluciones de problemas de ingeniería considerando los aspectos socioeconómicos |

| | | |
|---|---|---|
| calidad y seguridad del producto final.<br><br>**Ingeniería del Proceso de Software:** Aplica técnicas y metodologías de la Ingeniería de Software necesarias en el desarrollo y mantenimiento de componentes para conducir procesos de desarrollo a través de la realización de un conjunto coherente de actividades. | 1.3.1. Comparación entre el ciclo tradicional y el seguro.<br>1.3.2. Análisis de riesgo de arquitectura<br>1.3.3. Métricas<br>1.3.4. Administración de proyectos<br>1.4. Pruebas de software seguro<br>1.4.1. Análisis de código<br>1.4.2. Prueba de caja blanca<br>1.4.3. Prueba de penetración<br>1.4.4. Herramientas para pruebas de seguridad<br>**2. Diseño de software seguro**<br>2.1. Fundamentos de diseño seguro<br>2.1.1. Importancia<br>2.1.2. Terminología<br>2.1.3. Proceso<br>2.1.4. Principios<br>2.2. Aspectos relevantes en el diseño<br>2.2.1. Tipos de control de seguridad<br>2.2.2. Mecanismos de control de acceso<br>2.2.3. Métodos de encriptación<br>2.2.4. Prevención y detección de intrusos<br>2.2.5. Manejo de confianza<br>2.3. Estructuras y comportamientos<br>2.3.1. Patrones<br>2.3.2. Tácticas<br>2.3.3. Ejemplos prácticos<br>2.3.4. Técnicas de análisis y evaluación<br>2.3.5. Métricas<br>**3. Administración del desarrollo de software seguro**<br>3.1. Introducción<br>3.1.1. Naturaleza del software<br>3.1.2. Ciclo de vida del desarrollo de software<br>3.1.3. Elementos esenciales de seguridad<br>3.2. Especificaciones de seguridad<br>3.2.1. Acercamiento por capas<br>3.2.2. Amenazas y vectores de ataque<br>3.2.3. Políticas de seguridad<br>3.2.4. Cadena de suministros<br>3.3. Control del riesgo<br>3.3.1. Administración del riesgo<br>3.3.2. Respuesta a incidentes<br>3.3.3. Pruebas de seguridad | ● Valora los elementos de la construcción de software seguro<br>● Realiza la verificación y validación del software<br>● Implementa estrategias que aseguran la calidad del software<br>● Realiza pruebas de software |

| | | |
|---|---|---|
| | 3.3.4. Evolución de sistemas<br>3.4. Control de calidad<br>**4. Codificación de software seguro**<br>4.1. Principios y terminología<br>4.2. Vulnerabilidades<br>4.2.1. Buffer overflows<br>4.2.2. SQL Injection<br>4.2.3. Cross-site scripting<br>4.2.4. Autentificación y manejo de sesiones<br>4.2.5. Referencias a objetos<br>4.2.6. Exposición de datos<br>4.2.7. Problemas de configuración<br>4.3. Contramedidas<br>4.3.1. Estándares<br>4.3.2. Patrones<br>4.3.3. Depuración de datos<br>4.3.4. Autentificación<br>4.3.5. Manejo de sesiones<br>4.3.6. Encriptación<br>4.3.7. Administración de contraseñas<br>4.3.8. Control de acceso<br>4.3.9. Control de errores<br>4.3.10. Administración de archivos<br>4.3.11. Administración de memoria<br>4.4. Herramientas<br>4.4.1. Herramientas de análisis estático<br>4.4.2. Herramientas de análisis dinámico<br>4.4.3. Rastreo de vulnerabilidades<br>4.4.4. Seguridad en plataformas web<br>4.4.5. Seguridad en java | |
| **OBJETOS DE APRENDIZAJE** | **METODOLOGÍA**<br>(Estrategias, secuencias, recursos didácticos) | **EVIDENCIAS DE APRENDIZAJE.** |
| **1. Fundamentos de seguridad de software** | ● Videotutoriales<br>● Lecturas<br>● Foro de discusión | Mapa conceptual de terminología, actores y amenazas.<br>Resumen de buenas prácticas de seguridad a lo largo del ciclo de vida. |

| | | |
|---|---|---|
| **2. Diseño de software seguro** | ● Videotutoriales<br>● Lecturas<br>● Foro de discusión | Informe de resolución a un caso de estudio. |
| **3. Administración del desarrollo de software seguro** | ● Videotutoriales<br>● Lecturas<br>● Foro de discusión | Ensayo del proceso de administración que incluya valoración personal de los elementos del proceso. |
| **4. Codificación de software seguro** | ● Videotutoriales<br>● Lecturas<br>● Análisis de casos de estudio en foro de discusión | Informe de resolución a un caso de estudio. |

| **FUENTES DE INFORMACIÓN**<br>(Bibliografía, direcciones electrónicas) | **EVALUACIÓN DE LOS APRENDIZAJES**<br>(Criterios e instrumentos) |
|---|---|
| Software Security Engineering: A Guide for Project Managers, J.H. Allen et al., Addison-Wesley, 2008.<br><br>Software Security: Building security in, McGraw Gary, Pearson Education, ISBN 0-3211-35670-5<br><br>Building secure software, John Viega, Gary McGraw, Pearson Education, ISBN 0-21-72512-X<br><br>Secure software design, Theodor Richardson, Jones & Barlet Learning, ISBN 1-4496-2632-7 | La calificación final ordinaria se integra con los siguientes instrumentos:<br><br>**Fundamentos de seguridad de software**<br><br>● Mapa conceptual de terminología, actores y amenazas. **3%**<br>● Resumen de buenas prácticas de seguridad a lo largo del ciclo de vida. **10%**<br>● Participación en foros. **2%**<br>● Examen. **15%**<br><br>**Diseño de software seguro**<br><br>● Informe de resolución a un caso de estudio. **12%**<br>● Participación en foros. **2%**<br>● Examen. **15%**<br><br>**Administración del desarrollo de software seguro**<br><br>● Ensayo del proceso de administración que incluya valoración personal de los elementos del proceso. **10%**<br>● Foro. **2%**<br><br>**Codificación de software seguro**<br><br>● Informe de resolución a un caso de estudio. |

| | |
|---|---|
| | **12%**<br>Foro. **2%**<br>Examen. **15%**<br><br>Se evaluará mediante instrumentos tales como:<br>Listas de cotejo<br>Rúbricas<br>Exámenes en línea |

**Cronograma de Avance Programático**

| Objetos de aprendizaje. | | Semanas | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| I. | Fundamentos de seguridad | | | | | | | | | | | | |
| II. | Diseño de software seguro | | | | | | | | | | | | |
| III. | Administración del desarrollo de SW seguro | | | | | | | | | | | | |
| IV. | Codificación de software seguro | | | | | | | | | | | | |