



cætra

DUAL-USE TECHNOLOGIES ECOSYSTEM
REGION OF MURCIA



Programación Segura - Construyendo Resiliencia Cibernética (2ª Edición)

COLABORAN:



ORGANIZAN:



Contenido

Objetivos del Programa:	3
Descripción del Programa:	3
Estructura del Programa.....	4
Perfiles de participantes.....	9
Calendario.....	9
Formadores	10
Habilidades Adquiridas:	10

Objetivos del Programa:

La ciberseguridad es una cuestión crítica en los tiempos actuales debido a que ciberataques son cada vez más frecuentes, sofisticados y costosos.

Además, con la entrada en vigor de la Ley de Ciberresiliencia (CRA) en la Unión Europea, el desarrollo de software ha dejado de ser una cuestión meramente técnica para convertirse en una obligación legal. Esta normativa, junto con otras como DORA y NIS2, exige que todos los productos con componentes digitales, desde aplicaciones hasta dispositivos inteligentes, integren medidas de ciberseguridad desde su diseño y durante todo su ciclo de vida.

En este contexto, conocer y aplicar prácticas de programación segura ya no es opcional, es esencial para cumplir con la ley, proteger a los usuarios y garantizar la competitividad en el mercado europeo

Este curso de **50 horas lectivas** está diseñado para capacitar a profesionales del sector TIC en las **técnicas más actuales de programación segura** y en las claves de la **ciberresiliencia**, alineadas con la nueva normativa europea.

El objetivo final es el de formar a profesionales preparados para desarrollar software seguro desde el primer momento, con conocimientos sólidos y prácticos.

Se trata de una formación integral enmarcada en el programa CAETRA de la CARM, que impulsa la innovación y el emprendimiento tecnológico en el ámbito de la Defensa, Seguridad y Reconstrucción.

Descripción del Programa:

- **Duración total:** 6 semanas (mayo-junio 2026)
- **Horas lectivas:** 50 horas
- **Modalidad:** Híbrida (presencial+Online)
- **Estructura por módulos:** 5 módulos formativos
 - **Módulo 1:** Modelado de Amenazas y Seguridad por Diseño. **10 horas**
 - **Módulo 2:** Seguridad en Aplicaciones Web y OWASP Top Ten. **15 horas**
 - **Módulo 3:** Análisis de Código para la Detección de Vulnerabilidades. **10 horas**
 - **Módulo 4:** Hacking sobre Código y Pruebas de Penetración. **10 horas**
 - **Módulo 5:** Ley de Ciber Resiliencia Europea (CRA). **5 horas**
- **Número de Participantes:** 20 alumnos.

Estructura del Programa

Se trata de una formación integral enmarcada en el programa CAETRA de la CARM, que impulsa la innovación y el emprendimiento tecnológico en el ámbito de la Defensa, Seguridad y Reconstrucción.

Módulo 1: Modelado de Amenazas y Seguridad por Diseño

- **Conceptos Fundamentales del Modelado de Amenazas (2 horas)**
 1. Definición y propósito del modelado de amenazas en el ciclo de vida del desarrollo de software (SDLC).
 2. La importancia de una mentalidad de seguridad proactiva y el concepto de "cambiar la seguridad a la izquierda".
 3. Beneficios del modelado de amenazas: reducción de costos, retorno visible de la inversión, y evitar la necesidad de corrección posterior.
 4. Relación entre el modelado de amenazas y la "seguridad por diseño".
 5. Threat Hunting vs. Threat Modeling: comprender sus diferencias y cómo se complementan.
- **Metodologías y Marcos de Modelado de Amenazas (2 horas)**
 1. Introducción a diversas metodologías de modelado de amenazas.
 2. Análisis de diagramas de flujo de datos (DFD) como base para el modelado de amenazas.
 3. Identificación de activos, amenazas, y vulnerabilidades en diferentes arquitecturas de software.
 4. Establecimiento de controles de seguridad (contramedidas) para mitigar las amenazas.
- **Herramientas de Modelado de Amenazas (2 horas)**
 1. Exploración de diversas herramientas para hacer modelado de amenazas.
 2. Exploración en profundidad de IriusRisk como una herramienta de modelado de amenazas automatizada.
 3. Cómo incorporar el modelado de amenazas en cada etapa del ciclo de vida del desarrollo de software.
 4. Alineación del modelado de amenazas con regulaciones y marcos de seguridad.
- **Proceso de Creación de un Modelo de Amenazas (4 horas)**
 1. Creación de un modelo de amenazas para un caso de uso controlado.
 2. Análisis de los resultados, comentarios y dudas.
 3. Breve introducción a las certificaciones de IriusRisk.
 4. Dudas y preguntas sobre el curso.

Módulo 2: Seguridad en Aplicaciones Web y OWASP Top Ten (15 horas)

- **Auditorías de Aplicaciones Web (8 horas)**
 1. Explicación de los protocolos HTTP y HTTPS, y de su rol en las comunicaciones de las aplicaciones Web.
 2. Fases de una auditoría de seguridad de aplicaciones web.
 - a. Planificación de la auditoría.
 - b. Recolección de información.
 - c. Escaneos de vulnerabilidades.
 - d. Explotación de vulnerabilidades.
 - e. Generación del informe de resultados.
 3. Configuración de un entorno de pruebas, donde se ponga en práctica la explotación de vulnerabilidades explicadas.
- **Introducción a OWASP (1 hora)**
 4. ¿Qué es OWASP? El Proyecto Abierto de Seguridad de Aplicaciones Web y su misión.
 5. El OWASP Application Security Verification Standard (ASVS) como base para pruebas de seguridad.
 6. El Proyecto de Guía de Pruebas de Seguridad Web.
 7. La importancia de abordar la seguridad en aplicaciones web debido a su creciente uso y la frecuencia de ataques.
- **Análisis Detallado de las Diez Principales Vulnerabilidades de OWASP (5 horas)**
 1. Inyección SQL. Explicación del ataque, cómo ocurre (envío de datos no confiables a un intérprete), ejemplos, técnicas de prevención (validación de entradas, uso de sentencias preparadas).
 2. Secuencia de comandos en sitios cruzados (XSS). Tipos (almacenado, reflejado, basado en DOM), cómo se explotan, ejemplos de código malicioso, estrategias de mitigación (sanitización de entradas, codificación de salidas).
 3. Pérdida de autenticación y gestión de sesiones. Vulnerabilidades comunes (cierre de sesión inseguro, gestión débil de contraseñas, tiempos de desconexión inadecuados), cómo los atacantes pueden obtener acceso no autorizado, buenas prácticas (uso de cifrado, tokens seguros).
 4. Referencia directa insegura a objetos. Explicación de la vulnerabilidad, cómo los atacantes pueden acceder a recursos no autorizados, técnicas de prevención (uso de referencias indirectas, comprobación de acceso).
 5. Falsificación de peticiones en sitios cruzados (CSRF). Cómo los atacantes pueden generar peticiones falsas, ejemplos, métodos de mitigación (uso de tokens anti-CSRF).
 6. Configuración de seguridad defectuosa. Problemas comunes (software sin parches, errores de configuración del servidor, archivos por defecto, permisos inadecuados), impacto, recomendaciones para una configuración segura.
 7. Almacenamiento criptográfico inseguro. Errores comunes al aplicar cifrado (cifrado débil, almacenamiento inseguro de claves), importancia de proteger la información sensible, buenas prácticas (uso de algoritmos fuertes, gestión segura de claves).

8. Fallo de restricción de acceso a URL. Cómo los atacantes pueden acceder a páginas no autorizadas, importancia del control de acceso, recomendaciones (autorización basada en roles, denegación por defecto).
 9. Protección insuficiente en la capa de transporte. Uso inseguro de HTTP/HTTPS, ataques Man-in-the-Middle, mejores prácticas (uso exclusivo de HTTPS, configuración correcta de SSL/TLS, HSTS).
 10. Redirecciones y reenvíos no validados. Cómo los atacantes pueden redirigir a los usuarios a sitios maliciosos, estrategias para un uso seguro (evitar redirecciones basadas en parámetros de usuario, validación).
- **Otros Problemas de Seguridad Web (1 hora)**
 1. Entradas no validadas. Tipos de ataques derivados de la falta de validación, principios de validación de entradas.
 2. Manejo incorrecto de errores. Información sensible revelada a atacantes, buenas prácticas en el manejo de errores.
 3. Denegación de servicios (DoS). Cómo los atacantes pueden sobrecargar una aplicación, estrategias de mitigación (sin profundizar demasiado en ataques avanzados).

Módulo 3: Análisis de Código para la Detección de Vulnerabilidades (10 horas)

- **Introducción al Análisis de Código (2 horas)**
 1. Concepto y objetivos del análisis de código en la identificación temprana de vulnerabilidades.
 2. La importancia de integrar el análisis de código en el ciclo de vida del desarrollo (Secure SDLC y enfoque Shift-Left).
 3. Relación entre el análisis de código y la mitigación de las vulnerabilidades del OWASP Top 10.
 4. Análisis estático (SAST) vs. Análisis dinámico (DAST): diferencias, ventajas y desventajas.
 5. Introducción al Software Composition Analysis (SCA): detección de riesgos en componentes de terceros, dependencias de código abierto y gestión de vulnerabilidades conocidas (CVE).
- **Análisis Estático de Seguridad de Aplicaciones (SAST) (4 horas)**
 1. Funcionamiento de las herramientas SAST: análisis léxico, sintáctico, flujo de datos (data-flow), flujo de control y taint analysis.
 2. Tipos de vulnerabilidades detectables mediante SAST (inyecciones, XSS, uso inseguro de criptografía, hardcoded secrets, etc.).
 3. Ejemplos de herramientas SAST del mercado: Veracode, Fortify, Checkmarx, así como alternativas open source como Semgrep o SonarQube.
 4. Interpretación de los resultados del análisis SAST, priorización de hallazgos y gestión de falsos positivos.
 5. Integración de herramientas SAST en el proceso de desarrollo y en pipelines CI/CD (GitHub Actions, GitLab CI, Jenkins) .
 6. Prácticas básicas con una herramienta SAST de código abierto sobre un proyecto vulnerable de ejemplo.

- **Análisis Dinámico de Seguridad de Aplicaciones (DAST) (3 horas)**
 1. Funcionamiento de las herramientas DAST: interacción con la aplicación en ejecución y simulación de ataques.
 2. Tipos de vulnerabilidades que pueden detectar las herramientas DAST (inyecciones, problemas de autenticación y sesión, configuraciones inseguras del servidor, etc.).
 3. Ejemplos de herramientas DAST comunes en la industria: OWASP ZAP, Burp Suite, Acunetix o Netsparker.
 4. Interpretación de los resultados del análisis DAST y validación de vulnerabilidades.
 5. Consideraciones al utilizar herramientas DAST en diferentes entornos (desarrollo, preproducción y producción) y buenas prácticas para evitar impacto operativo.
- **Inteligencia Artificial Generativa en Auditoría y Remediación (1 hora)**
 1. Aplicaciones de la IA generativa en el triaje de hallazgos: priorización automática, detección de falsos positivos y clasificación contextual.
 2. Generación de propuestas de remediación: sugerencias de código seguro, explicación de la vulnerabilidad y justificación del fix.
 3. 19. Integración de la IA en el entorno de desarrollo: plugins, asistentes contextuales en el IDE y buenas prácticas para un uso responsable (protección de datos, revisión humana y gobernanza).

Módulo 4: Hacking sobre Código y Pruebas de Penetración (10 horas)

- **Introducción al Hacking Ético y las Pruebas de Penetración (2 horas)**
 1. Conceptos básicos del hacking ético y su rol en la seguridad.
 2. ¿Qué son las pruebas de penetración (pentesting)?
 3. Diferentes tipos de pruebas de penetración (caja negra, caja blanca, caja gris).
 4. Fases de una prueba de penetración.
 5. Consideraciones legales y éticas en las pruebas de penetración.
- **Explotación Práctica de Vulnerabilidades OWASP (6 horas - 1 hora por un subconjunto de vulnerabilidades clave)**
 1. Inyección SQL. Demostraciones conceptuales de cómo se puede explotar la inyección SQL para acceder a datos no autorizados.
 2. XSS. Ejemplos prácticos de cómo inyectar scripts maliciosos y sus posibles impactos.
 3. Vulnerabilidades de Autenticación. Simulación de intentos de eludir mecanismos de autenticación débiles.
 4. Referencia Directa Insegura a Objetos. Intentos de acceder a recursos utilizando identificadores manipulados.
 5. Configuración de Seguridad Defectuosa. Exploración de cómo configuraciones inseguras pueden ser aprovechadas.
 6. Falsificación de Peticiones en Sitios Cruzados (CSRF). Demostración de cómo se pueden generar peticiones no deseadas en nombre de un usuario autenticado.

Nota: Debido a las limitaciones de un curso formativo, estas demostraciones serán conceptuales y se realizarán en entornos controlados o utilizando código de ejemplo vulnerable, sin realizar ataques a sistemas reales sin permiso.

- **Herramientas Comunes de Hacking y Pentesting (2 horas)**
 1. Introducción a herramientas utilizadas para identificar y explotar vulnerabilidades (e.g., Burp Suite, OWASP ZAP, SQLMap - aclarando que las fuentes no especifican estas herramientas, pero son ejemplos comunes en la industria).
 2. Uso básico de estas herramientas para analizar el tráfico web y realizar pruebas básicas.
 3. ¿Qué es el Bug bounty y como me puede ayudar a identificar fallos de programación?

Módulo 5: Ley de Ciber Resiliencia Europea (CRA) (5 horas)

- **Introducción y Contexto de la Ley de Ciber Resiliencia (1 hora)**
 1. Entrada en vigor del Reglamento CRA (Ley de Ciber Resiliencia).
 2. Objetivo principal de la ley: proteger a consumidores y empresas asegurando altos estándares de seguridad en productos con elementos digitales.
 3. La necesidad de abordar el bajo nivel de ciberseguridad en productos digitales y la insuficiencia de información para los usuarios.
 4. El aumento de los riesgos cibernéticos debido al teletrabajo, dispositivos conectados y la digitalización.
 5. Reglamento de Inteligencia Artificial (AI Act) de la Unión Europea.
- **Alcance y Aplicabilidad de la Ley (1.5 horas)**
 1. ¿A quiénes afecta la ley? Fabricantes, desarrolladores, distribuidores, importadores y otros actores que comercializan productos digitales en la UE, tanto dentro como fuera de ella.
 2. ¿A qué productos y servicios digitales afecta? Definición amplia de "productos con elementos digitales" (hardware y software conectados directa o indirectamente), con excepciones específicas.
 3. La importancia de la marca CE para demostrar conformidad.
- **Requisitos y Obligaciones Clave (2 horas)**
 1. Requisitos de ciberseguridad para el diseño, desarrollo y producción.
 2. Obligaciones de los fabricantes: eliminación proactiva de vulnerabilidades, implementación de actualizaciones periódicas, soporte de seguridad y actualizaciones de software durante todo el ciclo de vida del producto (o 5 años).
 3. Notificación de vulnerabilidades a las autoridades competentes dentro de las 24 horas.
 4. Transparencia para los consumidores: proporcionar información precisa sobre las características de seguridad.
 5. Procesos de gestión de vulnerabilidades.
- **Implementación y Cumplimiento (0.5 horas)**
 1. Plazos de aplicación. Diferentes fechas para diferentes disposiciones (información de fabricantes, notificación de organismos, aplicación general).
 2. Plazo de adaptación para los fabricantes.
 3. Clasificación de productos según su nivel de riesgo (importantes, críticos, no clasificados) y sus implicaciones.
 4. Relación entre la Ley de Ciber Resiliencia y otras normativas (e.g., GDPR).

Perfiles de participantes

Requisitos de admisión y perfil participantes

Este curso está orientado a **profesionales en activo** con experiencia y en especial a:

- Desarrolladores de software
- Arquitectos de soluciones
- Técnicos y responsables de TI
- Especialistas en seguridad informática
- Cualquier profesional que participe en el desarrollo de productos digitales

Para participar en el Programa se priorizará por:

- Trabajadores en activo en el área TIC de empresas que desarrollen su actividad principal en un centro de trabajo ubicado en el ámbito geográfico de la Comunidad Autónoma de la Región de Murcia.
- Que tengan su trabajo en el área de desarrollo y cuenten con varios años de experiencia y en especial en desarrollo.
- Conocimientos de lenguajes de programación

En el caso de que haya más perfiles que cumplan los requisitos anteriores que plazas, se priorizará por orden de entrada.

Se podrá aceptar más de un trabajador por empresa siempre y cuando haya disponibilidad de plazas.

Calendario

50 horas (mayo – junio 2026)

3 jornadas por semana, en jueves y viernes, con el siguiente horario:

- Jueves mañana de 12:00 a 14:00 horas (2 horas)
- Jueves tarde de 15:30 a 19:30 horas (4 horas)
- Viernes de 9:00 a 13:00 horas (4 horas)

Distribución semana por módulo y modalidad

	Semana 1		Semana 2		Semana 3		Semana 4		Semana 5	Semana 6
	Jueves (21 mayo)	Viernes (22 mayo)	Jueves (28 mayo)	Viernes (29 mayo)	Jueves (4 junio)	Viernes (5 junio)	Jueves (11 junio)	Viernes (12 junio)	Jueves (18 junio)	Viernes (26 junio)
Módulo 1 (10h.)	P	P								
Módulo 2 (15h.)			O	O	O					
Módulo 3 (10h.)						O	O			
Módulo 4 (10h.)								O	O	
Módulo 5 (5h.)										P

P: Presencial
O: On Line

Formadores

Currículums

Álvaro Reyes, Analista de seguridad en IriusRisk

https://es.linkedin.com/in/alvaro-reyes?trk=public_post_feed-actor-name

Mercedes Muñoz Morillas, Pentesting Team Lead en aDvens

<https://www.linkedin.com/in/mikiminoru/>

José Manuel Núñez Marín, Red Team Operator en aDvens Iberia

<https://www.linkedin.com/in/jose-manuel-nu%C3%B1ez-marin-leadauditor/>

Bernardo Viqueira Hierro, Responsable Ciberseguridad DooingIT

<https://www.linkedin.com/in/bernardo-viqueira-hierro-0b882637/>

Aarón Flecha Menéndez, Consultor HACKRTU SL

<https://www.linkedin.com/in/aar%C3%B3n-flecha-men%C3%A9ndez-76773232/?originalSubdomain=es>

Habilidades Adquiridas:

Los participantes en esta formación obtendrán las siguientes habilidades:

- Aprender a incluir la seguridad desde el inicio del desarrollo de software, evitando errores costosos más adelante.
- Conocer las vulnerabilidades más comunes en aplicaciones web y cómo prevenirlas de forma eficaz.
- Saber cómo analizar el código de tus programas para detectar fallos de seguridad antes de que sea tarde.
- Verás ejemplos reales de ataques y aprenderás a protegerte ante ellos en un entorno seguro.
- Te pondrás al día con la nueva ley europea sobre ciberseguridad y lo que implica para tu trabajo.)

Además, los contenidos desarrollados en esta formación preparan al asistente para obtener certificaciones profesionales gratuitas, de Iriusrisk, muy valoradas en el sector como:

- Threat Modeling Foundations
- Threat Modeling Champion
- Y más...