



#### DESCRIPCIÓN DE LA ASIGNATURA

<b>Grado/Máster en:</b>	Master Universitario en INGENIERÍA INFORMÁTICA por la Universidad de Málaga
<b>Centro:</b>	Escuela Técnica Superior de Ingeniería Informática
<b>Asignatura:</b>	PROGRAMACIÓN SEGURA
<b>Código:</b>	204
<b>Tipo:</b>	Optativa
<b>Materia:</b>	ESPECIALIDAD EN CIBERSEGURIDAD
<b>Módulo:</b>	COMPLEMENTOS EN TECNOLOGÍAS INFORMÁTICAS
<b>Experimentalidad:</b>	
<b>Idioma en el que se imparte:</b>	Español
<b>Curso:</b>	2
<b>Semestre:</b>	1
<b>Nº Créditos</b>	4,5
<b>Nº Horas de dedicación del estudiante:</b>	112,5
<b>Nº Horas presenciales:</b>	33,8
<b>Tamaño del Grupo Grande:</b>	
<b>Tamaño del Grupo Reducido:</b>	
<b>Página web de la asignatura:</b>	

#### EQUIPO DOCENTE

<b>Departamento:</b>	LENGUAJES Y CIENCIAS DE LA COMPUTACIÓN
<b>Área:</b>	CIENCIA DE LA COMPUTACIÓN E INTELIGENCIA ARTIFICIAL

Nombre y Apellidos	Mail	Teléfono Laboral	Despacho	Horario Tutorías
Coordinador/a: JOSE ANTONIO MONTENEGRO MONTES	jmmontes@uma.es	952132808	3.2.40 - E.T.S.I. INFORMÁTICA	

#### RECOMENDACIONES Y ORIENTACIONES

Para cursar con éxito la asignatura es necesario poseer algunos conocimientos adquiridos en asignaturas anteriores:

- Seguridad de la Información: Conceptos básicos de seguridad de la información.
- Programación orientada a objetos: Programación en Java (la asignatura tiene prácticas de laboratorio obligatorias que deben programarse en Java).
- Sistemas Operativos: Conceptos básicos Gestión de Memoria y desarrollo entorno de desarrollo Linux.

#### CONTEXTO

Este curso cubre los principios y prácticas de la programación segura. Expondremos los modelos de seguridad, las amenazas, los principios de diseño y las prácticas de codificación segura. Un desarrollador con el adecuado conocimiento de estas técnicas minimizará las vulnerabilidades en el software, evitando que el software desarrollado pueda ser vulnerable y ser expuesto ante posibles ataques. Para el desarrollo de la asignatura desde el punto de vista teórico como práctico tendremos en cuenta las plataformas más representativas, desde las plataformas tradicionales hasta los dispositivos móviles, pasando por las plataformas web.

#### COMPETENCIAS

##### 1 Competencias generales y básicas.

###### Competencias básicas

- 1.2 CB7 - Que los estudiantes sepan aplicar los conocimientos adquiridos y su capacidad de resolución de problemas en entornos nuevos o poco conocidos dentro de contextos más amplios (o multidisciplinares) relacionados con su área de estudio
- 1.5 CB10 - Que los estudiantes posean las habilidades de aprendizaje que les permitan continuar estudiando de un modo que habrá de ser en gran medida autodirigido o autónomo

##### 2 Competencias específicas.

- 2.4 ET14 - Capacidad para diseñar, desarrollar, gestionar y evaluar mecanismos de certificación y garantía de seguridad en el tratamiento y acceso a la información en un sistema de procesamiento local o distribuido.

#### CONTENIDOS DE LA ASIGNATURA

##### Introducción a la programación segura

- 1.a. Descripción general de la revelación de vulnerabilidades.
- 1.b. Prácticas de programación segura.



#### Buenas prácticas de programación segura

- 2.a. Programación de proceso mono-usuario
- 2-b. Programación en Red

#### Programación segura con análisis estático

- 3. a. Análisis estático como parte del proceso de revisión de código.
- 3. b. Análisis estático al detalle

#### Herramientas

- 4. a. Soot. A framework for analyzing and transforming Java Applications
- 4. b. SAMATE - Software Assurance Metrics and Tool Evaluation
- 4. c. Polyspace Bug Finder

### ACTIVIDADES FORMATIVAS

#### Actividades presenciales

##### Actividades expositivas

Lección magistral

### ACTIVIDADES DE EVALUACIÓN

#### Actividades de evaluación presenciales

##### Actividades de evaluación del estudiante

Examen parcial  
Examen final  
Realización de trabajos y/o proyectos

### RESULTADOS DE APRENDIZAJE / CRITERIOS DE EVALUACIÓN

Nuestro objetivo es seguir en esta asignatura un enfoque práctico donde la mayoría de las sesiones se dedicarán al trabajo en proyectos y actividades individuales. Siguiendo algunos de los principios del aprendizaje denominado aula invertida, los estudiantes recibirán previamente a las clases material detallado de las sesiones, reemplazando en la medida de lo posible, las horas de clases teóricas por clases interactivas.

Propondremos para cada bloque un proyecto basándonos en la metodología de Aprendizaje Basado en Proyectos (PBL), que será utilizado para guiar el estudio sobre los contenidos. Los estudiantes recibirán una rúbrica para que puedan anticipar claramente en qué medida su proyecto cubre las expectativas. Trataremos de diseñar proyectos incrementales de tal manera que los estudiantes puedan reutilizar logros previos para cada nuevo proyecto, o al menos hacerlo interrelacionado. Los proyectos se centrarán en la programación segura en tres plataformas:

- Traditional Platform (Unix,Windows)
- Web
- Mobile Devices (Android)

### PROCEDIMIENTO DE EVALUACIÓN

La evaluación se basará en los siguientes elementos:

- a) Entrega de prácticas. Constituye el 15% de la nota final.
- b) Una evaluación sobre las prácticas. Constituyen el 15% de la nota final.
- c) Una evaluación final por convocatoria, en el que se evaluará el desarrollo de un proyecto donde la evaluación viene determinada por los siguientes criterios: Originalidad, Técnicas utilizadas, Exposición, Diseño Material Exposición. Será del 70% de la nota final.

Los alumnos que posean el reconocimiento de estudiante a tiempo parcial y/o el reconocimiento de deportista universitario de alto nivel podrán compensar la evaluación del tipo test mediante la entrega de trabajos voluntarios.

Este sistema de evaluación se aplicará en todas las convocatorias, incluidas las extraordinarias.

### BIBLIOGRAFÍA Y OTROS RECURSOS

#### Básica

- B. Chess and J. West. Secure Programming with Static Analysis. Addison Wesley, 2007.
- Fred Long, Dhruv Mohindra, Robert C. Seacord, Dean F. Sutherland, David Svoboda. The CERT Oracle Secure Coding Standard for Java. Addison Wesley, 2011.
- J. Viega and G. McGraw. Building Secure Software: How to Avoid Security Problems the Right Way. Addison Wesley, 2001.
- John Viega, Matt Messier - Secure programming cookbook, O'Reilly



Mark Dowd, John McDonald, Justin Schuh. The Art of Software Security Assessment: Identifying and Preventing Software Vulnerabilities. Addison Wesley, 2006.

Michael Howard - Writing secure code, MS Press

Michael Howard, Steve Lipner - Secure Development Lifecycle, MS Press

Secure Coding Guidelines for Java SE. <http://www.oracle.com/technetwork/java/seccodeguide-139067.html>

**DISTRIBUCIÓN DEL TRABAJO DEL ESTUDIANTE**

**ACTIVIDAD FORMATIVA PRESENCIAL**

Descripción	Horas	Grupo grande	Grupos reducidos
Lección magistral	33,8	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<b>TOTAL HORAS ACTIVIDAD FORMATIVA PRESENCIAL</b>	<b>33,8</b>		
<b>TOTAL HORAS ACTIVIDAD FORMATIVA NO PRESENCIAL</b>	<b>67,45</b>		
<b>TOTAL HORAS ACTIVIDAD EVALUACIÓN</b>	<b>11,25</b>		
<b>TOTAL HORAS DE TRABAJO DEL ESTUDIANTE</b>	<b>112,5</b>		

