

# 69168 - Fundamentals of Computing for Robotics, Graphics and Computer Vision

## Información del Plan Docente

**Año académico:** 2024/25

**Asignatura:** 69168 - Fundamentals of Computing for Robotics, Graphics and Computer Vision

**Centro académico:** 110 - Escuela de Ingeniería y Arquitectura

**Titulación:** 615 - Máster Universitario en Robótica, Gráficos y Visión por Computador / Robotics, Graphics and Computer Vision

**Créditos:** 3.0

**Curso:** 1

**Periodo de impartición:** Primer semestre

**Clase de asignatura:** Optativa

**Materia:**

## 1. Información básica de la asignatura

El curso tiene como objetivo fomentar prácticas efectivas de desarrollo de software y de computación, dentro del contexto de la Robótica, los Gráficos y la Visión por Computador. Durante el curso los estudiantes aprenderán los principios del diseño de sistemas informáticos y cómo desarrollar eficientemente algoritmos dirigidos a ellos, evitando los escollos que genera la falta de metodología en el desarrollo. Las técnicas exploradas en este curso van desde el bajo nivel y la gestión de memoria hasta abstracciones de alto nivel, como programación funcional u orientada a objetos. Además, también se explorarán conceptos generales de desarrollo de software, como el desarrollo basado en pruebas y el control de versiones.

Este curso está recomendado para estudiantes con cierta experiencia en programación que quieran mejorar sus habilidades para abordar con confianza las tareas informáticas durante el resto de los estudios de Maestría.

## 2. Resultados de aprendizaje

El estudiante debe ser capaz de:

- Comprender la arquitectura de los sistemas informáticos (hardware y software).
- Diseñar e implementar algoritmos simples para resolver problemas simples.
- Compilar programas y vincular bibliotecas con dichos programas.
- Comprender las interfaces del sistema: interfaz de programación de aplicaciones, interfaz binaria de aplicaciones, interfaz binaria de supervisor.
- Comprender la gestión de la memoria y cómo se accede a ella al desarrollar algoritmos.
- Mejorar la robustez de los algoritmos mediante depuración y metodologías de desarrollo seguras.
- Abstractar funcionalidad y representación de datos en algoritmos.
- Aplicar metodologías colaborativas para el desarrollo de software.
- Comprender los conceptos básicos de seguridad informática.

## 3. Programa de la asignatura

- Repaso de programación imperativa: composición secuencial, condicional e iterativa
- Subprogramas: funciones, parámetros, variables y abstracción funcional.
- Sistemas Informáticos: hardware y periféricos, sistemas operativos, bibliotecas y runtimes, aplicaciones.
- Gestión de memoria: memoria virtual, no gestionada o administrada, reserva de memoria (pila de activación y memoria dinámica), acceso indirecto a la memoria (punteros, referencias), punteros inteligentes
- Abstracción de datos: programación orientada a objetos, herencia, programación genérica
- Robustez, depuración y pruebas: metodologías de depuración manual, depuradores, excepciones, desarrollo basado en pruebas.
- Prácticas y herramientas de desarrollo: IDE, linters, verificadores de memoria, control de versiones.
- Seguridad básica: datos en reposo, datos en movimiento y datos en proceso.

## 4. Actividades académicas

La asignatura consta de 3 créditos ECTS que corresponden a una dedicación estimada del estudiante de 75 horas distribuidas de la siguiente manera:

- Clases teóricas: Se explicarán los conceptos teóricos de la materia y se desarrollarán ejemplos prácticos ilustrativos que apoyen la teoría cuando sea necesario. (10h)
- Clases de problemas: Se realizarán problemas y casos prácticos como complemento a los conceptos teóricos estudiados. (5h)
- Prácticas de laboratorio: Se realizarán una serie de trabajos guiados y tutorizados por el profesor. (15h)
- Estudio y asimilación de la teoría expuesta en las clases magistrales. (30h)
- Trabajo de aplicación práctica o investigación (13h).
- Pruebas de evaluación (2h).

## 5. Sistema de evaluación

**Evaluación continua:** para la evaluación continua, el sistema de evaluación es:

- **Trabajos prácticos (80%):** Se realizarán un conjunto de prácticas guiadas a lo largo del curso, con plazos fijos para entregar cada una de ellas. El resultado de estos será el código fuente que será evaluado de acuerdo con su calidad w.r.t. los principios, técnicas y procedimientos presentados durante las clases teóricas.
- **Exámenes orales (20%):** En cada trabajo práctico se realizarán preguntas a los estudiantes sobre su trabajo y cómo se vincula con los conceptos presentados durante las clases teóricas.

**Evaluación global:** para la evaluación global, el sistema de evaluación es:

- **Examen práctico (80%):** Consistente en un conjunto de ejercicios prácticos de programación, este examen se realizará en la fecha oficial del examen. Las soluciones de este examen serán evaluadas según su calidad w.r.t. los principios, técnicas y procedimientos presentados durante las clases teóricas.
- **Examen oral (20%):** Después de los ejercicios de programación, los alumnos recibirán preguntas de los profesores sobre sus soluciones propuestas y cómo se vinculan con los conceptos presentados durante las clases teóricas.

## 6. Objetivos de Desarrollo Sostenible

8 - Trabajo Decente y Crecimiento Económico  
9 - Industria, Innovación e Infraestructura