

Programa del curso EE-9302

Desarrollo de software para aplicaciones críticas

Escuela de Ingeniería Electromecánica
Carrera de Ingeniería Electromecánica con énfasis en Sistemas Ciberfísicos

I parte: Aspectos relativos al plan de estudios

1. Datos generales

Nombre del curso:	Desarrollo de software para aplicaciones críticas
Código:	EE-9302
Tipo de curso:	Teórico - Práctico
Obligatorio o electivo:	Electivo
Nº de créditos:	3
Nº horas de clase por semana:	4
Nº horas extraclase por semana:	5
Ubicación en el plan de estudios:	Curso electivo en Ingeniería Electromecánica con énfasis en Sistemas Ciberfísicos
Requisitos:	EE-8808 Fundamentos de ciberseguridad
Correquisitos:	Ninguno
El curso es requisito de:	Ninguno
Asistencia:	Obligatoria
Suficiencia:	No
Posibilidad de reconocimiento:	Sí
Aprobación y actualización del programa:	01/01/2026 en sesión de Consejo de Escuela 01-2026

2. Descripción general

El curso de *Desarrollo de software para aplicaciones críticas* es del tipo electivo y por esta razón no se incluye en los rasgos del plan de estudios.

Los aprendizajes que los estudiantes desarrollarán en el curso son: diseñar soluciones de software robustas aplicando estrategias de programación defensiva y tolerancia a fallos; implementar módulos funcionales utilizando herramientas y plataformas embebidas con soporte en tiempo real; verificar el comportamiento del software mediante pruebas automatizadas, análisis estático y simulación; y evaluar el cumplimiento de normas y estándares internacionales relevantes según el dominio de aplicación.

Para desempeñarse adecuadamente en este curso, los estudiantes deben poner en práctica lo aprendido en los cursos de: Microcontroladores, Aplicaciones de sistemas embebidos, y Automatización y digitalización industrial.

3. Objetivos

Al final del curso la persona estudiante será capaz de:

Objetivo general

- Desarrollar software confiable para aplicaciones críticas, incorporando principios de diseño seguro, verificación rigurosa y cumplimiento normativo en entornos embebidos y ciberfísicos.

Objetivos específicos

- Diseñar soluciones de software robustas aplicando estrategias de programación defensiva y tolerancia a fallos.
- Implementar módulos funcionales utilizando herramientas y plataformas embebidas con soporte en tiempo real.
- Verificar el comportamiento del software mediante pruebas automatizadas, análisis estático y simulación.
- Evaluar el cumplimiento de normas y estándares internacionales relevantes según el dominio de aplicación.

4. Contenidos

En el curso se desarrollarán los siguientes temas:

1. Introducción a sistemas software críticos
 - 1.1. Clasificación de aplicaciones críticas.
 - 1.2. Ciclo de vida del software en sistemas embebidos.
2. Modelado y especificación de requisitos
 - 2.1. Requisitos funcionales y no funcionales.
 - 2.2. Lenguajes de especificación formal
3. Arquitecturas y diseño seguro
 - 3.1. Principios de modularidad, aislamiento y redundancia.
 - 3.2. Patrones de diseño seguros.
4. Programación robusta

- 4.1. Técnicas de programación defensiva.
- 4.2. Control de errores y fallos.
- 5. Sistemas operativos en tiempo real (RTOS)
 - 5.1. Planificación, sincronización y manejo de recursos.
 - 5.2. Implementación de tareas periódicas y esporádicas.
- 6. Verificación, validación y pruebas
 - 6.1. Pruebas unitarias, de integración y de sistema.
 - 6.2. Pruebas en hardware-in-the-loop (HIL).
 - 6.3. Cobertura de código y análisis estático.
- 7. Normativas y certificación
 - 7.1. ISO 26262 (automotriz)
 - 7.2. DO-178C (aeroespacial)
 - 7.3. IEC 61508 (industrial)
- 8. Ciclo de certificación y documentación.
 - 8.1. Casos de estudio
- 9. Sistemas de control de vuelo (autopilotos, gestión de potencia).
- 10. Sistemas médicos embebidos (bombas de infusión, marcapasos).
- 11. Sistemas automotrices seguros (ADAS, frenos ABS).
- 12. Automatización industrial con redundancia y diagnóstico.
- 13. Análisis de fallas y mitigación: estudio de fallos históricos (Therac-25, Ariane 5, Toyota).

II parte: Aspectos operativos

5. Metodología En este curso, se utilizará el enfoque sistémico-complejo para la ejecución de las sesiones magistrales y se integrará la investigación práctica aplicada para las sesiones prácticas. Esta última se implementará mediante técnicas como el modelado, simulación, prototipado y la experimentación controlada.

Las personas estudiantes podrán desarrollar actividades en las que:

- Recibirán clases magistrales con material audiovisual y discusión en grupo sobre desarrollo de software para aplicaciones críticas
- Desarrollarán ejercicios prácticos en laboratorio utilizando RTOS y plataformas embebidas.
- Aplicarán técnicas de diseño robusto, programación defensiva y control de fallos.
- Analizarán casos de estudio reales para identificar riesgos, fallos y estrategias de mitigación.
- Evaluarán el cumplimiento de estándares internacionales.

Este enfoque metodológico permitirá a la persona estudiante desarrollar software confiable para aplicaciones críticas, incorporando principios de diseño seguro, verificación rigurosa y cumplimiento normativo en entornos embebidos y ciberfísicos

Si un estudiante requiere apoyos educativos, podrá solicitarlos a través del Departamento de Orientación y Psicología.

6. Evaluación La evaluación se distribuye en los siguientes rubros:

- Pruebas parciales: evaluaciones formales que miden el nivel de comprensión y aplicación de los conceptos clave del curso. Generalmente cubren una parte significativa del contenido visto hasta la fecha y pueden incluir problemas teóricos y prácticos.
- Tareas: evaluaciones que tienen el propósito de reforzar, aplicar o evaluar el aprendizaje de un tema específico. Pueden requerir investigación, resolución de problemas, desarrollo de habilidades prácticas o aplicación de conocimientos teóricos.
- Act. aprendizaje activo: actividad diseñada para que los estudiantes se involucren de manera directa y práctica en la construcción de su conocimiento, a través de la resolución de problemas, la discusión y la aplicación de conceptos teóricos en contextos reales o simulados.

Pruebas parciales (2)	60 %
Tareas (6)	15 %
Act. aprendizaje activo (1)	25 %
Total	100 %

De conformidad con el artículo 78 del Reglamento del Régimen Enseñanza-Aprendizaje del Instituto Tecnológico de Costa Rica y sus Reformas, en este curso la persona estudiante **no** tiene derecho a presentar un examen de reposición.

7. Bibliografía

- [1] P. A. Laplante, *Real-Time Systems Design and Analysis*, 4th. Wiley, 2011, ISBN: 9780470768648.
- [2] N. Storey, *Safety-Critical Computer Systems*. Addison-Wesley, 1996, ISBN: 9780201178199.
- [3] M. Wolf, *Computers as Components: Principles of Embedded Computing System Design*, 3rd. Morgan Kaufmann, 2012, ISBN: 9780123884367.
- [4] International Organization for Standardization, *ISO 26262: Road vehicles – Functional safety*, Available at <https://www.iso.org/standard/68383.html>, 2018.
- [5] RTCA, Inc. and EUROCAE, *DO-178C: Software Considerations in Airborne Systems and Equipment Certification*, Available at <https://www.rtca.org>, 2011.
- [6] International Electrotechnical Commission, *IEC 61508: Functional safety of electrical/electronic/programmable electronic safety-related systems*, Available at <https://webstore.iec.ch/publication/22273>, 2010.

8. Persona docente

El curso será impartido por:

Juan José Montero Jimenez, Ph.D.

Doctorado en Ingeniería Industrial e Informática. Universidad de Toulouse. Francia.

Máster en ciencias en Ingeniería Aeroespacial. ISAE-SUPAERO. Francia.

Licenciado en Ingeniería en Mantenimiento Industrial. Instituto Tecnológico de Costa Rica. Costa Rica

Correo: juan.montero@itcr.ac.cr Teléfono: 25509338

Oficina: 5 Escuela: Ingeniería Electromecánica Sede: Cartago