



Nombre de la materia :	<b>Programación en Tiempo Real</b>
Clave:	<b>CI0010-T</b>
No. De horas /semana :	<b>3</b>
Duración semanas:	<b>16</b>
Total de Horas :	<b>48</b>
No. De créditos :	<b>6</b>
Prerrequisitos :	<b>CI0000-T y CI0400-T</b>

### **Objetivo**

Que el estudiante identifique qué es un sistema de tiempo real, así como las restricciones que la programación de estos sistemas conlleva. Que conozca una plataforma de programación en tiempo real y la práctica en un sistema operativo de tiempo real.

Esta materia tiene aportaciones en el perfil del Ingeniero en Computación mediante el desarrollo de las siguientes competencias:

### **Competencias disciplinares**

- Identifica por sus características a los sistemas de tiempo real.
- Analiza las capacidades de los sistemas de tiempo real en aplicaciones como: multimedia, realidad aumentada, internet de las cosas (IoT), automatización y control, entre otras.
- Resuelve problemas usando metodología determinista y con restricción de tiempo.
- Aplica la metodología adecuada para realizar planificación basada en prioridades.
- Propone soluciones a problemas utilizando programación concurrente.
- Distingue los sistemas operativos de tiempo real de los sistemas operativos convencionales.
- Diseña y construye prototipos simples para la resolución de problemas.
- Argumenta la solución obtenida para un problema y la expresa de forma verbal, de manera precisa, coherente y creativa.
- Argumenta la solución obtenida para un problema y la expresa de forma escrita, mediante la producción de textos con base en el lenguaje científico.
- Identifica y relaciona las aplicaciones de los sistemas de tiempo real para resolver problemas que tengan impacto de transformación social y sostenibilidad.

### **Competencias genéricas**

- Participa en la solución analítica y práctica de problemas relacionados con su área de especialidad.
- Identifica, ordena e interpreta las ideas, datos y conceptos explícitos e implícitos en un texto.
- Utiliza las tecnologías de la información y comunicación para producir diversos materiales de estudio y productos, e incrementar sus posibilidades de formación.



- Aplica una disciplina de autoaprendizaje que le permite buscar soluciones a los problemas que se le presentan.
- Plantea la solución de problemas usando las herramientas de trabajo disponibles.
- Participa de manera efectiva en trabajos colaborativos diversos.

### Contenido sintético

Tema	Duración Horas
1. Introducción a los sistemas en tiempo real	4
Actividad 1	
2. Recursos de los sistemas en tiempo real	12
Actividad 2	
Actividad 3	
Examen Parcial 1	1
3. Sincronización	12
Actividad 4	
Actividad 5	
4. Planificación	6
Actividad 6	
Examen Parcial 2	2
5. Sistemas Operativos en Tiempo Real	4
Actividad 7	
6. Aplicaciones y prácticas de programación	7
7. Proyecto final (Examen final)	
Total	48

### Bibliografía básica

Burns, A.; Wellings, A., Sistemas de Tiempo Real y Lenguajes de Programación, Universidad de York, Addison Wesley, 2003

Abbott, D., Linux for Embedded and Real-time Applications, Second Edition, Prentice Hall, 2006

### Bibliografía complementaria

Buttazzo, G.; Hard Real-Time Computing Systems, Kluwer, 1997

Gallmeister, B. O., POSIX 4: Programming for the Real World. O'Reilly & Associates Inc. 2000.

### Manuales en línea para Arduino y Raspberry Pi:

<https://www.arduino.cc/>

<https://www.raspberrypi.org/>



### Otros enlaces complementarios:

<https://wiki.linuxfoundation.org/realtime/start>  
<http://www.oracle.com/technetwork/java/index.html>  
<https://developer.xamarin.com/guides/>  
<https://developer.android.com/studio/index.html?hl=es-419>  
<https://www.osadl.org/Single-View.111+M5c03315dc57.0.html>

### Metodología de enseñanza-aprendizaje:

Revisión de conceptos, análisis y solución de problemas en clase:	( X )
Lectura de material fuera de clase:	( X )
Ejercicios fuera de clase (tareas):	( X )
Investigación documental:	( X )
Elaboración de reportes técnicos o proyectos:	( X )
Prácticas de laboratorio en una materia asociada:	( )
Visitas a la industria:	( )

### Metodología de evaluación:

Asistencia:	
Tareas:	60%
Elaboración de reportes técnicos o proyectos:	20%
Exámenes de Academia o Departamentales	20%

### Contenido desarrollado

Temas	Duración Horas
1 Introducción a los sistemas en tiempo real	4
1.1 Qué es un sistema en tiempo real	1
1.2 Ejemplos de sistemas en tiempo real	1
1.3 Clasificación de sistemas en tiempo real	1
1.4 Restricciones de la programación en tiempo real	1
Actividad 1	
2 Recursos en sistemas de tiempo real	12
2.1 Identificación y gestión de recursos. Bloqueos y latencias	1
2.2 Modelo de Tarea simple	1
2.3 Tiempo. Funciones para manejo de tiempo	3
2.4 Acciones atómicas	1
2.5 Programación concurrente	6
Actividad 2	
Actividad 3	



Examen Parcial 1	1
3 Sincronización y comunicación	12
3.1 Variables compartidas.	2
3.2 Regiones críticas	1
3.3 Semáforos	3
3.4 Exclusión mutua	3
3.5 Paso de mensajes	3
Actividad 4	
Actividad 5	
4 Planificación	6
4.1 Planificación basada en prioridades	4
Modelo de tarea simple.	
Planificación estática y dinámica	
Algoritmo monotónico de frecuencia	
Herencia de prioridad	
4.2 Planificación basada en tiempos de respuesta	2
Test de planificabilidad basada en utilización	
Análisis de tiempo de respuesta	
Ejecución en el peor caso	
Actividad 6	
Examen Parcial 2	2
5 Sistemas Operativos en Tiempo Real	4
5.1 Diferencias entre sistemas operativos en tiempo real y convencionales	1
Estrategias en Linux para ejecución en tiempo real	
5.2 RT Linux	3
Actividad 7	
6 Aplicaciones y prácticas de programación	7
6.1 Lenguajes para tiempo real (C y C++ con POSIX. JavaRT, C#, ADA)	1
6.2 Temporizadores y señales	1
6.3 Interrupciones	1
6.4 Semáforos	1
6.5 Programación de tareas periódicas	1
6.6 Comunicación entre procesos	1
6.7 Aplicación para RTLinux	1
7 Proyecto Final (Examen final)	



Proyectos sugeridos:

- Programación de dispositivos embebidos (Raspberry Pi / Arduino).
- Programación de aplicaciones para dispositivos móviles (Android)
- Aplicaciones para IoT.
- SCADA + Arduino

NOTA: se sugiere que los proyectos finales resuelvan problemas reales, y que logren un impacto positivo para algún sector de la población o bien que se relacionen con la sostenibilidad ambiental.

**Programa propuesto por M.I. Rosalía Mora Juárez**

**Fecha de autorización por el H. Consejo Técnico: dd/mm/aaaa**

**Comentarios para la Academia y el H. Consejo Técnico:**

- El programa que aparece actualmente en la página de la FIE fue modificado hace más de 2 años, pero nunca se actualizó en la página. El presente documento es una nueva propuesta.
- Se sugiere incluir como pre-requisito las materias de Sistemas Operativos y Control Analógico 1, dado que se espera que los estudiantes tengan la capacidad de trabajar un proyecto de control con sistemas embebidos.
- Esta propuesta de programa se basa en el Modelo Educativo Nicolaita. Está pensado para trabajar por competencias (disciplinares y genéricas). Los temas se evalúan también por medio de rúbricas y en los porcentajes de evaluación se da más peso a las tareas o actividades, con esto se pretende que los estudiantes no memoricen, sino que desarrollen capacidad para resolver problemas de forma práctica y que aporten productos.
- Se incluye dos exámenes teóricos (los que indica el reglamento) y se tomarán como examen de academia.
- Para tener derecho a examen final se aplica el reglamento, por lo que el estudiante debe cumplir con el 80% de asistencias.
- Hay semestres en que no se puede cumplir con las 16 semanas programadas (debido a paros, tomas, huelgas, etc.) lo cual implica que se tengan que recortar temas y no se cumplía con el 100% del programa. Por esa razón, y para no comprometer el objetivo del curso, se proponen los siguientes cambios:

Programa anterior	Propuesta de modificación
-------------------	---------------------------



(página de la FIE)	
<ol style="list-style-type: none"><li>2. Restricciones de la programación en tiempo real. (2 horas)<ol style="list-style-type: none"><li>1. Tiempo.</li><li>2. Recursos físicos.</li><li>3. Bloqueos y latencia.</li></ol></li><li>6. Manejo de recursos. (6 horas)<ol style="list-style-type: none"><li>1. Identificación de recursos.</li><li>2. Programación concurrente.</li><li>3. Acciones atómicas.</li><li>4. Gestión de recursos.</li></ol></li></ol>	Se reordenan los temas 2 y 6 del programa antiguo. Esto se hace para dar más coherencia en el avance de temas y evitar temas redundantes.
<ol style="list-style-type: none"><li>9. Confiabilidad. (6 horas)<ol style="list-style-type: none"><li>1. Fallas y defectos.</li><li>2. Prevención y tolerancia.</li><li>3. N-versiones.</li><li>4. Excepciones.</li></ol></li></ol>	Se eliminó el tema 9 (Confiabilidad) debido a que éstos conceptos ya se tratan en otros cursos. Por lo general el tema 9 no se alcanza a cubrir. Se elimina ya que no compromete el objetivo del curso.

- Se redistribuyeron las horas de cada tema, de acuerdo a la experiencia de impartición del curso en los últimos semestres.
- Las siete actividades propuestas son problemas que el estudiante debe analizar para después desarrollar la programación adecuada y lograr la solución. Estas actividades son el equivalente a las “tareas” clásicas y se efectuarán extraclase, por lo que no se indican sus tiempos.
- Los reportes de actividades y prácticas se trabajan como productos y se integran en un portafolio de evidencias.
- El proyecto final equivale a un examen final práctico.
- Se sugiere que los proyectos finales resuelvan problemas reales, y que logren un impacto positivo para algún sector de la población o bien que se relacionen con la sostenibilidad ambiental.

**Aprobación por el H. Consejo Técnico de la FIE: 10 de abril de 2018**