

Carrera INGENIERIA EN INFORMATICA		
Asignatura 3660 - Sistemas Operativos Avanzados		
Trayecto Infraestructura		
Año académico 2023		
Responsables Ing. Esteban Carnuccio		
Carga horaria semanal 4 hs	Carga horaria total 64 hs	Créditos -----
Modalidad: Presencial		
Correlativas anteriores VIRTUALIZACIÓN DE HARDWARE	Correlativas posteriores PROYECTO FINAL DE CARRERA	
Conocimientos necesarios: -----		

Descripción de la asignatura

La asignatura Sistemas Operativos Avanzados está planteada como una materia que aporta conocimientos de sistemas operativos con diferentes orientaciones, de las que se ejecutan en computadoras tradicionales, a saber: Sistemas Operativos de tiempo real (SOTR), Embebidos, Móviles y Arquitecturas Reconfigurables.

Para ello se basa en el análisis y adquisición de conceptos sobre: las arquitecturas, requerimientos, servicios y posterior aplicación en casos de uso. De esta manera, se pretende brindar herramientas y saberes prácticos, que puedan ser utilizados en el ámbito profesional.

Metodología de enseñanza

Los temas que abarca la asignatura son dictados en forma iterativa e incremental. Esto se debe a que se instruye a los estudiantes con los conceptos elementales de cada uno de ellos, para que luego los apliquen en forma práctica. Para ello la cátedra propondrá dos trabajos prácticos, con propuestas generadas por parte de los docentes y que cada grupo de trabajo adaptará según sus intereses. Esta forma de trabajo hará que el alumno adquiera una sólida formación de habilidades para resolver problemas en el diseño de soluciones a clientes, junto a la exposición oral y escrita fomentando el trabajo en equipo y la discusión participativa.

Se promueve la investigación de algunas partes del trabajo en sitios sugeridos por la cátedra, para generar habilidades que son requeridas para su desarrollo como profesionales. Para aquellos alumnos que lo necesiten, pueden concurrir al laboratorio en horario extendido, donde siempre se encuentran docentes para consulta o para permitir el uso del material del laboratorio, tales como placas de desarrollo, sensores, actuadores, Notebooks para programación, la red conectada para trabajar con los servidores y libros para su consulta.

En la materia se utilizan diferentes herramientas tecnológicas pedagógicas de enseñanza. A través de material audiovisual y grabación de clases, lo que permite al alumno consultar lo enseñado en cualquier momento. Al mismo tiempo, se emplean las utilidades del sistema de gestión de aprendizaje propuesto por la universidad (MIEI), para que los alumnos puedan realizar consultas y entregas de trabajos.

Objetivos de aprendizaje

- Describir la arquitectura básica de un Sistema Operativo en Tiempo Real, justificando su importancia en las aplicaciones de ingeniería.
- Emplear herramientas que permitan manejar microcontroladores, conectividad, sistemas operativos embebidos y su relación con los sistemas de Tiempo Real
- Comprender y utilizar herramientas de desarrollo de aplicaciones en sistemas operativos móviles, que permitan la interconexión con otros sistemas
- Aprender sobre el funcionamiento de una Arquitectura IoT para su utilización
- Conocer y entender el desarrollo de software orientado a servicios y microservicios, sus problemas, sus soluciones y técnicas actuales, así como su impacto en la sociedad de la información.
- Reconocer el funcionamiento de distintos protocolos de intercambio de datos entre diferentes sistemas
- Comprender los conceptos relacionados con las arquitecturas de computadoras reconfigurables modernas, su uso y aplicación.

Contenidos mínimos

Sistemas Operativos de Tiempo Real, Sistemas Embebidos, Móviles y Portables. Programación concurrente en Sistemas Embebidos. Concepto de Arquitectura IoT. Concepto de Arquitecturas Reconfigurables. Concepto de Arquitectura Basadas en Servicios. Concepto de Microservicios.

Competencias a desarrollar

Genéricas

- Utilización de técnicas y herramientas de aplicación en la ingeniería en sistemas de información / informática.
- Desempeño en equipos de trabajo.
- Comunicación efectiva.
- Actuación profesional ética y responsable.
- Aprendizaje continuo.
- Identificación, formulación y resolución de problemas de ingeniería en sistemas de información/informática.
- Generación de desarrollos tecnológicos y/o innovaciones tecnológicas.
- Desarrollo de una actitud profesional emprendedora.
- Concepción, diseño y desarrollo de proyectos de ingeniería en sistemas de información / informática.

Específicas

- Especificación, proyecto y desarrollo de software.
- Establecimiento de métricas y normas de calidad de software.

Programa analítico

Unidad 1	Sistemas Operativos de tiempo real. Introducción a los SO de TR. Requisitos de los SO de TR. Características de los SO de TR. La arquitectura del Kernel. Políticas de planificación para tiempo real. Protocolos de sincronización de tiempo real. Las prioridades de los SOTR. La Gestión de las Tareas. El planificador y el controlador de interrupciones del reloj de Tiempo Real. Ejemplo de SO de TR. Introducción a la programación sobre un Sistema Operativo de Tiempo Real
Unidad 2	Sistemas Operativos embebidos. Concepto y características de sistemas embebidos. Estructura y componentes de un sistema embebido. Plataformas. Placas de desarrollo. Dispositivos de Entrada/Salida, Arquitectura, Procesadores, Mecanismos de comunicación (Ethernet, Wifi, Bluetooth, GSM, LTE, NFC, RFID, ZeeBig – Xbee, Lorawan, Sigfox, NB-IoT). Sensores y Actuadores. Aplicaciones en sistemas embebidos. Programación y depuración. Concepto de Bar Metal y Firmware. Prototipado y Producto Final.
Unidad 3	Internet de Las Cosas. Evolución de los sistemas embebidos hacia IoT. Concepto de Arquitectura Grid y Cluster. Concepto de M2M. Arquitectura básica de IoT. IoT en la Nube. Arquitectura en Capas. Concepto de Mist Computing, Edge Computing y Fog Computing. Utilización de servicios Cloud con IoT. Concepto de IIoT. Industria 4.0
Unidad 4	Sistemas Operativos Móviles. Dispositivos Móviles (celulares, GPS, Notebook, Tablets, etc.) y Portables (Smartwatch, Smartband, etc). Plataformas propietarias y Plataformas abiertas. SDK (Software Development Kit), licencias y lenguajes de programación para las plataformas. Sistema Operativo Android, componentes básicos del Framework de Android, uso del entorno de desarrollo. Implementación de una aplicación en un dispositivo físico o simulado. Servicios del Sistema Operativo. Uso de sensores, persistencia de datos. Mecanismos de comunicación (Bluetooth, Wifi, GSM, NFC). Geolocalización.
Unidad 5	Concurrencia en Sistemas Embebidos. Concurrencia en Sistemas Operativos de tiempo Real. Concurrencia en placas de prototipado. Concurrencia en Sistema Operativos Móviles. Creación de Procesos e Hilos. Sincronización entre Procesos e Hilos. Comunicación entre Procesos e Hilos.
Unidad 6	Concepto de Arquitecturas Basadas en Servicios. Concepto y estándares de SOA (Service Oriented Architecture). Concepto de Microservicios. Conceptos y protocolos básicos de servicios Web (Web Services). Composición y coordinación de servicios Web. Protocolos de intercambio de datos (XML, JSON, REST, SOAP, Productor/Consumidor (MQTT, Kafka)). Herramientas Cloud (Notificaciones Push, base de datos, etc). Ejemplos prácticos. Desarrollo de aplicaciones.
Unidad 7	Concepto de Arquitecturas Reconfigurables. Los dispositivos reconfigurables (FPGAs - Field Programmable Gate Array). Lenguajes de descripción hardware, hardware reconfigurable, programación (VHDL y Hardware Description Language.) y configuración de dispositivos reconfigurables. Herramientas de diseño. Técnicas de compilación para arquitecturas reconfigurables. Aplicaciones específicas de dispositivos reconfigurables (criptografía, compresión, procesamiento de video, robótica, comunicaciones, teléfonos móviles, DVD, videocámaras, PCs,

	impresoras, Routers, Gateways, PDAs, etc.). Arquitecturas Reconfigurables en IoT.
--	---

Planificación de actividades					
Semana	Clase	Actividad Detalle de la actividad a desarrollar	Tipo (indicar el tipo de actividad a desarrollar: teoría, practica, practica de laboratorio, trabajo de campo, otra)	Duración estimada	Unidad
Semana 1	Clase 1	Teoría Tiempo Real	Teoría	4 hrs.	1
Semana 2	Clase 2	Introducción a Sistemas Embebidos	Teoría/Práctica	4 hrs.	1 y 2
Semana 3	Clase 3	Teoría S.E.	Teoría	4 hrs.	2
Semana 4	Clase 4	Teoría S.E.	Teoría/Práctica	4 hrs.	2
Semana 5	Clase 5	Concurrencia S.E.	Teoría/Práctica	4 hrs.	5
Semana 6	Clase 6	Clase IoT	Teoría/Práctica	4 hrs.	3
Semana 7	Clase 7	1° Parcial		4 hrs.	
Semana 8	Clase 8	Dispositivos Móviles	Teoría/Práctica	4 hrs.	4
Semana 9	Clase 9	Dispositivos Móviles	Teoría/Práctica	4 hrs.	4
Semana 10	Clase 10	Dispositivos Móviles	Práctica	4 hrs.	5
Semana 11	Clase 11	Arquitectura SOA/Microservicios	Teoría/Práctica	4 hrs.	6
Semana 12	Clase 12	Arquitectura Reconfigurables	Teoría	4 hrs.	7
Semana 13	Clase 13	2° Parcial		4 hrs.	
Semana 14	Clase 14	Coloquio		4 hrs.	
Semana 15	Clase 15	Recuperatorio		4 hrs.	

Experiencias de laboratorio/Taller/Trabajos de Campo

Las experiencias de laboratorio están dadas por la confección de los trabajos prácticos. Con un punto extra en donde se busca que los estudiantes realicen la integración entre estos. El primer trabajo consiste en el diseño y la construcción de un sistema embebido, en donde los estudiantes traten de resolver una problemática de la vida real. Para ello deberán emplear diferentes mecanismos, para poder interactuar con el entorno utilizando sensores y actuadores. Por otra parte, el segundo trabajo práctico, consiste en desarrollar una aplicación para un dispositivo móvil. Dicho programa deberá hacer uso de distintos recursos y herramientas que ofrece esta plataforma, utilizando los conocimientos adquiridos en IoT durante la cursada. Los alumnos deberán integrar ambos trabajos de manera tal que puedan

intercambiar datos entre estos. Utilizando servicios en la nube, servidores externos u otro mecanismo de comunicación.

Metodología de Evaluación

- Esta asignatura se evaluará de acuerdo con la reglamentación vigente en la Universidad y la que se detalla en el "Reglamento de cursado y aprobación de la materia", dos evaluaciones parciales y un recuperatorio.
- En el primer parcial se evaluarán las Unidades 1, 2 y 3. Mientras que el segundo parcial se evaluarán las Unidades 4, 5, 6 y 7. El método de evaluación será escrito teórico/práctico de forma presencial. Se evaluarán las competencias mencionadas mediante enunciados específicos para su resolución. La devolución se realizará en la clase siguiente al examen explicando las respuestas correctas a los puntos evaluados.
- El recuperatorio será escrito teórico/práctico, en forma presencial, no integrador de forma equivalente a los exámenes parciales. En el cual el estudiante podrá rendir nuevamente el primer o segundo parcial.
- Para estas evaluaciones se constituirá un tribunal evaluador compuesto por el docente del curso y dos ayudantes que serán consultores.
- Los trabajos prácticos contribuyen a la comprensión de los temas a evaluar en los exámenes parciales, pero son de cumplimiento no obligatorio.
- Además, se requiere una asistencia a clase no inferior al 75%, se hará un seguimiento del trabajo realizado por cada integrante de grupo en clase.

Aprobación y cursada

A los fines de la aprobación de la materia, se considera "la última nota obtenida" en cada uno de los 2 exámenes rendidos (en primera instancia o recuperatorio).

- a) Por régimen de promoción, sin examen final, se considera la materia aprobada, cuando la calificación es igual o superior a 7 (siete) a través de exámenes parciales y recuperatorios, en las fechas indicadas en el cronograma.
- b) Si el alumno no llena los requisitos para promover (calificación superior o igual a 4 pero inferior a 7 puntos, queda en condición de cursada. Para su aprobación definitiva tendrá que rendir posteriormente un examen final. La validez de la cursada será de 5 turnos consecutivos de exámenes finales. Dichos turnos serán contados a partir del turno inmediato siguiente al periodo de cursada.
- c) El alumno que suma 3 (tres) aplazos en las evaluaciones y/o recuperatorio, pierde la materia y se considera desaprobado.

Examen Final

Los alumnos pueden rendir examen final bajo dos modalidades regular o libre.

- Para rendir examen como regular deberá tener la materia cursada y no haberse cumplido su vencimiento. Deberán rendir como regular los alumnos que obtengan entre cuatro y seis en los parciales o sus recuperatorios.
- Para rendir examen como libre tendrán que realizar previamente los trabajos prácticos propuestos en el último ciclo lectivo como complemento, rigiendo las mismas condiciones que en la cursada y es eliminatorio, para ello tendrá que comunicarse con el jefe de cátedra, con una anterioridad aproximada de 30 días, a la fecha programada por el Departamento de Ingeniería para el examen final, para establecer fecha y hora de entrega del tema y posterior evaluación.

La mesa examinadora se constituirá con al menos 2 docentes y considerará válidas las inscripciones que consten en las actas proporcionadas por la oficina de alumnos. Cada alumno rendirá el examen final con el programa vigente.

La modalidad del examen final contempla una exposición oral y/o escrita

Primera evaluación	Semana 7	Escrita	Duración: 2hrs, horario: 19 hrs
Segunda evaluación	Semana 13	Escrita	Duración: 2hrs, horario: 19 hrs
Recuperatorio	Semana 15	Escrita	Duración: 2hrs, horario: 19 hrs

Bibliografía obligatoria

Título	Autor	Editorial	Edición	Año
Operating System Concepts	Abraham Silberschatz, Peter Galvin, Greg Gagne	Wiley	7° Edición	2006
Operating Systems Internals And Design Principles	William Stallings	Prentice-Hall	6° Edición	2008
Sistemas de tiempo real : introducción al diseño	Gallina, Sergio H.	ACSE,	Primera	2016
CASE2015, Libro de trabajos.	Brengi D., Lutemberg A.	ACSE	Quinta	2015

Bibliografía complementaria recomendada

Título	Autor	Editorial	Edición	Año
Fundamentos de lógica digital con diseño VHDL	Brown, Stephen	McGraw-Hill	2da	2006

Sistemas Operativos Distribuidos	Tanenbaum A.	Prentice-Hall Hispanoamericana	1era	1993
Hard Real-Time Computing Systems	Giorgio C Buttazzo	Springer	2° Edición	2010

Otros recursos obligatorios (videos, enlaces, otros)	
Material y Apuntes de la cátedra	http://www.soa-unlam.com.ar/

Otros recursos complementarios (videos, enlaces, otros)	
Guía de desarrolladores de Android	https://developer.android.com/guide?hl=es-419
Introducción a FPGA	https://getmyuni.azureedge.net/assets/main/study-material/notes/electronics-communication-engineering-microelectronics-vlsi-design-introduction-to-fpga-notes.pdf
Arquitectura SOA	http://repositorio.utn.edu.ec/bitstream/123456789/587/1/CAPITULO S.pdf
Microservicios y Arquitectura REST	https://aws.amazon.com/es/microservices/ https://learn.microsoft.com/es-es/azure/architecture/microservices/
Desarrollo de Sistemas Embebidos	https://www.google.com/guíametodologicaparasudesarrollo http://www.ieec.uned.es/investigacion/Dipseil/PAC/archivos/Informacion de referencia ISE5 3 1.pdf
FreeRTOS	https://freertos.org/Documentation/161204 Mastering the FreeRTOS Real Time Kernel-A Hands-On Tutorial Guide.pdf
Sistemas Operativos Embebidos	https://www.researchgate.net/publication/365141635 Embedded Operating Systems/fulltext/6366770e37878b3e878515a0/Embedded-Operating-Systems.pdf
Internet de las Cosas	https://upcommons.upc.edu/bitstream/handle/2117/100921/LM08 R ES.pdf https://openaccess.uoc.edu/bitstream/10609/91207/7/mmedinabarTFM0119memoria.pdf
Computación Wearable	https://www.acta.es/medios/articulos/ciencias_y_tecnologia/041001.pdf
Programación Android Wear	https://developer.android.com/training/wearables?hl=es-419
Protocolo REST	https://ingenieria.upana.edu.gt/wp-content/uploads/2020/07/Edutek-dia-5-RESTful-api-RESTCONF2.pdf