

<b>Facultad de Ingeniería - Universidad Nacional de Cuyo</b>			
<b>P1- PROGRAMA DE ASIGNATURA</b>			
<b>Asignatura:</b>	<b>Sistemas Embebidos</b>		
<b>Profesor Titular:</b>	<b>Dr. Ing. Pablo Daniel Godoy</b>		
<b>Carrera:</b>	<b>Licenciatura en Ciencias de la Computación</b>		
<b>Año: 2023</b>	<b>Semestre: 9°</b>	<b>Horas Semestre: 80</b>	<b>Horas Semana: 5</b>

### ***DESCRIPCIÓN DE LA ASIGNATURA***

#### **Ubicación de la asignatura dentro de los diseños curriculares**

Los conocimientos que el Licenciado en Ciencias de la Computación debe adquirir durante su formación son suministrados en asignaturas se agrupan en bloques o áreas. La asignatura Sistemas Embebidos forma parte del área de Espacios Curriculares Optativos, cuyo objetivo es favorecer la formación integral del estudiante, enfatizar algún aspecto de la profesión y complementar la formación profesional.

Sistemas Embebidos se cursa en quinto año de la carrera, por lo cual los estudiantes poseen sólidos conocimientos de arquitecturas de computadoras, redes de computadoras, sistemas operativos, paradigmas de programación, algoritmos y estructuras de datos. En la asignatura sistemas embebidos, los estudiantes aplican dichos conocimientos para programar sistemas embebidos e integrarlos a otros sistemas de computación. Estos conocimientos permitirán a los estudiantes implementar aplicaciones de IoT completas.

#### **Objetivos**

Los objetivos de la asignatura Sistemas Embebidos se enuncian en base a la Ordenanza 40-2017, emitida por el Consejo Superior de la Universidad Nacional de Cuyo, que enumera las áreas de trabajo de los profesionales en Ciencias de la Computación, las expectativas de logros y contenidos mínimos para la asignatura Sistemas Embebidos.

En base a dichas ordenanzas, los objetivos de la asignatura son:

- Analizar las características de las diferentes arquitecturas, identificando los sistemas embebidos disponibles en la actualidad y sus características.
- Implementar protocolos específicos para comunicación y adquisición de datos.
- Identificar las herramientas para programar y depurar aplicaciones.
- Programar aplicaciones específicas para la integración de los sistemas embebidos.
- Crear software que haga uso de los servicios ofrecidos por los sistemas operativos en tiempo real para sistemas embebidos.
- Integrar sistemas embebidos a aplicaciones de IoT, aplicaciones web y aplicaciones en la nube.

## **CONTENIDOS**

### **Unidad 1: Introducción a sistemas embebidos**

**1.A** Arquitecturas de sistemas embebidos y microcontroladores. Componentes típicos. Características de arquitecturas de plataformas de hardware de sistemas embebidos. Tipos de sistemas embebidos. Ejemplos. Áreas de aplicación de sistemas embebidos.

**1.B** Programación de microcontroladores en lenguaje C y Python. Entornos de desarrollo. Integración con aplicaciones web y aplicaciones en la nube.

### **Unidad 2: Comunicación con el exterior**

**2.A** Puertos digitales. Conversores ADC y DAC. Comparadores analógicos. Ejemplos con microcontroladores basados en procesadores AVR y ARM.

**2.B** Protocolos de comunicación de uso común en sistemas embebidos cableados e inalámbricos: USB, I<sup>2</sup>C, SPI, UART y USART, CAN, RFID, NFC, ZigBee y LoRa. Comunicación serial con una computadora.

### **Unidad 3: Sistemas operativos para sistemas embebidos**

**3.A** Sistemas operativos para sistemas embebidos. Sistemas operativos en tiempo real. Características especiales. Ejemplo: FreeRTOS.

**3.B** Creación de aplicaciones en diferentes lenguajes (C, Python) que hagan uso de los servicios ofrecidos por los sistemas operativos para sistemas embebidos.

### **Unidad 4: Periféricos internos y funciones especiales**

**4.A** Timers y contadores de eventos. Watchdog. Interrupciones. Mecanismos de Reset.

**4.B** Técnicas para reducir el consumo de energía. Mecanismos de programación y debugging.

**4.C** Organización de memoria de sistemas embebidos. Almacenamiento no volátil.

**4.D** Creación de aplicaciones que hagan uso de periféricos propios de los sistemas embebidos en diferentes lenguajes de programación (C, Python).

### **Unidad 5: Introducción a IoT**

**5.B** Aplicaciones de IoT que emplean sistemas embebidos como parte de su arquitectura.

## **TRABAJOS PRÁCTICOS**

**Trabajo Práctico N°1:** GPIO. Conversores AD y DA. Comunicación con sistemas de computación y aplicaciones web.

**Trabajo Práctico N°2:** Protocolos de comunicación inalámbrica de bajo consumo.

**Trabajo Práctico N°3:** Sistemas operativos para sistemas embebidos. FreeRTOS.

**Trabajo Práctico N°4:** Temporizadores. Interrupciones. Memoria no volátil. Modos de bajo consumo de energía.

**Trabajo Práctico N°5:** Integración con aplicaciones en la nube e IoT.

### ***Trabajo Práctico integrador***

Consistirá en un trabajo práctico de implementación o investigación. Deberá integrar varias unidades de la asignatura. Los estudiantes deberán presentar un anteproyecto por escrito que deberá ser aprobado por los profesores de la cátedra.

En caso de desarrollar un trabajo de implementación, se deberá presentar el prototipo funcionando correctamente. En caso de desarrollar un trabajo de investigación, el mismo deberá tener formato de artículo científico.

Se reservarán al menos 10 horas de la carga horaria de la asignatura para que los estudiantes desarrollen sus trabajos prácticos integradores.

### ***METODOLOGÍA DE ENSEÑANZA***

#### **Estrategias metodológicas:**

El desarrollo de la asignatura incluye clases teóricas expositivas y trabajos prácticos de laboratorio. Durante las clases teóricas expositivas se introducirá a los estudiantes en los temas de cada una de las unidades del programa de estudio. Se integrarán y relacionarán los conocimientos de cada unidad, como también los conocimientos adquiridos en otras asignaturas. Se utilizarán soportes y material multimedia para la presentación de conceptos, gráficos, esquemas e imágenes y sistemas embebidos provistos por el docente a cargo de la asignatura. Se destinarán como máximo el 30% de la carga horaria semanal para el dictado de las clases teóricas.

El estudiante ampliará los conocimientos adquiridos en las clases teóricas mediante el estudio de hojas de datos y manuales de uso de los dispositivos y componentes de software que utilizará durante la realización de los trabajos prácticos de laboratorio.

Se destinará como mínimo el 70% de la carga horaria semanal para la realización de los trabajos prácticos de laboratorio. Los mismos requerirán a los estudiantes implementar aplicaciones que den solución a problemas planteados en las guías de los trabajos prácticos. Estas soluciones deberán ser completas, incluyendo el desarrollo del software y configuración de los sistemas embebidos, la integración con un sistema de cómputo y desarrollo de una aplicación web que permita mostrar los resultados e interactuar con los sistemas embebidos de forma local y remota.

Los problemas a resolver en los trabajos prácticos serán tales que admitan diferentes tipos de soluciones. Se recomendarán diferentes herramientas para la implementación de las soluciones. Se dejará al estudiante elegir las herramientas a utilizar.

<b>Actividad</b>	<b>Carga horaria</b>
Dictado de clases teóricas expositivas, presentación de ejemplos prácticos y resolución de ejercicios simples.	24
Formación práctica	

Formación Experimental – Laboratorio.	
Formación Experimental – Resolución de problemas de ingeniería.	46
Proyecto y diseño (Trabajo Práctico integrador)	10
<b>Total</b>	<b>80</b>

### **Recursos y materiales:**

Para la realización de los trabajos prácticos, los estudiantes tendrán acceso a los laboratorios de computación de la facultad de Ingeniería. El profesor a cargo de la asignatura se encargará de que los laboratorios se encuentren equipados con los elementos de hardware y software necesarios para la realización de los trabajos prácticos.

El profesor a cargo de la asignatura pondrá a disposición de los estudiantes equipamiento que utiliza en sus tareas de investigación científica, adquiridos mediante proyectos de investigación financiados por la Universidad Nacional de Cuyo, La Agencia Nacional de Promoción Científica y Tecnológica, y otros organismos de ciencia e investigación o empresas. Este equipamiento incluye:

- Placas de desarrollo Arduino UNO.
- Computadoras Raspberry Pi 3 y 4.
- Transceivers LoRa y XBee
- Diferentes placas de desarrollo, componentes electrónicos y sensores.

Los estudiantes tendrán acceso sin restricciones a todos estos dispositivos, pudiendo configurarlos, instalar aplicaciones, instalar y configurar sistemas operativos, conectar dispositivos externos a sus puertos o pines, etc.

Las herramientas de software a utilizar serán herramientas de software libre.

### **EVALUACIONES**

El sistema de evaluación, escalas de calificaciones, normas y pautas a aplicar en cada instancia de evaluación serán regidas por la Ordenanza 108/10 CS de la Universidad Nacional de Cuyo.

### **Sistema de acreditación:**

La asignatura podrá aprobarse por alguno de los siguientes métodos:

- Por examen final en condición regular o en condición libre.
- Por promoción directa.

### **Condiciones para obtener la Regularidad:**

- Realizar y aprobar todos los trabajos prácticos de la asignatura. Las condiciones de aprobación de cada trabajo práctico se enuncian en las guías de los mismos. Las condiciones excluyentes para la aprobación de los trabajos prácticos serán: i) presentar la solución a los problemas planteados funcionando correctamente y

cumpliendo las restricciones de operación indicadas en las guías de trabajos prácticos y ii) presentar un informe. Además, el estudiante deberá exponer la solución implementada al resto de los estudiantes.

- Realizar y aprobar el trabajo práctico integrador.

### **Condiciones para obtener la Aprobación de la asignatura mediante Promoción Directa:**

- Cumplir con las condiciones para obtener la regularidad de la asignatura.
- Aprobar un examen global integrador con nota igual o superior a 6. El mismo tendrá el mismo formato que el examen final en condición regular. Dicho examen se rendirá la última semana de clases, de acuerdo al calendario académico de la Facultad de Ingeniería. La nota final se computará como:  $[Nota\ trabajos\ prácticos] \cdot 0.6 + [Nota\ exámen\ global] \cdot 0.4$ .
- Tener aprobadas las asignaturas correlativas (Resolución CD-2017-RES-108 <https://digesto.ingenieria.uncuyo.edu.ar/regulations/6995>).

### **Examen Final en Condición Regular**

Examen teórico escrito o por plataforma formado por preguntas de opciones múltiples o de respuesta corta.

### **Examen Final en Condición Libre**

El examen final en condición libre estará dividido en dos partes:

- Examen teórico escrito formado por preguntas de opciones múltiples o de respuesta corta.
- Examen práctico oral y de desarrollo. En el mismo se realizarán preguntas sobre cada trabajo práctico. El estudiante deberá demostrar poseer los conocimientos suficientes para dar solución a los problemas planteados en todos los trabajos prácticos. Se realizarán preguntas de implementación, se solicitará la escritura de código, realización de diagramas y tomar decisiones de implementación en base al análisis del contenido de hojas de datos.

## **BIBLIOGRAFIA**

### **Bibliografía básica:**

Autor	Título	Editorial	Año	Ejemplares en biblioteca
Richard Barry	Mastering the FreeRTOS Real Time Kernel	Real Time Engineers Ltd.	2016	*1
Atmel Corporation	8-bit AVR Microcontrollers	Atmel Corporation	2016	*2

	ATmega328/P Datasheet complete			
Reyes Cortés, Fernando	Arduino : aplicaciones en robótica, mecatrónica e ingenierías	Marcombo	2015 (1° ed)	2

\*1 Disponible sin costo online en [https://www.freertos.org/Documentation/RTOS\\_book.html](https://www.freertos.org/Documentation/RTOS_book.html)

\*2 Disponible sin costo online en <https://ww1.microchip.com>

**Bibliografía complementaria:**


Autor	Título	Editorial	Año	Ejemplares en biblioteca
Reyes Cortés, Fernando	Arduino : aplicaciones en robótica, mecatrónica e ingenierías	Marcombo	2015 (1° ed)	2
Tojeiro Calaza, Germán	Taller de Arduino : un enfoque práctico para principiantes	Marcombo	2014 (1° ed)	1
Beiroa Mosquera, Rubén	Aprender Arduino, Electrónica y Programación	Alfaomega	2018	Biblioteca digital CID

Arduino S.r.l. website: <https://www.arduino.cc/>

The Raspberry Pi Foundation website: <https://www.raspberrypi.com/for-home/>

FreeRTOS "Real-time operating system for microcontrollers" website: <https://www.freertos.org/>

23/02/2023



Dr. Ing. Pablo Daniel Godoy

FECHA, FIRMA Y ACLARACIÓN TITULAR DE CÁTEDRA