



Ministerio de Cultura y Educación  
 Universidad Nacional de San Luis  
 Facultad de Ingeniería y Ciencias Agropecuarias  
 Departamento: Ingeniería  
 Area: Electrónica

(Programa del año 2025)  
 (Programa en trámite de aprobación)  
 (Presentado el 18/06/2025 13:54:23)

### I - Oferta Académica

Materia	Carrera	Plan	Año	Período
Computadores Digitales	ING. MECATRÓNICA	Ord	2025	2° cuatrimestre
		22/12 -10/2 2		
Computadoras Digitales	INGENIERÍA ELECTRÓNICA	Ord	2025	2° cuatrimestre
		19/12 -11/2 2		
Electrónica Digital 2	INGENIERÍA ELECTRÓNICA	OCD	2025	2° cuatrimestre
		N° 23/22		

### II - Equipo Docente

Docente	Función	Cargo	Dedicación
ACOSTA, GUILLERMO LUIS	Prof. Responsable	P.Adj Semi	20 Hs
PEÑALOZA, JUAN PABLO	Auxiliar de Práctico	A.1ra Simp	10 Hs

### III - Características del Curso

Credito Horario Semanal				
Teórico/Práctico	Teóricas	Prácticas de Aula	Práct. de lab/ camp/ Resid/ PIP, etc.	Total
6 Hs	Hs	Hs	Hs	6 Hs

Tipificación	Periodo
B - Teoria con prácticas de aula y laboratorio	2° Cuatrimestre

Duración			
Desde	Hasta	Cantidad de Semanas	Cantidad de Horas
04/08/2025	14/11/2025	15	90

### IV - Fundamentación

El curso de Electronica Digital 2 se fundamenta en la necesidad que el alumno de una carrera de grado con orientación en electrónica debe tener el conocimiento y la práctica básica en microprocesadores y computadoras necesarios para desenvolverse en el mundo tecnológico que nos rodea y que crece sin cesar en esa dirección.  
 Se desarrolla en base a la idea que una computadora puede considerarse como una jerarquía de niveles, cada uno de los cuales desempeña una función bien definida.

### V - Objetivos / Resultados de Aprendizaje

Interpretar el funcionamiento y la arquitectura de los microprocesadores y sus partes para comprender tecnológicamente

dispositivos y sistemas actuales.

Resolver problemas prácticos mediante computadoras y/o microcontroladores para aprovechar las ventajas que estas tecnologías presentan.

Diseñar y proyectar con dispositivos microcontroladores y FPGA en conjunto a los lenguajes de programación para resolver problemas reales.

## **VI - Contenidos**

### **1) Introducción.**

Organización estructurada de computadoras. Las distintas generaciones en la arquitectura de computadoras. Familias de computadoras.

### **2) Estudio de la CPU - Nivel de microarquitectura**

Trayectoria de datos. Microinstrucciones. Arquitecturas de ejemplo. Mejoras de diseño: Prefetching, Pipelining. Mejoras de rendimiento: Caché, Predicción de saltos, Ejecución fuera de orden, Ejecución especulativa.

### **3) Estudio del conjunto de instrucciones - Nivel ISA**

Tipos de datos. Formatos de instrucciones. Direccionamiento. Tipos de instrucciones. Flujo de control.

### **4) Sistemas operativos.**

Memoria virtual. Virtualización de hardware. Instrucciones de entrada/salida a nivel sistema operativo. Procesamiento paralelo.

### **5) Lenguaje ensamblador.**

Introducción al lenguaje ensamblador. Macros. El proceso de ensamblado. Enlazado y carga.

### **6) Programación de computadoras**

Paradigmas de programación. Paradigmas imperativos. Lenguajes para el paradigma imperativo. Ensamblador, C, C++. Ejercicios sobre programación de microcontroladores, y de microcontroladores conectados entre si y a computadoras de procesos. Programación en lenguaje Python. Ecosistema de librerías Python.

### **7) Procesadores ARM Cortex M3**

Arquitectura ARM. Generalidades. Arquitectura de buses. Registros. Mapa de Memoria. Pipelining. Acceso directo a memoria (DMA). Introducción a la familia STM32F1. Entorno de desarrollo. Depuración. Biblioteca HAL.

### **8) Manejo de entradas y salidas en ARM Cortex M3**

Introducción a GPIO. Estructura. Modos de funcionamiento. Flujo de ejecución. Interrupciones. El NVIC. Configuración y ciclo de vida. Implementación en STM32F1

### **9) Periféricos en ARM Cortex M3**

Configuración de reloj. Conversor analógico a digital. Modos de funcionamiento. Configuración. Timers. Modos de funcionamiento. Configuración e interrupciones. Comunicación serie USB-CDC. Conexión por puerto serie virtual. Implementación en STM32F1

### **10) Lenguajes de Descripción de Hardware**

Introducción a VHDL. Programación de FPGA. Ventajas.

### **11) Introducción a Bases de Datos y Tecnologías Web.**

Generalidades de Bases de Datos relacionales. Tipos de datos. Consultas SQL. Vinculación con Python. Introducción a HTML, CCS y Javascript.

## **VII - Plan de Trabajos Prácticos**

TP0 - Repaso de conceptos de electrónica digital.

TP1 - Programación avanzada en lenguaje C.

TP2 / Laboratorio 1: STM32 - Configuración de GPIO  
TP3 / Laboratorio 2: STM32 - Manejo de interrupciones  
TP4 / Laboratorio 3: STM32 - Manejo de ADC y Timers  
TP5 - Lenguajes de Descripción de Hardware (HDL) mediante FPGA.  
TP6 - Lenguajes de Programación de Alto Nivel (Python).

Se debe entregar informes de cada trabajo práctico, detallando la solución implementada a los problemas planteados y fundamentando la elección de dicha solución. Además, se debe entregar un proyecto final integrador en el que utilicen las tecnologías aprendidas para solucionar un problema a elección del estudiante junto con su informe, el cuál será defendido oralmente frente a los docentes y estudiantes

## VIII - Regimen de Aprobación

### A - METODOLOGÍA DE DICTADO DEL CURSO:

El dictado del curso será presencial. En todos los casos se prevé una clase teórica y una clase práctica semanal, donde esta última podrá ser de práctica de aula o laboratorio dependiendo de los contenidos del programa a dictarse en esa semana en particular. Los contenidos teóricos y prácticos serán puestos a disposición de los estudiantes a través de la plataforma Google Classroom provista por la UNSL.

### B - CONDICIONES PARA REGULARIZAR EL CURSO

Para acceder a la condición de regular, los estudiantes deberán cumplir con los siguientes requisitos:

- Entregar y aprobar con al menos 70 puntos, el 100 % de las actividades prácticas propuestas por el equipo docente.
- Aprobar con al menos 50 puntos el 100% de las evaluaciones parciales prácticas definidas, de acuerdo a la normativa vigente en la UNSL.
- Asistir al menos al 75 % de las clases prácticas de aula y al 100% de las clases prácticas de laboratorio

### C – RÉGIMEN DE APROBACIÓN CON EXÁMEN FINAL

El examen final para los estudiantes que se encuentren en condición regular consistirá en una evaluación oral y/o escrita sobre los contenidos teóricos de la asignatura. Los temas se sortearán al azar el día del examen.

### D – RÉGIMEN DE PROMOCIÓN SIN EXAMEN FINAL

El curso no contempla régimen de promoción sin examen final.

### E – RÉGIMEN DE APROBACIÓN PARA ESTUDIANTES LIBRES

Un estudiante en condición libre deberá presentar una carpeta con todos los trabajos prácticos de aula para ser evaluados antes de la fecha del examen final. Luego deberá acordar con los docentes de la asignatura una fecha, antes del examen final, donde presentará los prácticos de laboratorio funcionando. Una vez aprobados todos los prácticos (de aula y laboratorio), el estudiante será evaluado en un examen final oral sobre temas teóricos y prácticos que solicite la mesa examinadora.

## IX - Bibliografía Básica

- [1] Organización de computadoras: Un enfoque estructurado. 6ta edición, A. Tanenbaum, T. Austin. Pearson, 2013.
- [2] Tipo: Libro
- [3] Formato: Impreso
- [4] Disponibilidad: Biblioteca UNSL
- [5] Computer Organization and Design: The Hardware/Software interface. 3rd Edition, D. Patterson, J. Hennessy. Morgan Kauffman, 2005.
- [6] Tipo: Libro
- [7] Formato: Impreso
- [8] Disponibilidad: Biblioteca UNSL
- [9] Digital Design and Computer Architecture: ARM Edition S. Harris, D. Harris. Morgan Kauffman, 2016.
- [10] Tipo: Libro
- [11] Formato: Impreso
- [12] Disponibilidad: Biblioteca UNSL

## X - Bibliografía Complementaria

[1] Actel HDL Coding Style Guide Altec Corp. 2009.

[2] Tipo: Libro

[3] Formato: Digital

[4] Disponibilidad: [https://www.microsemi.com/document-portal/doc\\_download/130823-hdl-coding-style-guide](https://www.microsemi.com/document-portal/doc_download/130823-hdl-coding-style-guide)

[5] Mastering STM32 - A step-by-step guide to the most complete ARM Cortex-M platform, using the official STM32Cube development environment

[6] Tipo: Libro

[7] Formato: Digital

[8] Disponibilidad: <https://libgen.is/book/index.php?md5=9EF247BC7328AE96D937B3818A681558>

## XI - Resumen de Objetivos

- Interpretar el funcionamiento de un microprocesador.
- Proponer soluciones a problemas conocidos de su especialidad mediante la programación.
- Diseñar y proyectar con dispositivos de hardware programables para resolver problemas reales.

## XII - Resumen del Programa

Unidad 1: Introducción

Unidad 2: Estudio de la CPU - Nivel de microarquitectura

Unidad 3: Estudio del conjunto de instrucciones - Nivel ISA

Unidad 4: Sistemas Operativos

Unidad 5: Lenguaje Ensamblador

Unidad 6: Programación de computadoras con lenguajes de alto nivel.

Unidad 7: Procesadores ARM Cortex M3

Unidad 8: Manejo de entradas y salidas en ARM Cortex M3

Unidad 9: Periféricos en ARM Cortex M3

Unidad 10: Lenguajes de Descripción de Hardware

Unidad 11: Bases de Datos y Tecnologías WEB

## XIII - Imprevistos

En caso de no poder dictarse el curso en forma presencial debido al cumplimiento de normativas superiores se habilitará su dictado usando la plataforma Google Classroom siguiendo los mismos lineamientos teóricos y prácticos. Sin embargo se les adecuará los trabajos prácticos y de laboratorio al uso de softwares de simulación que serán debidamente entregados y explicados a los alumnos.

## XIV - Otros

Aprendizajes Previos:

Conoce los diferentes dispositivos electrónicos básicos, tanto digitales como analógicos.

Interpreta el álgebra de Boole

Evaluar componentes electrónicos disponibles según sus especificaciones.

Conocer la programación de lenguajes de alto nivel.

Detalles de horas de la Intensidad de la formación práctica.

Cantidad de horas de Teoría: 30 horas

Cantidad de horas de Práctico Aula: 0

Cantidad de horas de Práctico de Aula con software específico: 35 horas

Cantidad de horas de Formación Experimental: 10 horas

Cantidad de horas de Resolución Problemas Ingeniería con utilización de software específico: 0

Cantidad de horas de Resolución Problemas Ingeniería sin utilización de software específico: 0

Cantidad de horas de Diseño o Proyecto de Ingeniería con utilización de software específico: 15 horas

Cantidad de horas de Diseño o Proyecto de Ingeniería sin utilización de software específico: 0

Aportes del curso al perfil de egreso:

- 1.1. Identificar, formular y resolver problemas. (Nivel 3)
- 1.2. Concebir, diseñar, calcular, analizar y desarrollar proyectos (Nivel 2)
- 1.3. Planificar, gestionar, controlar, supervisar, coordinar, ejecutar y evaluar proyectos (Nivel 2)
- 2.1. Utilizar y adoptar de manera efectiva las técnicas, instrumentos y herramientas de aplicación (Nivel 2)
- 2.4. Aplicar conocimientos de las ciencias básicas de la ingeniería y de las tecnologías básicas. (Nivel 3)
- 2.5. Planificar y realizar ensayos y/o experimentos y analizar e interpretar resultados. (Nivel 2)
- 3.2. Comunicarse con efectividad en forma escrita, oral y gráfica. (Nivel 2)
- 3.3. Manejar el idioma inglés con suficiencia para la comunicación técnica. (Nivel 2)
- 3.5. Aprender de forma continua y autónoma, incluso aprovechando las inteligencias artificiales (Nivel 2)
- 3.8 Diseñar y proyectar con dispositivos de hardware programables para resolver problemas reales.

<b>ELEVACIÓN y APROBACIÓN DE ESTE PROGRAMA</b>	
	<b>Profesor Responsable</b>
Firma:	
Aclaración:	
Fecha:	