



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

(Programa del año 2026)
(Programa en trámite de aprobación)
(Presentado el 20/08/2026 19:09:40)

I - Oferta Académica

Materia	Carrera	Plan	Año	Período
Programación Informática	ING. MECATRÓNICA	OCD N° 19/22	2026	2° cuatrimestre
Programación Informática	INGENIERÍA ELECTRÓNICA	OCD N° 23/22	2026	2° cuatrimestre

II - Equipo Docente

Docente	Función	Cargo	Dedicación
CARLETO, JAVIER ALEJANDRO	Prof. Responsable	P.Adj Exc	40 Hs
DEMICHIELIS, JUAN PABLO	Prof. Colaborador	P.Adj Semi	20 Hs
GASULL, VIVIANA LUCIA	Prof. Colaborador	P.Adj Exc	40 Hs
FAJARDO, Matias Ivan	Auxiliar de Práctico	A.1ra Exc	40 Hs
VIDELA, NAHUEL NEHUEN	Auxiliar de Práctico	A.2da Simp	10 Hs

III - Características del Curso

Credito Horario Semanal				
Teórico/Práctico	Teóricas	Prácticas de Aula	Práct. de lab/ camp/ Resid/ PIP, etc.	Total
5 Hs	Hs	Hs	Hs	5 Hs

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

Duración			
Desde	Hasta	Cantidad de Semanas	Cantidad de Horas
03/08/2026	13/11/2026	15	75

IV - Fundamentación

La presente asignatura se fundamenta en la necesidad del ingeniero de conocer y aplicar las técnicas de resolución de problemas de complejidad media y alta mediante la programación aplicada, como así también, conocer y manejar herramientas informáticas para la resolución de problemas mediante la programación en lenguajes de alto nivel.

V - Objetivos / Resultados de Aprendizaje

Resultados de Aprendizaje:

1. Aplicar técnicas de diagramación y pensamiento lógico para la representación de soluciones algorítmicas estructuradas y no estructuradas, frente a problemas de carácter general, considerando diferentes ámbitos de aplicación, y actuando con pensamiento crítico y creativo.
2. Diseñar soluciones algorítmicas a problemas de complejidad media y alta para su aplicación en asignaturas de los ciclos

superiores, cumpliendo con las buenas prácticas de programación, y utilizando las ventajas de los entornos de programación con orientación numérica para su codificación.

3. Aplicar sentencias y bloques de programación para la codificación de algoritmos necesarios en los cursos posteriores y en su vida profesional, conociendo las reglas semánticas y sintácticas de lenguajes de programación determinados.

VI - Contenidos

Unidad Temática Nro. 1. INTRODUCCIÓN A LA PROGRAMACIÓN INFORMÁTICA

Objetivos específicos: Que el futuro profesional logre reconocer y recordar las estructuras necesarias para la resolución de problemas en forma algorítmica, y se interiorice en la clasificación de lenguajes de programación.

Desarrollo:

1-a.- Repaso de conceptos. Algoritmos. El Proceso de Resolución de Problemas. Etapas de la Resolución de Problemas.

Estrategia de resolución. Estructura general de un programa – modularidad – datos - constantes y variables

1-b.- Repaso de estructuras algorítmicas: Estructuras secuenciales, selectivas y repetitivas. Estructuras anidadas. Contadores y acumuladores.

1-c.-Lenguajes de programación – Clasificación – Paradigmas de programación

1-d – Herramientas de IA aplicadas a programación: prompting básico - uso responsable - IA como herramienta de debug y explicación de código, no de sustitución del razonamiento. - Por qué entender el algoritmo sigue siendo indispensable, aunque la IA lo escriba.

Unidad Temática Nro. 2. INTRODUCCIÓN A ENTORNOS DE PROGRAMACIÓN CON ORIENTACIÓN NUMERICA

Objetivo específico: que el futuro profesional logre conocer el funcionamiento básico del entorno MATLAB/OCTAVE como herramienta informática.

Desarrollo:

1-a.- Generalidades.

1-b.- Elementos y entorno operativo.

1-c.- Comandos matemáticos básicos.

1-d.- Manejo de la Ayuda

1-e.- Graficas: Comandos y alternativas.

1-f.- Utilización de archivos de comando, edición, depuración.

Unidad Temática Nro. 3. PROGRAMACIÓN EN ENTORNOS CON ORIENTACIÓN NUMERICA

Objetivos específicos: Que el futuro profesional logre codificar algoritmos en Matlab/Octave, y dominar esta herramienta para la utilización en asignaturas superiores. Que sea capaz de resolver problemas de complejidad media y alta mediante algoritmos

Desarrollo:

3-a.- Instrucciones de Matlab/Octave para programación: Entrada/Salida – Bifurcaciones – Iteraciones.

3-b.- Funciones en Matlab/Octave: Intrínsecas y de usuario

1-c.- Introducción a las estructuras de datos. Arreglos uni, bi y n-dimensionales. Operaciones: recorrido, asignación, actualización, desplazamiento, ordenamiento.

3-c.- Codificación de algoritmos en Matlab/Octave

1-e.- Resolución de Problemas de complejidad media y alta.

Unidad Temática Nro. 4. PROGRAMACIÓN EN C

Objetivo específico: Que el futuro profesional logre conocer el fundamento de la programación en C

Desarrollo:

4-a.- Introducción al lenguaje C. Concepto de lenguaje compilado. Proceso de compilación. Herramientas de desarrollo y depuración: programa fuente, programa objeto, programa ejecutable. Librería estándar. Convenciones de estilo. Comentarios y documentación.

4-b.- Estructura general de un programa en C. La función Main(). Componentes del lenguaje Variables y constantes,

operadores y expresiones. Palabras reservadas del lenguaje. Tipos. Conversión de tipos. Declaraciones. Operadores y precedencia.

Entrada y salida: Entrada y salida estándar. Entrada y salida con formato - scanf y printf.

4-c.-Control del flujo de ejecución: Bloques y sentencias. Condicionales: if-else, switch. Bucles: for, while, do-while. Operador ternario.

4-d.- Descomposición de problemas. Importancia de la modularización. Programación modular: Funciones y procedimientos: Funciones y valores de retorno. Variables externas, estáticas y registros. Alcance de variables. Recursividad. Estructura de bloques. El preprocesador de C. Directivas para el preprocesador.

4-e.-Punteros y arreglos: Punteros. Direcciones. Argumentos por valor y por referencia. Arreglos. Aritmética de direcciones. Arreglos multidimensionales. Inicialización.

4-f.-Estructuras: Introducción a las estructuras. Arreglos de estructuras. Definición de tipos, uniones y campos de bit.

VII - Plan de Trabajos Prácticos

Trab. Práctico Nro. 1:

Resolución de problemas mediante algoritmos

Trabajo práctico de auto revisión de los conocimientos y competencias aprehendidos en asignaturas previas. Tiene como objetivo recordar los conocimientos ya vistos. No tendrá evaluación. El equipo docente actuará de guía permanente y se realizarán los repasos pertinentes para la correcta aprehensión de los conocimientos y competencias.

Trabajo Práctico Nro. 2:

Introducción a Matlab/Octave

El trabajo práctico se desarrolla con la utilización de herramientas informáticas, software Matlab u Octave, donde el estudiante deberá realizar una serie de ejercicios propuestos con el fin de entender el manejo del software. El equipo docente actuará como guía permanente, tanto en forma individual como grupal. La evaluación del práctico se realiza en forma continua por parte del equipo docente.

Trabajo Práctico Nro. 3:

Diseño y codificación de algoritmos

Los estudiantes elegirán un entorno de desarrollo con orientación numérica (Matlab u Octave) en función de lo visto en la unidad 1. Procederán a la resolución individual de los problemas planteados mediante las técnicas de diagramación que consideren pertinentes y la codificación de los algoritmos diseñados. Los estudiantes deberán subir a la plataforma los problemas indicados como propuestos, para la revisión por parte del equipo docente.

Trabajo Integrador Nro. 1

Se desarrollará un aprendizaje basado en proyectos en el que los estudiantes, organizados en equipos, deberán programar en Matlab/Octave un juego de baja complejidad a elección. El equipo docente actuará como tutor a lo largo de todo el proceso, orientando el desarrollo e incorporando los conocimientos necesarios a medida que el proyecto avance. La evaluación se realizará mediante la presentación de los archivos programados y un informe descriptivo del desarrollo realizado.

Sobre el uso de herramientas de IA: se permite su utilización como asistente durante el desarrollo. Todo código generado total o parcialmente mediante IA deberá estar comentado indicando su origen. Durante la instancia de presentación y defensa, cada integrante del equipo deberá ser capaz de explicar el funcionamiento del código entregado. El criterio de evaluación no es el código en sí mismo, sino la comprensión que el estudiante demuestre sobre la solución desarrollada.

Trabajo Práctico Nro. 4:

Programación en C

TP A: Desarrollo de programas en un lenguaje de alto nivel usando en un entorno de desarrollo aplicando en cada caso la estructura de control más adecuada.

TP B: Resolución de programas propuestos usando la modularización y seleccionando las estructuras de datos acordes a los planteos.

Trabajo Integrador Nro. 2

Haciendo uso de la estrategia de aprendizaje basado en problemas, resolver un planteo que ponga en juego lo aprendido

mediante la programación en C, el mismo es de carácter evaluativo. Se resuelve de forma grupal y concluyen con la presentación de la solución encontrada al resto de la clase mediante una exposición/defensa.

VIII - Regimen de Aprobación

A - METODOLOGÍA DE DICTADO DEL CURSO:

El dictado de la asignatura, se realizará según lo detallado en el programa analítico y en general el dictado será teórico-práctico.

El estudiante dispondrá en forma permanente de todos los trabajos prácticos, guías de estudio, videos de las clases teóricas y auto evaluaciones, como así también consulta permanente mediante plataforma Classroom y contacto permanente con los docentes a través de un grupo de WhatsApp

Cada Unidad se comenzará con una clase teórica introductoria para que puedan comenzar con el práctico y se irá completando la teoría a medida que se avance en la resolución del mismo.

Se utilizarán metodologías de Aprendizaje Centrado en el Estudiante. Se utilizará la modalidad de clase invertida en algunos puntos, donde se guiará al estudiante al aprendizaje mediante recomendación de videos y material de lectura previa a la clase. En clase se realizará un repaso mediante actividades lúdicas para fortalecer el aprendizaje.

Para la Unidad 3 y la Unidad 4 se desarrollará un aprendizaje basado en proyecto donde los estudiantes en equipos de trabajo deberán programar en Matlab/Octave un juego a elección y un desarrollo para resolución de un problema en C respectivamente. En este caso los docentes actuarán de tutores para ir guiando al estudiante en el desarrollo del mismo, incorporando los conocimientos necesarios.

B - CONDICIONES PARA REGULARIZAR EL CURSO

Sólo podrán acceder a este régimen los estudiantes que cumplan con las condiciones requeridas para cursar la asignatura que estipula el régimen de correlatividades vigentes en el plan de estudios de la carrera y se encuentren debidamente inscriptos en este curso.

Las condiciones para regularizar son:

- Presentación de los ejercicios requeridos
- Aprobación de las evaluaciones parciales prácticas o sus recuperaciones.
- Aprobación de los trabajos prácticos integradores

Características de las evaluaciones:

- Para regularizar la asignatura, los estudiantes deberán aprobar la totalidad de las evaluaciones prácticas previstas. La evaluación se realizará a través de la resolución de problemas, de características similares a lo resuelto en el práctico.
- Las evaluaciones se realizarán en forma individual, fijándose tres instancias para cada evaluación.

Es decir, existirá para cada instancia de evaluación, PARCIAL, 1º RECUPERATORIO y 2º RECUPERATORIO

Cada instancia de evaluación se aprueba con una nota igual o superior a 7 puntos sobre 10.

Estas evaluaciones podrán reemplazarse con evaluación continua a través trabajos prácticos y ejercicios propuestos, dependiendo de la cantidad de estudiantes y la dinámica del grupo.

C – RÉGIMEN DE APROBACIÓN CON EXÁMEN FINAL

El examen versará sobre la totalidad del programa, contemplando los aspectos teóricos.

La modalidad del examen final podrá ser escrita u oral de acuerdo a como lo decida el tribunal evaluador

D – RÉGIMEN DE PROMOCIÓN SIN EXAMEN FINAL

Sólo podrán acceder a este régimen los estudiantes que cumplan con las condiciones requeridas para cursar la asignatura que estipula el régimen de correlatividades vigentes en el plan de estudios de la carrera y se encuentren debidamente inscriptos en este curso.

Condiciones para promocionar el curso sin examen final (modalidad teórico-práctica):

- Presentación de los ejercicios requeridos
- Aprobación del 100% de las evaluaciones parciales teóricas o sus recuperaciones, con un mínimo de 7 (siete) puntos.
- Aprobación de las instancias de evaluación parciales prácticas o sus recuperaciones.
- Aprobación de los trabajos prácticos integradores

Características de las evaluaciones:

· Las evaluaciones constarán de dos etapas, una teórica y una práctica (Evaluadas en forma conjunta). La primera, realizarán a través de un examen donde el estudiante deberá exponer o responder las preguntas que se le formulen acerca de los temas contenidos en las Unidades Temáticas evaluadas. La segunda se realizará a través de la resolución de problemas, de características similares a lo resuelto en el práctico.

· Las evaluaciones se realizarán en forma individual, fijándose tres instancias para cada evaluación.

Es decir, existirá para cada instancia de evaluación, PARCIAL, 1º RECUPERATORIO y 2º RECUPERATORIO

Pudiendo alcanzarse la condición de promoción en cualquiera de las instancias.

Cada instancia de evaluación se aprueba con una nota igual o superior a 7 puntos sobre 10.

Estas evaluaciones prácticas podrán reemplazarse con evaluación continua a través trabajos prácticos y ejercicios propuestos, dependiendo de la cantidad de estudiantes y la dinámica del grupo.

La nota final en la materia surgirá del promedio de todas las notas obtenidas en los distintos exámenes, teóricos, prácticos y de los trabajos prácticos integradores.

E – RÉGIMEN DE APROBACIÓN PARA ESTUDIANTES LIBRES

Sólo podrán acceder a este régimen los estudiantes que registraron su inscripción anual en el período establecido y aquellos que estén comprendidos en alguna de las siguientes opciones;

a. Los estudiantes que estando inscriptos en el curso como promocionales o regulares, no cumplieron con los requisitos estipulados en el programa para esas categorías.

b. Los estudiantes no inscriptos para cursar, que cumplen con las correlativas requeridas para rendir el curso.

c. los estudiantes que han obtenido la regularización en el curso, pero el plazo de su validez ha vencido.

Para rendir la asignatura como estudiante libre, éste deberá inscribirse en los turnos de exámenes estipulados en el calendario de la Universidad, al igual que los estudiantes regulares.

Características de las evaluaciones:

El examen versará sobre la totalidad del último programa, contemplando los aspectos teóricos y prácticos del curso.

El examen constará de una parte práctica y una teórica que se realizan en el mismo día.

Para aprobar el curso el estudiante deberá obtener como calificación mínima de 4 (cuatro) puntos como promedio de las notas obtenidas en la instancia práctica y en la teórica, no pudiendo ser menor a 4 (cuatro) en cada una de ellas.

La modalidad del examen final podrá ser escrita u oral de acuerdo a como lo decida el tribunal evaluador.

Se aconseja al estudiante que desee rendir un examen libre ponerse en contacto previo con el responsable del curso para recabar mayor información.

IX - Bibliografía Básica

[1] Resúmenes de Clase - Apuntes desarrollados por el Equipo Docente (2026) - Autores: Javier A. Carletto – Juan Pablo Demichelis. Tipo: Resumen de la Asignatura. Formato: Digital. Disponibilidad: Material de estudio en la plataforma de la asignatura.

[2] Duque Betancourt, D. F., Saint-Priest Velásquez, Y. E., Segovia, P. y Loaiza, D. F. (2017). Algoritmos y Programación en Pseudocódigo. Tipo: Libro universitario. Formato: Digital — 1ª ed., 2017, Universidad Santiago de Cali. Disponibilidad: DOAB (Directory of Open Access Books): <https://directory.doabooks.org/handle/20.500.12854/80883>. ResearchGate: https://www.researchgate.net/publication/367855444_Algoritmos_y_programacion_en_pseudocodigo

[3] Gagliano, G. M., Alarcón, C. I., Angelone, L. M. y otros (2014). Elementos esenciales para programación: Algoritmos y Estructuras de Datos. Tipo: Libro de texto abierto (licencia CC BY-SA 3.0). Formato: Digital — 1ª ed., 2014, Iniciativa Latinoamericana de Libros de Texto Abiertos (LATIn) / Universidad Nacional de Rosario. Disponibilidad: <https://rephip.unr.edu.ar/items/da61c18d-24e1-4427-a5e3-c2e9b66eeaaaf>

[4] Aguilar Vidal, V. (2022). GNU Octave para Ingeniería. Tipo: Apunte universitario. Formato: Digital — 2022. Disponibilidad: https://www.researchgate.net/publication/365187455_GNU_OCTAVE_para_Ingenieria

[5] Referencia de funciones básicas de MATLAB 2026 - MathWorks. Tipo: Ficha de Ayuda. Formato: Digital. Disponible en:

https://la.mathworks.com/campaigns/offers/matlab-basic-functions-reference-cheat-sheet.html?s_v1=64560&elqem=5130750

_EM_ROW_DIR_26-03_MOE-CG-ESP&rec_id=8fb2e0668ab84a56ba4c3db1c6705b97&elqTrackId=2bec21944fcd4b51a96d6010a34fbf75&elq=8fb2e0668ab84a56ba4c3db1c6705b97&elqaid=64560&elqat=1&elqCampaignId=24910&elqak=8AF55878E09B4C684192B510C153D773E29C8AB5E73ACC7E138B479D17A5C71FB3D9#table3

[6] Aprenda Matlab 7.0 como si estuviera en primero. Autores: Javier García de Jalón, José Ignacio Rodríguez, Jesús Vidal. Tipo: Apunte Escuela Técnica Superior de Ingenieros Industriales, Universidad Politécnica de Madrid. Autores: 2005. Formato: Digital. Disponibilidad: Distribución libre web.

[7] El lenguaje de programación C (The C programming language). Brian W. Kernighan, Dennis M. Ritchie. Ed. Prentice Hall, 2ª edición. Tipo: Libro. Formato: Impreso. Disponibilidad: Área de Computación.

[8] Editorial Universidad de Almería (2024). Lecciones de Programación en C para estudiantes de ingeniería. Tipo: Manual universitario. Formato: Digital — 1ª ed., 2024, Universidad de Almería. Disponibilidad: https://editorial.ual.es/libro/lecciones-de-programacion-en-c-para-estudiantes-de-ingenieria_158922/

X - Bibliografía Complementaria

[1] Sitios Webs recomendados 2026. Tipo: Recopilación de sitios webs, publicaciones, videotutoriales y otros. Formato: Digital. Disponibilidad: Distribución libre web.

[2] Centro de Ayuda de Matlab. Tipo: Documentación oficial del software. Formato: Digital. Disponibilidad: <https://la.mathworks.com/help/matlab/getting-started-with-matlab.html>

[3] Pinales Delgado, Francisco y Velázquez Amador, César. (2014). Algoritmos resueltos con Diagramas de Flujo y Pseudocódigo. Tipo: Libro - Problemario universitario. Formato: Digital — Universidad Autónoma de Aguascalientes. Disponibilidad: <https://editorial.uaa.mx/docs/algoritmos.pdf>.
https://editorial.uaa.mx/catalogo/ccb_algoritmos_9786078285969.html

[4] Solución de problemas de ingeniería con Matlab – Delores M. Etter. Editorial Prentice Hall, 1997. Tipo: Libro. Formato: Impreso. Disponibilidad: Biblioteca Villa Mercedes / Área de Computación.

[5] Aprenda a programar como si estuviera en primero. Autores: Iker Aguinaga, Gonzalo Martínez, Javier Díaz. Tipo: Apunte Libre de la Escuela Superior de Ingenieros de San Sebastián. Formato: Digital. Disponibilidad: Distribución libre web.

[6] Pérez López, C. (2002). Matlab y sus aplicaciones en las ciencias y la ingeniería. Pearson Educación. Tipo: Libro. Formato: Digital. Disponibilidad: <https://elibro.net/es/lc/unsl/titulos/45293>

[7] Metodología de la Programación. Diagramas de flujo algoritmos y programación estructurada – Luis Joyanes Aguilar – McGraw Hill – 1987. Tipo: Libro. Formato: Impreso. Disponibilidad: Biblioteca Villa Mercedes.

[8] Comunidad Stack Overflow. Empezando con el lenguaje C. Tipo: Manual colaborativo (licencia CC BY-SA). Formato: Digital. Disponibilidad: <https://manual-informatica.com/programacion/5-empezando-el-lenguaje-c.html>

XI - Resumen de Objetivos

1. Aplicar técnicas de diagramación y pensamiento lógico
2. Diseñar soluciones algorítmicas a problemas de complejidad media y alta
3. Aplicar sentencias y bloques de programación

XII - Resumen del Programa

Unidad Temática Nro. 1. INTRODUCCIÓN A LA PROGRAMACIÓN INFORMÁTICA

Unidad Temática Nro. 2. INTRODUCCIÓN A ENTORNOS DE PROGRAMACIÓN CON ORIENTACIÓN NUMERICA

Unidad Temática Nro. 3. PROGRAMACIÓN EN ENTORNOS CON ORIENTACIÓN NUMERICA

Unidad Temática Nro. 4. PROGRAMACIÓN EN C

XIII - Imprevistos

--- En el caso que algunos estudiantes no puedan disponer del software necesario para resolver los prácticos en sus hogares se utilizarán aplicaciones para celular, software alternativo y herramientas on line que emulan el software necesario. Si alguna de las técnicas centradas en el estudiante no se pudiese, se reemplazará dichas técnicas por las metodologías tradicionales. En casos de fuerza mayor que compliquen la presencialidad, se utilizarán clases síncronas no presenciales.

XIV - Otros

Aprendizajes Previos:

Analizar y descomponer situaciones problemáticas
Conocer los componentes y estructuras básicas de un algoritmo
Usar pseudocódigo y técnicas de diagramación.
Poseer pensamiento lógico y algorítmico para la resolución de problemas

Detalles de horas de la Intensidad de la formación práctica.

Cantidad de horas de Teoría:20

Cantidad de horas de Práctico Aula: (Resolución de prácticos en carpeta) 15

Cantidad de horas de Práctico de Aula con software específico: (Resolución de prácticos en PC con software específico propio de la disciplina de la asignatura) 20

Cantidad de horas de Resolución Problemas Ingeniería con utilización de software específico: (Resolución de Problemas de ingeniería con utilización de software específico propio de la disciplina de la asignatura) 20

Aportes del curso al perfil de egreso:

1.1. Identificar, formular y resolver problemas. (Nivel 2)

2.1. Utilizar y adoptar de manera efectiva las técnicas, instrumentos y herramientas de aplicación. (Nivel 2)

3.5. Aprender en forma continua y autónoma. (Nivel 1)

ELEVACIÓN y APROBACIÓN DE ESTE PROGRAMA	
	Profesor Responsable
Firma:	
Aclaración:	
Fecha:	