



Universidad Nacional del Nordeste



Facultad de Ciencias Agrarias

RESOLUCIÓN N° 1 3 6 5 2 / 2 3
CORRIENTES, 2 5 AGO 2023

VISTO

El Expediente N° 07-2023-01655 por el cual la Secretaria Académica E.E. (Dra.) Laura Itatí GIMÉNEZ, eleva para su consideración el programa de la Asignatura “Informática y Programación”, del Nuevo Plan de Estudios de la Carrera de Grado Ingeniería Industrial según Resolución N° 340/23 CS, y

CONSIDERANDO

Que dicho programa fue analizado por la Comisión de Gestión y Evaluación Curricular de Ingeniería Industrial.

Lo resuelto en la sesión del día 25 de agosto de 2023.

Por ello;

**EL CONSEJO DIRECTIVO DE LA FACULTAD DE CIENCIAS AGRARIAS
RESUELVE:**

ARTÍCULO 1º.- Aprobar el programa de la Asignatura Obligatoria “Informática y Programación”, de la Carrera Ingeniería Industrial que, como anexo, forma parte integrante de la presente Resolución.

ARTÍCULO 2º.- Regístrese, notifíquese y archívese.

E.E. (DRA.) LAURA ITATÍ GIMÉNEZ
SECRETARIA ACADEMICA

ING. AGR.(DR.) MARIO HUGO URBANI
BECANO

PROGRAMA ANALÍTICO Y DE EXAMEN

1. IDENTIFICACION

Facultad	Ciencias Agrarias
Carrera	Ingeniería Industrial
Departamento	Matemáticas y Estadística
Asignatura	Informática y Programación
Carácter	Obligatoria
Año	1er año
Régimen	Cuatrimestral
Carga horaria Total	80

Carga horaria semanal (horas reloj)	
Teórica	
Práctica	3
Teórico -práctica	2
Total	5

2. OBJETIVOS GENERALES DE LA ASIGNATURA

- Comprender los principios de la tecnología informática y sus aplicaciones.
- Identificar problemas algorítmicos.
- Comprender el proceso de desarrollo de software.
- Resolver problemas aplicando soluciones algorítmicas y estructuras de datos

3. ARTICULACIÓN CON OTRAS ASIGNATURAS

Anteriores

- Álgebra y Geometría Analítica: ya que la resolución de problemas algorítmicos y la comprensión del proceso de desarrollo de software requieren habilidades matemáticas básicas, como álgebra y lógica matemática.

Posteriores

- Estadística Aplicada: Los modelos estadísticos pueden ser implementados mediante la programación. Esto permitiría a los estudiantes implementar y experimentar con diferentes modelos de forma más eficiente y efectiva.
- Organización Industrial I: los sistemas industriales complejos, como los sistemas de control de producción o los sistemas de monitoreo de calidad, pueden ser programados para lograr una mayor precisión y control en la operación de los mismos. La programación también puede utilizarse para analizar grandes cantidades de datos en la Organización Industrial, lo que puede ayudar a la toma de decisiones informadas en cuanto a la producción y el uso de recursos.

4. NIVEL DE APOORTE DE LAS COMPETENCIAS DE EGRESO

Categoría	Competencia	0	1	2	3
CGT 1	Identificar, formular y resolver problemas de ingeniería.		X		
CGT 2	Concebir, diseñar y desarrollar proyectos de ingeniería (sistemas, componentes, productos o procesos).		X		
CGT 3	Gestionar, planificar, ejecutar y controlar proyectos de ingeniería (sistemas, componentes, productos o procesos).		X		
CGT4	Utilizar de manera efectiva las técnicas y herramientas de la ingeniería.		X		
CGT5	Contribuir a la generación de desarrollos tecnológicos y/o innovaciones tecnológicas.		X		
CGSA-6	Desempeñarse de manera efectiva en equipos de trabajo.		X		
CGSA-7.	Comunicarse con efectividad.		X		
CGSA-9.	Aprender en forma continua y autónoma.		X		
CE1	Diseñar, proyectar, calcular, modelar y planificar las operaciones y procesos de producción, distribución y comercialización de productos (bienes y servicios).		X		
CE4	Dirigir, gestionar, optimizar, controlar y mantener las operaciones, procesos e instalaciones requeridas para la producción, distribución y comercialización de productos (bienes y servicios).		X		
CE5	Evaluar la sustentabilidad técnico-económica y ambiental de las operaciones, procesos e instalaciones requeridas para la producción, distribución y comercialización de productos (bienes y servicios).		X		
CE6	Gestionar y certificar el funcionamiento, condiciones de uso, calidad y mejora continua de las operaciones, procesos e instalaciones requeridas para la producción, distribución y comercialización de productos (bienes y servicios).		X		

5. CONTENIDOS POR UNIDAD

Unidad 1: Informática, Ciudadanía Digital y Seguridad

Objetivo: Comprender los conceptos fundamentales de la tecnología informática

Definición de la informática. Ciudadanía digital: información sensible, privacidad de la información, métodos seguros para el resguardo y envío de datos en Internet.

Unidad 2: Algoritmos, programas, eventos y procedimientos

Objetivo: Identificar y aplicar algoritmos adecuados para resolver problemas.

Algoritmos secuenciales, nociones de estado, autómatas, programación por eventos, variables.



Universidad Nacional del Nordeste



Facultad de Ciencias Agrarias

13652 / 23

Procedimientos y funciones: nociones, redundancia. Parámetros y argumentos.

Estructuras de Control (secuencia, selección e iteración). Estructuras selectivas, estructuras repetitivas. Propositiones, conectivos lógicos, tablas de verdad, equivalencias lógicas, leyes de De Morgan y simplificación de expresiones lógicas. Variables, tipos de datos, funciones. Programación por eventos.

Unidad 3: Representación de la Información y Estructuras de Datos

Objetivo: Comprender los conceptos fundamentales de la representación de la información. Identificar y aplicar las estructuras de datos

Conceptos de datos, información, representación, codificación, decodificación, errores, redundancia, compresión de datos y formatos de archivo.

Sistemas de numeración. Representación de datos: bits, bytes, cadenas de caracteres, imágenes, sonidos. Operaciones y expresiones con datos.

Arreglos: conceptos, características, operaciones.

Unidad 4: La Computadora

Objetivo: Comprender los conceptos fundamentales de la tecnología informática, como el hardware, y software

Componentes físicos de una computadora. Funciones de los componentes principales: procesador, memoria, disco duro, periféricos.

Sistemas operativos: conceptos de sistema operativo, funciones, administración de recursos (procesos, memoria, dispositivos de entrada/salida) y seguridad informática.

Aplicaciones informáticas: herramientas ofimáticas, aplicaciones gráficas, bases de datos y aplicaciones web.

Unidad 6: Desarrollo de Software

Objetivo: 1. Comprender el proceso de desarrollo de software y aplicar metodologías de desarrollo ágiles. Crear programas en lenguaje C para resolver problemas prácticos. Identificar y resolver errores en el código. Diseñar y desarrollar programas utilizando técnicas de depuración y pruebas. Implementar aplicaciones utilizando bibliotecas y frameworks de terceros. Adquirir habilidades para trabajar en equipo y colaborar en proyectos de programación.

Fases del proceso de desarrollo de software: análisis, diseño, implementación, pruebas, mantenimiento.

Herramientas de desarrollo de software: entornos de programación, editores de código, compiladores, depuradores

Tipos de lenguajes de programación: alto nivel, bajo nivel, compilados, interpretados

Sintaxis y semántica de un lenguaje de programación

6. MODALIDAD DE LAS ACTIVIDADES DE APRENDIZAJE Y ESTRATEGIAS DIDÁCTICAS

1. Estrategias de Enseñanza

Las clases estarán organizadas de la siguiente manera: clases teórico-prácticas y prácticas. Se utilizarán las siguientes estrategias:

- a. Enfoque en la resolución de problemas: se fomentan la resolución de problemas algorítmicos y el diseño de programas informáticos, a través de



Universidad Nacional del Nordeste



Facultad de Ciencias Agrarias

13652 / 23

la presentación de problemas y ejercicios que impliquen el uso de algoritmos y estructuras de datos.

- b. Enseñanza basada en proyectos: se proponen proyectos que involucren la creación de soluciones informáticas que utilicen diferentes lenguajes de programación, y así fomentar el trabajo en equipo y el desarrollo de habilidades de programación.
- c. Aprendizaje cooperativo: se fomentan dinámicas en grupo, donde los estudiantes puedan compartir sus conocimientos y experiencias, resolver problemas en conjunto y desarrollar habilidades de trabajo en equipo.
- d. Uso de recursos digitales: se emplean recursos digitales como videos, tutoriales, juegos educativos y simuladores para ayudar a los estudiantes a comprender mejor los conceptos y aplicaciones de la tecnología informática.
- e. Enseñanza basada en casos: se presentan casos de estudio que involucren problemas reales de la industria informática, y así ayudar a los estudiantes a desarrollar habilidades de análisis y resolución de problemas en un contexto real.

2. Actividades de aprendizaje

Basado en los contenidos y estrategias de enseñanza propuestos, se diseña un conjunto de actividades de aprendizaje para ayudar a los estudiantes a alcanzar los objetivos de aprendizaje, detallados a continuación:

- a. Clases expositivas: el profesor explica los conceptos fundamentales de la informática y proporcionar ejemplos para ilustrarlos.
- b. Resolución de problemas: los estudiantes trabajan en problemas algorítmicos, aplicando soluciones y estructuras de datos para encontrar la respuesta correcta.
- c. Ejercicios de programación: los estudiantes practican escribiendo y resolviendo programas utilizando un lenguaje de programación específico.
- d. Estudio de casos: los estudiantes estudian casos reales en los que se hayan utilizado algoritmos y estructuras de datos para resolver problemas. Los estudiantes pueden analizar los casos de estudio y discutir cómo se aplicaron las soluciones.
- e. Trabajo en equipo: los estudiantes trabajan en grupos para resolver problemas o desarrollar programas. El trabajo en equipo puede mejorar las habilidades de comunicación y colaboración de los estudiantes.

7. EVALUACIÓN

1. Criterios

Los criterios de evaluación para las actividades de aprendizaje relacionadas con los contenidos y estrategias de enseñanza mencionados anteriormente serán:

1. Comprender los principios de la tecnología informática y sus aplicaciones:



Universidad Nacional del Nordeste



Facultad de Ciencias Agrarias

13652 / 23

- 1.1. Identificar las principales tecnologías de la información y su impacto en la sociedad.
- 1.2. Comprender el funcionamiento de las redes y sus aplicaciones en diferentes contextos.
- 1.3. Identificar las herramientas de software utilizadas en diferentes ámbitos y su función.
2. Identificar problemas algorítmicos:
 - 2.1. Identificar problemas algorítmicos y proponer soluciones.
 - 2.2. Analizar y comparar diferentes algoritmos para resolver un mismo problema.
 - 2.3. Identificar las principales características de los algoritmos eficientes.
3. Comprender el proceso de desarrollo de software:
 - 3.1. Identificar las diferentes etapas del proceso de desarrollo de software.
 - 3.2. Comprender el papel de las diferentes personas involucradas en el proceso de desarrollo de software.
 - 3.3. Identificar los principales modelos de ciclo de vida del software y sus características.
4. Resolver problemas aplicando soluciones algorítmicas y estructuras de datos:
 - 4.1. Identificar las principales estructuras de datos y sus características.
 - 4.2. Seleccionar la estructura de datos adecuada para resolver un problema dado.
 - 4.3. Implementar algoritmos eficientes para resolver problemas en un lenguaje de programación específico.

2. Instrumentos

1. Evaluación formativa: se utilizar la evaluación formativa para monitorear el progreso de los estudiantes, identificar áreas de oportunidad y brindar retroalimentación constante para mejorar el aprendizaje.
2. Exámenes escritos: incluirán preguntas de opción múltiple, preguntas de respuesta corta y preguntas de ensayo. Estos exámenes evaluarán el conocimiento de los estudiantes sobre los conceptos fundamentales, así como su capacidad para resolver problemas y aplicar algoritmos y estructuras de datos.
3. Trabajos prácticos: los trabajos prácticos incluirán la implementación de algoritmos, el desarrollo de pequeñas aplicaciones y la resolución de problemas relacionados con la informática. Estos trabajos evaluarán la capacidad de los estudiantes para aplicar los conceptos teóricos a situaciones prácticas.

8. REGIMEN DE ACREDITACIÓN

1. Condiciones para regularizar la materia

- Registrar un 70 % de asistencia a clases teóricas y prácticas.
- Aprobar de 2 (dos) evaluaciones parciales escritas individuales, con el 60%. Con dos recuperatorio cuya aprobación le otorga de condición de alumno regular.

2. Condiciones para promoción de la materia



Universidad Nacional del Nordeste



Facultad de Ciencias Agrarias

1 3 6 5 2 / 2 3

- Registrar un 80 % de asistencia a clases teóricas-prácticas y prácticas.
 - Aprobar de 2 (dos) evaluaciones parciales escritas individuales, con el 60%. Con un recuperatorio cuya aprobación le otorga de condición de alumno regular.
 - Aprobar con el 60% los Trabajos prácticos.
3. **Condiciones para rendir examen final:** (detallar modalidad: oral, escrito, en máquina, etc)
1. Regular: examen oral de los contenidos del programa, de los cuales se sortearán tres para desarrollar.
 2. Libre:
 - a. Instancia de evaluación práctica: aprobar con el 60% los ejercicios propuestos.
 - b. Instancia de evaluación teórica: examen oral de los contenidos del programa, de los cuales se sortearán tres para desarrollar si supera la evaluación práctica.

9. BIBLIOGRAFIA:

- a. Bibliografía Específica
 - Bucker, A. B. (2017). Computer science handbook. CRC Press.
 - Sedgewick, R., & Wayne, K. (2019). Algorithms (4th ed.). Addison-Wesley.
 - Downey, A. (2012). Think Python: How to think like a computer scientist. O'Reilly Media.
 - Gaddis, T. (2018). Starting out with C++: From control structures through objects. Pearson Education.
 - Kernighan, B. W., & Ritchie, D. M. (1988). The C Programming Language (2nd ed.). Prentice Hall.
 - Knuth, D. E. (1997). The art of computer programming, volume 1: Fundamental algorithms (3rd ed.). Addison-Wesley.
 - Brookshear, J. G. (2014). Computer science: An overview (12th ed.). Pearson.
- b. Bibliografía Complementaria:
 - Bancila, M. (2018). Modern C Programming Cookbook: Over 100 recipes to help you overcome your difficulties with C programming and gain a deeper understanding of the working of modern C. Packt Publishing.
 - Griffiths, D., & Griffiths, D. (2012). Head First C: A Brain-Friendly Guide. O'Reilly Media.
 - Meyers, S. (2014). Effective Modern C++: 42 Specific Ways to Improve Your Use of C++11 and C++14. O'Reilly Media.
 - Perry, G., & Miller, D. (2013). C Programming Absolute Beginner's Guide (3rd Edition). Que Publishing.
 - Shaw, Z. A. (2015). Learn C the Hard Way: Practical Exercises on the Computational Subjects You Keep Avoiding (Like C) (Zed Shaw's Hard Way Series). Addison-Wesley Professional.

10. PROGRAMA DE TRABAJOS PRÁCTICOS.

La programación es un proceso creativo, motivador y constructivo que permite a los estudiantes crear sus propios artefactos o soluciones a problemas planteados por la cátedra utilizando métodos o "recetas". En este sentido, las actividades se centrarán en desarrollar en los estudiantes habilidades que aporta la programación y el pensamiento computacional en el proceso de solución de problemas, como analizar problemas, organizar y representar datos de manera lógica, automatizar soluciones mediante pensamiento algorítmico, usar abstracciones y modelos, comunicar procesos y resultados, reconocer patrones, generalizar y transferir.

Las actividades prácticas se dividirán, en términos generales, en:

Contenidos	Porcentaje de carga horaria
Introducción al pensamiento computacional, nociones de autómatas, de estado, eventos.	10%
Representación y almacenamiento de la información. Seguridad y codificación de los datos.	10%
Fundamentos de la programación, estructuras básicas, algoritmia, programación textual.	50%
Algoritmos y estructura de datos	30%

En cada nuevo contenido se partirán de conocimientos previos del alumno, la cátedra propondrá un problema que el estudiante intentará resolver con sus conocimientos y habilidades, luego se expondrá el método y concepto computacional para este tipo de problemas, de este modo el estudiante aprenderá haciendo, learning by doing o aprendizaje por descubrimiento/experimento.

Se trabajará también con el concepto de programación de aparos o XP, utilizando lenguaje de programación textual (ej: C, Arduino, Python) y simuladores de prototipado y ejecución en microcontroladores y/o en hardware específico según los recursos con que se cuente.

Las evaluaciones serán en proceso, observando la evolución de la clase en cuanto a los objetivos planificados, así como sumativa en cada examen parcial teórico práctico más abajo detallado.

Trabajo Práctico N° 1 (Unidad 1: Informática, Ciudadanía Digital y Seguridad)

Objetivo general:



Universidad Nacional del Nordeste



Facultad de Ciencias Agrarias

13652 / 23

- Generar conciencia sobre riesgos que se presentan en Internet.
- Brindar herramientas para enfrentarlos.

Objetivos específicos:

- Identificar información sensible que se publica en Internet.
- Concientizar sobre los riesgos que supone la exposición de información.
- Generar conciencia sobre el alcance de las publicaciones en medios digitales.
- Configurar opciones de seguridad de redes sociales
- Presentar el abuso de phishing.
- Reflexionar sobre el robo de identidad.

Trabajo Práctico N° 2 (Unidad 1: Informática, Ciudadanía Digital y Seguridad)

Objetivo general:

- Establecer criterios para crear contraseñas seguras.
- Comprender la diferencia entre texto plano y texto cifrado.
- Distinguir los mecanismos de cifrado simétricos y asimétricos

Objetivos específicos:

- Tomar conciencia de la importancia de usar contraseñas seguras.
- Distinguir entre contraseñas seguras y vulnerables.
- Identificar características que hacen a la solidez de una contraseña.
- Introducir la necesidad de mecanismos de cifrado.
- Comprender los principios del cifrado simétrico.
- Reconocer ejemplos de uso habituales de cifrado simétrico.
- Introducir los métodos de cifrado asimétrico.
- Comprender la diferencia entre claves públicas y privadas.

Trabajo Práctico N° 3 (Unidad 2: Algoritmos y programación)

Objetivo general:

- Introducir la noción de programa.
- Presentar la programación dirigida por eventos.
- Realizar una práctica de programación en la computadora

Objetivos específicos:

- Introducir la idea de programa como una descripción del comportamiento que se espera que tenga una máquina.
- Programar una aplicación sencilla.
- Manejar eventos.

Trabajo Práctico N° 4 (Unidad 2: Algoritmos y programación)

Objetivo general:

- Presentar la importancia de las interfaces gráficas.
- Separar el diseño de la interfaz de una aplicación de la programación de su comportamiento.
- Publicar aplicaciones.

Objetivos específicos:

- Reconocer la importancia de las interfaces gráficas.
- Familiarizarse con herramientas para diseñar interfaces.
- Diferenciar presentación y lógica de una aplicación.
- Publicar aplicaciones



Universidad Nacional del Nordeste



Facultad de Ciencias Agrarias

1 3 6 5 2 / 2 3

Trabajo Práctico N° 5 (Unidad 2: Algoritmos y programación)

Objetivo general:

- Introducir la noción de procedimiento.
- Presentar a los procedimientos como herramienta para evitar redundancia

Objetivos específicos:

- Presentar la noción de procedimiento
- Definir procedimientos.
- Refactorizar programas

Trabajo Práctico N° 6 (Unidad 2: Algoritmos y programación)

Objetivo general:

- Introducir las nociones de parámetro y argumento.
- Producir soluciones generales que resuelvan diversos problemas particulares.

Objetivos específicos:

- Presentar las nociones de parámetro y argumento
- Definir procedimientos con parámetros.
- Invocar procedimientos usando argumentos

Trabajo Práctico N° 7 (Unidad 2: Algoritmos y programación)

Objetivo general:

- Introducir las alternativas condicionales.
- Utilizar alternativas condicionales en un proyecto de programación

Objetivos específicos:

- Utilizar alternativas condicionales en un programa.
- Presentar la noción de aleatoriedad
-

Trabajo Práctico N° 8 (Unidad 3: Representación de la Información y Estructuras de Datos)

Objetivo general:

- Introducir el uso de variables.
- Presentar las listas como estructura de datos.
- Utilizar repeticiones para resolver problemas

Objetivos específicos:

- Ejercitar el razonamiento lógico deductivo.
- Presentar procesos repetitivos.
- Introducir el uso de variables.
- Diferenciar variables locales y globales.
- Construir un programa con repeticiones.
- Introducir el uso de listas.
- Construir una aplicación partiendo de una versión previa.
- Presentar los operadores lógicos de disyunción y conjunción.
- Introducir las funciones.

Trabajo Práctico N° 9 (Unidad 3: Representación de la Información y Estructuras de Datos)



Universidad Nacional del Nordeste



Facultad de Ciencias Agrarias

13652 / 23

Objetivo general:

- Comprender un sistema binario.
- Representar información numérica.
- Representar información textual.

Objetivos específicos:

- Presentar un sistema de codificación binario
- Representar números en un sistema binario.
- Representar información textual con un sistema binario.
- Reconocer y corregir ambigüedades de un sistema de representación.

Trabajo Práctico N° 10 (Unidad 4: La computadora)

Objetivo general:

- Presentar el modelo de von Neumann.
- Reconocer que las computadoras actuales son descritas por este modelo.

Objetivos específicos:

- Deconstruir la noción habitual de computadora.
- Presentar la arquitectura de von Neumann.
- Reconocer que las computadoras actuales se ajustan al modelo de von Neumann.
- Conocer una computadora por dentro.
- Identificar los componentes de una computadora.
- Identificar variables que determinan el rendimiento de la CPU.
- Dimensionar los volúmenes de información que almacenan los dispositivos de memoria.
- Conocer el mecanismo de caching.
- Conocer las principales características de los procesadores

Trabajo Práctico N° 11 (Unidad 4: La computadora)

Objetivo general:

- Observar cómo la computadora ejecuta programas en código de máquina.
- Presentar la categorización de lenguajes de alto y bajo nivel.
- Comprender las traducciones que hacen falta para que un programa pueda ejecutarse.

Objetivos específicos:

- Conocer la compilación de programas.
- Diferenciar lenguajes compilados y lenguajes interpretados.
- Distinguir entre lenguajes de alto y bajo nivel.
- Reproducir la ejecución de un programa en lenguaje de máquina.
- Identificar los componentes internos del procesador.
- Conocer el ciclo de instrucción.

Trabajo Práctico N° 12 (Unidad 4: La computadora)

Objetivo general:

- Comprender los sistemas operativos como un conjunto de piezas de software.
- Reconocer los principales servicios que provee un sistema operativo.
- Observar cómo los sistemas operativos administran el uso de la CPU y la memoria al ejecutarse más de un programa en forma concurrente.



Universidad Nacional del Nordeste



Facultad de Ciencias Agrarias

13652/23

Objetivos específicos:

- Reconocer a los sistemas operativos como un conjunto de componentes de software.
- Identificar algunos servicios que brindan de los sistemas operativos.
- Conocer los sistemas operativos multitareas.
- Observar que los programas no se ejecutan en forma simultánea.

Ámbito de formación práctica	Carga horaria total	
	Presencial	No Presencial
Formación experimental		
Resolución de problemas abiertos de ingeniería	20	
Actividades de proyecto y diseño	20	
Otras actividades	20	
Carga horaria total	60	

- Las actividades de formación práctica se llevarán a cabo tanto en la sala de informática como en el domicilio del estudiante. Posteriormente, se brindará la oportunidad para que los estudiantes puedan exponer sus soluciones, depurar su código y aclarar cualquier duda que puedan tener en un ambiente presencial.
- La carga horaria semanal de las actividades prácticas, son de tres (3) horas presenciales, y un (1) hora no presencial, con actividad virtual sincrónica o asincrónica según disponibilidad tecnológica o del tema a desarrollar.

	Carga horaria	
	Presencial	No Presencial
Carga horaria semanal total	5	
Carga horaria semanal destinada a la formación práctica	3	1

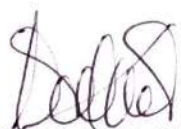
1. PROGRAMA DE EXAMEN:

a. Regular

- Modalidad: oral
- Temas: contenidos de la materia

b. Libre

- Modalidad:
 - Evaluación práctica: escrita
 - Evolución teórica: oral
- Temas contenidos de la materia


E.E. (DRA.) LAURA ITATÍ GIMÉNEZ
SECRETARIA ACADEMICA


ING. AGR.(DR.) MARIO HUGO URBANI
BECANO