

UADE	PROGRAMACIÓN AVANZADA	3.4.057
Departamento al que pertenece		Modalidad
Dpto. De Tecnología Informática Ciclo Inicial		Presencial
Carga horaria		Fecha de aprobacion en el Consejo de Facultad y N° de Acta
68 hs		No cargado aún
Carrera(s) en la(s) que se dicta		Codigos(s) carrera(s)
Ingeniería Electromecánica		3805
Ingeniería Electromecánica		3824
Ingeniería En Telecomunicaciones		3905
Licenciatura En Organización De La Producción		4205
Ingeniería Industrial		4505
Doble Titulación En Ingeniería Industrial Y Electromecánica		453824
Doble Titulación En Ingeniería Industrial Y Electromecánica		45389
Licenciatura En Tecnología Electromecánica		7505
Licenciatura En Tecnología De Las Comunicaciones		7605
Ingeniería Electrónica		9513
Ingeniería Electrónica		9524

I. Objetivos de aprendizaje

Que el alumno logre:

- Aplicar la heurística para el desarrollo de algoritmos de mediana complejidad.
- Resolver problemas utilizando metodologías estructuradas.
- Identificar y modelar problemas de ingeniería resolubles con estructuras de datos abstractos (pilas, colas y listas)

II. Aporte a los ejes transversales

Ing. Electromecánica e Industrial:

La materia de *Programación Avanzada* es fundamental en su formación, ya que desarrolla competencias clave para la identificación, formulación y resolución de problemas complejos.

Para los estudiantes de Ingeniería Electromecánica, se enfoca en la creación de software para modelar, simular y optimizar sistemas, fomentando la generación de innovaciones tecnológicas y la integración eficiente de hardware y software en proyectos de automatización y control. Además, promueve el uso de herramientas avanzadas para el diseño y análisis de sistemas electromecánicos.

En Ingeniería Industrial, la programación avanzada permite optimizar procesos productivos mediante el uso de algoritmos y modelos de simulación, facilitando la gestión y automatización de operaciones en la producción, distribución y comercialización de bienes y servicios. La aplicación de software especializado contribuye a la planificación eficiente de recursos, la toma de decisiones basada en datos y la mejora continua de los procesos industriales.

Esta formación integral permite a los estudiantes aplicar técnicas y herramientas para concebir y desarrollar proyectos, fortaleciendo su capacidad para afrontar desafíos tecnológicos en su ámbito profesional, ya sea en la optimización de sistemas electromecánicos o en la gestión eficiente de procesos industriales.

Ing. Electrónica

El aporte a los ejes transversales para los estudiantes de la carrera de Ingeniería Electrónica en la materia de Programación Avanzada es un punto crítico, debido a que fortalece su capacidad para el diseño, implementación y optimización de sistemas electrónicos complejos. Destaca su contribución en ejes clave como el proyecto, diseño y cálculo de dispositivos para la generación, transmisión y procesamiento de señales, así como en la creación de circuitos y sistemas para la comunicación. A través de la programación, los estudiantes desarrollan soluciones innovadoras que integran hardware y software, como sistemas embebidos y dispositivos lógicos programables, fundamentales para la automatización y el control.

Además, se abordan competencias relacionadas con la construcción, mantenimiento y operación de circuitos analógicos y digitales, aplicados al manejo de señales y energía eléctrica. Esto permite a los estudiantes participar activamente en el desarrollo tecnológico, garantizando una sólida formación técnica en sistemas de alta complejidad.

III. Contenidos

Unidad I: Modificadores de almacenamiento y operadores

Modificadores de tipo almacenamientos y modificadores de modo de almacenamiento. Su uso en argumentos de funciones. Alcance de las variables. Operadores para el manejo de bits. Enumeradores y definiciones de tipos (typedef). Cadenas de caracteres. Funciones de biblioteca para el manejo de cadenas de caracteres.

Unidad II: Punteros

Punteros: conceptos generales, punteros a los distintos tipos de datos. Pasaje de parámetros a funciones por dirección. Arreglos de punteros. Punteros a funciones. Aritmética de punteros. Punteros a arreglos uni y multidimensionales. Pasaje de parámetros en la línea de comandos. Memoria dinámica.

Unidad III: Tipos de datos definidos por el usuario

El tipo de dato agregado estructura, declaración, tipos de estructuras, estructuras anidadas, restricciones. Variables de tipo estructura, punteros a estructura. Operadores de acceso. Campos de bit como miembros de estructuras, declaración y utilización. Uniones.

Unidad IV: Archivos

Archivos y flujos. Flujos estándar. Archivos con y sin formato. Funciones para manejo de archivos con formato. Acceso secuencial y acceso directo. Concepto de registro y campo.

Unidad V: Lista, Pilas y Colas

Estructuras auto-referenciadas. Listas enlazadas, simples y dobles. Listas enlazadas circulares. Pilas y Colas. Distintas implementaciones: memoria estática y memoria dinámica.

IV. Estrategias de enseñanza

- Aprendizaje basado en problemas
- Aprendizaje vivencial
- Exposición y diálogo con preguntas
- Trabajo en Laboratorios

V. Formación práctica

TP N° 1: Introducción – Funciones, matrices y vectores:

Actividades:

- Realización de ejercicios con funciones que sirven para la creación de bloques de código reutilizables que realizan tareas específicas y ayudan a organizar y modular el código, facilitando su mantenimiento y comprensión.
- Utilización de vectores tal que estructuras de datos que almacenan una secuencia ordenada de elementos del mismo tipo y permiten el acceso y manipulación eficiente de datos en una sola dimensión. Al mismo tiempo aplicación de matrices como una extensión de los vectores a dos o más dimensiones, utilizadas para representar y operar con datos en forma de tablas.

Ámbito: Laboratorio de informática

TP N° 2: Cadena de caracteres

Actividades:

1. Ejercitación que involucra la manipulación de secuencias de caracteres, como la concatenación, división, búsqueda y reemplazo de subcadenas, fundamentales para procesar y analizar texto. Las cadenas de caracteres se utilizan en una amplia variedad de contextos, desde la entrada y salida de datos hasta la comunicación entre sistemas y el manejo de archivos.
2. Actividades prácticas de laboratorio que incluyen operaciones como convertir mayúsculas a minúsculas, eliminar espacios en blanco y comparar cadenas. Utilización de Expresiones regulares, tal que herramientas poderosas para buscar y manipular patrones complejos dentro de cadenas.

Ámbito: Laboratorio de informática

TP N° 3: Estructuras estáticas de datos

Actividades:

1. Ejercitación para aplicar estructuras estáticas de datos es decir aquellas cuyo tamaño se define en tiempo de compilación y no puede cambiar durante la ejecución del programa. Ejemplos comunes incluyen arrays y estructuras de datos fijas. Ofrecen acceso rápido y predecible a los elementos, y suelen ser más eficientes en términos de memoria.

Ámbito: Laboratorio de informática

TP N° 4: Archivos binarios y de texto

Actividades:

1. Ejercitación en archivos de texto, que Almacenan datos en formato legible por humanos, utilizando caracteres de texto. Son fáciles de crear y editar con cualquier editor de texto y archivos binarios, que son los que almacenan datos en formato binario, no legible directamente por humanos, pero son más eficientes en términos de almacenamiento y velocidad de acceso.
2. Utilización de archivos de texto para configuraciones, logs y datos que necesitan ser leídos o editados por personas y de archivos binarios para almacenar datos complejos como imágenes, audio, video y estructuras de datos que requieren acceso rápido y eficiente.
3. Práctica de conversión y manipulación entre archivos de texto y binarios según las necesidades de la aplicación, utilizando funciones específicas de programación.

Ámbito: Laboratorio de informática

TP N° 5: Estructuras dinámicas de datos

Actividades:

1. Ejercitación para aplicar estructuras dinámicas de datos, es decir, permiten cambiar su tamaño durante la ejecución del programa, adaptándose a las necesidades de almacenamiento. Ejemplos incluyen listas enlazadas, pilas, colas y árboles. Estas estructuras ofrecen flexibilidad y eficiencia en el uso de memoria, ya que pueden crecer y reducirse según sea necesario.

Ámbito: Laboratorio de informática

VI. Recursos

- Dispositivos digitales: Tablet, Wacom, celular, PC, notebooks
- Laboratorio
- Pizarrón
- Situaciones problemáticas
- Trabajos prácticos
- Otros: Software: Visual Studio

VII. Evaluación

◦ Requisitos de aprobación

- a. Cumplimiento mínimo del 75% de asistencia.
- b. Aprobación de cada una de las evaluaciones parciales y/o recuperatorios previstos con una calificación mínima de 4 (cuatro) puntos.

◦ Modalidad de evaluación

Dos instancias de evaluación parcial y un examen final

◦ Régimen de aprobación

Promoción: Los alumnos que obtuviesen una calificación mínima de 8 (ocho) puntos en cada una de las dos evaluaciones parciales previstas, obtendrán la aprobación de la asignatura. Si el estudiante debe rendir recuperatorio, no puede acceder al régimen de promoción. Los alumnos que aprueben las evaluaciones parciales y/o recuperatorios con un mínimo de 4 (cuatro) puntos, y que no logren la calificación necesaria para la promoción directa, podrán rendir examen final de la asignatura en los 11 (once) turnos de exámenes finales consecutivos posteriores a la aprobación de la cursada.

Los actos de deshonestidad académica o cualquier situación de indisciplina serán sancionados según el régimen disciplinario correspondiente.

VIII. Bibliografía

◦ Obligatoria

- Brombach, L. (2021). *Practical robotics in C++: Build and program real autonomous robots using Raspberry Pi*. BPB.
- Schildt, H. (1996). *C: Manual de referencia*. McGraw-Hill.
- Deitel, H. M., & Deitel, P. J. (1995). *Como programar en C/C++*. Prentice Hall.
- Joyanes Aguilar, L. (2020). *Fundamentos de programación: Algoritmos, estructuras de datos y objetos*. McGraw-Hill.
- Kernighan, B. W., & Ritchie, D. M. (1991). *El lenguaje de programación C*. Prentice Hall.

◦ Complementaria

- Joyanes Aguilar, L., & Zahonero Martínez, I. (2004). *Algoritmos y estructuras de datos: Una perspectiva en C*. McGraw-Hill.

IX. Detalles adicionales

No cargados aún

X. Cronograma

Nº Clase	Nº/s Unidad/es	Tema	Observaciones
1	1	Lenguaje de programación ¿C¿. Estructura de un programa. Directivas. Tipos de datos. Operadores. Estructuras de control. Sentencias de entrada y salida.	-
2	2	Funciones y punteros. Pasaje de parámetros por valor y por dirección.	-
3	2	Familiarización con el uso del entorno de desarrollo.	-
4	3	Arreglos. Vectores. Matrices. Ordenamiento. Búsqueda binaria.	-
5	3	Cadenas de caracteres. Funciones de Biblioteca para el manejo de cadenas.	-
6	4	Estructuras.	-
7	4	Arreglos de estructuras. ABM.	-
8	4	Archivos. Generalidades. Simulacro.	-
9	1, 2, 3, 4	PRIMER PARCIAL	-
10	5	Archivos binarios. Acceso directo. Ordenamiento de archivos.	-
11	5	Asignación dinámica de memoria (ADM). Arreglos de punteros a estructuras.	-
12	5	Archivos de texto. E/S con formato sobre archivos de texto. Ejercitación.	-
13	5	Listas enlazadas simples. Pilas y colas dinámicas.	-
14	5	Ejercitación. Simulacro.	-
15	5	SEGUNDO PARCIAL	-
16	5	Ejercitación con listas enlazadas.	-
17	5	Ejercitación. Recuperatorio y final adelantado.	-
18	1, 2, 3, 4, 5	FINAL REGULAR	-

La cantidad de clases estará sujeta al turno de la materia.