

INGENIERÍA EN MECATRÓNICA									
COMPUTACIÓN II									
DEPARTAMENTO		Ciencias de la Computación, Informática y Sistemas de Representación							
PLAN DE ESTUDIOS		CARÁCTER				DICTADO			
2015		X	Obligatoria		Optativa		Anual	X	Cuatrimstral
AÑO	MÓDULO	RÉGIMEN				CUATRIMESTRE DE CURSADO			
2	3		Teórica	X	Teórica-Práctica	X	Primero		Segundo
CARGA HORARIA TOTAL				75	CANTIDAD DE SEMANAS				15

DISTRIBUCIÓN DE LA CARGA HORARIA	Carga horaria
Bloques	Presencial
Ciencias Básicas de la Ingeniería	75
Tecnologías Básicas	0
Tecnologías Aplicadas	0
Ciencias y Tecnologías Complementarias	0
TOTAL	75

CARGA HORARIA DESTINADA A LAS ACTIVIDADES DE FORMACIÓN PRÁCTICA	Carga horaria
	Presencial
Instancias supervisadas de Formación Práctica	42,0
Proyecto Integrador	10,5
Práctica Profesional Supervisada	0
TOTAL	52,5

CARGA HORARIA SEMANAL	Presencial
Teoría	1,5
Formación Práctica	3,5
TOTAL	5



1. Fundamentación.

En Computación II se propone conocer el lenguaje con todas sus funcionalidades y características, ya que resultará imprescindible para los futuros ingenieros mecatrónicos, los cuales se encontrarán en todo momento con la necesidad de programa en C para hacer mover y controlar los sistemas mecánicos que diseñen.

El lenguaje C es conocido como un lenguaje de nivel medio, ya que combina elementos de lenguajes de alto nivel con la funcionalidad del ensamblador. Además, el código C es muy transportable, haciendo que el software escrito para un tipo de computadora o microcontrolador se pueda adaptar a otro de una forma más sencilla.

Como lenguaje de tipo medio, C, permite la manipulación de bits, bytes, y direcciones los cuales son los elementos básicos con los que trabajan las computadoras y microcontroladores. Esta posibilidad hace de C un lenguaje adecuado para la programación a nivel de sistema donde son comunes estas operaciones lo que permite usar C en lugar del lenguaje ensamblador en muchas situaciones, de hecho es usado por casi la totalidad de los fabricantes de microcontroladores para realizar las capas de abstracción de hardware (del inglés, HAL Hardware Abstraction Layers) las cuales dan acceso a la arquitectura de los mismos.

Como complemento introducir al alumno en la programación orientada a objeto (POO), va a completar los conocimientos necesarios que abarcan a la mayoría de los lenguajes de programación que se utilizan en la mecatrónica.

2. Objetivos.

Se espera que los estudiantes puedan:

- 1) Comprender la estructura y los aspectos característicos del lenguaje C como lenguaje estructurado para una correcta escritura del mismo.
- 2) Entender los tipos de datos y sentencias de control utilizadas por C para poder realizar programas más eficientes.
- 3) Manipular operaciones con bits, bytes y puntero de forma eficiente, de modo que permitan realizar programas adecuados para la posterior programación de microcontroladores.
- 4) Desarrollar la habilidad de resolución de problemas para poder abordar y crear soluciones eficientes utilizando algoritmos y estructura de datos adecuados.

3. Competencias y Descriptores.

La asignatura asegura el desarrollo de los contenidos mínimos previstos en el Plan de Estudios, los descriptores de conocimiento y los ejes y enunciados multidimensionales y transversales previstos en la Resolución CD N° 525/23.

Dicha Resolución del Consejo Directivo dispone, entre otras cuestiones, la contribución



de cada asignatura a la matriz de tributación de los descriptores de conocimiento. Asimismo, establece el aporte mínimo que cada asignatura deberá realizar para el desarrollo de las competencias y/o ejes y enunciados multidimensionales y transversales previstos en la normativa ministerial correspondiente.

La contribución de esta asignatura a los contenidos mínimos y a los descriptores de conocimiento, puede evidenciarse en los contenidos analíticos detallados en el apartado específico de este Programa Analítico.

En cuanto a las competencias y/o ejes y enunciados multidimensionales y transversales, los mismos se desarrollan en la intensidad prevista en la Resolución CD N° 525/23, siendo abordados de manera integral, articulando la metodología empleada, los contenidos analíticos, las actividades de formación práctica, las formas de evaluación, junto a los demás aspectos intervinientes en el proceso de enseñanza y aprendizaje llevado a cabo por esta asignatura.

Por ello, el presente Programa Analítico se ajusta a la normativa dictada por la unidad académica para dar cumplimiento a la normativa nacional vigente vinculada a la carrera.

4. Contenidos.

4.1. Contenidos Mínimos.

Programación en Lenguaje C. Introducción a la Programación orientada a objetos. Lenguaje Phyton.

4.2. Contenidos Analíticos.

TEMA I: Sintaxis general. Tipos de datos. Entrada/Salida. Operadores. Decisiones y bucles. Construcción if. Construcción if-else. Construcción switch. Construcción for. Construcción while. Construcción do-while.

TEMA II: Arreglos, Cadenas y Punteros: Arreglos. Enumeraciones. Estructuras. Uniones. Campos de Bits. Cadenas. Punteros. Operadores de punteros (* , &). Aritmética de punteros. Inicialización de punteros. Punteros y arreglos. Punteros y cadenas. Puntero a puntero. Arreglos y punteros de estructuras. Punteros a funciones. Conversiones de tipo y casting.

TEMA III: Funciones: Forma general de una función. Ámbito de las funciones. Argumentos y parámetros. Variables globales y locales. Argumentos y parámetros de funciones: llamada por valor, llamada por referencia. Funciones que devuelven valores no enteros. Prototipos de funciones. Valor retornado y argumentos de una función. Funciones y estructuras. Estructuras como parámetros y como valor de retorno de una función. Funciones y estructuras que permiten operar con la fecha y hora. Funciones recursivas.

TEMA IV: Archivos: Accesos a disco. Archivos de texto y binarios. Entrada/Salida a nivel del sistema. El operador Sizeof. Redireccionamiento. Archivos. Concepto. Archivos de texto y binarios. Trabajo sobre archivos. Apertura y cierre de un archivo. Escritura de un archivo. Lectura de un archivo. Búsqueda y modificación. Operaciones con archivos. Altas, modificaciones, consultas. Baja lógica de elementos de un archivo binario. Copia y



comparación de archivos.

TEMA V: Asignación Dinámica de Memoria: Punteros estáticos y dinámicos. Gestión dinámica de memoria. Estructuras dinámicas de datos: Colas, Pilas y Listas enlazadas. Insertar y eliminar elementos. Generación, recorrido y destrucción de una estructura dinámica de datos. Variables avanzadas.

TEMA VI: Introducción a la Programación Orientada a Objetos: Conceptos de clases, objetos, atributos, métodos, eventos. Lenguaje Python.

5. Metodología de enseñanza y de aprendizaje.

Se involucra al estudiante en la programación en C por medio de ejercicios y proyectos propuestos para cada unidad que provoquen en él, cuestiones que lo lleven a investigar, diseñar y desarrollar soluciones que le permitan adquirir técnicas de resolución de problemas y fomentar el aprendizaje continuo. Estos ejercicios o proyectos estarán debidamente organizados en forma tal que vayan poniendo a prueba todos los puntos necesarios para poder realizar los programas.

Así, se les presentarán en clases todos los elementos, esto se hará, a través de la discusión de las diferentes formas de realizar lo que uno desea con las herramientas que el lenguaje C brinda para trabajar la resolución de un problema. Se seguirá el contenido de las unidades, las cuales contienen los distintos tópicos organizados en orden de los conocimientos necesarios.

En cada unidad se proponen ejercicios de ejemplo que contemplan la mayoría de las posibilidades. La discusión del ejemplo pondrá en evidencia los problemas que pueda encontrar el alumno.

Los ejercicios o proyectos propuesto en cada unidad alimentarán la generación de un proyecto final integrador para que el alumno pueda trabajar en la identificación del problema y encontrar la resolución del mismo. Este proyecto deberá estar debidamente documentado ayudando a fomentar su capacidad de comunicación. Se buscará que el estudiante tenga una participación activa de su aprendizaje, lo anime a experimentar y aplicar las soluciones, realizando debates en grupos y estudiando los distintos casos de resolución.

6. Descripción de las actividades Teóricas y de Formación Práctica.

6.1. Actividades Teóricas.

Las actividades teóricas se complementan inmediatamente con las actividades prácticas, se busca de esta forma que el alumno incorpore los tópicos dados en la teoría mediante trabajos prácticos, se entiende que para la programación esto es relevante ya que siempre se necesita poder visualizar el resultado de lo aprendido experimentado que el programa realizado hace lo esperado.



6.2. **Actividades de Formación Práctica.**

Las actividades de formación práctica se realizarán para abordar todos los contenidos teóricos dados, para esto se realizarán por medio de trabajos prácticos que podrán enmarcarse como trabajos de laboratorio, resoluciones de problemas o ejercicios prácticos. En estos trabajos el estudiante podrá utilizar las distintas estructuras y componentes básicos del lenguaje, interpretar el uso de arreglos y estructuras, aplicar funciones y punteros, resolver problemas en donde tenga que utilizar el manejo de archivos, e integrar todos items dados en la teoría mediante un proyecto integrador.

6.3. **Ámbitos donde se desarrollan las actividades de Formación Práctica.**

Para la realización de las clases teórica-prácticas se necesitará:

- Sala de computación

Respecto del aula esta debe ser un área o mesas amplias con buena distribución de tomacorrientes para que se puedan ubicar los alumnos que llevan su notebook, de forma de liberar los lugares con pc de escritorio del laboratorio. El software para el desarrollo de las actividades será brindado por la cátedra, estará disponible en el Campus para los alumnos y deberá estar instalado en las computadoras del laboratorio.

7. **Articulación con otros espacios.**

Computación II es una materia que aporta verticalmente conocimientos esenciales para la mecatrónica sobre todo en la temática de automatización y control con microcontroladores siendo este un dispositivo fundamental en cualquier sistema de control. Si bien es una materia de las ciencias básicas necesita de que los alumnos tengan una introducción a la arquitectura de las computadoras para entender los tipos de datos base que usa C, esto se articula bien con *Computación I* materia que trata este tema, y con *Informática y Sistemas de Representación* en donde se pone al alumno en la resolución de pseudocódigo herramienta necesaria para que los alumnos puedan abordar no solo el lenguaje de programación C, sino también, cualquier otro en el que el estudiante quiera especializarse.

8. **Formas de evaluación.**

Se utilizará una evaluación formativa, para poder estar en todas y cada una de las etapas del proceso de aprendizaje. Así se podrá guiar a los estudiantes para que logren sus objetivos a medio y largo plazo. Se incitará a los alumnos a asumir un papel activo para que sean autosuficientes y, en el futuro puedan continuar con el proceso de aprendizaje con las habilidades y conocimientos adquiridos, cuando los estudiantes saben lo que están aprendiendo y cómo lo están estudiando, les resulta más fácil lograrlo por sí mismos. Para ello los docentes guiaremos y animaremos a los estudiantes a seguir por la buena dirección y mantener el ritmo.



Se utilizará una evaluación inicial para determinar el grado de conocimiento de los estudiantes para adaptar las estrategias de enseñanza y los estudiantes puedan acomodar su enfoque de aprendizaje. El objetivo principal de estas evaluaciones para el aprendizaje es impulsar el deseo de los estudiantes de aprender, acentuando el desarrollo y el logro académico en lugar del fracaso. Se utilizarán para la evaluación proyectos, presentaciones y actividades grupales. En primera instancia se evaluará al alumno por medio de ejercicios que indiquen el grado de conocimiento de la materia, a partir de aquí se seleccionarán ejercicios niveladores en el caso que sea necesario. Luego se llevará una evaluación continua del alumno por medio de ejercicios propuestos de forma que le permitan ver al alumno dónde necesita enfocarse. Por último se realizará un trabajo final que involucre todos los temas dados para valorar todo aquello que el alumno ha aprendido y cómo logró integrar esos conocimientos en su día a día.

En todo momento se proporcionarán al estudiante detalles sobre cómo pueden mejorar su aprendizaje, se les brindará habilidades, ejemplos para que puedan resolver los problemas. Este hecho hace que muchos alumnos mejoren su rendimiento pero, sobre todo, que muchos otros no den por perdido el curso. También, se va a poner en claro en las clase que es lo se va a enseñar, de esta forma el estudiante puede saber de antemano que va a aprender y pueda comprobarlo al final de las mismas. En todo momento se tratará de vincular lo que se está enseñando con su uso en la vida real así los estudiantes adquieren otra perspectiva; entienden mejor lo que están aprendiendo y para qué sirve.

9. Condiciones de Regularidad y Promoción.

9.1. Condiciones de Regularidad.

- **Resolución CD N° 200/12 - Artículo 14° - Inciso c):** Cátedras con un sistema de clases teóricas-prácticas 70% de asistencia para las clases dictadas.
- **Resolución CD N° 200/12 - Artículo 15° (los alumnos inscriptos en esta unidad académica, que presenten certificado de trabajo) - Inciso c):** Cátedras con un sistema de clases teóricas-prácticas 60% de asistencia para las clases dictadas.
- **Resolución CD N° 200/12 - Artículo 14° y 15° - Inciso d):** Haber presentado los informes de las actividades prácticas establecidos en la Programación de Cátedra.
- **Para esta asignatura esto último incluye:** La presentación que acredite la realización del 100% de los Trabajos Prácticos y Cuestionarios (Actividades de Formación Práctica Planificadas) será condición necesaria para regularizar y promocionar la asignatura.

9.2. Condiciones de Promoción.

- Tener aprobada la asignatura *Computación I*.
- La asignatura será promocionable con 2 (DOS) parciales, el último como proyecto integrador.



RESOLUCION C.D. N° 667/23

- Se deben cumplir los requisitos académicos para ser alumno regular del Reglamento Académico de la facultad (Resolución CD N° 200/12), de asistencia y actividades prácticas obligatorias.
- Para esta asignatura: La presentación que acredite la realización del 100% de los Trabajos Prácticos y Cuestionarios (Actividades de Formación Práctica Planificadas) será condición necesaria para regularizar y promocionar la asignatura.
- Obtener en ambos Parciales una calificación mayor o igual a 6 (SEIS).
- Existirá sólo una instancia recuperatoria: únicamente en los casos que, en sólo UNO de los parciales la/el estudiante haya obtenido una calificación menor o igual a 5 (CINCO) y mayor o igual a 4 (CUATRO), y estén cumplimentadas todas las otras exigencias para la promoción.
- En los casos de promoción, la nota final surgirá del promedio de las evaluaciones con calificación mayor o igual a 6 (SEIS).

10. Bibliografía.

Título	Autores	Editorial	Año de Edición
Fundamentos de informática y programación en C	Diego R. Llanos Ferraris	Paraninfo	2017
100 problemas resueltos de programación en lenguaje C para ingeniería	Ignacio Alvarado Aldea, José María Maestre Torreblanca, Carlos Vivas Venegas y Ascensión Zafra Cabeza	Paraninfo	2017
Aprende a programar en Python	Ángel Arias y Miguel Ángel Benitez.	Las Vegas	2021
Fundamentos de programación: Algoritmos, estructura de datos y objetos	Joyanes Aguilar, Luis	Mc. Graw Hill	2008
Estructura lógica y diseño de programas.	Cohen, Alan	Madrid Paraninfo	1986
Electrónica e informática aplicada	Rocha Díaz, Alberto Eduardo Ariel	Librería y editorial Alsina	2014