

Carrera: Ingeniería Electrónica**Asignatura:** Informática I**Planificación a partir del Ciclo Lectivo 2024****1. Datos administrativos de la asignatura**

Nivel en la carrera	1	Duración	Anual
Plan	2023		
Bloque curricular:	Ciencias Básicas de la Ingeniería		
Carga horaria presencial semanal (hs. cátedra):	5	Carga Horaria total (hs. reloj):	120
Carga horaria no presencial semanal (hs. reloj) (si correspondiese)		% horas no presenciales (hs. reloj) (si correspondiese)	

2. Presentación, Fundamentación

Cuando el estudiante comienza el cursado de esta materia, inicia su experiencia en la vida universitaria y está en la fase inicial de su proceso de ambientación a la carrera de Ingeniería. En Informática I aprenderá herramientas fundamentales como los diferentes sistemas de representación numérica, las bases metodológicas de desarrollo de software que le permiten diseñar algoritmos para resolver problemas, la utilización del lenguaje de programación C para implementar estos algoritmos, y los sistemas básicos de intercomunicación entre plataformas. A su vez, esta materia aplica estas competencias conforme se van desarrollando en la resolución de problemas que involucren los conocimientos adquiridos en las asignaturas del mismo nivel (Álgebra, Análisis I y II, y Física I) aplicando los mismos a resolución de problemas reales.

Esta materia es parte del área *Técnicas Digitales*, siendo correlativa de las demás materias del área, en forma mediata o inmediata. La relación con las demás materias del área se puede describir como sigue:

- A Informática II le aporta el bagaje de conocimiento referente a analizar y resolver problemas orientados a ingeniería por medio de un lenguaje de programación estructurado como C, permitiéndole asumir tal dominio y concentrarse en programación orientada a objetos.
- A Técnicas Digitales I le aporta la introducción a los sistemas de numeración y la aritmética binaria como así también las funciones lógicas booleanas.
- A Técnicas Digitales II le suministra el conocimiento general en bloques del hardware sobre el que es posible desarrollar fácilmente conceptos como sistemas basados en microprocesadores y sus interfaces, manejo de memoria, programación en bajo nivel, etc.

- El conocimiento de programación en lenguaje C es complementario para el desarrollo de contenidos de Técnicas digitales III tales como procesamiento digital de señales y redes. El abanico de temas de Redes de datos, que abarca: TCP, definición del protocolo OSI, congestión de datos, ruteo, voz sobre IP, etc. requieren conocimientos sólidos no solo de hardware sino también de la estructura de los dispositivos de entrada/salida y sus controladores.

Los contenidos de esta asignatura son relevantes para todo el espectro de materias que tratan temas tecnológicos en el plan de estudios. El análisis se sistematiza por áreas de conocimiento, destacando algunos ejemplos en los ejes de temas relacionados:

1. Área Matemática, Área Física y Área Química: Destreza de cálculo por la aplicación de programas computacionales escritos en C.
2. Área Electrónica: Manejo de conceptos de sentencias secuenciales, variables, arreglos, etc. para el uso de herramientas informáticas de cálculo como Octave.
3. Área Teoría de circuitos: A lo anterior se suman fundamentos para el desarrollo de programas en C para sistemas embebidos que implementen filtros digitales.
4. Área Sistemas de control: A lo anterior se suman fundamentos para el desarrollo de programas en C para los sistemas embebidos de control de motores de corriente continua sin escobillas.
5. Trabajo Final: Desarrollo del Proyecto Final utilizando sistemas embebidos programados en lenguaje C.

3. Relación de la asignatura con las competencias de egreso de la carrera

En la tabla siguiente se establece la relación de la asignatura con las competencias de egreso: Específicas, Genéricas Tecnológicas y Genéricas Sociales, Políticas y Actitudinales de la carrera.

Se incluyen las competencias de egreso a las que tributa, aportes reales y significativos de la asignatura, y en qué nivel (no aporta, bajo, medio, alto).

Competencias	Nivel
Competencias genéricas tecnológicas (CG):	
CG.1. Identificar, formular y resolver problemas de ingeniería.	Alto
CG.2. Concebir, diseñar y desarrollar proyectos de ingeniería.	Bajo
CG.3. Gestionar, planificar, ejecutar y controlar proyectos de ingeniería.	Bajo
CG.4. Utilizar de manera efectiva las técnicas y herramientas de aplicación en la ingeniería.	Medio
CG.5. Contribuir a la generación de desarrollos tecnológicos y/o innovaciones tecnológicas.	Bajo
Competencias genéricas sociales, políticas y actitudinales (CG)	
CG.6. Desempeñarse de manera efectiva en equipos de trabajo.	Medio
CG.7. Comunicarse con efectividad.	Medio

CG.8. Actuar con ética, responsabilidad profesional y compromiso social, considerando el impacto económico, social y ambiental de su actividad en el contexto local y global.	Medio
CG.9. Aprender en forma continua y autónoma.	Medio
CG.10. Actuar con espíritu emprendedor.	Bajo
CG.11 Fundamentos para el desarrollo de una actitud profesional emprendedora.	No aporta
Competencias Específicas de la carrera	
CE 1.1. Diseñar, proyectar y calcular sistemas, equipos y dispositivos de generación, transmisión y/o procesamiento de campos y señales analógicos y digitales; circuitos integrados; hardware de sistemas de cómputo de propósito general y/o específico y el software a él asociado; hardware y software de sistemas embebidos y dispositivos lógicos programables; sistemas de automatización y control; sistemas de procesamiento y de comunicación de datos y sistemas irradiantes, para brindar soluciones óptimas de acuerdo a las condiciones técnicas, legales, económicas, humanas y ambientales.	Bajo
CE 1.2. Plantear, interpretar, modelar y resolver los problemas de ingeniería descritos.	Bajo
CE 1.3. Plantear, interpretar, modelar, analizar y resolver problemas, diseño e implementación de circuitos y sistemas electrónicos.	No aporta
CE 1.4. Diseñar, proyectar y calcular circuitos y sistemas digitales.	No aporta
CE 1.5. Diseñar, proyectar y calcular circuitos y sistemas para la generación, recepción, transmisión, procesamiento y conversión de campos y señales para sistemas de comunicación.	No aporta
CE 1.6. Diseñar, proyectar y calcular circuitos y sistemas de control.	No aporta
CE 1.7. Diseñar, proyectar y calcular circuitos y sistemas electrónicos aplicados a la generación, manejo, amplificación, procesamiento, instrumentación y acondicionamiento de energía eléctrica y señales de distinta naturaleza.	No aporta
CE 2.1. Proyectar, dirigir y controlar la construcción, implementación, mantenimiento y operación de lo mencionado anteriormente.	No aporta
CE 3.1. Validar y certificar el funcionamiento, condición de uso o estado de los sistemas mencionados anteriormente.	No aporta
CE 4.1. Proyectar y dirigir lo referido a la higiene y seguridad en la actividad profesional de acuerdo con la normativa vigente.	No aporta
CE 5.1. Diseñar, Proyectar, Calcular y Aplicar dispositivos semiconductores, aplicando estrategias conceptuales y metodológicas asociadas a los principios de cálculo, diseño y simulaciones, con el objeto de optimizar con sentido innovador, responsabilidad profesional y compromiso social, los recursos existentes.	No aporta
CE 6.1. Diseñar, proyectar, calcular, implementar e instalar equipamiento electrónico y su interconexión, aplicados a sistemas de energía, empleando criterios de eficiencia energética y seguridad eléctrica, con responsabilidad económica y social.	No aporta
CE 7.1 Diseñar, Proyectar, Calcular e Instalar sistemas, subsistemas, equipos, componentes,	No aporta

partes, y piezas electrónicas para control, medición, regulación y protección de máquinas eléctricas en redes de baja tensión y sistemas de generación y distribución de energía eléctrica, para brindar soluciones en el marco de las normas vigentes, aplicando criterios de eficiencia energética, seguridad eléctrica, y cuidado del medio ambiente.	
CE 8.1. Diseñar, Proyectar, Calcular e Implementar sistemas, subsistemas, equipos, componentes, partes y piezas electrónicas, de navegación o señalización de vehículos, aplicando criterios técnicos, de seguridad y regulatorios vigentes, y estrategias conceptuales y metodológicas asociadas a los principios de cálculo y diseño con sentido innovador.	No aporta
CE 9.1. Evaluar el impacto ambiental de sistemas, subsistemas, equipos, componentes, partes, y piezas relacionadas con la actividad profesional establecida por sus actividades reservadas y los alcances, de acuerdo con la normativa vigente y aplicando estrategias conceptuales y metodológicas asociadas a los principios de las buenas prácticas profesionales, con el objeto de resguardar el medio ambiente.	No aporta
CE 10.1. Realizar estudios, tareas y asesoramientos, relacionados con la actividad profesional establecida por sus actividades reservadas y los alcances, aportando sus saberes, competencias y/o técnicas, para brindar soluciones óptimas y eficientes en el marco de las normas vigentes y las condiciones técnicas, legales, económicas, humanas y ambientales establecidas.	No aporta
CE 10.2 Realizar pericias, tasaciones y arbitrajes relacionados con su actividad profesional, respetando marcos normativos y jurídicos con el objeto de asesorar a las partes o a los tribunales de Justicia.	No aporta
CE 10.3 Evaluar aspectos económicos, financieros y de inversiones, para la determinación de proyectos, bienes y servicios, relacionados con su actividad profesional, analizando variables micro y macroeconómicas e interpretando la realidad económica en el contexto nacional e internacional.	No aporta

4. Contenidos Mínimos

- Estructura de un sistema computacional.
- Sistemas de numeración y aritmética binaria.
- Interpretación y resolución de problemas, herramientas de representación de algoritmos.
- Lenguaje de programación estructurada.
- Contenedores de datos complejos.
- Uso del lenguaje en aplicaciones de bajo nivel.

5. Objetivos establecidos en el DC

Que los y las estudiantes sean capaces de:

- Aprender las bases y elementos constitutivos de un sistema de cómputo tanto en sus componentes de hardware como de software y sus sistemas de representación numérica.
- Adquirir las bases metodológicas de desarrollo de software que les permitan diseñar algoritmos para resolver problemas determinados, discriminando entre las diferentes alternativas para resolverlo cuál es la más eficiente con fundamento teórico.
- Implementar dichos algoritmos en forma práctica utilizando lenguajes de programación estructurada, empleando diferentes recursos de éste, manejo de

- datos complejos y recursos de bajo nivel.
- Dominar el funcionamiento, las posibilidades, y el empleo de las diferentes herramientas de desarrollo.

6. Resultados de aprendizaje	
Los siguientes resultados de aprendizaje se promueven en el desarrollo de la asignatura	
Identificador de RA	Redacción
RA1	Identificar los elementos básicos desde el punto de vista del hardware y del software considerando una arquitectura genérica para analizar los principios de funcionamiento de las computadoras.
RA2	Reconocer las ventajas tecnológicas de los sistemas de numeración en bases no decimales, particularmente la binaria, teniendo en cuenta la "naturaleza binaria" del hardware para el procesamiento en la computadora.
RA3	Reconocer las herramientas de desarrollo y control de versiones adecuadas a partir de una consigna dada para la resolución de problemas.
RA4	Escribir programas simples pero completos en lenguaje C a partir de problemas de sentencias secuenciales para familiarizarse con tipos fundamentales de datos, operadores aritméticos y funciones de entrada y salida estándar.
RA5	Reconocer patrones de algoritmos básicos en estructuras más complejas para la simplificación de la resolución de problemas.
RA6	Reconocer fragmentos de código factibles de ser modularizados teniendo en cuenta su uso para simplificar los programas y aumentar su legibilidad.
RA7	Reconocer las ventajas del uso de arreglos teniendo en cuenta el beneficio de agrupar elementos de datos relacionados del mismo tipo para almacenar, ordenar y buscar elementos en listas o tablas de valores.
RA8	Utilizar punteros teniendo en cuenta su capacidad de almacenar direcciones de memoria para modificar o usar indirectamente variables o funciones apuntadas por estos elementos cuando son pasados como argumentos en llamados a función.
RA9	Utilizar tipos de datos derivados teniendo en cuenta que permiten hacer uso del hardware a bajo nivel, reduciendo los requerimientos de memoria a la vez que aumentan la legibilidad para introducirse en la escritura software complejo como sistemas operativos, drivers de dispositivos, software de redes, etc.

7. Relación de los RA y las competencias

En la tabla siguiente se indica con X la tributación de cada Resultado de Aprendizaje con las competencias de egreso: específicas, genéricas tecnológicas, sociales, políticas y actitudinales de la carrera.

RA	CE1.1	CE1.2	CE1.3	CE1.4	CE1.5	CE1.6	CE1.7	CE2.1	CE3.1	CE4.1	CE5.1	CE6.1	CE7.1	CE8.1	CE9.1	CE10.1	CE10.2	CE10.3
RA1	X	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
RA2	X	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
RA3	X	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
RA4	X	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
RA5	X	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
RA6	X	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
RA7	X	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
RA8	X	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

RA9	X	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-----	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

RA	CG1	CG2	CG3	CG4	CG5	CG6	CG7	CG8	CG9	CG10	CG11
RA1	-	-	-	X	-	X	X	-	X	-	-
RA2	X	-	-	X	-	X	X	X	X	-	-

RA3	X	-	X	X	X	X	X	X	X	X	-
RA4	X	X	X	X	X	X	X	X	X	X	-
RA5	X	X	X	X	X	X	X	X	X	X	-
RA6	X	X	X	X	X	X	X	X	X	X	-
RA7	X	X	X	X	X	X	X	X	X	X	-
RA8	X	X	X	X	X	X	X	X	X	X	-
RA9	X	X	X	X	X	X	X	X	X	X	-

8. Asignaturas correlativas previas

Para cursar y rendir debe tener cursadas:

- Asignatura/s:
 - No posee materias correlativas previas

Para cursar y rendir debe tener aprobada:

- Asignatura/s:
 - No posee materias correlativas previas

9. Asignaturas correlativas posteriores

Indicar las asignaturas correlativas posteriores:

- Asignatura/s:
 - 8- Informática II. 16- Técnicas Digitales I. 17- Dispositivos electrónicos. 19- Electrónica Aplicada I. 37- Trabajo Final.

10. Programa analítico

Este programa analítico contempla los contenidos mínimos, previstos en el DC vigente, y aquellos que se consideran necesarios para desarrollar los resultados de aprendizaje propuestos.

Unidad Nº: 1

Título: Estructura de una computadora.

Contenidos: Antecedentes históricos. Definición de unidades fundamentales (bit y byte) y sus múltiplos. Definición de memoria. Capacidad de memoria. Tipos de memoria. Buses. Unidad Central del Proceso (CPU). Unidad Aritmética y Lógica (ALU). Unidad de Control. Contador de Programa. Dispositivos de entrada/salida (I/O). Ejecución de instrucciones. Fase de búsqueda de instrucciones. Secuencia de Instrucciones: Programa. Conceptos de hardware y software. Interacción entre ambos elementos.

Herramientas de desarrollo: El compilador. El enlazador (Linker). Su relación con el hardware y el sistema operativo. Introducción al desarrollo de software: Programa fuente, programa objeto, programa ejecutable. Fases de compilación y vinculación de programas en el entorno de un sistema operativo.

Carga horaria: 10h

Unidad Nº: 2

Título: Sistemas de numeración y representación numérica. Aritmética binaria.

Contenidos: Introducción. Los sistemas de numeración y su evolución histórica. Sistemas de numeración decimal, binario, octal y hexadecimal. Pasajes entre sistemas de números enteros y positivos. Representación de números signados: Convenio de signo y magnitud. Convenio de complemento a uno. Convenio de complemento a dos. Operaciones de adición y de sustracción utilizando el convenio de complemento a dos.
Representación de números fraccionales. Notación punto fijo y punto flotante. Precisión y truncado. Errores en notación de punto flotante. Representación según formato IEEE 754. Representación de caracteres: Binario Codificado Decimal (BCD), ASCII, Unicode.

Carga horaria: 10h

Unidad Nº: 3

Título: Introducción al lenguaje C.

Contenidos: Elementos del lenguaje C. Introducción a la sintaxis del lenguaje C. Primer ejemplo: Hola Mundo. Identificación de los elementos de sintaxis. Uso del compilador. Tipos de datos, tamaño y declaraciones. Constantes. Declaraciones. Operadores aritméticos, relacionales y lógicos. Cast. Jerarquía de operadores. Operadores de evaluación (expresiones condicionales). Operadores de asignación. Precedencia. Preprocesador. Archivos de cabecera. Encabezado stdio.h. Entrada y salida con formato. Funciones básicas de entrada salida: scanf, printf, getch, getchar.

Carga horaria: 15h

Unidad Nº: 4

Título: Introducción a la programación estructurada.

Contenidos: Interpretación de enunciados. Ideas sobre programas y datos. Algoritmos. Estructuras básicas de programación. Secuencia. Selección. Iteración. Salto incondicional. Eliminación del salto incondicional. Primer paradigma de programación: programación estructurada. Diagramas de flujo del programa: símbolos de operación, comentarios, decisión, líneas de flujo, conexión. Pseudocódigo. Implementación de algoritmos sencillos.

Carga horaria: 15h

Unidad Nº: 5

Título: Control de flujo en lenguaje C.

Contenidos: Implementación de estructuras de selección: simple (if), doble (if-else) y múltiple (switch-case). Estructuras de repetición: while, do-while y for. Sentencias break y continue. Bucles anidados. Control de flujo con el preprocesador: compilación condicional. Implementación de algoritmos en lenguaje C.

Carga horaria: 15h

Unidad Nº: 6

Título: Funciones en lenguaje C.

Contenidos: Programación modular. Definición de una función. Prototipo de una función. Argumentos. Archivos de cabecera. Variables locales y globales. Clases de almacenamiento: externas, estáticas, registros y automáticas. Reglas de alcance (scope). Proposición return. Llamada por valor y por referencia. Recursividad. Condiciones para implementar funciones recursivas. Ejemplo: factorial de un número. Recursividad vs. Iteración.

Carga horaria: 15h

Unidad Nº: 7

Título: Arreglos en lenguaje C.

Contenidos: Arreglos. Declaración de arreglos. Concepto de vector (arreglo de una dimensión) y de matriz (arreglo de dos dimensiones). Inicialización de los arreglos. Algoritmos de ordenamiento. Ordenamiento por más de un criterio. Algoritmos de búsqueda en arreglos. Algoritmos de búsqueda e inserción en arreglos. Ejemplos de uso de arreglos: desarrollo de histogramas, cálculo de media y desvío. Pasaje de arreglos como argumentos a funciones.

Carga horaria: 15h

Unidad Nº: 8

Título: Punteros en lenguaje C.

Contenidos: Concepto de puntero. Concepto de dirección. Operadores unarios. Aritmética de punteros. Relación entre punteros y arreglos. Inicialización de punteros. Implementación de llamadas a función por referencia. Punteros a puntero. Arreglo de punteros. Ordenamiento de estructuras utilizando arreglo de punteros. Inicialización de punteros y reserva de espacio en memoria: malloc y free. Punteros vs. Arreglos multidimensionales. Argumentos por línea de comandos. Punteros a función.

Carga horaria: 20h

Unidad N°: 9

Título: Estructuras y uniones en C. Campos de bit.

Contenidos: Estructuras de datos. Bases de las estructuras. Sintaxis: Definición de estructura. Acceso a los miembros de una estructura. Punteros a estructuras. Arreglos de estructuras. Funciones y estructuras. Estructuras autoreferenciadas. Typedef. Uniones. Definición. Acceso a los miembros de una unión. Campos de bit. Definición. Acceso a los miembros de un campo de bits. Constantes de enumeración. Aplicaciones relacionadas.

Carga horaria: 15h

Unidad N°: 10

Título: Manejo de archivos en C.

Contenidos: Concepto de flujo de datos (streams). Creación, apertura y cierre de archivos de texto. Archivos binarios. Funciones para lectura y escritura de bajo nivel. Acceso secuencial vs. acceso aleatorio. Uso de las funciones de archivos para acceder a dispositivos de E/S. Concepto "todo es un archivo". Ejemplos.

Carga horaria: 15h

Unidad N°: 11

Título: Uso del lenguaje C en aplicaciones de bajo nivel.

Contenidos: Operadores a nivel de bit. Operadores lógicos. Operadores de desplazamiento. Aplicaciones relacionadas. E/S mapeada en memoria. Puertos GPIO. Modelo de programación de un puerto. Controlador de dispositivo (device driver). Interfaces de programación APIs.

Carga horaria: 15h

Carga horaria por tipo de formación práctica de toda la asignatura

Tipo de formación práctica	Horas reloj
Formación experimental	50
Análisis y resolución de problemas de ingeniería y estudios de casos	10

Bibliografía Obligatoria:

- Deitel, H.M., & Deitel, P.J. (2016). *C How to Program: with an introduction to C++*. (8th ed.). Essex, UK: Pearson.
- Bryant, R.E., & O'Hallaron, D.R. (2011). *Computer Systems: A Programmer's Perspective*. (2nd ed.). Essex, UK: Pearson.
- Harris, S.L., & Harris, D.M. (2015). *Digital Design and Computer Architecture. ARM edition*. (1st ed.). San Francisco, CA: Morgan Kaufmann.

Bibliografía optativa y otros materiales a utilizar en la asignatura:

- Kernighan, B.W., & Ritchie, D.M. (1988). *The C programming language*. (2nd ed.). Englewood Cliffs, NJ: Prentice Hall.

11. Metodología de enseñanza

El principal desafío de la materia es introducir a estudiantes con poca o nula experiencia en programación al uso de la computadora en bajo nivel y a la programación en C. Para alcanzar esta meta se debe ofrecer el tratamiento riguroso de la teoría y la práctica, pero también enseñar de forma paciente los principios de la programación estructurada. Se ofrecerá una recopilación de ejemplos, ejercicios y proyectos que se toman de diferentes campos, a fin de proporcionar al estudiante la oportunidad de resolver problemas interesantes del mundo real. Se hará hincapié en la claridad de la programación, mediante el uso del paradigma de programación estructurada.

Uno de los principales temas a través de esta asignatura será como administrar la complejidad. La técnica crítica para administrar la complejidad es la abstracción: ocultar detalles cuando ellos no son importantes. Un sistema puede verse desde diferentes niveles de abstracción. El dictado se hará siguiendo el camino que va desde lo simple hasta llegar a lo más complejo.

Las clases se organizarán en secciones teóricas/prácticas en el aula y en laboratorio con computadoras. En cada clase los estudiantes deberán aplicar directamente los conceptos vistos anteriormente. Por otra parte, se procurará que cada tema haga referencia y se sustente en los temas ya abordados tanto en el dictado de la asignatura como en las otras asignaturas que el estudiante curse paralelamente.

Las mediaciones pedagógicas utilizadas en este curso serán:

Lección Magistral Participativa: Mientras el Profesor expone los conceptos teóricos, el estudiante atenderá, realizará preguntas, tomará notas, etc. Cuando el profesor detenga su exposición, el estudiante participará en la resolución de un problema en la pantalla. Se debatirá sobre los resultados obtenidos y posibles variaciones y mejoras. Este proceso se repetirá con cada tema.

Resolución de Ejercicios: El estudiante, de manera autónoma, desarrollará las soluciones adecuadas mediante la aplicación de algoritmos provistos por el profesor. El estudiante se retroalimentará con los valores esperados consignados en la Guía de Ejercicios.

Resolución de Problemas: El Estudiante, de manera autónoma o en grupo, descubrirá las características concretas del problema. Esta tarea le supone recuperar los saberes previos y

relacionar los mismos con las actividades que demanda la resolución.

Aprendizaje Cooperativo en Grupos Pequeños: Los estudiantes identificarán las metas del trabajo. Luego para lograrlas, trabajarán en forma colaborativa y participarán de manera comprometida en la consecución de los objetivos de aprendizaje. Al finalizar deberán hacer una presentación relámpago desde el lugar informando los resultados obtenidos.

Aprendizaje Basado en Problemas: El estudiante deberá abordar un problema complejo el cual será una simulación de un problema de la vida profesional que le llevará varios días resolver. La formación necesaria para resolver el problema será limitada, por lo que deberá investigar sobre temas específicos.

Formación Experimental en Laboratorios de Acceso Local: Los estudiantes desarrollarán la Formación Experimental en laboratorios equipados con computadoras con el software necesario para realizar las actividades propuestas.

Trabajo Autónomo: El estudiante asumirá la organización de su trabajo y la responsabilidad del aprendizaje de diferentes saberes según su propio ritmo. Ello implica que deberá autorregular el proceso personal de aprendizaje y deberá conocer cuáles estrategias son posibles de utilizar para afrontar la actividad. El Profesor participará en esta modalidad generando actividades motivadoras, orientando al estudiante cuando sea necesario.

Presentaciones Escritas: El estudiante, en forma individual, presentará un informe los resultados obtenidos en los trabajos prácticos. El estudiante se deberá expresar de manera concisa, clara y precisa, identificando el tema central y los puntos clave, explicando y argumentando, y manejando las herramientas informáticas apropiadas.

Presentaciones Orales: El estudiante, en forma individual, presentará oralmente en un coloquio relámpago los resultados obtenidos en los trabajos prácticos.

Aprendizaje Basado en Proyectos: El estudiante, de manera autónoma deberá movilizar, integrar y aplicar aprendizajes ya desarrollados durante toda la materia y materias del mismo nivel para desarrollar su primer proyecto. Abordarán problemas o temas reales, no simulados.

12. Recomendaciones para el estudio

Se recomienda:

1. La lectura previa del material bibliográfico indicado antes del desarrollo de la teoría ya sea como repaso o como motivador del nuevo desarrollo.
2. Acceder al aula virtual donde estará disponible el material de las distintas unidades tanto de las actividades prácticas como de las teóricas.
3. Mantener comunicación directa con el cuerpo docente de la asignatura a través de los canales institucionales habilitados: correo electrónico institucional y autogestión académica.
4. Predisposición a estudiar y trabajar en equipos respetando distintas formas de pensar.

13. Metodología de evaluación

El modelo de enseñanza basado en competencias implica la aplicación de metodologías e instrumentos de evaluación que permiten conocer, a docentes y estudiantes, el nivel de

desarrollo de las competencias que aborda la asignatura.

Se implementarán evaluaciones formativas y sumativas durante todo el cursado, las cuales se describen a continuación:

- Cada resultado de aprendizaje tendrá:
 - a) Una autoevaluación por parte del estudiante que consistirá en un examen mediado por tecnología de carácter asíncrono.
 - b) Un Trabajo Práctico de aula o de programación acorde al contenido y evaluado por una rúbrica.
- Habrá tres parciales teóricos/prácticos correspondientes a:
 - 1ro) Unidades 1, 2 y 3.
 - 2do) Unidades 4, 5, 6 y 7.
 - 3ro) Unidades 8, 9 y 10
- Habrá un trabajo final integrador para aquellos estudiantes que aspiren a la Aprobación Directa.

A continuación, se detallan todos los Resultados de Aprendizajes con sus contenidos a desarrollar para alcanzarlos, la mediación pedagógica, metodologías y estrategias de evaluación, tiempo en horas reloj.

Resultados de Aprendizaje	Contenidos según programa	Mediación Pedagógica	Metodología y Estrategias de Evaluación	Tiempos en hora reloj
RA 1: Identificar los elementos básicos desde el punto de vista del hardware y del software considerando una arquitectura genérica para analizar los principios de funcionamiento de las computadoras	Unidad 1. Estructura de una computadora.	Lección magistral participativa. Aprendizaje cooperativo en grupos pequeños. Formación experimental en laboratorios de acceso local. Trabajo autónomo. Presentaciones escritas.	Autoevaluación con examen mediado por tecnología de carácter asíncrono. Presentación del TP1.	Presenciales: 8 Teoría y práctica: 6 Laboratorio: 2 Extra áulicas: 4
RA 2: Reconocer las ventajas tecnológicas de los sistemas de numeración en bases no	Unidad 1. Estructura de una computadora. Unidad 2: Sistemas de numeración y representación numérica. Aritmética binaria.	Lección magistral participativa. Aprendizaje cooperativo en grupos pequeños. Resolución de ejercicios. Trabajo autónomo. Presentaciones escritas. Presentaciones orales.	Autoevaluación por medio de examen mediado por tecnología de carácter asíncrono. Presentación del TP2.	Presenciales: 8 Teoría y práctica: 8 Laboratorio: 0 Extra áulicas: 4

<p>decimales, particularmente la binaria, teniendo en cuenta la “naturaleza binaria” del hardware para el procesamiento en la computadora</p>				
<p>RA 3: Reconocer las herramientas de desarrollo y control de versiones adecuadas a partir de una consigna dada para la resolución de problemas.</p>	<p>Unidad 3. Introducción al lenguaje C. Unidad 4. Introducción a la programación estructurada. Unidad 5. Control de flujo en lenguaje C. Unidad 6. Funciones en lenguaje C. Unidad 7. Arreglos en lenguaje C. Unidad 8. Punteros en lenguaje C. Unidad 9. Estructuras y uniones en C. Campos de bit. Unidad 10. Uso del lenguaje C en aplicaciones de bajo nivel. Unidad 11. Manejo de archivos en C.</p>	<p>Lección magistral participativa. Resolución de problemas. Resolución basada en problemas.</p>	<p>Autoevaluación por medio de examen mediado por tecnología de carácter asíncrono.</p>	<p>Presenciales: 6 Teoría y práctica: 3 Laboratorio: 3 Extra áulicas: 4</p>

<p>RA 4: Escribir programas simples pero completos en lenguaje C a partir de problemas de sentencias secuenciales para familiarizarse con tipos fundamentales de datos, operadores aritméticos y funciones de entrada y salida estándar.</p>	<p>Unidad 3. Introducción al lenguaje C.</p>	<p>Lección magistral participativa. Aprendizaje cooperativo en grupos pequeños. Resolución de ejercicios. Resolución de problemas. Aprendizaje basado en problemas. Formación experimental en laboratorios de acceso local. Trabajo autónomo. Presentaciones escritas.</p>	<p>Autoevaluación por medio de examen mediado por tecnología de carácter asíncrono. Presentación del TP3.</p>	<p>Presenciales: 12 Teoría y práctica: 6 Laboratorio: 6 Extra áulicas: 12</p>
<p>RA 5: Reconocer</p>	<p>Unidad 4. Introducción a la programación estructurada. Unidad 5. Control de flujo en</p>	<p>Lección magistral participativa. Aprendizaje cooperativo en grupos pequeños.</p>	<p>Autoevaluación por medio de examen mediado por tecnología de carácter</p>	<p>Presenciales: 12 Teoría y práctica: 6 Laboratorio: 6</p>

<p>patrones de algoritmos básicos en estructuras más complejas para la simplificación de la resolución de problemas</p>	<p>lenguaje C. Unidad 6. Funciones en lenguaje C. Unidad 7. Arreglos en lenguaje C. Unidad 8. Punteros en lenguaje C. Unidad 9. Estructuras y uniones en C. Campos de bit. Unidad 10. Uso del lenguaje C en aplicaciones de bajo nivel. Unidad 11. Manejo de archivos en C.</p>	<p>Resolución de ejercicios. Resolución de problemas. Aprendizaje basado en problemas. Formación experimental en laboratorios de acceso local. Trabajo autónomo. Presentaciones escritas.</p>	<p>asíncrono. Presentación de TP4 al TP11.</p>	<p>Extra áulicas: 6</p>
<p>RA 6: Reconocer fragmentos de código factibles de ser modularizados teniendo en cuenta su uso para simplificar los programas y aumentar su legibilidad</p>	<p>Unidad 5. Control de flujo en lenguaje C. Unidad 6. Funciones en lenguaje C. Unidad 7. Arreglos en lenguaje C. Unidad 8. Punteros en lenguaje C. Unidad 9. Estructuras y uniones en C. Campos de bit. Unidad 10. Uso del lenguaje C en aplicaciones de bajo nivel. Unidad 11. Manejo de archivos en C.</p>	<p>Lección magistral participativa. Aprendizaje cooperativo en grupos pequeños. Resolución de ejercicios. Resolución de problemas. Aprendizaje basado en problemas. Formación experimental en laboratorios de acceso local. Trabajo autónomo. Presentaciones escritas. Presentaciones orales.</p>	<p>Autoevaluación por medio de examen mediado por tecnología de carácter asíncrono. Presentación del TP6 al TP11</p>	<p>Presenciales: 12 Teoría y práctica: 6 Laboratorio: 6 Extra áulicas: 10</p>
<p>RA 7: Reconocer las ventajas del uso de arreglos</p>	<p>Unidad 7. Arreglos en lenguaje C. Unidad 8. Punteros en lenguaje C. Unidad 9. Estructuras y</p>	<p>Lección magistral participativa. Aprendizaje cooperativo en grupos pequeños. Resolución de ejercicios. Resolución de problemas.</p>	<p>Autoevaluación por medio de examen mediado por tecnología de carácter asíncrono. Presentación del TP7 al</p>	<p>Presenciales: 16 Teoría y práctica: 8 Laboratorio: 8 Extra áulicas: 10</p>

<p>teniendo en cuenta el beneficio de agrupar elementos de datos relacionados del mismo tipo para almacenar, ordenar y buscar elementos en listas o tablas de valores</p>	<p>uniones en C. Campos de bit. Unidad 10. Uso del lenguaje C en aplicaciones de bajo nivel. Unidad 11. Manejo de archivos en C.</p>	<p>Aprendizaje basado en problemas. Formación experimental en laboratorios de acceso local. Trabajo autónomo. Presentaciones escritas. Presentaciones orales.</p>	<p>TP11.</p>	
<p>RA 8: Utilizar punteros teniendo en cuenta su capacidad de almacenar direcciones de memoria para modificar o usar indirectamente variables o</p>	<p>Unidad 8. Punteros en lenguaje C. Unidad 9. Estructuras y uniones en C. Campos de bit. Unidad 10. Uso del lenguaje C en aplicaciones de bajo nivel. Unidad 11. Manejo de archivos en C.</p>	<p>Lección magistral participativa. Aprendizaje cooperativo en grupos pequeños. Resolución de ejercicios. Resolución de problemas. Aprendizaje basado en problemas. Formación experimental en laboratorios de acceso local. Trabajo autónomo. Presentaciones escritas. Presentaciones orales.</p>	<p>Autoevaluación por medio de examen mediado por tecnología de carácter asíncrono. Presentación del TP8 al TP11.</p>	<p>Presenciales: 22 Teoría y práctica: 10 Laboratorio: 12 Extra áulicas: 20</p>

<p>funciones apuntadas por estos elementos cuando son pasados como argumentos en llamados a función</p>				
<p>RA 9: Utilizar tipos de datos derivados teniendo en cuenta que permiten hacer uso del hardware a bajo nivel, reduciendo los requerimientos de memoria a la vez que aumentan la legibilidad para introducirse en la escritura</p>	<p>Unidad 8. Punteros en lenguaje C. Unidad 9. Estructuras y uniones en C. Campos de bit. Unidad 10. Uso del lenguaje C en aplicaciones de bajo nivel. Unidad 11. Manejo de archivos en C.</p>	<p>Lección magistral participativa. Aprendizaje cooperativo en grupos pequeños. Resolución de ejercicios. Resolución de problemas. Aprendizaje basado en problemas. Aprendizaje basado en proyectos. Formación experimental en laboratorios de acceso local. Trabajo autónomo. Presentaciones escritas. Presentaciones orales.</p>	<p>Autoevaluación por medio de examen mediado por tecnología de carácter asíncrono. Presentación del TP9 al TP11.</p>	<p>Presenciales: 24 Teoría y práctica: 12 Laboratorio: 12 Extra áulicas: 20</p>

software complejo como sistemas operativos, drivers de dispositivos, software de redes, etc.				
--	--	--	--	--

14. Condiciones de aprobación

La evaluación se realizará mediante tres exámenes parciales que comprenden desarrollos teóricos y prácticos correspondientes a:

- 1ro) Unidades 1, 2 y 3.
- 2do) Unidades 4, 5, 6 y 7.
- 3ro) Unidades 8, 9 y 10.

Los exámenes deberán exigir al estudiante aplicar razonamientos y desarrollos explícitos de cada tema. Las partes teóricas y prácticas en cada parcial estarán claramente diferenciadas. El sistema de calificación para las evaluaciones será de 1 a 10, y se aprobará con el 60 % del examen total resuelto.

Habrá un trabajo final integrador para aquellos estudiantes que aspiren a la aprobación directa.

Para regularizar

Para lograr la condición de **Regular** se deberán:

- 1) Aprobar los exámenes parciales, en instancias directas o a través de los recuperatorios correspondientes
- 2) Aprobar el 80% de todos los trabajos prácticos antes de la última semana de clases (es una condición obligatoria haber aprobado el TP5 y TP8)
- 3) Superar el 75% de asistencia.

Para Aprobación Directa

Para lograr la condición de **Aprobación Directa** se deberán:

- 1) Aprobar los exámenes parciales superando el 60% tanto en la parte teórica como en la parte práctica
- 2) Aprobar el 100% de los trabajos prácticos, incluyendo el TP10 (Integrador)
- 3) Superar el 75% de asistencia

Recuperatorio

El estudiante que fuere aplazado en **una** de las evaluaciones podrá rendir un examen recuperatorio. Los temas a evaluar serán coincidentes con los temas correspondientes al examen no aprobado. En este caso se pueden dar las siguientes situaciones:

- El examen es aprobado en cuyo caso se logra la condición de **Regular**.
- El examen es desaprobado en cuyo caso la nueva condición es **Libre**.

En caso de alcanzar la condición de Regular la calificación del recuperatorio reemplazará a la del examen desaprobado, momento en el cual se podría alcanzar la condición de Aprobado Directo si se cumplieran los requisitos de dicha condición.

Por el contrario, quien haya sido aplazado en **dos o tres** parciales podrá rendir un parcial recuperatorio integrador donde se pueden dar las siguientes situaciones:

- El examen es aprobado en cuyo caso se logra la condición de **Regular**.
- El examen es desaprobado en cuyo caso la nueva condición es **Libre**.

Si el examen es aprobado, la calificación obtenida reemplaza todas las calificaciones desaprobadas (dos o tres) pero la condición Regular es la definitiva (en la medida que se cumplan los requisitos para tal condición).

Si luego de las tres instancias de evaluación durante el año, con los tres parciales aprobados, uno y solo uno de los parciales no alcanzara el 60% en la parte teórica o en la parte práctica, se

dará la opción al estudiante de recuperar en una única instancia definitiva donde se pueden dar las siguientes situaciones:

- El examen es aprobado superando el 60% tanto en la parte teórica como en la parte práctica logrando la **Aprobación Directa** (en la medida que se cumplan los requisitos para tal condición).
- El examen es aprobado, pero no se alcanza el 60% en la parte teórica o en la parte práctica, en cuyo caso se mantiene la condición de **Regular** (en la medida que se cumplan los requisitos para tal condición).
- El examen es desaprobado, en cuyo caso la nueva condición es **Libre**.

Estudiantes que asistan a todas las instancias de evaluación propuestas, pero que en las mismas no alcancen los requisitos previstos para la regularidad, tendrán la condición de **Libre**.

Estudiantes que no asistan a todas las instancias de evaluación propuestas, no presenten las actividades de formación práctica, o no cumplan con la asistencia requerida para la regularización, tendrán la condición de **Abandonó**.

15. Modalidad de examen

Los estudiantes con condición **Regular** deberán rendir un examen teórico/práctico de carácter escrito con diversas modalidades de respuestas como: opciones múltiples, completar espacios en blancos, verdadero/falso con justificación y desarrollo de contenido solicitado.

Todos los temas estarán acompañados de sus respectivos puntajes. Para aprobar se deberá obtener un mínimo de 60% sobre el total del examen.

16. Recursos necesarios

Los espacios físicos necesarios son:

- Aula
- Laboratorio Informático

Recursos tecnológicos de apoyo:

- Proyector multimedia
- PC con software de desarrollo integrado para programar en C (Ej. Zinjal, VS Code, etc.)
- Placas de desarrollo Raspberry Pi

Anexo I: Plantel docente de la asignatura				
Titular	Luis Eduardo Toledo	Dedicación:	2DS	
Asociado	Claudio Paz	Dedicación:	Full time	
Adjunto:	Especifique Nombre y Apellido completo.	Dedicación:		
Jefe de Trabajos Prácticos	Martin Nievas	Dedicación:	2 DS	
Jefe de Trabajos Prácticos	Silvia Carrera	Dedicación:	1 DS	
Auxiliar de 1ra.		Dedicación:		
Auxiliar de 2da.	Especifique Nombre y Apellido completo.	Dedicación:	Especifique la cantidad de dedicciones.	

FIRMA (Jefe o encargado de cátedra).

Anexo II: Cronograma de clases/trabajos prácticos/evaluaciones (por comisión)

COMISIÓN: Indique la comisión.			
Nro. de Semana	Fecha	Tema	Tipo de Actividad
1	Indique la fecha	Describa el tema trabajado	Seleccione el tipo de actividad.

FIRMA (de cada docente que conforman la comisión).