

**Carrera: Ingeniería Electrónica**  
**Asignatura: Informática II**  
**Planificación a partir del Ciclo Lectivo 2025**

1. Datos administrativos de la asignatura			
Nivel en la carrera	2	Duración	Anual
Plan	2023		
Bloque curricular:	Tecnologías Aplicadas		
Carga horaria presencial semanal (hs. cátedra):	5	Carga Horaria total (hs. reloj):	120
Carga horaria no presencial semanal (hs. reloj) (si correspondiese)	0	% horas no presenciales (hs. reloj) (si correspondiese)	0

2. Presentación, Fundamentación
<p>Según la Ordenanza del Consejo Superior de UTN N° 1849, la asignatura “Informática II” forma parte del Bloque de las Tecnologías Aplicadas y del Área de las Técnicas Digitales de la carrera de Ingeniería Electrónica.</p> <p>Según las Competencias Específicas consignadas en el Diseño Curricular de la carrera de Ingeniería Electrónica, la asignatura “Informática II” debe lograr que el estudiante adquiera las capacidades para el desarrollo de software asociado a equipos y dispositivos de transmisión y/o procesamiento de señales analógicas y digitales, y de comunicación de datos de sistemas de cómputo de propósito general (PC) y/o específico (Sistemas Embebidos). Con vista a ello, en las diversas unidades temáticas se enseñará a interpretar, modelar y resolver problemas de Ingeniería mediante la utilización de las Ciencias Básicas e Informática.</p>

3. Relación de la asignatura con las competencias de egreso de la carrera
<p>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).</p>

<b>Competencias</b>	<b>Nivel</b>
<b>Competencias genéricas tecnológicas (CG):</b>	
CG.1. Identificar, formular y resolver problemas de ingeniería.	Bajo
CG.2. Concebir, diseñar y desarrollar proyectos de ingeniería.	No aporta
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.	Bajo
CG.5. Contribuir a la generación de desarrollos tecnológicos y/o innovaciones tecnológicas.	No aporta
<b>Competencias genéricas sociales, políticas y actitudinales (CG)</b>	
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.	Bajo
CG.9. Aprender en forma continua y autónoma.	Bajo
CG.10. Actuar con espíritu emprendedor.	No aporta
<b>Competencias Específicas de la carrera</b>	
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.	Medio
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.	Bajo
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, 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.	No aporta
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

Los Contenidos Mínimos de la asignatura “Informática II” indicados en el Diseño Curricular de Ingeniería Electrónica –plan 2023– de la Ordenanza del C.S. No 1849 de la UTN son:

- Programación avanzada utilizando el lenguaje de programación introducido en Informática I.
- Introducción a las estructuras dinámicas de datos.
- Herramientas de construcción de software.
- Control de periféricos y/o comunicación con microcontroladores.
- Desarrollo de aplicaciones usando interfaces de usuario gráficas.
- Desarrollo de aplicaciones utilizando cálculo numérico.
- Introducción a un lenguaje de programación avanzado.
- Introducción a los fundamentos de los sistemas operativos avanzados.

## 5. Objetivos establecidos en el DC

El Diseño Curricular de Ingeniería Electrónica –plan 2023– de la Ordenanza del C.S. No 1849 de la UTN establece que los y las estudiantes sean capaces de:

- Diseñar y desarrollar aplicaciones para resolución de problemas complejos aplicados a ingeniería electrónica.
- Utilizar fluidamente herramientas de documentación, de control de versiones, y de automatización de la construcción de un programa o biblioteca a partir de las fuentes.

## 6. Resultados de aprendizaje

Los siguientes resultados de aprendizaje se promueven en el desarrollo de la asignatura

Identificador de RA	Redacción
RA1	Utilizar el sistema operativo GNU/Linux teniendo en cuenta que la línea de comandos resulta la herramienta más adecuada para el desarrollo de software.
RA2	Utilizar el compilador del lenguaje C y herramientas auxiliares teniendo en cuenta las etapas de construcción, la documentación y control de versiones para el desarrollo de programas de PC y de microcontroladores.
RA3	Desarrollar programas en lenguaje C con acceso a registros de hardware teniendo en cuenta el mapa de memoria para la programación de periféricos de microcontroladores.
RA4	Desarrollar programas de PC y microcontroladores teniendo particular consideración del tipo de dato a utilizar con la finalidad de intercambiar información entre ellos a través del puerto serie.
RA5	Desarrollar programas en el lenguaje C++ de línea de comandos y con interfaz gráfica de usuario según el paradigma de programación orientada a objetos para el procesamiento y comunicación de datos.
RA6	Desarrollar programas de cálculo numérico en base al lenguaje C y C++ para su aplicaciones en ingeniería.
RA7	Reconocer las estructuras dinámicas de datos teniendo en cuenta tanto estructuras lineales como no lineales para su aplicaciones en ingeniería.

### 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	-	-	-	-	-	-	-	-	-	-
RA2	-	-	-	-	-	-	-	X	-	-	-	-	-	-	-	-	-	-
RA3	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
RA4	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
RA5	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
RA6	X	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
RA7	X	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

RA	CG1	CG2	CG3	CG4	CG5	CG6	CG7	CG8	CG9	CG10
RA1	-	-	-	X	-	X	X	-	X	-
RA2	-	-	-	X	-	X	X	-	-	-
RA3	X	-	X	-	-	X	X	X	-	-
RA4	X	-	X	X	-	X	X	X	-	-
RA5	X	-	X	X	-	X	X	X	-	-
RA6	-	-	-	-	-	-	-	-	X	-
RA7	-	-	-	-	-	-	-	-	X	-

### 8. Asignaturas correlativas previas

Para cursar y rendir debe tener cursadas:

- Asignatura/s:
  - 1 – Informática I
  - 2 – Álgebra y Geometría Analítica
  - 3 – Análisis Matemático I

Para cursar y rendir debe tener aprobada:

- Asignatura/s:
  -

### 9. Asignaturas correlativas posteriores

Indicar las asignaturas correlativas posteriores:

- Asignatura/s:
  - 18 – Legislación
  - 22 – Técnicas Digitales II
  - 29 – Técnicas Digitales III
  - 36 – Economía
  - 37 – Proyecto 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 N°: 1

#### Título: Introducción al sistema operativo GNU/Linux

#### Contenido:

Descripción del sistema operativo GNU/Linux. Historia y contexto. Distribuciones. Licencias. Conceptos de software y hardware de código abierto. Componentes de un sistema Linux (Kernel, Shell, entornos gráficos, etc.). La Shell de Linux, ingreso y egreso. Usuarios y grupos. Páginas de manuales. Variables de entorno. Programación en C bajo el SO GNU/Linux. Ejecución de programa desde la Shell y paso de parámetros a la función main. Sistemas de archivos y permisos, directorio raíz, actual y padre. Caminos/rutas/paths absolutos y relativos. Atributos y tipos de archivos. Permisos y modificación de permisos. Redirección de salida

estándar, error y entrada estándar. Pipes/tuberías (IPC). Procesos, definición, PID, PPID. Procesos en primer plano y segundo plano. Organización de un proceso en memoria, segmentos (text, data, stack y heap). Señales (IPC).

**Carga horaria por Unidad:** 20 horas.

## **Unidad N°: 2**

### **Título: Herramientas para el desarrollo de software**

#### **Contenido:**

Herramientas de compilación, gcc y g++. Programas con varios archivos fuentes. Separación en archivos .h y .c (.cpp). Etapas de compilación: preprocesado, compilado, ensamblado y enlazado. Construcción por etapas, banderas del compilador. El preprocesador: macros y compilación condicional; macros con valor, paso de valor por línea de comandos. Uso y construcción de bibliotecas estáticas y dinámicas. Banderas del compilador para incluir archivos de cabecera y enlazado con bibliotecas. Banderas del compilador de estándares del lenguaje C. Construcción de proyectos de software con la herramienta make. Archivo Makefile y reglas: objetivos, listas de dependencias y comandos. Herramientas de control de versiones y documentación.

**Carga horaria por Unidad:** 20 horas.

## **Unidad N°: 3**

### **Título: Programación de microcontroladores en lenguaje C**

#### **Contenido:**

Introducción a microcontroladores y sistemas embebidos. Arquitectura de un microcontrolador ( $\mu$ C), microprocesador vs. microcontrolador. Herramientas de construcción: toolchain y compilador cruzado. Herramienta de grabación de la memoria del  $\mu$ C. Programación de microcontroladores en lenguaje C. Programación de E/S digitales mediante el acceso a registro. Calificador volatile para el acceso a hardware. Calificador const. Calificador const con punteros. Operadores a nivel de bits.

**Carga horaria por Unidad:** 15 horas.

## **Unidad N°: 4**

### **Título: Comunicación entre un microcontrolador y la PC**

#### **Contenido:**

Comunicación de la PC con otros dispositivos/periféricos, comunicación paralela y serie. Transmisión síncrona y asíncrona. Receptor/transmisor asíncrono universal (UART). Trama de comunicación serie asíncrona (bits de inicio, datos, paridad y parada). Descripción de trama

(8N1, 5E2, etc.). Errores de trama. Half-Duplex, Full-Duplex y Simplex. El estándar RS-232. Principales características eléctricas, mecánicas y funcionales. Puerto serie en la PC, adaptadores a USB-serie. Dispositivos tty y acceso a dispositivos. Aplicaciones de terminal de línea de comandos y gráficas. Programación del puerto serie de la PC en GNU/Linux, configuración con termios. Programación en lenguaje C de la UART de un microcontrolador mediante el acceso a registros. Comunicación serie entre un microcontrolador y la PC.

**Carga horaria por Unidad:** 25 horas.

## **Unidad N°: 5**

### **Título: El lenguaje de programación C++**

#### **Contenido:**

El lenguaje de programación C++. Introducción y estándares. Entrada/salida. Tipo de dato bool. Referencias y parámetros por referencia. Funciones inline. Sobrecarga de funciones. Plantillas de funciones. Asignación dinámica de memoria, operadores new y delete. Programación orientada a objetos. Conceptos de clases y objetos. Definición de clases. Separación entre la interfaz y la implementación. Espacio de nombres. Reutilización de software, ejemplos de bibliotecas de clases. Control de acceso a miembros. Funciones miembros. Constructor y destructor. Constructor copia. El puntero this. Uso del calificador const: datos miembros const, funciones miembros const y objetos const. Inicialización de datos miembros const. Composición de clases. Herencia de clases. Plantillas de clases. Biblioteca estándar de C++ (cadenas, vectores, etc.). Funciones amigas. Miembros de clase protected. Diagramas de clases UML. Sobrecarga de operadores, función operador. Sobrecarga de operadores de inserción y extracción de flujos. Entrada/salida estándar (cout/cin).

**Carga horaria por Unidad:** 40 horas.

## **Unidad N°: 6**

### **Título: Programación de entornos gráficos**

#### **Contenido:**

Introducción a la programación de entornos de interfaz gráfica de usuario (GUI). Ejemplo de aplicaciones CLI y GUI. Herramientas: qmake, QtCreator y QtDesigner. Programación gráfica con las bibliotecas Qt. Programación por eventos. Programas ejemplos con Widget básicos. Funciones setter y getter de atributos de objetos. Jerarquía de clases de Qt. Aplicaciones como clases C++. Señales y ranuras, ejemplos. Programación de puerto serie de la PC con las bibliotecas Qt.

**Carga horaria por Unidad:** 20 horas.

## Unidad N°: 7

### Título: Aplicación de la PC al cálculo numérico

#### Contenido:

Representaciones numéricas en una PC. Aproximaciones y errores. Expansión en series. Generación de números aleatorios. Búsqueda de raíces, diferenciación e integración. Álgebra matricial. Bibliotecas de funciones y clases para cálculo numérico.

**Carga horaria por Unidad:** 10 horas.

## Unidad N°: 8

### Título: Introducción a las estructuras dinámicas de datos

#### Contenido:

Estructuras dinámicas de datos. Estructuras lineales y no lineales. Listas enlazadas, operaciones permitidas. Listas doblemente enlazadas y anillo lógico. Estructura de pila y operaciones con una pila (push y pop). Estructura de colas y operaciones permitidas (enqueue y dequeue). Árboles, definición y características: nodo padre, sub-árboles, nodo hoja, camino, niveles y grado. Árbol binario, definición recursiva. Árbol de búsqueda binario. Recorrido de un árbol binario.

**Carga horaria por Unidad:** 10 horas.

### Carga horaria por tipo de formación práctica de toda la asignatura

Tipo de formación práctica	Horas reloj
Formación experimental	42 hs
Análisis y resolución de problemas de ingeniería y estudios de casos	20 hs
Formulación, análisis y desarrollo de proyectos.	0 hs

### Bibliografía Obligatoria:

- Perez Paina, G. (2022). Apunte de cátedra: Unidad "Control de periféricos". *Comunicación serie entre PC y placa Arduino*. <https://www.profesores.frc.utn.edu.ar/electronica/informaticaii/>
- Deitel, H., Deitel, P., & Pérez, J. (2004). *Cómo programar en C/C++ y Java*. Pearson Educación.
- Stroustrup, B., & García, J. (2001). *El lenguaje de programación C++*. Pearson Educación.
- Prieto, S., & Población, O. (2008). *Linux. Guía Práctica*. RA-MA S.A. Editorial y Publicaciones.

- Allende, S., & Serna, M. (2021). *Sistemas Operativos: Linux*. Jorge Sarmiento Editor – Universitatis.
- Gough, B. (rev.2011). *Una Introducción a GCC (para los compiladores de GNU gcc y g++)*. GNU. <https://www.nongnu.org/gccintro-es/gccintro.es.pdf>
- Diván, M. (2007). *Algoritmos y estructuras de datos con C/C++*. Santa Rosa, Argentina: Universidad Nacional de La Pampa.
- Aguilar, L. (2006). *Programación en C++*. Algoritmos, estructuras de datos y objetos. McGraw-Hill Interamericana de España S.L.
- Ascheri, M. E., & Pizzaro, R. A. (2008). *Cálculo Numérico*. EdUNLPam.
- S. D. Rajan. (2021). *Object-Oriented Numerical Methods via C++*. 2nd Edition. <https://faculty.engineering.asu.edu/rajan/object-oriented-numerical-methods-via-c/>

### **Bibliografía optativa y otros materiales a utilizar en la asignatura:**

- Schildt, H. (2004). *C Manual de Referencia*. McGraw-Hill Interamericana de España S.L.
- Matthew, N., & Stones, R. (2011). *Beginning linux programming*. Wiley.
- Press, W. H., Teukolsky, S. A., Vetterling, W. T., & Flannery, B. P. (2007). *Numerical Recipes in C. The Art of Scientific Computing*. Cambridge University Press. <http://www.cambridge.org/9780521880688>

## **11. Metodología de enseñanza**

El modelo de formación que le da sustento al diseño curricular de la carrera Ingeniería Electrónica debe atender simultáneamente varias dimensiones (Ordenanza C.S. N° 1849) de las cuales vale mencionar aquí el balance entre teoría y práctica, tanto en la incorporación de habilidades, conceptos e información, como en el enfoque para la resolución de problemas no explícitos. Esto con el fin de construir un aprendizaje gradual, partiendo de un saber anterior y aprendiendo a través de actividades de laboratorio, buscando así el saber hacer y el aprender a aprender. Teniendo esto en cuenta, la propuesta metodológica se articula en torno a la resolución de problemas de complejidad creciente, para lo cual se prevé la presentación de casos que enfrenten a los estudiantes a situaciones de la vida profesional.

Para obtener resultados satisfactorios, la metodología propuesta incluye los siguientes recursos didácticos que apoyarán los procesos de enseñanza y aprendizaje. Los recursos se diferenciarán según el tipo de actividad que apoyen, a saber: desarrollo de clases, gestión, comunicación y evaluación. Se prevé que las clases se desarrollen mayormente en el laboratorio y adquieran un formato teórico-práctico. De este modo, cada clase será planificada en dos segmentos: el primero de ellos de carácter expositivo, presentará los temas y contenidos del día, y el segundo estará

orientado a la práctica del estudiante. Los recursos didácticos que apoyarán el segmento expositivo de la clase consistirán en filminas que presentarán los contenidos de manera resumida, ordenada y en relación, y situaciones problemáticas que ejemplificarán tales contenidos y que adelantarán el tipo de trabajo esperado en el segmento práctico. Además de las filminas, el desarrollo de las clases se apoyará con distintos programas informáticos, que serán pertinentes según la unidad temática que se desarrolle; por ejemplo: la shell de Linux, editores de código fuente, compiladores, IDEs, etc. Por otro lado, el segmento práctico se desarrollará en torno a las guías de ejercicios, resolución de problemas y estudio de casos. Las filminas y las guías de actividades prácticas estarán disponibles en la página web de la asignatura, en conjunto con contenidos complementarios e información adicional incluyendo documentos de referencia, links a páginas webs, videos, etc.

A continuación se presenta una breve descripción de los recursos didácticos a utilizar:

- a. Lección magistral participativa: mientras el profesor expone un concepto teórico y realiza demostraciones el estudiante atiende, realiza preguntas, toma notas, copia un esquema, etc. Luego de que el profesor detenga su exposición el estudiante realiza un ejercicio rutinario y/o una evaluación diagnóstico breve en pequeños grupos de debate sobre una situación planteada por el profesor. Esta actividad facilita la comprensión de temas complejos, sintetizados en forma estructurada y organizada.
- b. Resolución de ejercicios: el estudiante (individual o grupalmente) desarrolla las soluciones adecuadas o correctas mediante la ejercitación de rutinas, la aplicación de procedimientos y la interpretación de los resultados a partir de las consignas provistas por el profesor o auxiliar docente. El estudiante no debe memorizar ningún saber, ya que en todo momento tiene a disposición diferentes recursos (presentaciones de clase, apuntes, libros de texto, guías de ejercicios resueltos, conexión a internet, etc.). El docente debe enseñar tanto el saber conocer como el saber hacer necesarios para la resolución de los ejercicios, asegurándose de que los estudiantes tengan una o más referencias que le permitan verificar los resultados esperados. Esta actividad complementa la exposición magistral.
- c. Resolución de problemas: esta tarea le supone recuperar los saberes previos y relacionarlos con las actividades que demanda la resolución. Selecciona un procedimiento de resolución adecuado al contexto del problema, eligiendo entre varias alternativas de acuerdo a las características del mismo. Esta actividad le permite al estudiante desarrollar

el pensamiento complejo y promueve el desarrollo de estrategias de planificación, organización y gestión de tiempos y recursos de aprendizaje

- d. Estudio de casos: los estudiantes, trabajando de forma grupal, deben analizar en forma intensiva un problema propuesto por el profesor y resolverlo aplicando diferentes procedimientos alternativos de solución. Los estudiantes son guiados hacia el pensamiento complejo por el profesor quien debe conocer en profundidad el caso preparando actividades para el grupo, observando, reconduciendo el análisis y si fuera necesario realizando alguna síntesis final.

Como recursos didácticos que mediarán la vinculación de conceptos y problemas se elaborarán guías de actividades de complejidad creciente. Cada guía contará con ejercicios básicos de aplicación de los conceptos aprendidos y problemas cuya solución requiera la integración de tales conceptos de una manera creativa. Al mismo tiempo se pretende que estas guías estimulen la capacidad de auto-aprendizaje en los estudiantes, en tanto se verán en situación de indagar aquellos conceptos requeridos por el caso y que no fueron adelantados en clase a tal fin.

Como parte de la propuesta metodológica el docente se posiciona en el rol de orientador en la resolución de las actividades, considerando que es en este trabajo cuando el estudiante reconocerá las dificultades propias del proceso de aprendizaje.

## 12. Recomendaciones para el estudio

Se recomienda la asistencia a clases tanto para asimilar los conceptos teóricos de cada uno de los temas del programa analítico como para que los docentes puedan acompañar el desarrollo de las actividades prácticas propuestas. Es importante también desarrollar un trabajo cooperativo. Se recomienda complementar el desarrollo de actividades y estudio de los conceptos teóricos con horas adicionales a las asignadas a clases, como así también acceder periódicamente al Autogestión de la Facultad y al Aula Virtual, que serán los medios de comunicación utilizando durante el cursado de la materia.

## 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 prevén técnicas de evaluación sumativa y formativa. La evaluación sumativa permitirá calificar los rendimientos para la acreditación, mientras que la evaluación formativa se realizará durante los procesos de enseñanza y aprendizaje con la finalidad de reorientarlos. La evaluación formativa permitirá también que los estudiantes puedan reconocer sus procesos de aprendizaje y tomar decisiones al respecto, adquiriendo mayor autonomía en los mismos. Las técnicas de evaluación propuestas se orientan a:

- Calificar los aprendizajes de los estudiantes para acreditar la aprobación de la asignatura en las condiciones de regular o aprobación directa.
- Reconocer las principales dificultades de los estudiantes en sus procesos de aprendizaje, para ofrecer instancias de apoyo y realizar ajustes en la programación de las semanas asignadas a cada unidad temática.
- Observar los contenidos que presenten mayores problemas de comprensión, lo que permitirá ajustar la propuesta metodológica planteando nuevos recursos didácticos.
- Valorar los recursos didácticos para reeditar aquellos que mejor hayan contribuido a la construcción de conocimientos y revisar los que hayan presentado dificultades en su implementación o que no se consideren del todo pertinentes.

En relación con lo anterior, las técnicas de evaluación que propone la cátedra son:

1. Evaluaciones opcionales:

- Estas evaluaciones consistirán en: i) cuestionarios múltiples-opciones, ii) actividades de autoevaluación y/o iii) coevaluación.
- La evaluación mediante cuestionarios múltiples-opciones se llevará a cabo a comienzo de la clase teórica y consistirá de 20 preguntas de múltiples-opciones con 5 opciones cada una y una sola correcta. Cada evaluación tendrá una duración de 20 minutos. Esta evaluación será síncrona y presencial, mediada por la tecnología (Aula Virtual).
- Las actividades de autoevaluación y coevaluación se llevarán a cabo de forma asíncrona mediada por la tecnología (Aula Virtual) con un plazos de 15 días para su entrega luego de su formulación.

2. Exámenes teóricos-prácticos:

- La parte teórica de la evaluación consiste en preguntas conceptuales de múltiples opciones (similares a las aquellas correspondientes a las Evaluaciones opcionales) con un puntaje de 3/10; más ejercicios de carácter conceptual con puntaje de 2/10.

- La parte práctica de la evaluación consiste en resolver ejercicios y/o problemas de programación similares a los vistos en clases y tendrá un puntaje de 5/10.
- Los exámenes serán síncronos y presenciales, mediados por la tecnología (Aula Virtual).
- Se consideran tres instancias de evaluación: 2 parciales y 1 integradora.
- Es posible la recuperación de los 2 exámenes parciales si se han realizado todas las Evaluaciones continuas opcionales.

### 3. Trabajo práctico grupal:

- Se enmarca dentro de la metodología de estudios de casos mencionada anteriormente.
- Será calificado como aprobado/no aprobado.

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
<p>RA 1: Utilizar el sistema operativo GNU/Linux teniendo en cuenta que la línea de comandos resulta la herramienta más adecuada para el desarrollo de software.</p>	<p><b>Unidad 1: Introducción al sistema operativo GNU/Linux.</b></p> <ul style="list-style-type: none"> <li>- Sistema operativo GNU/Linux. Componentes.</li> <li>- La Shell del SO GNU/Linux.</li> <li>- Variables de entorno.</li> <li>- Sistemas de archivos y permisos.</li> <li>- Redirección de entrada y salida.</li> <li>- Señales y procesos.</li> </ul>	<p><b>Lección magistral participativa</b></p> <ul style="list-style-type: none"> <li>- Prestar atención al docente, al material presentado; tomar notas relevantes y destacar conceptos clave.</li> <li>- Hacer preguntas para aclarar dudas; contribuir a la discusión y compartir ideas; responder preguntas del profesor.</li> <li>- Lectura del material con anticipación.</li> </ul> <p><b>Resolución de ejercicios</b></p> <p>Realizar ejercicios prácticos en clase para aplicar conceptos teóricos.</p>	<ul style="list-style-type: none"> <li>- Reconoce los alcances y posibilidades del sistema operativo GNU/Linux como herramienta de desarrollo de software.</li> <li>- Accede a fuente de información confiables sobre el sistema operativo GNU/Linux.</li> <li>- Utiliza de manera fluida la Shell de GNU/Linux como herramienta de programación.</li> <li>- Accede a la página de manuales de comandos.</li> </ul>	<ul style="list-style-type: none"> <li>- Presenciales: 15 hs</li> <li>- Teoría y práctica: 7 hs</li> <li>- Laboratorio: 8 hs</li> <li>- Extra áulicas: 19 hs</li> </ul>

			<p><b>Evaluación formativa</b></p> <ul style="list-style-type: none"> <li>- Evaluación continua opcional.</li> </ul> <p><b>Evaluación sumativa</b></p> <ul style="list-style-type: none"> <li>- Exámenes teóricos-prácticos.</li> </ul>	
<p>RA 2: Utilizar el compilador del lenguaje C y herramientas auxiliares teniendo en cuenta las etapas de construcción, la documentación y control de versiones para el desarrollo de programas de PC y de microcontroladores.</p>	<p><b>Unidad 2: Herramientas para el desarrollo de software.</b></p> <ul style="list-style-type: none"> <li>- Etapas de construcción de un programación y banderas del compilador.</li> <li>- Programas con varios archivos fuentes.</li> <li>- Preprocesamiento. Macros y compilación condicional.</li> <li>- Uso y construcción de bibliotecas estáticas y dinámicas.</li> <li>- Herramientas de construcción automática (make).</li> <li>- Control de versiones.</li> </ul>	<ul style="list-style-type: none"> <li>- Lección magistral participativa.</li> <li>- Resolución de ejercicios.</li> </ul>	<ul style="list-style-type: none"> <li>- Identifica y organiza las etapas de construcción de un programa.</li> <li>- Organiza la construcción de un proyecto de software de múltiples archivos fuentes aplicados a diferentes arquitecturas de procesadores.</li> <li>- Selecciona y aplica funciones del compilador adecuadas para la construcción de software en sus diferentes etapas.</li> <li>- Reconoce el alcance de la herramienta para la construcción, documentación y control</li> </ul>	<ul style="list-style-type: none"> <li>- Presenciales: 15 hs</li> <li>- Teoría y práctica: 7 hs</li> <li>- Laboratorio: 8 hs</li> <li>- Extra áulicas: 19 hs</li> </ul>

	<ul style="list-style-type: none"> <li>- Documentación.</li> </ul>		<p>de versiones de programas.</p> <ul style="list-style-type: none"> <li>- Trabaja en forma grupal para resolver el Estudio de Caso propuesto.</li> </ul> <p><b>Evaluación formativa</b></p> <ul style="list-style-type: none"> <li>- Evaluación continua opcional.</li> <li>- Estudio de casos.</li> </ul> <p><b>Evaluación sumativa</b></p> <ul style="list-style-type: none"> <li>- Exámenes teóricos-prácticos.</li> </ul>	
<p>RA 3: Desarrollar programas en lenguaje C con acceso a registros de hardware teniendo en cuenta el mapa de memoria para la programación de periféricos de microcontroladores.</p>	<p><b>Unidad 3: Programación de microcontroladores en lenguaje C.</b></p> <ul style="list-style-type: none"> <li>- Arquitectura de un microcontrolador.</li> <li>- Herramientas: toolchain y compilador cruzado.</li> <li>- Programación de microcontroladores en lenguaje C.</li> <li>- Operadores a nivel de bits.</li> <li>- Programación de E/S digitales mediante el acceso a registro.</li> </ul>	<p>Lección magistral participativa.</p> <p>Resolución de ejercicios.</p> <p>Resolución de problemas.</p> <p><b>Estudio de casos</b></p> <ul style="list-style-type: none"> <li>- Trabajar en problemas realistas para explorar y aprender sobre conceptos y temas específicos.</li> <li>- Realizar investigaciones independientes sobre temas específicos a resolver.</li> <li>- Trabajar en equipos para fomentar la colaboración y la distribución de tareas.</li> <li>- Presentación y exposición del trabajo desarrollado a fin de compartir sus experiencias.</li> </ul>	<ul style="list-style-type: none"> <li>- Identifica y organiza las partes de un programa en lenguaje C (entrada / salida / procesamiento / comunicación) para un microcontrolador</li> <li>- Organiza la implementación y verificación de programas de microcontroladores con manejo de dispositivos de entrada/salida.</li> <li>- Trabaja en forma grupal para resolver el EdC propuesto.</li> </ul>	<ul style="list-style-type: none"> <li>- Presenciales: 11 hs</li> <li>- Teoría y práctica: 5 hs</li> <li>- Laboratorio: 6 hs</li> <li>- Extra áulicas: 14 hs</li> </ul>

	- Calificador const y volatile.		<p><b>Evaluación formativa</b></p> <ul style="list-style-type: none"> <li>- Evaluación continua opcional.</li> <li>- Estudio de casos.</li> </ul> <p><b>Evaluación sumativa</b></p> <ul style="list-style-type: none"> <li>- Exámenes teóricos-prácticos.</li> </ul>	
<p>RA 4: Desarrollar programas de PC y microcontroladores teniendo particular consideración del tipo de dato a utilizar con la finalidad de intercambiar información entre ellos a través del puerto serie.</p>	<p><b>Unidad 4: Comunicación entre un microcontrolador y la PC.</b></p> <ul style="list-style-type: none"> <li>- Comunicación de la PC con otros dispositivos/periféricos.</li> <li>- El estándar RS-232.</li> <li>- Programación de puerto serie de la PC con adaptador USB.</li> </ul>	<ul style="list-style-type: none"> <li>- Lección magistral participativa.</li> <li>- Resolución de ejercicios.</li> <li>- Resolución de problemas.</li> <li>- Estudio de casos.</li> </ul>	<ul style="list-style-type: none"> <li>- Organiza la implementación y verificación de programas de PC y Sistemas Embebidos para la comunicación serie entre ellos.</li> <li>- Genera alternativas en el uso de diferentes tipos de datos para la comunicación entre un Sistema Embebido y una PC.</li> <li>- Reconoce el alcance y limitaciones de la comunicación serie (UART-TTL y RS-232).</li> <li>- Trabaja en forma grupal para resolver el Estudio de Caso propuesto.</li> </ul> <p><b>Evaluación formativa</b></p> <ul style="list-style-type: none"> <li>- Evaluación continua opcional.</li> </ul>	<ul style="list-style-type: none"> <li>- Presenciales: 19 hs</li> <li>- Teoría y práctica: 8 hs</li> <li>- Laboratorio: 11 hs</li> <li>- Extra áulicas: 24 hs</li> </ul>

			- Estudio de casos.  <b>Evaluación sumativa</b> - Exámenes teóricos-prácticos.	
RA 5: Desarrollar programas en el lenguaje C++ de línea de comandos y con interfaz gráfica de usuario según el paradigma de programación orientada a objetos para el procesamiento y comunicación de datos.	<p><b>Unidad 5: El lenguaje de programación C++.</b></p> <p><b>Unidad 6: Programación de entornos gráficos en C++.</b></p> <ul style="list-style-type: none"> <li>- Programación orientada a objetos.</li> <li>- Clase en lenguaje C++.</li> <li>- Constructor y destructor.</li> <li>- Herencia.</li> <li>- Sobrecarga de operadores.</li> <li>- Biblioteca de clases de C++.</li> <li>- Programación de aplicaciones con interfaz gráfica de usuario utilizando la biblioteca Qt.</li> </ul>	<ul style="list-style-type: none"> <li>- Lección magistral participativa.</li> <li>- Resolución de ejercicios.</li> <li>- Resolución de problemas.</li> <li>- Estudio de casos.</li> </ul>	<ul style="list-style-type: none"> <li>- Identifica y organiza las partes de un programa en lenguaje C++ (entrada / salida / procesamiento / comunicación) de línea de comandos y con interfaz gráfica de usuario (GUI).</li> <li>- Genera alternativas de solución en problemas a resolver mediante programación en lenguaje C++.</li> <li>- Accede a fuente de información adecuada de biblioteca de clases C++.</li> <li>- Trabaja en forma grupal para resolver el Estudio de Caso propuesto.</li> </ul> <p><b>Evaluación formativa</b></p> <ul style="list-style-type: none"> <li>- Evaluación continua opcional.</li> <li>- Estudio de casos.</li> </ul> <p><b>Evaluación sumativa</b></p>	<ul style="list-style-type: none"> <li>- Presenciales: 39 hs</li> <li>- Teoría y práctica: 18 hs</li> <li>- Laboratorio: 21 hs</li> <li>- Extra áulicas: 48 hs</li> </ul>

			- Exámenes teóricos-prácticos.	
RA 6: Desarrollar programas de cálculo numérico en base al lenguaje C y C+ para su aplicaciones en ingeniería.	<b>Unidad 7: Aplicación de la PC al cálculo numérico.</b>  - Representación numérica en una PC.  - Métodos numéricos.	- Lección magistral participativa.  - Resolución de ejercicios.	- Reconoce el alcance y las limitaciones de las soluciones numéricas a problemas de ingeniería.  - Utiliza bibliotecas de funciones y clases para la resolución de problemas aplicando el cálculo numérico.  <b>Evaluación sumativa</b> - Exámenes teóricos-prácticos.	- Presenciales: 6 hs  - Teoría y práctica: 3 hs  - Laboratorio: 3 hs  - Extra áulicas: 8 hs
RA 7: Reconocer las estructuras dinámicas de datos teniendo en cuenta tanto estructuras lineales como no lineales para su aplicaciones en ingeniería.	<b>Unidad 8: Introducción a las estructuras dinámicas de datos.</b>  - Estructuras dinámicas de datos: lineales y no lineales.  - Listas, pilas y colas.  - Árboles binarios.	- Lección magistral participativa.  - Resolución de ejercicios.	- Reconoce el alcance de las EDD.  - Utiliza funciones de biblioteca y clases para elaborar programas que hagan uso de EDD.  <b>Evaluación sumativa</b> - Exámenes teóricos-prácticos.	- Presenciales: 6 hs  - Teoría y práctica: 3 hs  - Laboratorio: 3 hs  - Extra áulicas: 8 hs

#### 14. Condiciones de aprobación

En el Reglamento de Estudio para las Carreras de Grado en la Universidad Tecnológica Nacional (Ordenanza del Consejo Superior N° 1549) se establecen dos regímenes de aprobación de las asignaturas: **aprobación directa** y **aprobación no directa o con examen final**. Las condiciones para la aprobación directa son:

- Cumplir con los requisitos de inscripción a la materia según diseño curricular.
- Asistir a clases (con inasistencia máxima del 25% y en caso excepcionalmente del 40% a solicitud del estudiante ante la Secretaría Académica).
- Cumplir con las actividades de formación práctica.
- Aprobar las instancias de evaluación (las calificaciones de las instancias de evaluación se expresan en números enteros dentro de la escala del 1 (uno) al 10 (diez), **aprobando con un mínimo de 6 (seis)**).
- El estudiante que no apruebe alguna de las instancias de evaluación, tendrá al menos una instancia de recuperación.
- La calificación se expresa en número entero y en caso de promedios con decimales se redondeará al valor más próximo. La nota promedio de las instancias de evaluación aprobadas así obtenida será la calificación definitiva de aprobación directa.

#### **Condición de Libre**

No cumplió con las actividades prácticas y cuenta con menos del 75% de las asistencias a las clases.

#### **Condición de Abandono**

Asistencia menor del 75%. No vino nunca o sólo a las primeras clases teóricas y de las actividades de formación práctica.

#### **Condición de Regular**

La condición de regular se alcanza mediante la aprobación de dos exámenes parciales teóricos-prácticos, pudiendo recuperar un solo parcial, más la aprobación del trabajo práctico grupal. Es posible la recuperación de los dos exámenes parciales en caso de haber realizado todas las Evaluaciones continuas opcionales. La aprobación de la asignatura se completa con un examen final teórico-práctico integrador.

## **Condición de Aprobación Directa**

La condición de aprobación directa se alcanza mediante el cumplimiento de la condición de regular más la aprobación de un examen integrador teórico-práctico con opción a un examen recuperatorio. La calificación final de la aprobación directa será el **promedio entre los 3 exámenes** (2 exámenes parciales y el examen integrador).

## **Trabajo práctico grupal**

La aprobación del trabajo práctico grupal es requisito para lograr la condición de regular de la asignatura. Este trabajo se deberá llevar a cabo de forma grupal y consiste en el desarrollo del programa para: i) un sistema embebido y ii) una PC que se comuniquen entre ellos mediante comunicación serie. Los requerimientos del desarrollo serán planteados por el/los docente/s de la cátedra.

## **Objetivos**

El trabajo práctico grupal tiene por objetivo poner a los estudiantes ante la situación de resolver problemas de la vida real, integrando los conocimientos de informática y programación adquiridos hasta el momento. Además, se prevé que con el desarrollo de dicho trabajo los estudiantes desarrollen las capacidades de auto-aprendizaje, búsqueda de material bibliográfico, trabajo en equipo y exposición en público.

## **Materiales y herramientas**

- Utilizar exclusivamente el Sistema Operativo GNU/Linux y el compilador gcc.
- Utilizar exclusivamente los lenguajes de programación C y C++.
- El sistema embebido a utilizar deberá estar basado en microcontroladores, preferentemente la plataforma Arduino.

## **Requisitos**

1. Deberá constituirse un **grupo de trabajo** integrado por **tres miembros**.
2. El grupo deberá proponer una idea proyecto sobre un problema de la vida real. Los problemas a resolver pueden ser del ámbito productivo, educativo, entretenimiento, vigilancia y seguridad, asistencia a personas mayores o con capacidades diferentes, etc.
3. En base a la idea proyecto propuesto el/los docente/s de la cátedra elaborarán un documento de requerimientos que detalle el desarrollo a llevar a cabo por el grupo de trabajo.

4. El desarrollo consistirá de un sistema embebido compuesto de un microcontrolador y dispositivos de entrada/salida, el cuál deberá comunicarse con la PC a través del puerto serie. El microcontrolador y la PC deberán ejecutar programas implementados por los integrantes del grupo utilizando los lenguajes de programación C y/o C++.
5. La presentación del trabajo incluye:
  - a. El equipo funcionando.
  - b. Una presentación (filminas) para la exposición oral. La exposición oral deberá presentar los programas desarrollados y tener una duración de 15 min. más 5 min. para responder preguntas.
6. El grupo de trabajo deberá entregar al docente las filminas en formato pdf y el código fuente de todos los programas elaborados.
7. En la evaluación del trabajo, cualquiera de los integrantes del grupo deberá estar en condiciones de responder sobre los programas desarrollados. No cumplir con este requisito implica la reprobación de dicho integrante.

## 15. Modalidad de examen

El examen final es de carácter integrador cubriendo todos los temas propuesto por el programa analítico de la materia. Se evaluarán tres temas incluyendo conceptos teóricos y prácticos. El examen será síncrono y presencial, mediado por tecnología (Aula Virtual).

## 16. Recursos necesarios

Los recursos necesarios para el dictado de las clases teóricas-prácticas son:

1. Aulas con PC con sistema operativo GNU/Linux, preferentemente la última versión LTS (Long-Term Support) de Ubuntu.
2. Herramientas de construcción y gestión de software: gcc, make, git, doxygen.
3. Herramientas de compilación cruzada para microcontroladores AVR basada en GCC (avr-gcc) y de grabación (avrdude).
4. IDE de desarrollo Arduino.
5. Placas de desarrollo Arduino, preferentemente el modelo Arduino UNO.
6. Bibliotecas de programación gráfica Qt e IDE de desarrollo (Qt Creator).

<b>Anexo I: Plantel docente de la asignatura</b>			
Titular	Especifique Nombre y Apellido completo.	Dedicación:	Especifique la cantidad de dedicaciones.
Asociado	Especifique Nombre y Apellido completo.	Dedicación:	Especifique la cantidad de dedicaciones.
Adjunto:	Gonzalo Fernando Perez Paina Juan Herencia Silvia Alejandra Carrera	Dedicación:	DE DS DS
Jefe de Trabajos Prácticos	José Luis Martinez Nestor Palomeque	Dedicación:	DS DS
Auxiliar de 1ra.	Especifique Nombre y Apellido completo.	Dedicación:	Especifique la cantidad de dedicaciones.
Auxiliar de 2da.	Especifique Nombre y Apellido completo.	Dedicación:	Especifique la cantidad de dedicaciones.

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	Describe el tema trabajado	Seleccione el tipo de actividad.

FIRMA (de cada docente que conforman la comisión).