

Programación II

Planificación Ciclo lectivo 2023

Datos administrativos de la asignatura			
Departamento:	Bioingeniería	Carrera	Bioingeniería
Asignatura:	Programación II		
Nivel de la carrera	2	Duración <small>(Anual o Cuatrimestral)</small>	Anual
Bloque curricular	Tecnologías Básicas		
Área curricular	Digital		
Carga horaria presencial semanal:	3,75	Carga Horaria total	120
Carga horaria no presencial semanal (si correspondiese).	0	% horas no presenciales (si correspondiese)	0
Profesor Titular/Asociado/Adjunto	Ing. Darío Kuruc	Dedicación (horas):	Simple
Auxiliar/es JTP	Ing. Walter Allaltune		Simple
Auxiliares de 1°			

Presentación, Fundamentación

Programación II es la continuidad natural de Programación I, en la cual los estudiantes adquieren las bases de la programación estructurada. En Programación II se abordan enfoques avanzados de programación como es la programación gobernada por eventos y, de manera introductoria, la programación orientada a objetos.

Utilizando un kit de desarrollo, empleando como excusa el uc LPC845 de la firma NXP, se enfrenta la necesidad de aprender alguna técnica de modelado (Máquinas de Estados) que permita resolver problemas básicos de sistemas de control a nivel de aplicación sin contar con un Sistema Operativo que administre los recursos.

Se analiza la concurrencia de procesos y la manera de administrarlos en ese contexto. Asimismo, se aborda el manejo de los recursos del hardware incursionando en la lectura e interpretación de las hojas de datos y manuales del microcontrolador en cuestión, lo cual permite llevar a cabo una programación a muy bajo nivel, configurando los registros del dispositivo por medio de técnicas de operadores a nivel de bits, uniones y/o campos de bits.

Se trabaja a nivel de los registros para la configuración y uso del GPIO, el Systick, las interrupciones, la comunicación serie y la conversión AD/DA. Todo ello, en conjunto con el desarrollo de una interfaz gráfica sencilla sobre el framework Qt, permite alcanzar hacia fin de año el desarrollo integral de un sistema de control con su propia interfaz de hardware y con una interfaz gráfica en la PC desde la cual también puede operarse.

Los aportes de la asignatura al perfil de egreso son:

- sólidas bases científicas, técnicas, tecnológicas.
- resolución de problemas en forma sistémica, abstrayendo modelos generales de casos particulares.
- formación continua, a partir del abordaje autónomo de contenido.
- integración de conocimientos a partir de experimentación y ensayos y la síntesis de problemas complejos.
- capacidades comunicativas que le permitan interactuar en equipos profesionales.
- solvencia en el manejo de herramientas vigentes en el campo electrónico de la bioingeniería y las TICs.

La relación de la asignatura con los alcances del título no son particulares, sino que están distribuidos en diversos aspectos en el sentido de que los sistemas de control favorecen el cumplimiento de los objetivos de diseño, operación y proyecto de lo alcanzado en las actividades reservadas y alcances.

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

Competencias específicas de la carrera (CE)		Competencias genéricas tecnológicas (CT)		Competencias genéricas sociales, políticas y actitudinales (CS)	
CE 1	1	CT 1	0	CS 1	0
CE 2	1	CT 2	0	CS 2	0
CE 3	0	CT 3	0	CS 3	0
CE 4	0	CT 4	0	CS 4	0
CE 5	0	CT 5	1	CS 5	1
CE 6	0				

CE 7	0				
CE 8	0				
CE 9	0				
CE 10	0				

CE 1 Diseñar, Calcular y Proyectar instalaciones, equipamiento e instrumental biomédico, aplicando conocimiento integral y tecnologías adecuadas para atender la demanda de la población y las variables económicas características de la bioingeniería.

CE 2 Proyectar, dirigir y controlar la construcción, operación y mantenimiento de instalaciones, equipamientos e instrumental de tecnología biomédica, procesamiento de señales biomédicas y sistemas derivados de biomateriales utilizados en el área de la salud.

CT5: Contribuir a la generación de desarrollos tecnológicos y/o innovaciones tecnológicas.

CS5: Actuar con espíritu emprendedor.

Propósito
<ul style="list-style-type: none"> ▪ Brindar conocimientos y técnicas de programación, análisis, diseño y desarrollo de aplicaciones de diversa naturaleza para el estudio y modificación de los sistemas electrónicos biomédicos, desde lo teórico hasta la implementación, mediante conceptos analíticos, herramientas computacionales y ensayos y experimentos.
Objetivos establecidos por el diseño curricular
<ul style="list-style-type: none"> • Diseñar y desarrollar aplicaciones para resolución de problemas complejos aplicados a bioingeniería. • 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.
Resultados de aprendizaje
<p>RA1: [Diseña y desarrolla] [aplicaciones] [para resolución de problemas complejos] [aplicados a la bioingeniería].</p>

RA2: [Utiliza] fluidamente [herramientas de documentación, de control de versiones, y de automatización de la construcción de un programa o biblioteca] [a partir de las fuentes].

RA3: [Aplica] [los principios de la programación orientada a objetos] [utilizando un lenguaje apropiado] [para resolver problemas electrónicos en bioingeniería].

RA4: [Evalúa] [la pertinencia del uso de un lenguaje estructurado o uno orientado a objetos] [para la resolución de problemas electrónicos en bioingeniería] [a través de los conceptos y criterios de selección].

RA5: [Desarrolla] [interfaces gráficas] [para realizar aplicaciones] [donde se necesite la interacción mediante puerto serie entre la PC o un dispositivo móvil y un microcontrolador].

RA6: [Conoce] [las estructuras avanzadas de datos] [utilizadas más frecuentemente] [para el desarrollo de programas].

RA7: [Programa] [sistemas embebidos en bajo nivel] [para el desarrollo de programas que utilicen acceso a registros de periféricos], [mediante lenguaje de programación].

RA8: [Aplica] [técnicas de comunicación y programación] [para el manejo de periféricos de uso común, como displays, conversores AD y DA y puertos serie], [para resolver aplicaciones comunes en sistemas embebidos].

Asignaturas correlativas previas

Para cursar y aprobar debe tener cursada:

- Programación I

Asignaturas correlativas posteriores

- Técnicas Digitales II

Programa analítico. Unidades temáticas

Programa analítico. Unidades temáticas		
Unidades, Contenidos	Carga horaria	Carga horaria práctica

<p>UNIDAD 1: El Lenguaje C++ y las Estructuras Avanzadas de Datos</p> <p>Introducción a la programación orientada a Objetos - Lenguaje C++ como evolución respecto del lenguaje C - Tipos abstractos de datos: Clases - Clases y Objetos - Miembros públicos y privados - constructores y destructores - Concepto de encapsulamiento. Entrada/Salida en C++.</p> <p>Operador visibilidad - Operadores para gestión dinámica de memoria: new y delete - Sobrecarga de funciones y operadores. Funciones y clases friend</p>	20	5
<p>- uso del apuntador this - calificador const - especificador de clase de almacenamiento: extern y static. Herencia: tipos (hincapié en herencia pública) - clase base y derivada – miembro protected – constructores y destructores en la clase derivada. Contenedores de datos – Aplicaciones con Estructuras Avanzadas de Datos: Colas, Pilas, Listas. Funciones Virtuales y Virtuales puras – Clases Abstractas – Concepto de Interface</p>		
<p>UNIDAD 2: Entornos Gráficos</p> <p>Introducción a la programación en entornos gráficos - Bibliotecas y entornos de desarrollo: Caso particular de QT y el QT Creator. -Introducción a la programación gobernada por eventos y las señales (Signals and Slots) de QT. - Formularios Básicos: Cuadros de Diálogo y aplicaciones de formularios sencillas. - Widgets más comunes: Button, Check box, Radio button, Menu bar, Toolbar, Scrollbar, Text box, Combo box, Label y Otros. Ejemplos de aplicación: Clientes de sistemas computacionales (PC , bases de datos, Sistemas embebidos, etc.) comunicación serie, representaciones gráficas (data logger)</p>	10	2

<p>UNIDAD 3: Introducción a los microcontroladores y los Sistemas Embebidos.</p> <p>Concepto de Sistema Embebido. Presentación de periféricos elementales: Puertos de entrada/salida, Contador, Timer, UART, etc. IDE para microcontroladores - Compilación modular - Metodología de proyectos – Diagrama de capas: drivers, primitivas, aplicación – portabilidad. Concepto de Sistema Operativo en Tiempo Real. DIAGRAMA DE CAPAS: Metodología de trabajo. Aplicación - Primitivas – Drivers.</p>	5	1
<p>UNIDAD 4: Introducción a la arquitectura de un Microcontrolador.</p> <p>Caso de uso: Cortex M0+ y su programación en C++</p> <p>Microcontrolador: Análisis de un circuito inteligente básico. Integración de periféricos en un solo chip. Familia CórteX M0+. Presentación del mapa de memoria y periféricos. Técnicas de acceso. Identificación de Registros del Core y de sus periféricos. Especificadores de memoria. La Switch Matrix (SWM), registros y operación.</p>	5	1
<p>UNIDAD 5: Introducción a GPIO.</p> <p>GPIO (General Purpose Input Output): Configuración de los puertos de entrada/salida de propósito general: tratamiento como entrada, tratamiento como salida, tratamiento bidireccional. Registros asociados. Lectura de entradas digitales simples - Acción sobre salidas digitales. Consideraciones teóricas de las entradas y las salidas de los microcontroladores. Desarrollo de funciones utilitarias en C++ para el manejo de GPIO.</p>	5	1
<p>UNIDAD 6: Pooling (encuesta) e Interrupciones</p> <p>Idea general - Estrategias de atención - Funciones de interrupción - Pooling vs. Interrupciones - Registros asociados - Vector de interrupciones - Prioridades - Distribución de tiempos de ejecución para el programa principal y las funciones de interrupción - Mínimo tiempo admisible entre interrupciones - Pasaje de información entre el programa principal y las funciones de interrupción. Fuentes interrupción externas (en pines).</p>	10	2

<p>UNIDAD 7: Contadores / Temporizadores</p> <p>Contador - Características principales – Temporizador: Caso particular de contador - Utilización como Contador (acumuladores de eventos) - Utilización como Temporizador (Base de Tiempo) - Multiplicadores de la base de Tiempo (ticks). Implementación - Registros asociados - Modos de funcionamiento. scheduler: Principios de funcionamiento. Caso de estudio: El systick del core del Cortex M.</p>	5	1
<p>UNIDAD 8: GPIO avanzado.</p> <p>Desarrollo del firmware y primitivas asociadas a Display de 7 segmentos y a la lectura y depuración mediante técnicas de absorción de transitorios de teclados lineales, teclados matriciales y entradas digitales de microswitchs. Escritura de salidas digitales. Display LCD.</p>	15	3
<p>UNIDAD 9: Programación Gobernada por Eventos</p> <p>Introducción a los diagramas secuenciales con múltiples acciones temporizadas. Desarrollo de funciones facilitadoras: TimerStart(), TimerStop(), TimerClose(), etc. Máquina de Estados: Diagrama de globos.</p>	20	5
<p>Implementación con switch-case. Implementación con múltiples if. Implementación con punteros a función. Modelización y Síntesis de problemas mediante la utilización de máquinas de estado - Casos de automatismos independientes dentro de un mismo equipo - Utilización de máquinas de estados en paralelo: ventajas. Análisis de casos de aplicación típicos. Resolución de problemas mediante la combinación de Máquinas de estado y Diagramas secuenciales.</p>		

<p>UNIDAD 10: Comunicación Serie.</p> <p>Necesidad de la comunicación serie - Serializadores y paralelizadores – Comunicación serie asincrónica: Conceptos. Velocidad de transmisión - Registros asociados en el microcontrolador - Modos de operación – Buffers de Rx y Tx: Pilas y Colas circulares - Implementación de protocolos punto a punto y multipunto - Estrategias de programación por pooling e interrupciones. Desarrollo del firmware y primitivas asociadas a la comunicación serie.</p>	20	5
<p>UNIDAD 11: Introducción a la Medición y Generación de Señales Analógicas</p> <p>Introducción a la medición de magnitudes analógicas (temperatura, presión, humedad, etc.) - Conversores ADC: y DAC características principales - Registros asociados. Estrategias de programación por pooling y por interrupciones - Interpretación de los valores obtenidos (Tablas, filtros, etc.) - Eliminación de valores espurios: Filtros de media móvil y de mediana. Desarrollo del firmware y primitivas asociados a la medición y generación de señales analógicas.</p>	10	3
<p>UNIDAD 12: Métodos numéricos elementales</p> <p>Aplicaciones algorítmicas haciendo uso de métodos numéricos tradicionales. Soluciones aproximadas. Precisión. Errores.</p> <p>Ecuaciones diferenciales ordinarias: Ecuaciones con condiciones iniciales. Integración por Taylor. Método de Euler, Euler Gauss. Euler Richardson. Métodos de Runge-Kutte. Método predictor corrector. Estabilidad y convergencia. Sistemas de ecuaciones diferenciales. Ecuaciones de orden</p>	10	2
<p>superior. Ecuaciones con condiciones de contorno. Métodos de diferencias finitas y elementos finitos para ecuaciones diferenciales.</p>		
<p>Metodología de enseñanza</p>		

Se trata de promover procesos cognitivos que permitan un abordaje integral para poder ir avanzando hacia la construcción de un proyecto integrador cuya presentación se realizará al final del curso. Se privilegia un enfoque grupal desde el punto de vista de la interacción de los estudiantes.

Como estrategia metodológica durante la clase, se utilizan las presentaciones como disparador temático en soportes de diapositivas para poner en disponibilidad ideas previas por parte de los estudiantes a modo de fomentar aprendizajes significativos.

En función de los recursos, los alumnos deben realizar trabajos prácticos de laboratorio para aplicar los elementos conceptuales, metodológicos y prácticos que desarrollaron.

Sintéticamente, la clase se diagrama con los siguientes elementos.

- Lectura previa de la bibliografía recomendada para desarrollo de una actividad de seguimiento.
- El profesor presenta casos de estudio de la especialidad y otros significativos de otras disciplinas a fin de propiciar el análisis y la discusión sobre los mismos en conjunto con los estudiantes.
- Espacio de práctica, destinada fundamentalmente a la implementación en software de las resoluciones analíticas.
- Clases invertidas en temas abordados en otras materias y que resultan de aplicación directa en esta.
- Especialistas invitados para el desarrollo de temas directamente vinculados a su desarrollo profesional.
- Uso de dispositivos académicos para la verificación de resultados y ensayos de sistemas dinámicos en laboratorios físicas de la carrera.
- Exposición de trabajos de elaboración grupal ante auditorio especializado. Evaluación entre pares.
- Intercambio de información, actividades de seguimiento por cuestionario y ejercitación a través del CVG, de uso intensivo.
- Uso de software de cálculo numérico para la verificación y simulación de los temas adecuados.
- Uso de procesadores de texto (Word, Latex) y de presentaciones (PowerPoint, Latex) para la elaboración de informes y exposiciones frente a auditorio y formato poster.

Recomendaciones para el estudio

- Asistencia a clase, participación en las consignas propuestas y socialización de dudas y dificultades para el aprendizaje conjunto.

- Lectura previa de los contenidos de acuerdo con la bibliografía recomendada.
- Resolución de las guías de ejercitación propuestas.
- Desarrollo de código de simulación propio para adquirir la sintaxis y técnicas de programación con el objetivo de la reutilización futura del código desarrollado.
- Hábito de estudio de, al menos, 4 hs fuera de clase, para la preparación de las actividades periódicas de teoría y práctica y seguimiento de los contenidos.
- Búsqueda de la relación de los temas de la materia con conocimientos previos adquiridos en otras asignaturas y de acuerdo con lo facilitado por la cátedra.
- Fluidez en el manejo de procesadores de texto (Word, Latex) y de presentaciones (PowerPoint, Latex) para la elaboración de informes y exposiciones frente a auditorio y formato poster.

Metodología de evaluación

- Evaluación diagnóstica a través del CVG de los temas previos requeridos para el abordaje de los temas de la materia. Útil para el alumno para comprender la solidez en los temas cotidianos de la materia y útil para la cátedra para diagramar la evolución de la profundidad de tratamiento de los temas para finalizar con los resultados de aprendizaje en igualdad de adquisición por los regulares o aprobados de la materia.
- Evaluación periódica antes de la clase para la construcción de la parte teórica-expositiva sobre las dudas de la lectura y la profundización en detalles de los temas de cada clase y el estudio de casos de la especialidad en el contexto de los temas de la clase o la unidad temática. Esta evaluación consiste en preguntas de opción múltiple, verdadero o falso y a desarrollar a partir de la información disponible en la lectura bibliográfica guiada. Su resultado es Aprobado o No Aprobado. Para aprobar cada instancia individualmente se debe resolver satisfactoriamente el 70% de las consignas planteadas sin errores conceptuales graves. Para aprobar el conjunto de estas actividades se deben aprobar las actividades asignadas obligatoriamente (entre 5 y 10 por cursante).
- Evaluación periódica luego de finalizar cada unidad temática de acuerdo con lo identificado en el cronograma. Esta evaluación consiste en problemas analíticos y de resolución numérica, simulación y derivación de conclusiones de análisis. Su resultado es Aprobado o No Aprobado. Para aprobar cada instancia individualmente se debe resolver satisfactoriamente el 70% de las consignas planteadas sin errores conceptuales graves. Para aprobar el conjunto de estas actividades se deben aprobar las actividades asignadas obligatoriamente (entre 4 y 6 por cursante).

- Trabajos prácticos expositivos a organizar en el transcurso del año. Trabajo grupal. El informe es revisado por la cátedra para evitar errores conceptuales previo a la exposición. La exposición es evaluada por los grupos pares en base a los aspectos puntuales propuestos por la cátedra. La actividad es desarrollada a través del CVG y la exposición con la modalidad expositiva con presentación y ejercicio resuelto en pizarra.
- Trabajos prácticos de laboratorio basados en la ejecución de una guía de actividades propias del dispositivo de experimentación y ensayo. Trabajo grupal, de ejecución individual por grupo, para que todos los grupos hagan la actividad de laboratorio exhaustiva y activamente.
- Clase externa virtual expuesta por profesionales con desempeño en el área de la temática particular, evaluado mediante un cuestionario conceptual sobre los temas expuestos.
- Trabajo integrador de análisis, diseño, desarrollo, implementación y ensayo de una aplicación software que habilita la aprobación directa o por final de la materia. Trabajo grupal, con nivel de objetivos acorde al número de participantes. En el transcurso del año lectivo, los cursantes o grupo proponen o toman un caso para trabajar y con la guía de la cátedra se propone un esquema de actividades que permitirán concretar el sistema. Los avances son controlados por la cátedra para orientar en las dificultades encontradas propias del salto a la implementación. Se entrega la aplicación elaborada e informe de análisis, diseño, desarrollo, implementación y ensayo, a la vez que es expuesto sintéticamente frente a la cátedra y docentes invitados afines a la temática.

Cronograma de clases / trabajos prácticos / exámenes (tentativo)

CLASE	Contenidos/Actividad	Modalidad
1	Presentación general de la materia (tanto cuestiones administrativas como académicas). - Charla inicial motivadora TPO 10 min Programación Orientada a Objetos. Lenguajes aptos para el POO. Lenguaje C++. Declaración de variables. Tipos abstractos de datos - Clases. Clases y Objetos. Miembros: Públicos y privados. Constructores No parametrizados y parametrizados. Destructores. Referencias, constructor de copia - Entrada/Salida en C++.	Presencial
2	Operador visibilidad - Operadores para gestión dinámica de memoria: New y delete - puntero this - miembros static - friend - Sobrecarga de operadores. Ejercitación	Presencial
3	Herencia, Miembros protegidos. Funciones Virtuales - Ejercitación integradora.	Presencial

4	TPC: Ejercitación C++ usando Estructuras Avanzadas de Datos tradicionales (Colas, Pilas, LSE, LDE)	Presencial
5	Introducción a la programación de entornos visuales. Ejemplo introductorio. Ejercitación sobre Qt: Widgets: Cuadros de dialogo - Button - Check box - Radio button - Menu bar – Toolbar – Scrollbar - Text box - Combo box – Label	Virtualidad sincrónica
6	Clase práctica Qt	Presencial
7	Presentación de la idea general de un ""cto. inteligente"" y del método de trabajo.¿Qué son los micros? - Arquitecturas - ¿Qué son los Periféricos? - ¿Qué es un Sistema Embebido? - Intro al cortex M0+ LPC845. Repaso operadores a nivel de bits - uniones y campos de bits. - Descripción del Kit Infotronic. Presentacion del IDE MCUXpresso	Presencial
8	Acceso al HW - GPIO del LPC845 - Registros Asociados. Switch Matrix (SWM). Acceso a Harsware: ¿Cómo lo hago en C++? - Diagrama de capas.	Presencial
9	Concepto de interrupción, - su implementación en Cortex M0+. Interrupciones Externas - Registros asociados. Ejemplos de uso sencillos. - ¿Cómo lo hago en C++?	Presencial / Virtualidad sincrónica
10	Necesidad de una base de tiempo: Systick Timer - Registros Asociados. Ejemplos de uso sencillos - Estrategias avanzadas de temporización (maquinaria de timers) ¿Cómo lo hago en C++?	Presencial / Virtualidad sincrónica
11	TPL1: Uso de IDE - Acceso a hardware - interrupciones - Systick	Presencial
12	GPIO avanzadas: Visualización 7seg - Técnicas de multiplexación. ¿Cómo lo hago en C++?	Presencial / Virtualidad sincrónica
13	GPIO avanzadas: Entradas Digitales y Teclados. Técnicas de anti rebote (debounce) ¿Cómo lo hago en C++?	Presencial / Virtualidad sincrónica
14	TPL2: Teclado, display, entradas digitales	Presencial
15	PARCIAL 1	Presencial
16	Biblioteca Standard de plantillas (STL) de C++ - Clases contenedoras - Ejercitación.	Presencial / Virtualidad sincrónica
17	Introducción a la Programación Gobernada por Eventos, Conceptos básicos y codificación de ejemplos basados en eventos y acciones simples. Presentación del software uModelFactory	Presencial

18	Maquinas concurrentes. Estados compuestos. Utilización de funciones de temporización.	Virtualidad sincrónica
19	TPC: Programación Gobernada por Eventos en C++	Presencial
20	TPL3: Ejercitación Maquina de Estados	Presencial
21	Comunicación serie: Generalidades. Serializadores y Paralelizadores. Norma RS232. Protocolo Asincrónico. Registros Asociados. Uso de QserialPort	Presencial
22	Buffers de Rx y Tx: Colas circular. Primitivas y drivers. Implementación de protocolos punto a punto y multipunto.	Presencial
23	TPL4: Práctica comunicación serie asincrónica	Presencial
24	Ejercitación integradora - Vinculación con a PC a través de la clase QserialPort	Presencial
25	Conversor Análogo Digital (ADC) y Conversor Digital Análogo (DAC): Generalidades - ADC/DAC Cortex M0+ - Estrategias de programación. Interpretación de los valores obtenidos. Tablas, Filtros. Generación de señales analógicas. Diagrama de capas	Presencial
26	Presentacion MDE + GUI del TPO Display de cuarzo liquido	Presencial
27	Ejercitación ADC/DAC + Serie	Presencial
28	Ejercitación Integradora: PC (QSerialPort) - UART - ADC - DAC	Presencial
29	Métodos numéricos tradicionales. Aplicaciones algorítmicas. Soluciones aproximadas. Precisión. Errores.	Presencial / Virtualidad sincrónica
30	Desarrollo de aplicaciones utilizando cálculo numérico: casos de estudio con ecuaciones no lineales, sistemas de ecuaciones lineales, ecuaciones diferenciales ordinarias e integración. Evaluación de herramientas de software: software simbólico, software numérico.	Presencial
31	Desarrollo de aplicaciones utilizando cálculo numérico: Ecuaciones diferenciales ordinarias: Ecuaciones con condiciones iniciales. Integración por Taylor. Método de Euler, Euler Gauss. Euler Richarson. Métodos de Runge-Kutte. Método predictor corrector. Estabilidad y convergencia. Sistemas de ecuaciones diferenciales. Ecuaciones de orden superior. Ecuaciones con condiciones de contorno. Métodos de diferencias finitas y elementos finitos para ecuaciones diferenciales.	Presencial

	32	PARCIAL 2	Presencial
--	----	------------------	------------

Recursos necesarios

- Aula adecuada en espacio para la cantidad de inscriptos
- Espacio en el CVG
- Pizarra, fibrones
- Proyector
- PC docente
- Laboratorio de informática
- Laboratorio con mesa de trabajo y enchufe para el equipo eléctrico
- Software
- Plataforma de videoconferencia para las clases externas virtuales (Zoom, Teams)

Referencias Bibliográficas (citadas según normas APA)

Bibliografía obligatoria, optativa y otros materiales del curso

- [1] Red de desarrolladores de Qt en Español - MediaWiki, (06/03/19) <https://wiki.qt.io/Main/es> .
- [2] M. J. Pont, (2002). Embedded C. Addison-Wesley.
- [3] Anónimo. (2010). “APRENDA Qt4 DESDE HOY MISMO, https://www.academia.edu/21989349/APRENDA_Qt4_DESDE_HOY_MISMO
- [4] B. W. Kernighan and D. Ritchie. (1991) El lenguaje de programación C. Prentice-Hall.
- [5] J. M. GOMEZ, A. MARTIN PRAT, X. MOLINERO ALBAREDA, P. P. VAZQUEZ ALCO CER, and F. XHAF A.(2006) Programación en C++ para ingenieros. Madrid: Paraninfo SA..
- [6] NXP® Semiconductors(Oct. 28, 2020) “Hoja de Datos LPC84x,” . <https://www.nxp.com/docs/en/data-sheet/LPC84x.pdf>
- [7] H. M. DEITEL and P. J. DEITEL (2008), C/C++ Cómo programar, 6ta Edicion, 6th ed. Madrid: Pearson. www.deitel.com
- [8] B. Hall (2020). “Beej’s Guide to Network Programming Using Internet Sockets,” p. 134, http://beej.us/guide/bgnet/pdf/bgnet_usl_c_1.pdf

Función docencia

- Participación de los auxiliares docentes en el dictado de clases teóricas bajo supervisión del profesor.
- Participación en la Maestría en Docencia Universitaria.
- Participación en congresos y seminarios del área de conocimiento de la materia.

Reuniones de asignatura y área

Reuniones de cátedra:

- A cada fecha de final: Mayo, Julio, septiembre, diciembre, febrero.
- Informe estadístico de actividades evaluadas y ejecutadas, revisión del cronograma de clases, proyección de actividades próximas.
- Intercambio sobre el curso de las clases, prácticas, laboratorios.

Reuniones de área:

- Finales de julio y diciembre. Ante requerimiento, reunión en febrero.
- Curso de clases, ejecución, dificultades y deserción.
- Planificaciones para el próximo ciclo lectivo.
- Intercambio sobre criterios comunes y particulares al área y las asignaturas.
- Intercambio sobre las comisiones de cada año.

Atención y orientación a las y los estudiantes

- Orientación y consultas fuera del horario de clase a través del CVG.
- Orientación y consultas dentro del horario de clase como parte del intercambio dinámico de la exposición.
- Las evaluaciones teóricas no se recuperan individualmente, sino que se asignan en periodicidad de acuerdo con el desempeño académico del alumno, procurando que el número de actividades sea posible de la aprobación del conjunto de actividades. Se resuelven en horario de clase (30').
- Las evaluaciones prácticas no se recuperan individualmente, sino que se asignan en periodicidad de acuerdo con el desempeño académico del alumno, procurando que el número de actividades sea posible de la aprobación del conjunto de actividades. Se resuelven posterior al fin de la unidad temática, plazo de entrega de 1 semana.
- Los trabajos de laboratorio no resueltos satisfactoriamente se rehacen en nueva fecha a coordinar. La fecha y horario particular es coordinada entre grupo y cátedra para la realización individual por grupo del trabajo. Idéntica organización se toma para la repetición de la actividad ante su desaprobación.
- Los trabajos expositivos tienen una revisión preliminar de manera de no incurrir en la desaprobación de la actividad y para que la socialización de los resultados sea constructiva a todo el curso.
- El aprendizaje autónomo se da en la preparación de los contenidos para cada clase, que luego son validados y profundizados en la clase del día.

ANEXO 1: FUNCIÓN INVESTIGACIÓN Y EXTENSIÓN (si corresponde)

En este Anexo 1 (a completar si correspondiese) la cátedra detallará las actividades previstas respecto a la función docencia en el marco de la asignatura

No previsto

Lineamientos de investigación de la cátedra

Para introducir a las y los estudiantes a las actividades de investigación que realiza la cátedra. Se recomienda incorporar al Programa analítico de la asignatura los lineamientos de investigación en los cuales la asignatura este participando.

No corresponde

Lineamientos de extensión de la cátedra

Para introducir a las y los estudiantes a las actividades de Extensión que realiza la cátedra. Se recomienda incorporar al Programa analítico de la asignatura los programas de Extensión en los cuales la asignatura este participando

No corresponde

Actividades en las que pueden participar las y los estudiantes

Incluir todas aquellas instancias en las cuales las y los estudiantes puedan incorporarse como participantes activos tanto en proyectos de investigación como de extensión, en la asignatura o mediante el trabajo conjunto con otras asignaturas

Eje: Investigación

Proyecto	Cronograma de actividades
-	-

Eje: Extensión

Proyecto	Cronograma de actividades
-	-