



**CARRERA: Ing. Electrónica**

**PROGRAMA ANALÍTICO DE ASIGNATURA**

**NOMBRE DE LA ACTIVIDAD CURRICULAR: TÉCNICAS DIGITALES III**

Año Académico: 2024

Área: Técnicas Digitales

Bloque: Tecnologías Aplicadas

Nivel: 5to

Tipo: Obligatoria

Modalidad: Anual

**Cargas horarias totales:**

<i>Horas reloj</i>	<i>Horas cátedra</i>	<i>Horas cátedra semanales</i>
120	160	5

**COMPOSICIÓN DEL EQUIPO DE CÁTEDRA:**

Profesor Titular: Ing. Alejandro Furfaro  
Profesor Adjunto: Ing. Juan Andrés Montenegro  
Ing. Darío Alpern  
JTP: Ing. Diego García  
Ing. Mariano Gonzalez  
Ing. Nudelman, Gustavo  
Ing. Christian Nigri  
ATP 1°: Ing. Luciano Ferreyro  
Ing. Hernán Zbucki  
Ing. Zugman, Gabriel

**FUNDAMENTACIÓN**

Técnicas Digitales III es una asignatura de Tecnología Aplicada de acuerdo con lo establecido en el Diseño Curricular del Plan 2023 (Ord. CS N° 1849). Por esta razón es



que es imprescindible que los fundamentos teóricos se expresen claramente a través de contenidos concretos adecuados para su aplicación práctica.

La asignatura, siendo la última asignatura obligatoria del área de técnicas digitales, busca establecer bases conceptuales sólidas que permitan al estudiante comprender de manera profunda el funcionamiento de un sistema compuesto por uno o más cores de Procesamiento (sus aspectos y principios fundamentales de Arquitectura y Organización), el Sistema Operativo que permite administrar las aplicaciones, y los Protocolos de conectividad que le permiten interactuar con el entorno, haciendo especial hincapié en los principios fundamentales que le proporcionen una base de conocimiento permanente y le permitan al estudiante, una vez graduado continuar su aprendizaje en estos ítems o cualquier otro incluso de áreas diferentes de conocimiento, aplicando las metodologías de análisis en profundidad de los problemas y sus metodologías de resolución.

En los cursos se estudiará en detalle los conceptos básicos y fundamentales de diseño de sistemas de cómputo de propósito general, que permitan al estudiante desempeñarse en diseño lógico, y programación de bajo nivel en cualquier rango de equipamiento que use microprocesadores capaces de correr sistemas operativos real time, o de propósito general, diseñar interfaces de hardware para entrada salida y sus correspondientes device drivers, empleando microprocesadores de alta gama de procesamiento paralelo (SMP o AMP), de propósito general o embebidos, determinando claramente y en la práctica la relación de sus recursos de arquitectura de hardware con el sistema operativo, y el rango de aplicaciones de sistemas con alto paralelismo a nivel de instrucción

#### COMPETENCIAS DE EGRESO ESPECÍFICAS A LAS QUE CONTRIBUYE:

Competencia	Competencias de Actividades reservadas			Competencias de Alcances
	Baja	Media	Alta	
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,			X	



económicas, humanas y ambientales.				
CE 1.2: Plantear, interpretar, modelar y resolver los problemas de ingeniería descritos en CE 1.1.		X		
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.				X
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.				X

#### COMPETENCIAS DE EGRESO GENÉRICAS A LAS QUE CONTRIBUYE:

Competencia	Baja	Media	Alta
Concebir, diseñar y desarrollar proyectos de Ingeniería Electrónica		X	
Utilizar de manera efectiva las técnicas y herramientas de aplicación de la ingeniería Electrónica.			X
Contribuir a la Generación de desarrollos tecnológicos y/o innovaciones tecnológicas.			X
Desempeñarse de manera efectiva en equipos de trabajo.		X	
Aprender en forma continua y autónoma.		X	
Actuar con espíritu emprendedor		X	

#### OBJETIVOS (APRENDIZAJE/LOGROS A ALCANZAR)

Después de haber cursado Técnicas Digitales III se busca que los alumnos sean capaces de:

- Diseñar e implementar sistemas de procesamiento digital de señales para un rango de aplicaciones diverso.
- Comprender la arquitectura de una red de datos, sus protocolos asociados, y su aplicación a la interconexión de dispositivos.



- Desarrollar proyectos de software en sistemas computacionales que involucren un Sistema Operativo de propósito general, su relación con los recursos de hardware de la CPU, y sus requerimientos particulares.
- Comprender el funcionamiento de las micro-arquitecturas de CPUs complejas en las que se implementen los principios de paralelismo a nivel de instrucción y a nivel de datos, para distinguir su campo de aplicación, así como las fortalezas de dichas tecnologías.
- Comprender los modelos de arquitectura y organización de Microprocesadores con paralelismo a nivel de datos para su aplicación efectiva en el desarrollo aplicaciones de Procesamiento digital de señales, imágenes, y video explotando estas capacidades para mejorar el rendimiento del sistema.
- Diseñar el software de arranque “bare metal” para un sistema multitarea aplicable a un procesador embebido o de alto desempeño, y su interacción con el setup del sistema operativo, para emplear los recursos de arquitectura de programación de sistemas de un procesador con Paralelismo a Nivel de Instrucciones, aplicando los recursos más avanzados en las herramientas de desarrollo de software.
- Relacionar los recursos de Arquitectura de Sistemas de los Procesadores de Propósito General con la implementación de las funciones del sistema operativo que se apoyan directamente en estos recursos, para distinguir la necesidad de su utilización y optimizar su uso.
- Comprender la arquitectura interna de un sistema operativo multitasking, en los aspectos de más bajo nivel, para poder desarrollar drivers de dispositivos de E/S existentes en el sistema, y los recursos de programación necesarios para desarrollar aplicaciones accesibles a través de una red de datos.
- Comprender las herramientas y recursos de desarrollo que permiten personalizar el kernel de un sistema operativo, para adaptarlos a las necesidades de una solución de ingeniería determinada, utilizando la cantidad de recursos mínima necesaria y optimizando así el consumo de recursos y de energía eléctrica, construyendo una imagen lista para arrancar el sistema y operar las aplicaciones desarrolladas

## **CONTENIDOS**

### **Contenidos mínimos**

- 1) Procesamiento Digital de Señales y su relación con el hardware.
- 2) Redes de Datos. Protocolos.
- 3) Sistema Operativo de propósito general y su relación con el hardware.



## **Contenidos analíticos**

### **Unidad 1: Arquitectura y Organización de un Computador.**

Introducción a los sistemas computacionales de 32 y 64 bits de propósito general. Diferencias entre los sistemas embebidos, y las computadoras personales y servers back end.

Concepto de Arquitectura y organización de un computador. Stack de Componentes:

- Hardware
- Firmware
- Sistema Operativo
- Aplicaciones.

Conceptos generales de diseño y funcionamiento.

Descripción del arranque de un sistema de cómputo de propósito general: Inicialización y Bootloader (Firmware) + Gestor de arranque (disco) + Sistema operativo (disco).

Descripción del arranque de un sistema embebido: Inicialización + Bootloader + Kernel (Firmware - Imagen en ROM).

Estructura de un sistema de propósito general para escritorio o Server Back end. El POST: Inicialización del sistema y bootstrap loader.

Introducción a un bootloader de un sistema embebido. First Stage Boot Loader. Compilación, customización del hardware, y personalización del kernel.

### **Unidad 2: Necesidad de un Sistema Operativo**

#### **Introducción**

Concepto de administración de recursos de hardware, y Funciones de un Sistema Operativo.

Arquitectura genérica de un Sistema Operativo: Device Drivers, Kernel, e Interfaz de Usuario, Mecanismos de acceso al Kernel desde las aplicaciones: System Calls, o APIs.

Descripción de la arquitectura UNIX y su evolución en versiones: BSD (Berkeley Software Distribution), y ATT. Origen y evolución de Linux. Conceptos rectores de intercomunicación de aplicaciones y uso de entrada/salida. Aplicación del concepto "everything is a file".



### **Estándar POSIX:**

Motivación de la estandarización. Sistemas operativos POSIX compatibles. Su Aplicación en Linux. Linux para sistemas Embebidos personales y back end. Ventajas de POSIX: Portabilidad de aplicaciones. Productividad en el desarrollo de sistemas aportada por el uso de estándares.

### **Revisión de temas**

Breve Revisión de conceptos de File System, Procesos, IPC's y threads, impartidos en Informática I y en Informática II.

### **Formatos y reglas de ejecución de aplicaciones**

ELF (Executable and Linkable Format). Estructuras principales y datos del encabezado. Acceso al encabezamiento ELF de objetos y binarios ejecutables.

### **Unidad 4: Recursos de Arquitectura de soporte al Sistema Operativo en Procesadores de 32/64 bits.**

#### **Gestión de memoria. Conceptos generales:**

Memoria Virtual: Breve descripción del sistema de administración por parte del sistema operativo.

Administración de memoria Física por Segmentación. Características de los segmentos. Administración de los segmentos. Traslación de una dirección.

Administración de memoria Física por Paginación. Características de las Páginas. Ventajas de la paginación en la administración de Memoria Virtual. Mecanismo de traducción de una dirección virtual a física. Estructura de traducción en niveles jerárquicos. Administración de las páginas por tareas. Traducción de direcciones. Translation Lookside Buffer: Funcionamiento y contenido.

Lineamientos de diseño para el subsistema de administración de memoria de un kernel multitarea. Protección de memoria para cada tarea mediante los recursos de Paginación. Estructura de traducción por cada tarea. Páginas que deben estar visibles para cada tarea en su estructura de traducción.

#### **Procesador ARMv7.**

Traducción de direcciones virtuales a físicas de 32 bits. Descriptores de Página, Estructura jerárquica de traducción en 2 niveles. Directorio de Tablas de Páginas, Tablas de Páginas, Contenido y Acceso, Descripción de los recursos del Procesador dedicados al manejo de la Paginación.



## **Unidad 5: Procesamiento Digital de Señales**

Concepto de Procesamiento Digital de Señales.

Características de los algoritmos de Procesamiento de Señales. Aplicación del análisis discreto de Fourier a partir de señales digitalizadas.

Arquitecturas de hardware para procesamiento de estos algoritmos en tiempo real. Procesadores Digitales de Señal. Paralelización de Buses de datos y operandos. Abandono del modelo Von Newman. Arquitecturas Harvard para DSP.

Análisis de las microarquitecturas x86 y ARMv7 recientes desde esta óptica. Conclusiones y aspectos destacables. Uso de los procesadores mencionados como DSP. Limitaciones del Cortex-M.

### **Modelo SIMD.**

Paralelismo a nivel de datos. Concepto de procesadores vectoriales.

Características distintivas de los algoritmos DSP. Concepto de aritmética saturada y de desborde. Aplicaciones en cada caso.

Tipos de datos empaquetados enteros y de punto flotante.

Aplicaciones de Filtrado de audio y señales en general. Efectos de sonido sobre señales de audio. Convolución de señales

### **Procesamiento de imágenes.**

Representación de imágenes digitalizadas. Algoritmos de detección de bordes, morfologías, filtrado y detección de patrones en Imágenes. Convolución 2D

### **Extensiones de instrucciones ARMv7 para DSP:**

Extensiones NEON. Ejemplos de aplicación en el procesamiento digital de señales con NEON.

## **Unidad 6: Organización y Microarquitectura**

Conceptos de organización y relación con el hardware.

Organización jerárquica de memoria. Memoria cache: Funcionamiento, Principio de vecindad. Cache de Mapeo Directo, desventajas. Cache Asociativo. Cache multinivel: L1, L2, L3. Políticas de escritura en un cache. Implementación de coherencia de Caches en sistemas SMP: recursos de hardware involucrados y protocolos para su aseguramiento. Protocolo M.E.S.I. Diagrama de estados.



Ejemplos y análisis de Organización de caches en procesadores de arquitecturas homogéneas (Intel AMD) y heterogéneas (ARM).

Máquina básica de ejecución de Instrucciones. Paralelismo a nivel de Instrucción (ILP) y paralelismo a Nivel de datos (DLP). Evolución del ILP: Pipeline de instrucciones, Procesadores Superescalares, Ejecución Fuera de Orden, Ejecución especulativa, Procesadores Multithread, Implementaciones prácticas Hyperthreading (Intel), ARM, Niagara (SUN).

Tecnología Multicore y Manycore. Diferencias con Multithread. Combinaciones Multicore / Multithread.

### **Unidad 7: Introducción a los Sistemas de Multi Procesamiento.**

Multi Procesamiento Simétrico y Asimétrico. Características distintivas y a Aplicación de cada tipo. Requerimientos para un sistema operativo.

Visión introductoria de los Protocolos de inicialización de un SMP basado en procesadores x86, y de los recursos asociados a Multiprocesamiento: Local APIC e IO/APIC, funcionamiento, descripción, activación, inicialización y mensajes entre procesadores. Introducción al diseño de un scheduler de tareas para sistemas Multiprocesador.

### **Unidad 8: Tópicos avanzados sobre Sistemas Operativos**

Sistema de arranque de Linux. Consideraciones particulares para Inicializar un sistema Linux embebido.

#### **Tratamiento directo con el hardware: El Boot Loader.**

Proyectos de Boot Loaders open source para sistemas embebidos: U-Boot (Universal Boot Loader). Plataformas soportadas. Tarjetas de desarrollo soportadas. Funciones que debe realizar un bootloader:

- Inicialización del procesador.
- Inicialización del controlador de DRAM,
- Inicialización del Hardware de E/S.

Estructura de directorios de U-Boot. Criterios para adaptación a diseños propios. Traspaso del control al Sistema Operativo para que inicie su carga y ejecución.

#### **File Systems.**

Breve revisión de File System, y sus características distintivas en Linux. Soporte en Linux a FS no nativos: caso NTFS. Montaje y desmontaje de FS.





Concepto avanzados: Implementaciones para disco y flash ROM. Concepto de bloque. Block device. Gestión de bloques inválidos, garbage colector y gestión de errores.

File system virtuales. Ejemplo: /proc. Información relevante.

Estructura de un File System: Elementos constitutivos: Suprbloque, I-Nodo, Fichero, Directorio. Flash Translation Layer (FTL). Diferentes File Systems nativos de Linux: File Systems: ext4, ext3, ext2, CRAMFS, JFFS2, YAFFS/YAFFS2, SQUASHFS, LOGFS, UBIFS.

### **Device Drivers**

Módulos de kernel. Enlace dinámico vs enlace estático. Ambiente de programación del kernel. Mapeo de funciones de acceso a archivo sobre el Hardware de E/S. Criterios para el desarrollo de Devices Drivers para sistemas embebidos. La entrada de Linux sobre ARM. La solución al caos: Device Tree. Entornos de programación. Herramientas de desarrollo. Ejemplos prácticos de aplicación: Desarrollo de un driver completo : Implementación práctica de sus principales funciones. Uso de dispositivos en Linux para el procesamiento digital de señales: Desarrollo de aplicaciones de procesamiento de audio e imágenes en tiempo real.

### **Personalización del kernel. BSP's**

Como construir un kernel customizado a las necesidades de un sistema embebido. Que es un BSP, que problemas resuelve y cuales son sus ventajas. Diferentes BSP's Android, OpenEmbeddedd, OpenWrt, Yocto Project.

Ventajas del uso de Yocto. Arquitectura de Yocto: Metadata: Capas, Recetas, Clases. Bitbake: Archivos de Configuración. Creación de una imagen de kernel a medida de un sistema. Construcción de diferentes imágenes-

### **Unidad 9: Sistemas de Redes.**

Introducción a los sistemas de Comunicaciones de Datos. Principios Básicos. Entorno distribuido. Modelo Cliente-Servidor (Client-Server). Modelo ISO-OSI.

Modelos LAN y WAN. LANs Standards.

Standards de capa 2:

Ethernet (IEEE 802.3), principales características Estructura del frame Ethernet. Dirección MAC. Colisiones: tratamiento. Tamaño del paquete. Diferentes adaptadores. Métodos de Control de Transmisión y Acceso al medio físico.

Protocolos de comunicación en capa 3 y 4:

TCP/IP Esquema de Addressing, Paquetes, Protocolos. Formato de una Dirección IP. Asignación de una Dirección IP a los Nodos de la Red. Mapeo de la Dirección IP a la



Dirección MAC. Resolución de dirección de enlace (física para la determinación de un nodo a nivel de capa 2: Protocolo ARP.

Búsqueda y Utilización de Servicios en la Red. TCP y UDP. Naming y Addressing en TCP/IP. Concepto de port. Relación con el proceso que implementa el servicio. Registro de los servicios en el sistema operativo. Archivos de configuración para resolver servicios. Well known ports. Concepto y descripción de los principales ports. Técnicas de programación: Llamadas standard al sistema operativo y estructuras auxiliares de uso en la determinación de los puertos. Uso de funciones para conversión de formatos network a string, o byte order. Ejemplos de aplicación Sockets: Técnicas y funciones de programación de aplicaciones que se intercomunican a través de una red de datos. Desarrollo de aplicaciones servidores concurrentes. Interacción del stack TCP/IP con el sistema Operativo.

#### DISTRIBUCIÓN DE CARGA HORARIA ENTRE ACTIVIDADES TEÓRICAS Y PRÁCTICAS

Modalidad organizativa de las clases	Horas Reloj totales presenciales	Horas reloj virtuales totales	Horas totales
<b>Teórica</b>	48	0	48
<b>Formación práctica</b>	72	0	72

Tipo de prácticas	Horas Reloj totales presenciales	Horas reloj virtuales	Lugar donde se desarrolla la práctica
Formación experimental	20		Laboratorios 105, 109 y 110 del Dpto. de Ing. Electrónica UTN.BA sede Medrano
Problemas abiertos de Ingeniería (ABP)	24		Laboratorios 105, 109 y 110 del Dpto. de Ing. Electrónica UTN.BA sede Medrano
Proyecto y diseño	28		Laboratorios 105, 109 y 110 del Dpto. de Ing. Electrónica UTN.BA sede Medrano
Otras:			
Práctica supervisada			
<b>Total de horas</b>	72		



## **ESTRATEGIAS DE ENSEÑANZA Y ACTIVIDADES DE APRENDIZAJE**

### **ESTRATEGIAS DE ENSEÑANZA Y ACTIVIDADES DE APRENDIZAJE**

Se procura continuar imprimiendo en los alumnos la impronta general del estado del arte de la actividad profesional, tanto en ámbitos empresariales como científicos iniciada en el curso piloto de Técnicas Digitales II. Al igual que en el curso mencionado, se proporciona un abordaje general, evitando la impronta personal del docente, ya que se pretende una mirada sin sesgo personal. El esfuerzo es entender la industria de sistemas de cómputos mas allá de los muros de nuestra propia actividad y de las fronteras de la Argentina, proporcionando al estudiante una formación que lo prepare para superar las limitaciones evidentes de la industria local, y convertirlo en un agente de cambio capaz de impulsar el desarrollo que el país requiere.

Por otra parte se procura proporcionar capacidad para trabajo por proyectos desarrollados por equipos multidisciplinarios conformados por especialistas en las diferentes áreas disciplinares en virtud de la complejidad tecnológica que cada área ha alcanzado de la mano de la Investigación científica y el desarrollo tecnológico consecuente. Se considera importante que la asignatura pueda desarrollar esta competencias en los estudiantes a través de proyectos y aplicaciones que eventualmente puedan propender a la Integración Horizontal de contenidos con el resto de las asignaturas del 5to. Nivel del plan de estudios, brindándoles una visión mayor de las perspectivas de aplicación de las tecnologías digitales de alta prestación en otras áreas de la ingeniería Electrónica cuyos contenidos son compatibles con ésta asignatura para la implementación de proyectos de espectro mas amplio: como así también con las asignaturas relacionadas de los niveles previos en base a un proyecto de Área consistente que pueda hilar competencias y conocimientos a lo largo de la formación a medida que el estudiante transita por los diferentes niveles de la carrera.

#### **Soporte tecnológico**

Las consultas y comunicaciones se pueden efectuar en cualquier momento de cualquier día de la semana. Para ello utilizamos un server en Discord que nos permite a los docentes responder consultas y resolver dudas de manera no presencial. De este modo un estudiante puede consultar a los docentes cualquier problema que lo estanca en el avance de su trabajo.

Combinado con Discord utilizamos el servidor gitlab de la Facultad, para que los estudiantes entreguen sus trabajos. Además se podrán subir versiones con problemas para que los docentes podamos verificar los reportes de problemas y consultas en Discord y podamos orientarlos en su resolución entre clases, evitándoles esperar a la clase siguiente para resolver sus dudas.



Teniendo en cuenta el requisito de evaluación continua para la promoción, establecido en el nuevo reglamento de estudios de la UTN implementado en el ciclo lectivo 2017, se implementa un sistema de evaluación continua. Se divide la asignatura en dos instancias de evaluación, una por cada cuatrimestre. Las dos se basarán en la metodología de entrega individual de un Trabajo Práctico integrador que ha sido desarrollado en etapas incrementales en las cuales los docentes pueden evaluar a cada estudiante de manera continua. Al final del cuatrimestre el Trabajo Práctico final será la resultante de dicho desarrollo incremental y debe estar en condiciones de funcionamiento en el repositorio git. Por cada entrega parcial durante el cuatrimestre se realiza un examen escrito que permita verificar que el estudiante comprende los conceptos teóricos detrás del código que termina de entregar.

### **Trabajos Prácticos**

Los Trabajos Prácticos se diseñan como secuencias incrementales de funcionalidades. Se parte de un esquema minimalista de arranque, o de los Trabajos prácticos producidos en el curso piloto de Técnicas Digitales II, según si el tema a abordar inicia en esta asignatura o completa un tema iniciado en el piloto de Técnicas Dlgitales II. En cualquier caso, en cada paso se le agregan funciones y prestaciones. El trabajo práctico final de cada secuencia o trayecto formativo será la resultante de un proceso de desarrollo gradual en el que cada paso significa aplicar en la práctica un nuevo concepto teórico.

Los trabajos prácticos se entregan en el repositorio gitlab. La parte práctica debe ser desarrollada por el alumno en forma individual, parte del trabajo se ejecuta en clase en presencia de los docentes y otra parte fuera del horario de clase asistido por los docentes a demanda mediante el soporte tecnológico de discord y gitlab.

Se espera de esta forma que el Trabajo Práctico final de cada trayecto formativo obre como elemento de aprendizaje efectivo ya que requiere del alumno tiempo extra al de clase (es una estrategia que le exige el necesario tiempo de aplicación y estudio sin el cual es imposible aspirar a aprobar los trabajos prácticos como mínimo o promocionar la asignatura como meta superior).

La guía de trabajos integradores incluyen los siguientes hitos de desarrollo:

Primer Cuatrimestre:

Trayecto 1

Se toma el producto del Trayecto 1 del curso piloto de Técnicas Digitales II y se lo continúa a partir del entregable aprobado, con los siguientes pasos.

- Habilitación del Coprocesador 15 del procesador CORTEX-A de la Zynq-7000.



- Diseño de un conjunto de tablas de página que permitan ubicar al kernel en páginas propias, e implementar un sistema funcional de administración de memoria para las tareas implementadas en el curso piloto de Técnicas Digitales II.
- Implementar el sistema de excepciones por fallos en el acceso a páginas.
- Implementar un sistema de demanda de páginas para las tareas.

#### Trayecto 2

- Habilitación del Coprocesador 14.
- Desarrollo de una tarea simple que utilice instrucciones Neon
- Modificar el scheduler para implementar lazy fpu en el context switch

#### Trayecto 3

- Análisis de trabajos científicos sobre Micro Arquitectura relacionados con los tópicos cubiertos por la asignatura
- Implementación de algunos de los bloques simples en la FPGA de la Zynq-7000.

#### Trayecto 4

- Implementación de un server TCP IP concurrente partiendo de los programas de **sockets** vistos en Informática I.
- Implementación de señales para el control de la finalización de las instancias hijo del server.
- Instalación del server en el SoC Zynq-7000 booteando LINUX
- Desarrollo de un cliente que pruebe la funcionalidad del server.
- Exploración con Wireshark de las transacciones analizando los paquetes que componen cada mensaje en cada capa del protocolo.

#### Segundo Cuatrimestre:

#### Trayecto 5

- Puesta en marcha de la FPGA y del SoC Zync-7000 bajo Linux.



- Tomar las aplicaciones del Trayecto 3 del curso piloto de TDII y agregarles procesamiento digital de señales utilizando el set de instrucciones NEON. Uso de *pydfax* para cálculo de los coeficientes de los filtros a implementar.

#### Trayecto 6

- Diseño de un driver de caracter para el dispositivo customizado en Verilog con acceso por DMA y conexión al Bus mediante conectividad AXI provista dentro de la IP de Xilinx, desarrollado en el Trayecto 4 del curso piloto de TDII.

#### Evaluación

Por cada trayecto de formación práctica se evalúa además como mínimo un examen escrito.

De este modo se evalúa cada componente del trabajo práctico no solo desde su implementación sino desde la comprensión cabal de los conceptos teóricos que le dan al desarrollo de cada competencia el necesario sólido sustento del conocimiento, condición indispensable para un profesional de excelencia.

Al final de cada cuatrimestre las calificaciones surgen de la combinación ponderada de las evaluaciones de cada trayecto en sus aspecto práctico y teórico (examen escrito asociado).

En caso de obtener 8 o mas puntos en cada examen parcial el participante promocionará la asignatura.

Puede recuperar uno de los dos parciales en caso que éste no llegue a esa calificación. Nunca ambos. La nota definitiva será la del recuperatorio, ***aun si esta resultase menor a la calificación que se quiso recuperar. Ejemplo: Si se busca recuperar un 7 para convertirlo en 8 y promocionar y la nota del recuperatorio es menor de 7, queda la del recuperatorio como definitiva. Aun si desaprueba.***

#### ARTICULACIÓN HORIZONTAL Y VERTICAL CON OTRAS MATERIAS

##### Articulación con las materias del 5to. Nivel

El 5to. Nivel de la carrera se caracteriza por la cantidad de asignaturas de tecnología aplicada que requieren un esfuerzo muy alto para regularizar o promocionar la asignatura.

En diferentes relevamientos realizados con estudiantes, hemos podido constatar las siguientes opiniones representativas de la mayor parte de los encuestados:

1. Las materias de este nivel son muy diversas y requieren todas de un esfuerzo muy alto . No es posible cursarlas todas en el mismo ciclo lectivo exitosamente.
2. Los contenidos de estas asignaturas son sumamente amplios, y constituyen en algunos casos áreas de especialidad.
3. Por lo general un sistema de cómputo termina siendo un componente esencial en cualquier proyecto que se realice en el resto de las asignaturas de la carrera.



Procuramos articular nuestros prácticos con algún concepto de otras asignaturas de la carrera y en particular del 5to Nivel, tratando de no saturar a los estudiantes en función del feedback que nos proporcionan de este nivel de la carrera

En particular trabajamos con mas énfasis las interfaces de Adquisición de datos ya que las mismas pueden ser aplicadas en proyectos que involucren acondicionamiento de señales débiles o en algún tipo de modulación digital (Electrónica Aplicada III), o pueden limitar en banda un espectro de señal de entrada para desarrollar filtros digitales (Teoría de Circuitos II), utilizando las instrucciones del modelo de Ejecución SIMD (NEON). En el caso de Medidas Electrónicas I, nos concentramos especialmente en las interfaces ya descriptas a la entrada de los conversores AD.

Al abordar Micro Arquitectura estudiamos cuestiones de tecnología de integración que nos permiten articular con Tecnología Electrónica, y estudiamos también la problemática del consumo de energía con algunos puntos de contacto con Electrónica de Potencia, en especial al analizar sistemas de cómputo portátiles, en los cuales la duración de la batería es un aspecto crítico en el diseño.

### **Articulación con el Área**

Uno de los principales objetivos de este programa de estudio y de los contenidos propuestos tanto para los tópicos teóricos como los recursos seleccionados para su implementación práctica es convertir a esta asignatura en un eslabón fundamental dentro del Área Técnicas Digitales.

En las asignaturas Informática I e Informática II se desarrollan competencias de programación con lenguajes estructurados (C), y orientados a Objetos (C++), con base en SoCs de gama media y baja, gestionados con Microcontroladores. Se trabaja con la Arquitectura ARMV7 perfil M (que en rigor en algunos modelos como el Cortex M0 es ARMv6).

No obstante los estudiantes comprenden como programar cualquier dispositivos mas o menos básico de Entrada Salida, y programan sus handlers de interrupción de manera fluida. Los proyectos de Informática II dan prueba sobrada de esta competencias.

Informática I está trabajando en una modificación en la implementación práctica de la primer parte de la asignatura que trabaja exclusivamente en lenguaje C, sobre la misma plataforma de microcontrolador que se utiliza en Informática II de modo de darle a esta segunda asignatura del área un mejor vuelo de aplicación.

Aun sin esta modificación que aun se trabaja a nivel conceptual en Informática I, no tiene sentido seguir trabajando con microcontroladores en la carrera, ya que los microprocesadores tienen un rango de aplicaciones muy superior y además una complejidad para su comprensión y aplicación que justifica ampliamente su abordaje a lo largo de dos cursos.

Por otra parte en la asignatura Técnicas Digitales I aborda el diseño de los bloques básicos de sistemas lógicos con el enfoque de Diseño RTL de modo de poderlos utilizar como bloques constitutivos de sistemas lógicos de complejidad creciente, y su descripción se efectúa a través



de Lenguaje de Descripción de Hardware (Actualmente VHDL, aunque se plantea en 2024 migrar a Verilog). La filosofía de construir sistemas mas complejos partiendo de los anteriores permite que en varios de sus cursos se llegue a diseñar un core sencillo de procesador, su interfaz con memoria y su pipeline de ejecución.

Técnicas Digitales II introduce al estudiante en el primer curso de Microprocesadores de la carrera cuyo plan de trabajos prácticos se desarrolla sobre el mismo SoC FPGA.

Técnicas Digitales III en la presente versión piloto articula de manera muy clara con Técnicas Digitales II, aliviando respecto de la versión "Regular" de la materia un 30% de su carga de esfuerzo, asegurando al final de los cinco cursos del Área Técnicas Digitales las mismas competencias en los estudiantes.

Con esta articulación es posible darle mas tiempo de maduración a temas de gran importancia conceptual y de amplia aplicación práctica.

#### CRONOGRAMA ESTIMADO DE CLASES

Clase N°	Descripción	Modalidad
Clase 1	Presentación de la asignatura. Objetivos. Modalidad. <b>Teórica:</b> <b>Material de soporte Audiovisual:</b> Video Clases Teóricas 8 y 9. <b>Temario:</b> Modelo de programación de sistemas. Gestión de memoria dinámica. Paginación. Traducción de páginas. Uso de niveles jerárquicos de Traducción. Memoria Virtual: Funcionamiento, Recursos del Procesador para uso del Sistema Operativo. Habilitación del coprocesador 15 en ARMv7.	Presencial
Clase 2	<b>Teórica:</b> <b>Material de soporte Audiovisual:</b> Video Clase Teórica 10. <b>Temario:</b> Administración de memoria en CORTEX-A y CORTEX-R. Administración de memoria dinámica. Funcionamiento de la Unidad de Traducción de Paginas de un CORTEX - A. Estructuras de tablas. Formatos Long y Short Descriptor. Tablas, Descriptores, Registros de uso del sistema. Excepciones asociadas. ----- <b>Laboratorio:</b> Tomar el Trabajo Práctico de TDII. Habilitación del coprocesador 15. Estructura de traducción multinivel, Descriptor corto.	Presencial
Clase 3	<b>Laboratorio.</b> Administrador de memoria por paginación.	Presencial
Clase 4	<b>Laboratorio:</b> Continuación del administrador de memoria por paginación.	Presencial





Clase 5	<p><b>Evaluación Teórica Trayecto 1 (20 minutos)</b></p> <p><b>Teórica:</b> <b>Material de soporte Audiovisual:</b> Video Clase Teórica 11 <b>Temario:</b> ARMv7 – Extensiones Multimedia. Modelo SIMD. NEON. Registros, Formatos Numéricos. Aritmética saturada vs. Aritmética de desborde. Procesamiento Digital de Señales. Ejemplos de Aplicación.</p> <hr/> <p><b>Laboratorio:</b> Habilitación de Coprocesador 14. Análisis del context switch. Agregado de la función conocida como <i>lazy fpu</i> para resguardar el contexto de registros SIMD solo cuando se requiere minimizando su carga de procesamiento hacia desde memoria.</p>	Presencial
Clase 6	<p><b>Laboratorio:</b> Incorporación de dos tareas adicionales simples pero que utilicen instrucciones SIMD.</p>	Presencial
Clase 7	<p><b>Evaluación Teórica Trayecto 2 (20 minutos)</b></p> <p><b>Teórica:</b> <b>Material de soporte Audiovisual:</b> Video Clase Teórica 19 <b>Temario:</b> Introducción a la Tecnología y Organización de los Sistemas de cómputo. Tendencias en tecnología de Integración. Scaling. Lithography. Frecuencia de clock. Performance. Energía. Régimen estático y dinámico. Problemas de las corrientes de leakage. Métricas. Perspectiva de evolución. Micro Arquitectura. Memoria caché: Funcionamiento, Principio de vecindad. Cache de Mapeo Directo, desventajas. Cache Asociativo de 2 vías (extensión del concepto a 4 y 8 vías). Cache L1, L2, L3. Inserción, búsqueda, y escritura</p> <hr/> <p><b>Laboratorio:</b> Continuación del administrador de memoria por paginación con el agregado de las nuevas tareas.</p>	Presencial
Clase 8	<p><b>Teórica:</b> <b>Material de soporte Audiovisual:</b> Video Clase Teórica 19 <b>Temario:</b> Micro Arquitectura. SMP: Coherencia, recursos de hardware y protocolos para su aseguramiento. Protocolos MSI, MESI, MESIF y MOESI. Diagrama de estados. Ejemplos de Caché en ARM AMD e Intel.</p> <hr/>	Presencial



	<b>Laboratorio:</b> Continuación del administrador de memoria por paginación.	
Clase 9	<b>Teórica:</b> <b>Material de soporte Audiovisual:</b> Video Clase Teórica 19 <b>Temario:</b> Memorias DRAM. Funcionamiento. Organización Interna. Estructura de una celda. Refresco, Realimentación. Arquitecturas: Evolución. Tecnologías. Standard JEDEC. Configuración. Protocolos de acceso. Controlador de memoria. Funciones. Ejemplo: Procesador Sitara XAM3359AZCZ100, (CORTEX-A8) en la tarjeta BeagleBone Black. Registros de configuración del controlador integrado. Parámetros característicos. Código de inicialización. Análisis de hojas de datos de memorias y relación con la inicialización del controlador. ----- <b>Laboratorio:</b> Continuación del administrador de memoria por paginación	Presencial
Clase 10	<b>Teórica:</b> <b>Material de soporte Audiovisual:</b> Video Clase Teórica 20 <b>Temario:</b> Máquina básica de ejecución de Instrucciones. Paralelismo a nivel de Instrucción y paralelismo a Nivel de datos. Evolución del ILP: Pipeline de instrucciones, Procesadores Superescalares, Ejecución Fuera de Orden, Ejecución especulativa, Procesadores Multithread, Implementación de Intel, Hyperthreading. Tecnología Multicore y Manycore. Implementaciones en IA-32. Diferencias con Multithread y mix con Multithread. ----- <b>Laboratorio:</b> Continuación del administrador de memoria por paginación.	Presencial
Clase 11	<b>Laboratorio:</b> Continuación del administrador de memoria por paginación.	Presencial
Clase 12	<b>Laboratorio:</b> Continuación del administrador de memoria por paginación.	Presencial
Clase 13	<b>Teórica:</b> <b>Material de soporte Audiovisual:</b> Video Clase Teórica 14 <b>Temario:</b> Redes de datos. OSI: Modelo de capas. Capa 2 Ethernet. Frame 802.3 y 802.11. TCP / IP. Internetworking, servicios, ports, sockets. Modelo cliente servidor. Three way handshake, funciones de conexión asociadas. Wireshark.	Virtualidad Sincrónica



Clase 14	<b>Laboratorio:</b> Taller de programación en Linux: sockets. Cliente-servidor sencillo. Programación Concurrente: Uso de <i>fork ()</i> y Pthreads.	Presencial
Clase 15	<b>Evaluación Teórica Trayecto 3 (Entrega Trabajo de análisis de Microarquitectura)</b> <b>Laboratorio:</b> Agregar al server algoritmos básicos con instrucciones NEON. Taller de programación DSP. Algoritmos para procesar señales e imágenes, escritos en ASM utilizando instrucciones NEON, aptos para ser invocados desde programas escritos en Assembler C y C++.	Presencial
Clase 16	<b>Laboratorio:</b> Taller de Programación en Linux: sockets. Desarrollo de servidores concurrentes. Manejo de ports TCP y UDP. Aplicaciones. Funciones de resolución de nombres, y manejo de direcciones en formato network. Espera de múltiples eventos desde un mismo hilo o proceso. Introducción de elementos de programación avanzada en Linux: IPCs, espera de múltiples eventos bloqueantes. Control de parentesco. de procesos concurrentes.	Presencial
Clase 17	<b>Teórica:</b> <b>Material de soporte Audiovisual:</b> Video Clases Teóricas 12 y 13. <b>Temario:</b> Concepto de administración de recursos de hardware, y Funciones de un Sistema Operativo. Arquitectura genérica de un Sistema Operativo: Device Drivers, Kernel, e Interfaz de Usuario, Mecanismos de acceso al Kernel desde las aplicaciones: System Calls, o APIs. Descripción de la arquitectura UNIX y su evolución en versiones: BSD (Berkeley Software Distribution), y ATT. Origen y evolución de Linux. Conceptos rectores de intercomunicación de aplicaciones y uso de entrada y salida. Aplicación del concepto “everything is a file”. Estándar POSIX. Arquitectura de Linux. Gestión de Memoria y de procesos. <i>fork ()</i> , <i>execp ()</i> . Gestión de procesos, Process control Block ( <i>task_struct</i> ). Estados de los procesos. Señales.	Presencial
Clase 18	<b>Evaluación Teórica Trayecto 4 (20 minutos)</b> <b>Laboratorio:</b> Taller de Programación: Introducción a la Programación paralela. Aplicar al trabajo práctico final el uso de Threads. Sincronización, Mutex, Variables de condición, etc.	Virtualidad Sincrónica



Clase 19	<p><b>Teórica:</b> <b>Material de soporte Audiovisual:</b> Video Clases Teóricas 15 y 16 <b>Temario:</b> File Systems: Conceptos y definiciones. Funciones Generales - Funciones avanzadas. Alto nivel de abstracción: Virtual File System: Estructuras <i>file</i> e <i>i-node</i>. Implementaciones: extensión 2, 3 y 4. y JFFS2, File Systems para Flash. Breve Revisión de conceptos de File System, Procesos, IPC's y threads. ELF (Executable and Linkable Format). Estructuras principales y datos del encabezado. Acceso al encabezamiento ELF de objetos y binarios ejecutables. Acceso al hardware en Linux. Acceso desde modo User: el sysfs. Primitivas del kernel para dispositivos. Uso de POSIX. Device Drivers: Clasificación. Linux Driver Model: Visión jerárquica del hardware. Kernel Objects. Programación de kernel modules.</p>	Presencial
Clase 20	<p><b>Laboratorio:</b> Taller de programación. Evolución del server para incorporarle funciones de procesamiento digital de señales.</p>	Presencial
Clase 21	<p><b>Teórica:</b> <b>Material de soporte Audiovisual:</b> Video Clase 17 <b>Temario:</b> Char Devices. Desarrollo de un char device. Números Mayor y Menor, enlace con las POSIX syscall. Registro de un driver. Manejo de Interrupciones. Interacción con el kernel para esperar eventos y poner a dormir al proceso invocante. Problemática para acceso a hardware heterogéneo. El caso ARM. Device Tree. Generación, funcionamiento.</p>	Presencial
Clase 22	<p><b>Laboratorio:</b> Programación de Kernel Modules y primeros pasos para el Driver del dispositivo de la FPGA traído del Trabajo Práctico del curso piloto de Técnicas Digitales II.</p>	Presencial
Clase 23	<p><b>Teórica:</b> <b>Material de soporte Audiovisual:</b> Video Clase 18 <b>Temario:</b> Platform Devices. Estructuras y jerarquías asociadas. Funciones del api del kernel .</p> <p>-----</p> <p><b>Laboratorio:</b> Programación de Drivers.</p>	Presencial
Clase 24	<p><b>Laboratorio:</b> Programación de Drivers.</p>	Presencial



Clase 25	<b>Laboratorio:</b> Programación de Drivers.	Presencial
Clase 26	<b>Evaluación Teórica Trayecto 5 (30 minutos)</b> <b>Laboratorio:</b> Programación de Drivers.	Presencial
Clase 27	<b>Laboratorio:</b> Programación de Drivers.	Presencial
Clase 28	<b>Laboratorio:</b> Programación de Drivers.	Presencial
Clase 29	<b>Laboratorio:</b> Programación de Drivers. Ajustes al server.	Presencial
Clase 30	<b>Laboratorio:</b> Programación de Drivers. Ajustes al server.	Presencial
Clase 31	<b>Laboratorio:</b> Programación de Drivers. Ajustes al server.	Presencial
Clase 32	<b>Laboratorio:</b> Programación de Drivers. Ajustes al server.	Presencial

## BIBLIOGRAFÍA OBLIGATORIA

Hohl, W., & Hinds, C. (2015). ARM assembly language: Fundamentals and techniques (Second Edition). CRC Press.

Pyeatt, L. D. (2016). Modern Assembly Language Programming with the ARM Processor. Elsevier. <https://doi.org/10.1016/B978-0-12-803698-3.00001-2>. Disponible en formato electrónico en <https://developer.arm.com/documentation/ddi0406/latest/>

Arm Limited. (2014). Instruction Set Assembly Guide for Armv7 and earlier Arm architectures Reference Guide Disponible en formato electrónico en <https://developer.arm.com/documentation/100076/latest/>

Arm Limited. (2011). Cortex-A9 Technical Reference Manual. Disponible en formato electrónico en <https://developer.arm.com/documentation/ddi0388/latest/>

Arm Limited. (2014). NEON Programmer's Guide. Disponible en formato electrónico en <https://developer.arm.com/documentation/den0018/latest/>

Arm Limited. (2020). ARM NEON Intrinsics Reference. Architecture specification. Disponible en formato electrónico en



<https://developer.arm.com/documentation/ihl0073/a/?resultof=%22%76%6d%6f%76%71%5f%6e%5f%66%33%32%22%20>

John L. Hennessy David A. Patterson (2017): Computer Organization. A Quantitative Approach. 6th. Edition. Ed Morgan Kaufman.

Bruce Jacobs, Spencer W. NG, David T Wang. (2007): Memory Systems. Cache, DRAM, Disks. Ed. Elsevier Morgan Kaufman

Daniel P. Bovet & Marco Cesati (2005): Understanding the Linux Kernel 3<sup>rd</sup>. Edition. Ed. O'Reilly.

John Madieu (2022): Linux Device Drivers Development. 2nd Edition Packt Publishing. Capítulos 2 a 9

Alessandro Rubini (2005): Linux Device Drivers 3<sup>rd</sup>. Edition. Ed. O'Reilly. Disponible en formato electrónico en <http://www.xml.com/ldd/chapter/book/bookindexpdf.html>. Capítulos 1, 2, 3, 9

Douglas Comer (2013): Internetworking with TCP/IP. Tomo I. 6th. Edition. Ed. Pearson. Capítulos 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 22, 23, 24

Douglas Comer - Richard Stevens (2000): Internetworking with TCP/IP. Volume III. Ed Prentice Hall. Capítulos 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15

### **BIBLIOGRAFÍA COMPLEMENTARIA**

William B. Giles (1991): Assembly Language Programming for the Intel 80XXX Family. Ed Macmillan Coll Div

Richard Stevens (2003): UNIX Network Programming, Volume 1, Third Edition: Networking APIs: Sockets and XTI. Ed Addison Wexley

Richard Stevens (1994): The Protocols (TCP/IP Illustrated, Volume 1). Ed. Addison Wexley