



Ministerio de Educación
Universidad Tecnológica Nacional
Rectorado



**APRUEBA CURSOS DE ACTUALIZACIÓN DE POSGRADO DEL DOCTORADO EN
INGENIERÍA, MENCIÓN COMPUTACIÓN**

Buenos Aires, 3 de septiembre de 2020

VISTO la Resolución N° 412/19 del Decano Ad-Referéndum del Consejo Directivo de la Facultad Regional Mendoza, a través de la cual solicita la aprobación y autorización de implementación de los Cursos de Actualización de Posgrado “Computación de Altas Prestaciones”, “Análisis de Software Seguro” y “Modelado y Análisis de Sistemas con Métodos Formales” para el Doctorado en Ingeniería, mención Computación, y

CONSIDERANDO:

Que el Consejo Superior autorizó por Resolución N° 1450/17 el dictado de la carrera de Doctorado en Ingeniería, mención Computación, en la Facultad Regional Mendoza.

Que los Cursos propuestos responden a la necesidad de brindar a docentes, investigadores y graduados de la Universidad conocimientos científicos actualizados dirigidos a doctorandos en Ingeniería.

Que la Facultad Regional Mendoza cuenta con un plantel de profesores de elevado nivel académico y profesional, además de una prolongada y amplia experiencia en el dictado de cursos y seminarios vinculados a los propuestos.

Que la Comisión de Posgrado de la Universidad ha analizado los antecedentes que acompañan la solicitud y avala la presentación, y la Comisión de Ciencia, Tecnología y Posgrado recomienda su aprobación.



Ministerio de Educación
Universidad Tecnológica Nacional
Rectorado

Que el dictado de la medida se efectúa en uso de las atribuciones otorgadas por el Estatuto Universitario.

Por ello,

EL CONSEJO SUPERIOR DE LA UNIVERSIDAD TECNOLÓGICA NACIONAL

ORDENA:

ARTÍCULO 1°.- Aprobar el currículum de los Cursos de Actualización de Posgrado “Computación de Altas Prestaciones”, “Análisis de Software Seguro” y “Modelado y Análisis de Sistemas con Métodos Formales”, para el Doctorado en Ingeniería, mención Computación, que figura en el Anexo I y es parte integrante de la presente Ordenanza.

ARTÍCULO 2°.- Autorizar el dictado de los mencionados Cursos en la Facultad Regional Mendoza, y avalar la propuesta del Cuerpo Docente que figura en el Anexo II y es parte integrante de la presente Ordenanza, en el marco de lo establecido por la Ordenanza N° 1313 y la Resolución N° 1450/17.

ARTÍCULO 3°.- Establecer que la propuesta mencionada en el Artículo precedente quedará supeditada al cronograma de dictado de las correspondientes actividades académicas de la Facultad Regional.

ARTÍCULO 4°.- Regístrese. Comuníquese y archívese.

ORDENANZA N° 1777

UTN
DO
f.c.r.
l.p.

ING. MIGUEL ÁNGEL SOSA
Secretario General

ING. HÉCTOR EDUARDO AIASSA
RECTOR



Ministerio de Educación
Universidad Tecnológica Nacional
Rectorado



ORDENANZA N° 1777

ANEXO I

CURSOS DE ACTUALIZACIÓN DE POSGRADO
DOCTORADO EN INGENIERÍA, MENCIÓN COMPUTACIÓN

1. COMPUTACIÓN DE ALTAS PRESTACIONES

A. FUNDAMENTACIÓN Y JUSTIFICACIÓN

La computación de altas prestaciones (High Performance Computing, HPC) es una herramienta que permite resolver numerosos problemas en diversas disciplinas de ciencia e ingeniería. Aplicaciones como la predicción meteorológica, el diseño de fármacos o la simulación de fluidos requieren de los llamados supercomputadores, sistemas informáticos con una gran potencia de cálculo, para procesar grandes cantidades de datos numéricos. La gran complejidad de estos supercomputadores dificulta cada vez más su programación eficiente e impone una creciente necesidad de expertos en este campo. En este curso se muestra el tipo de problemas que puede resolver la computación de altas prestaciones, los principios de diseño y organización de los supercomputadores, así como los modelos y herramientas de programación, depuración y análisis de prestaciones más empleados. Se presenta la arquitectura de un supercomputador, analizando los distintos niveles de paralelismo existentes (unidades funcionales, hilos hardware, núcleos, procesadores), su jerarquía de memoria y la interconexión entre los distintos recursos del sistema. También se aborda la aceleración de aplicaciones mediante el uso eficiente de los recursos computacionales y de la jerarquía de memoria. En esencia, se trata de explotar el conocimiento de la interacción hardware/software mediante una programación consciente



Ministerio de Educación
Universidad Tecnológica Nacional
Rectorado



de la arquitectura. La demanda de expertos en supercomputación es creciente tanto en el mundo de la investigación como en la industria. Sin embargo, la implementación eficiente de aplicaciones para computación de altas prestaciones es una tarea que presenta interesantes retos. Es fundamental conocer los niveles hardware y software y su interacción para explotar los recursos de un supercomputador de forma eficiente. La razón de ser de este curso es conocer el estado del arte de la computación de alta prestaciones, sus aplicaciones industriales, científicas y tecnológicas, y los problemas abiertos que existen en la actualidad. Desde el punto de vista de la formación para la investigación, este curso abre dos vías. Por una parte, introduce los desafíos que surgen al diseñar un supercomputador. Por otra parte, presenta las características y limitaciones de las actuales herramientas de programación, compilación, depuración y análisis de prestaciones de aplicaciones, sugiriendo líneas abiertas de trabajo.

B. OBJETIVOS

- Adquirir conocimientos para el diseño y programación eficiente de supercomputadores.
- Conocer las tendencias de futuro asociadas a la arquitectura y organización del supercomputador.
- Evaluar las tecnologías de programación, compilación y análisis de prestaciones que se usan en supercomputación.
- Entender sobre conceptos avanzados en arquitectura y organización de computadoras.

C. CONTENIDOS

Simulación numérica: Análisis relevante de los fenómenos físicos en la electrónica digital.

Presentación de las técnicas de aproximación numérica. Uso de simuladores numéricos.



Ministerio de Educación
Universidad Tecnológica Nacional
Rectorado



Arquitectura y Tecnología de Supercomputadores: Evolución y estado actual del paralelismo dentro del procesador. Optimización energética en las CPUs de alto rendimiento. Nuevas técnicas en la jerarquía de memoria. Segmentación y arquitectura superescalar. Tendencias actuales en los sistemas multiprocesador de memoria compartida. Coherencia, consistencia. Tendencias actuales en los sistemas multiprocesador de memoria distribuida. Nuevos desafíos en las Redes de interconexión. Arquitecturas específicas para muy alto rendimiento. Extensiones multimedia, GPGPUs. Uso de benchmarks.

Programación paralela: Estado del arte de las métricas y herramientas de análisis de rendimiento. Impacto de la vectorización en el rendimiento en las nuevas tecnologías. Caso de Estudio de máximo rendimiento en paralelización de sistemas de memoria compartida. OpenMP. Caso de Estudio de máximo rendimiento en paralelización de sistemas de memoria distribuida. MPI. Caso de Estudio de máximo rendimiento en paralelización híbrida.

D. DURACIÓN

El curso tendrá una carga horaria de CINCUENTA (50) horas.

E. METODOLOGÍA

El régimen de cursado previsto es presencial. El cursado prevé la combinación de clases teóricas - expositivas y actividades prácticas en laboratorio.

F. EVALUACIÓN

Para la aprobación del curso se requerirá, además del 80% de asistencia, la ejecución de



Ministerio de Educación
Universidad Tecnológica Nacional
Rectorado



los trabajos prácticos y la aprobación de un examen final individual.

2. ANÁLISIS DE SOFTWARE SEGURO

A. FUNDAMENTACIÓN Y JUSTIFICACIÓN

El desarrollo de software es uno de los pilares actuales en los sistemas informáticos. Cada vez más los sistemas dependen de elementos software desarrollados por ingenieros informáticos, por programadores, e incluso por equipos multidisciplinares. Sin embargo, muchos de estos desarrollos tienen defectos en su forma, tanto a nivel de diseño como a nivel de implementación. Estos defectos, que pueden derivar en vulnerabilidades, pueden llegar a explotarse comprometiendo así la seguridad del sistema. Cada vez es más frecuente la publicación de vulnerabilidades en software ampliamente utilizado, como OpenSSL (recientemente, vulnerabilidades POODLE y Heartbleed) o la consola de Unix Bash (recientemente, vulnerabilidad Shellshock). Para minimizar estos defectos potencialmente explotables en sistemas informáticos, conviene conocer las vulnerabilidades, especialmente las más complejas y frecuentes, introducidas durante el desarrollo de código, así como los mecanismos para evitarlas. En este curso, además, se analizarán los conceptos y detalles técnicos detrás de los ataques, así como posibles defensas para su evitación. Por último, se presentará la metodología para análisis de software desarrollado con intención maliciosa, con el objetivo de lograr analizar un incidente de seguridad acontecido en un sistema informático y cuantificar su impacto.

La seguridad de los sistemas informáticos es un pilar fundamental para garantizar el buen funcionamiento del propio sistema. Si este sistema se ve comprometido, puede ocasionarse pérdidas de negocio, filtración de datos sensibles de clientes, o incluso perjuicios contra otros seres humanos, si se piensa en fallos de seguridad acontecidos en



Ministerio de Educación
Universidad Tecnológica Nacional
Rectorado



centrales nucleares, presas hidráulicas, e incluso centrales de distribución eléctrica. Para evitar estas situaciones, es necesario, en primer lugar, reconocer cualquier vulnerabilidad que pueda introducirse durante la fase de implementación. A tal efecto, en este curso se detallarán los principios de programación segura para la identificación, reconocimiento, y aplicación de métodos de defensa ante las vulnerabilidades más comunes (ej., desbordamiento de búffer, format string, TOCTOU). Para además poder realizar una buena defensa, siguiendo los principios de El Arte de la Guerra, es necesario entonces conocer los fundamentos técnicos detrás de la mayoría de los ataques informáticos. En este curso se profundizará en estos ataques, entendiendo cómo y por qué se producen. Después, se describirán técnicamente las soluciones propuestas para evitar estos ataques.

Muchas organizaciones sufren ataques intencionados a sus sistemas informáticos. Estos incidentes suelen ser en forma de ejecutables orientados a la realización de alguna actividad maliciosa. Un ejemplo muy actual de este tipo de problemas es el temido “ransomware”: este malware (del inglés, malicious software) se dedica a comprometer el sistema de la víctima mediante el cifrado de todos sus ficheros, exigiendo posteriormente un pago para la liberalización de la información de la víctima. Para finalizar el curso, se introducirán los conceptos de qué es un software malicioso y cómo funciona, así como la explicación sobre cómo, dado un incidente, se puede manejar y analizar para dar respuesta al incidente de manera apropiada y en tiempo.

B. OBJETIVOS

Objetivos Generales

- Elaborar y defender argumentos y resolver problemas dentro del área de seguridad informática y de las comunicaciones.



Ministerio de Educación
Universidad Tecnológica Nacional
Rectorado



- Reunir e interpretar datos relevantes dentro del área de seguridad informática y de las comunicaciones.
- Emitir juicios sobre temas relevantes de índole social, científica o ética desde la perspectiva de la ciberseguridad.
- Transmitir soluciones al entorno industrial y empresarial en el campo de la ciberseguridad.
- Ser capaz de desarrollar proyectos de seguridad informática y de las comunicaciones.

Objetivos Específicos

- Identificar en código fuente las vulnerabilidades más comunes que pueden suceder por una mala implementación.
- Reconocer posibles escenarios de ataque a un sistema informático.
- Incorporar soluciones defensivas frente a posibles ataques a sistemas informáticos.

C. CONTENIDOS

Análisis avanzado de software: Tipos de análisis. Novedades técnicas. Tendencias. Herramientas avanzadas. Criterios de selección.

Programación segura: Vulnerabilidades y contramedidas. Estado del arte. Novedades tecnológicas. Nuevas visiones para la clasificación. Vulnerabilidades de complejidad baja, media y elevada. Análisis de código y reconocimiento de vulnerabilidades. Programación segura. Contramedidas.

Técnicas avanzadas de explotación de vulnerabilidades: Defensas. Tendencias actuales. Tipos de explotación software. Secuestro del flujo de control de ejecución. Mecanismos de defensa (a nivel hardware y software).

Técnicas de análisis de código malicioso: Tipos complejos de malware y ciclo de vida.



Ministerio de Educación
Universidad Tecnológica Nacional
Rectorado



Detección temprana y reacción. Metodologías de análisis. Construcción de entornos de laboratorio para análisis.

D. DURACIÓN

El curso tendrá una carga horaria de CINCUENTA (50) horas.

E. METODOLOGÍA

El régimen de cursado previsto es presencial. El cursado prevé la combinación de clases teóricas - expositivas y actividades prácticas en laboratorios.

F. EVALUACIÓN

Para la aprobación del curso se requerirá, además del 80% de asistencia, la ejecución de los trabajos prácticos y la aprobación de un examen final individual.

3. MODELADO Y ANÁLISIS DE SISTEMAS CON MÉTODOS FORMALES

A. FUNDAMENTACIÓN Y JUSTIFICACIÓN

El bienestar de la sociedad actual depende en gran medida de sistemas electrónicos, hardware y software, que controlan cada vez más aspectos de nuestra vida. Por un lado, el correcto funcionamiento de estos sistemas permite a la sociedad seguir desarrollándose a un ritmo superior a cualquier tiempo pasado. Por otro lado, sus fallos causan un decremento en la productividad y bienestar de la sociedad. Los fallos en sistemas críticos causan inseguridad social y pueden incluso perjudicar la salud de las personas. Garantizar un correcto funcionamiento, así como su calidad de servicio y disponibilidad, son tareas indispensables.



Ministerio de Educación
Universidad Tecnológica Nacional
Rectorado



Actualmente, los componentes electrónicos y los sistemas software están presentes de forma ubicua en cada vez más aspectos de nuestra vida: centrales eléctricas, comunicaciones, transporte o finanzas son sólo algunos ejemplos. Garantizar un correcto funcionamiento, así como su calidad de servicio y disponibilidad, es indispensable. Además, la detección de potenciales errores de diseño antes del inicio de la fase de desarrollo e implementación del producto permite ahorrar costes y esfuerzo tanto en tiempo como en dinero. Adicionalmente, el análisis de las propiedades no funcionales del sistema, como son las prestaciones y la fiabilidad, resulta imprescindible si durante el diseño queremos conocer las potencialidades del producto, por ejemplo, en cuanto a calidad de servicio (QoS) o su disponibilidad. Por tanto, el modelado y estudio de un sistema proporciona grandes ventajas a la vez que ayuda a evitar futuros fallos en producción y sus costosos remedios. En este curso veremos cómo modelar un sistema, especificar el comportamiento deseable que debe seguir mediante propiedades expresadas con formalismos lógicos y redes de Petri; y su posterior análisis cualitativo y cuantitativo mediante técnicas de verificación.

B. OBJETIVOS

Objetivo General:

Adquirir conocimientos avanzados sobre el modelado formal de sistemas (industriales o no), y su análisis cualitativo y cuantitativo.

Objetivos Específicos:

- Aprender a modelar sistemas (industriales o no) concurrentes mediante lenguajes formales: redes de Petri y lógica.
- Aprender a especificar de manera formal las propiedades deseables que deben



Ministerio de Educación
Universidad Tecnológica Nacional
Rectorado



aparecer en el modelo de nuestro sistema.

- Aprender a modelar el comportamiento del sistema para su posterior análisis cuantitativo.
- Aprender a evaluar e interpretar los resultados de la evaluación: contraejemplos en el caso cualitativo y cálculo de métricas en el caso cuantitativo (throughput, tiempo de servicio, tiempo medio de fallo, disponibilidad).
- Conocer las herramientas software existentes de verificación y evaluación de sistemas.

C. CONTENIDOS

Lógica temporal: Extensión de lógicas de primer orden con operadores modales, sintaxis y semántica de los operadores; extensiones cuantitativas y probabilistas para la lógica temporal. Clasificación y detalle de las principales lógicas temporales según sus propiedades: probabilidad, tiempo discreto vs continuo, reloj implícito vs explícito, lógica lineal vs ramificada, predicados sobre variables booleanas o reales (p.e., LTL, CTL, PCTL, CSLTa, STL, etc.). Estado del arte actual, últimos avances y aplicaciones.

Modelado con Redes de Petri: Modelado y análisis con redes de Petri; extensión de las Redes de Petri con tiempo; extensiones cuantitativas para el análisis de prestaciones: Redes de Petri estocásticas generalizadas. Árboles de fallo: Modelado y análisis con árboles de fallo; extensiones cuantitativas para el análisis de fiabilidad.

Modelado con Lenguajes Formales: Evaluación de la calidad del software: paradigma de ingeniería del software dirigida por modelos (Model-driven software engineering - MDSE). Modelado del software con el Lenguaje de Modelado Unificado (Unified Modeling Language - UML); Lenguajes de Modelado Específicos de Dominio (Domain Specific Modeling Languages - DSML); técnicas de creación de perfiles ("profiling") en UML;



Ministerio de Educación
Universidad Tecnológica Nacional
Rectorado



técnicas de transformación de modelo del software en modelo formales.

Verificación y análisis de modelos: Verificación de propiedades lógicas. Definición del modelo a analizar: sistemas basados en estados vs eventos. Sistemas de estados discretos, híbridos y continuos, sistemas estocásticos (Redes de Petri, autómatas temporizados y variantes). Técnicas verificación (model checking) según la tipología del modelo y la lógica elegida: - Model checking explícito, compilación del espacio de estados vs evaluación dinámica. - Model checking simbólico, algoritmos de punto fijo y estructuras de datos. Ordered Binary Decision Diagrams (ORBDDs). - Model checking cuantitativo, probabilístico y estadístico. Algoritmos.

Herramientas: Software de verificación y análisis cuantitativo con modelos formales (ej., GreatSPN). Herramientas avanzadas de soporte a MDSE: DICE-Simulation.

D. DURACIÓN

El curso tendrá una carga horaria de CUARENTA Y CINCO (45) horas.

E. METODOLOGÍA

El régimen de cursado previsto es presencial. El cursado prevé la combinación de clases teóricas - expositivas y actividades prácticas de laboratorio.

F. EVALUACIÓN

Para la aprobación del curso se requerirá, además del 80% de asistencia, la ejecución de los trabajos prácticos y la aprobación de un examen final individual.



Ministerio de Educación
Universidad Tecnológica Nacional
Rectorado



ORDENANZA N° 1777

ANEXO II

**CURSOS DE ACTUALIZACIÓN DE POSGRADO
DEL DOCTORADO EN INGENIERÍA, MENCIÓN COMPUTACIÓN
FACULTAD REGIONAL MENDOZA**

Cuerpo Docente

1. COMPUTACIÓN DE ALTAS PRESTACIONES

- Dr. Víctor VIÑALS YÚFERA (Pasaporte PAE819279) *Responsable*
- Dr. Jesús ALASTRUEY BENEDÉ (Pasaporte A1803641900)

2. ANÁLISIS DE SOFTWARE SEGURO

- Dr. Ricardo RODRÍGUEZ (Pasaporte PAK330219)

3. MODELADO Y ANÁLISIS DE SISTEMAS CON MÉTODOS FORMALES

- Dr. José MERSEGUER HERNAIZ (Pasaporte Y29163316) *Responsable*
- Dra. Simona BENARDI (Pasaporte X6287399-G)
- Dr. José Ignacio REQUERO (Pasaporte AAJ437947)
