



Ministerio de Cultura y Educación  
Universidad Nacional de San Luis  
Facultad de Ciencias Físico Matemáticas y Naturales  
Departamento: Informatica  
Area: Area II: Sistemas de Computacion

(Programa del año 2026)  
(Programa en trámite de aprobación)  
(Presentado el 29/05/2026 16:55:58)

### I - Oferta Académica

Materia	Carrera	Plan	Año	Período
() COMPUTACIÓN PARALELA EN GPU	LIC.CS.COMP.	RD-3 -1/20 23	2026	1° cuatrimestre

### II - Equipo Docente

Docente	Función	Cargo	Dedicación
MIRANDA, NATALIA CAROLINA	Prof. Responsable	P.Adj Exc	40 Hs
PICCOLI, MARIA FABIANA	Prof. Co-Responsable	P.Asoc Exc	40 Hs

### III - Características del Curso

Credito Horario Semanal				
Teórico/Práctico	Teóricas	Prácticas de Aula	Práct. de lab/ camp/ Resid/ PIP, etc.	Total
Hs	2 Hs	Hs	2 Hs	4 Hs

Tipificación	Periodo
B - Teoria con prácticas de aula y laboratorio	1° Cuatrimestre

Duración			
Desde	Hasta	Cantidad de Semanas	Cantidad de Horas
11/03/2026	23/06/2026	15	60

### IV - Fundamentación

El auge de las nuevas tecnologías y la constante demanda de mejores resultados en las aplicaciones hace imprescindible el estudio de nuevas técnicas y metodologías de programación. La enseñanza de la disciplina Sistemas Paralelos tiene como objetivo proveer las herramientas básicas para permitir al estudiante considerar la computación paralela en las nuevas tecnologías o tecnologías dedicadas como son las GPU. Además, le permitiría iniciarse en el campo de la investigación científica y en desarrollos propios de su área de interés.

En el Plan de Estudio de la Licenciatura/Ingeniería se introduce el tema en distintas asignaturas a través de dos tópicos fundamentales: el hardware y el software. La presente asignatura pretende establecer como lograr computación paralela y desarrollar aplicaciones paralelas eficientes y adecuadas a ejecutarse en cualquier arquitectura.

### V - Objetivos / Resultados de Aprendizaje

El procesador gráfico, GPU, surge como una alternativa válida para la computación de alto desempeño de aplicaciones de propósito general. Las GPU comienzan a ser una alternativa de bajo costo para el desarrollo de aplicaciones de muy alto rendimiento que tradicionalmente han sido exclusivas de las supercomputadoras. Así mismo, con la aparición del lenguaje CUDA, la programación de las GPU para el desarrollo de aplicaciones de propósito general se ha facilitado enormemente. Son objetivos de este curso: conocer la potencia actual del procesador gráfico y su utilización para la resolución de aplicaciones de propósito general. Familiarizarse con las formas más actuales de programación gráfica para propósito

general: CUDA y OpenCL. Comparar arquitecturas multi-core y many-core.

Durante el dictado de la asignatura se abordan los siguientes ejes transversales:

- 1- Identificación, formulación y resolución de problemas de informática.
- 2- Utilización de técnicas y herramientas de aplicación en la informática.
- 3- Concepción, diseño y desarrollo de proyectos de informática.
- 4- Fundamentos para el desempeño en equipos de trabajo.
- 5- Generación de desarrollos tecnológicos y/o innovaciones tecnológicas.
- 6- Fundamentos para la comunicación efectiva.
- 7- Fundamentos para la acción ética y responsable.
- 8- Fundamentos para el aprendizaje continuo.

Estos ejes se abordan a través de los distintos prácticos y actividades de la materia. En su resolución se propone el trabajo en equipo, la exposición y defensa de las soluciones propuestas aplicando el modelo paralelo de GPU y los beneficios que se obtienen en la performance de las distintas aplicaciones.

## **VI - Contenidos**

### **Unidad temática 1: GPU: Introducción a GPGPU**

- GPU Programación y Arquitectura
- Características de la GPU.
- Arquitectura de GPU: N-vidia, ATI, Tesla, Fermi, Kepler
- Pipeline Gráfico
- GPGPU: Computación de Propósito General en GPU.
- Paradigmas de Computación Paralela: Modelo de Memoria Compartida, Paralelismo de Datos.

### **Unidad temática 2: Programación de GPU a través de CUDA**

- Introducción a CUDA
- Modelo de programación de CUDA
- Características Básica.
- Diseño de programas en GPU
- Threading.
- Thread, Bloques, Grid.
- Sincronización de Threads
- Ejemplos.

### **Unidad temática 3: Modelo de Memoria de GPU**

- Modelo de Memoria de GPU.
- Jerarquías de Memoria: Registros, Memoria Compartida, Memoria constante, Texturas, Memoria Global.
- Estructuras de datos básicas.
- Creación de Estructuras de datos en la GPU.
- Ejemplos.

### **Unidad temática 4: Modelo GPU-CPU**

- Modelo CPU-GPU.
- Transferencia de datos GPU-CPU.
- Modelos de programación paralela en GPU: Reducciones, Map, Scan.
- Ejemplos

### **Unidad temática 5: Performance en GPU**

- Optimización de la performance.
- Colisiones de Memoria
- Accesos a Memoria Coalesced.

## Unidad temática 6: Otros Lenguajes de Programación de GPU

- Otros modelos de programación GPGPU: OpenCL

## VII - Plan de Trabajos Prácticos

Para cada unidad se deja disponible el material correspondiente a los contenidos de la unidad, las diapositivas de clase, el apunte teórico y su correspondiente trabajo práctico en el repositorio digital.

A través de distintos problemas simples, resueltos en paralelo y otros a resolver, el estudiante se aproximará en el primer caso a las facilidades provistas por la biblioteca respecto a la programación paralela; y segundo al diseño e implementación de soluciones paralelas a los problemas planteados.

Respecto a las distintas arquitecturas de GPU, se analiza cada una de las existentes, reconociendo características generales y las nuevas tecnologías que incorpora una arquitectura mas moderna con respecto a las anteriores.

En cuanto a la programación, se usa la herramienta CUDA. Los ejercicios se presentan en forma progresiva incorporando cada concepto y añadiendo mayor complejidad a medida que se avanza. Desarrollando el razonamiento crítico en el estudiante para aportar una o varias soluciones a un problema dado.

Práctico 1: Arquitecturas de GPU- Distintas Generaciones.

Objetivo: Introducir al estudiante en el conocimiento y uso de la GPU.

Comparar y analizar las nuevas características y ventajas que incorporan cada una de las arquitecturas de la GPU.

Comparación con la CPU

Temas: GPU Programación y Arquitectura. Características de la GPU. Arquitectura de GPU: N-vidia, ATI, Tesla, Fermi, Kepler. Pipeline Gráfico.

Práctico 2: Programación Básica - Memoria Global

Objetivo: Iniciar al estudiante en la programación de GPU y la resolución de problemas siguiendo el modelo SIMD y MIMD propios de la GPU.

Temas: Modelo de Programación. Grid, Bloques y Threads. Uso de Memoria Global y Local. Sincronización por barrera y exclusión.

Práctico 3: Memoria Shared y de Constante.

Objetivo: Introducir al estudiante en el uso de la jerarquía de memoria de GPU.

Temas: Memoria Global, de Textura y de Constante. Memoria Shared, Local y de Registro. Accesos y costos de acceso.

Práctico 4: Análisis de Desempeño.

Objetivo: Analizar el desempeño de las distintas soluciones propuestas. Comparación con soluciones secuenciales y paralelas en CPU.

Tema: Tiempo de ejecución y Aceleración. Análisis de Colisiones de memoria e influencia de transferencias por PCI-Express.

Proyecto Integrador

Objetivo: Integrar todos los conceptos vistos en la materia.

Temas: Programación, Jerarquía de Memorias y Análisis de Performance.

Los ejes transversales:

1- Identificación, formulación y resolución de problemas de informática. Alto

Cómo se aborda

Se entrega una serie de ejercicios con casos reales, ficticios o típicos de Ingeniería/Ciencias de la Computación. Los

problemas tienen diferentes complejidades y su desarrollo es en aula presentándose diferentes casos y situaciones, dando lugar a distintos análisis del tema. Esta nómina constituye una guía general para la presentación de varias actividades prácticas vinculadas a los diferentes módulos de aprendizaje. Estos casos se tratarán a lo largo de los diferentes módulos de la actividad curricular, con los problemas y aportes correspondientes. Los trabajos, se pueden desarrollar en grupo a lo más de dos estudiantes, a fin de lograr una propuesta de resolución consensuada.

Cómo se evalúa

Se trabaja con evaluación continua o formativa. Cada estudiante/grupo entrega por cada práctico, algunos ejercicios resueltos. El docente realiza una devolución. Posteriormente, en forma conjunta (estudiantes y docentes como moderadores), se cotejan las propuestas de solución encontradas, intentando establecer un diálogo y debate de las propuestas presentadas, donde cada par puede opinar, aportar con sugerencias o casos análogos, consultar y aceptar o refutar las soluciones. Finalmente, se le pide al estudiante la entrega de un ejercicio desarrollado, el cual es evaluado cualitativamente, observando si ha logrado incorporar el marco teórico y sus fundamentos para la resolución del problema. De esta forma, se observa el grado de apropiación de los contenidos, por cada unidad temática, a los efectos de que pueda profundizar y/o revisar los conceptos adquiridos.

## 2- Utilización de técnicas y herramientas de aplicación en la informática.

Cómo se aborda

Mediante el desarrollo e implementación de los prácticos y el proyecto final.

Para los prácticos se hace entrega de una serie de ejercicios con casos reales, ficticios o típicos de Ingeniería/Ciencias de la Computación. Los problemas tienen diferentes complejidades y su desarrollo es en Laboratorio; presentándose diferentes casos y situaciones, dando lugar a distintos Sistemas Paralelos. Los trabajos, se pueden desarrollar en grupo a lo más de dos estudiantes, a fin de lograr una propuesta de resolución consensuada.

Cómo se evalúa

Se trabaja con evaluación continua o formativa. Cada estudiante/grupo entrega por cada práctico algunos ejercicios resueltos. El docente realiza una devolución. En caso de pertinencia, se pueden cotejar las propuestas de solución encontradas, intentando establecer un diálogo y debate de las propuestas presentadas, donde cada par puede opinar, aportar con sugerencias o casos análogos, consultar y aceptar o refutar las soluciones.

En el proyecto final, mediante las soluciones encontradas, la arquitectura del sistema, funcionalidades, su desempeño y optimizaciones realizadas. También los posibles trabajos futuros que se podrían realizar.

## 3- Concepción, diseño y desarrollo de proyectos de informática.

Alto

Cómo se aborda

Mediante el desarrollo del proyecto Final. Al finalizar los prácticos, se presenta al estudiante un caso de la vida real o ficticia, sobre el cual desarrollará el proyecto completo de un sistema paralelo para ser desarrollado en la GPU. Al finalizar, el/la estudiante hace entrega de un informe del proyecto, incorporando los aspectos relacionados vistos en los prácticos. Para su implementación eligen lenguaje de programación base para las herramientas paralelas.

Cómo se evalúa

Realiza una presentación formal ante sus pares y equipo docente de su proyecto PF implementado. En esta presentación, sus pares interactúan con el presentador, mediante consultas y cuestionamientos acerca del sistema presentado. De esta forma, la interacción entre pares, la participación y exposición pública, el trabajo en equipo, las indagaciones de sus pares y la defensa fundamentada, el compartir y defender cada una de las decisiones tomadas, la preparación previa a su presentación, entre otras acciones, generan ámbitos de tensión y confianza, que contextualiza al estudiante en posibles escenarios profesionales, y que desde cualquiera de los roles que interpreta, lo interpela a sí mismo sobre sus capacidades adquiridas en la formación. La evaluación se hace para el presentador y para las/los estudiantes pares. En el caso del presentador, conlleva una evaluación cuantitativa; en el caso de los pares, una evaluación cualitativa acerca de su interés y actuación comunicacional con el presentador.

Este Proyecto Final es el resultado final de varios contenidos curriculares, la formación para la actuación profesional, la motivación al trabajo en equipo o intergrupalo, el desarrollo comunicacional oral y escrito, y alcances de su PF con otras materias del plan de estudios.

## 4- Fundamentos para el desempeño en equipos de trabajo.

Alto

Cómo se aborda

En el aula, mediante la conformación de grupos de trabajo para el desarrollo de los prácticos y proyecto final. Los equipos se conforman por a lo más dos personas, para el trabajo mancomunado de cada actividad.

Cómo se evalúa

Se observa la integración, el liderazgo, el respeto mutuo, la participación en los desarrollos, el discurso y fundamentos ante los cuestionamientos, la distribución de las tareas, la resolución de conflictos, los abordajes a los problemas, entre otros.

5- Generación de desarrollos tecnológicos y/o innovaciones tecnológicas.

Medio

Cómo se aborda

Mediante el desarrollo e implementación de los prácticos y proyecto final.

Cómo se evalúa

Se observan las soluciones encontradas en cada práctico y, principalmente en el proyecto final, mediante las soluciones encontradas, la arquitectura del sistema, funcionalidades, su desempeño y optimizaciones realizadas. También los posibles trabajos futuros que se podrían realizar.

6- Fundamentos para la comunicación efectiva.

Medio

Cómo se aborda

Mediante i) la lectura de apuntes de la actividad curricular, libros y artículos científicos; ii) entrega de informes escritos de los trabajos prácticos, iii) presentaciones orales de los prácticos. Debates y defensas orales.

Cómo se evalúa

Mediante correcciones y devoluciones correctivas las presentaciones escritas y orales.

7- Fundamentos para la acción ética y responsable.

Alto

Cómo se aborda

Al inicio de la materia, se entrega un cronograma con las actividades teóricas y prácticas, evaluaciones y recuperaciones, y otras actividades de interés (congresos, disertaciones, etc.) del cuatrimestre completo. También, mediante la entrega de trabajos prácticos individuales/grupales, la participación en equipos de trabajo, simulación de presentaciones formales, la actuación como par evaluador, entre otras actividades.

Cómo se evalúa

Observando si existe una visión crítica en torno a su ejercicio profesional, el cual pudiera comprometer el interés público poniendo en riesgo de modo directo la salud, la seguridad, los derechos, los bienes o la formación de los habitantes.

Mediante el seguimiento del cumplimiento del cronograma propuesto y de las actividades planteadas, de su relación con el usuario, del análisis de sus trabajos prácticos con respecto a la seguridad, al medio ambiente y posibles daños ante sus posibles soluciones, resguardos de la información conforme la ley, entre otros.

8- Fundamentos para el aprendizaje continuo.

Alto

Cómo se aborda

Mediante el análisis de situaciones reales, que conduzcan a la conceptualización de casos, a relacionar con contenidos previos los temas y las herramientas provistas de los temas actuales (teórico o práctico) y a otros temas disciplinares de interés.

Cómo se evalúa

Todos los prácticos conllevan una corrección informada de las actividades para que cada estudiante observe el nivel de apropiación de los contenidos.

## VIII - Regimen de Aprobación

Acerca del dictado de la Materia:

Las clases tendrán la modalidad teórico-práctica.

Acerca de la regularización de la materia:

El estudiante debe:

- 1- Tener como mínimo un 80% de asistencia a las clases teórico-prácticas.
- 2- Por cada práctico, el equipo docente exigirá la entrega de un ejercicio tipo.
- 3- Desarrollo de un trabajo práctico evaluativo consistente de:
  1. El diseño de una aplicación paralela en GPU.
  2. Confección de un informe. Dicho informe deberá: especificar los aspectos teóricos y tener un análisis detallado y justificado de las decisiones tomadas.
  3. Exposición del trabajo realizado.

Los puntos 3.1 y 3.2 tendrán tantas recuperaciones como la reglamentación vigente exige.

Acerca de la aprobación de la materia:

Por promoción: Si la nota obtenida en el trabajo práctico evaluativo obtiene una nota mayor o igual a 7 y el desempeño del estudiante durante la cursada fue satisfactorio. Además, se tomará un coloquio oral de la materia. La nota final será el promedio de todas las instancias evacuativas.

Por examen regular: Si el estudiante regularizó la materia, puede acceder a un examen regular, el cual puede ser oral o escrito y versará sobre los temas del programa.

Dadas las características de la materia y la importancia de los desarrollos prácticos, no se permiten estudiantes libres.

## **IX - Bibliografía Básica**

- [1] • Buck I. “Gpu computing with nvidia cuda”. ACM SIGGRAPH 2007 courses ACM, 2007. New York, NY, USA.
- [2] • Chen W. y Hang H. “H.264/avc motion estimation implementation on compute unified device architecture (cuda)”. In IEEE, editor, IEEE International Conference on Multimedia. 2008.
- [3] • Barlas, G. Multicore and GPU Programming An Integrated Approach. ISBN: 978-0-12-417137-4. Morgan Kaufmann. 2015.
- [4] • CUDA Corporation. CUDA C++ Programming Guide. Release 12.6. 2024.
- [5] • Guil N. y Ujaldón M. “La gpu como arquitectura emergente para supercomputación”. In XIX Jornadas de Paralelismo de Castellon. 2008.
- [6] • Kirk, D.,Hwu, W.. “Programming Massively Parallel Processors: A Hands-on Approach”. 4th Ed. ISBN: 978-0-323-91231-0. Elsevier. 2023.
- [7] • Lloyd D., Boyd C., y Govindaraju N. “Fast computation of general fourier transforms on gpus”. In IEEE International Conference on Multimedia and Expo. 2008.
- [8] • Nottingham A. y Irwin B. “Gpu packet classification using openssl: a consideration of viable classification methods”. In Research Conf. of the South African Inst. of Comp. Sc. and Inf. Technologists. ACM, 2009.
- [9] • NVIDIA. “Nvidia cuda compute unified device architecture, programming guide version 2.0”. In NVIDIA. 2008a.
- [10] • NVIDIA. “Nvidia geforce 8800 gpu architecture overview”. In NVIDIA. 2006.
- [11] • NVIDIA. Nvidia geforce gtx 200 gpu architectural overview. In NVIDIA. 2008b.
- [12] • Piccoli, María Fabiana. Computación de Alto Desempeño en GPU. Edulp. ISBN 978-950-34-0759-2. 2011.
- [13] • Ryoo S., Rodrigues C., Bagsorkhi S., Stone S., Kirk D., y Hwu W. Optimization principles and application performance evaluation of a multithreaded gpu using cuda. In ACM. ACM, 2008.
- [14] • Sanders, J., Kandrot, E.. “Cuda by Example: An Introduction to General- Purpose Gpu Programming”. ISBN: 0131387685. Addison-Wesley Professional. 2010.
- [15] • Soyata, T. GPU Parallel Program Development Using CUDA. ISBN 13: 978-1-4987-5075-2. CRC Press. 2018.
- [16] • Volkov V. “Benchmarking gpus to tune dense linear algebra”. In ACM/IEEE conference on Supercomputing. IEEE Press, Piscataway, NJ, USA, 2008.

## X - Bibliografía Complementaria

[1]

## XI - Resumen de Objetivos

Conocer la potencia actual del procesador gráfico y su utilización para la solución de aplicaciones de propósito general. Familiarizarse con las formas más actuales de programación gráfica para propósito general: CUDA y OpenCL.

## XII - Resumen del Programa

- GPU: Introducción a GPGPU
- Programación de GPU a través de CUDA
- Modelo de Memoria de GPU
- Modelo GPU-CPU
- Performance en GPU
- Otros Lenguajes de Programación de GPU

## XIII - Imprevistos

El plan de trabajo es adecuado para realizar el desarrollo por medio de un repositorio digital.

## XIV - Otros

mails de contacto: natalia.miranda@gmail.com, spgpu@gmail.com, mfpiccoli@gmail.com

### ELEVACIÓN y APROBACIÓN DE ESTE PROGRAMA

ELEVACIÓN y APROBACIÓN DE ESTE PROGRAMA	
	Profesor Responsable
Firma:	
Aclaración:	
Fecha:	