

	<b>FORMATO DE SYLLABUS</b>	Código: AA-FR-003	
	Macroproceso: Direccionamiento Estratégico	Versión: 01	
	Proceso: Autoevaluación y Acreditación	Fecha de Aprobación: 27/07/2023	

<b>FACULTAD:</b>	<b>FACULTAD DE INGENIERÍA</b>		
<b>PROYECTO CURRICULAR:</b>	Maestría en Ciencias de la Información y las Comunicaciones	<b>CÓDIGO PLAN DE ESTUDIOS:</b>	

**I. IDENTIFICACIÓN DEL ESPACIO ACADÉMICO**

**NOMBRE DEL ESPACIO ACADÉMICO: COMPUTACIÓN PARALELA**

Código del espacio académico:	79503004	Número de créditos académicos:	X			
Distribución horas de trabajo:	HTD	48	HTC	16	HTA	128
Tipo de espacio académico:	Asignatura	X	Cátedra			

**NATURALEZA DEL ESPACIO ACADÉMICO:**

Obligatorio Básico	Obligatorio Complementario	Electivo Intrínseco	X	Electivo Extrínseco	
--------------------	----------------------------	---------------------	---	---------------------	--

**CARÁCTER DEL ESPACIO ACADÉMICO:**

Teórico	Práctico	Teórico-Práctico	X	Otros:	Cuál: _____
---------	----------	------------------	---	--------	-------------

**MODALIDAD DE OFERTA DEL ESPACIO ACADÉMICO:**

Presencial	Presencial con incorporación de TIC	Virtual	Otros:	Cuál: _____
------------	-------------------------------------	---------	--------	-------------

**II. SUGERENCIAS DE SABERES Y CONOCIMIENTOS PREVIOS**

Para el buen desarrollo del curso, se considera necesario e indispensable que el estudiante tenga dominio de los temas programación en C y C++, Algebra Lineal y tener conocimiento avanzado en cálculo.

**III. JUSTIFICACIÓN DEL ESPACIO ACADÉMICO**

La aparición de nuevas arquitecturas de procesadores (ARM, Gpus, FPGAs, SoC) ha cambiado la manera como entendemos el mundo de la programación, especialmente cuando estas nuevas arquitecturas se mezclan creando las así denominadas arquitecturas heterogéneas de computación.

La programación paralela por tanto capacita al estudiante para desarrollar algoritmos eficientes en cuanto a distribución, utilización de datos y la manera como puede dividir un gran problemas en unidades pequeñas de cooperan y que pueden ser ejecutadas paralelamente para que el disminuir el tiempo total del ejecución.

Si bien los principios generales de la programación paralela pueden ser aplicados a cualquier arquitectura, las implementaciones de los mismos principios varian algunas veces de manera radical entre cada una de ellas, requiriendo conocer aspectos relevantes del hardware, del modo de transferencia y de los algoritmos y patrones que pueden ser utilizados.

Entendiendo que la programación paralela, requiere lenguajes que sean capaces de expresar los diversos niveles y formas de paralelismo, se hace necesario dominar por tanto los lenguajes y frameworks más populares al igual que los nuevos lenguajes emergentes, disponiendo así de las capacidades y habilidades necesarias para desarrollar programas eficientes que aprovechen el poder de cómputo en el ambiente multicore/manycore y aceleradores.

**IV. OBJETIVOS DEL ESPACIO ACADÉMICO (GENERAL Y ESPECÍFICOS)**

**OBJETIVO GENERAL:** Conocer los elementos teóricos y prácticos básicos para que el futuro ingeniero sea capaz de desarrollar software en arquitecturas paralelas utilizando los lenguajes de programación, frameworks y librerías más populares que son utilizadas en el ambito científico actual.

**OBJETIVOS ESPECIFICOS:**

- 1- Conocer los diferentes lenguajes utilizados en programación paralela C,C++, Fortran, Scala, Python
- 2- Dominar los principales frameworks y librerías que son utilizadas para el desarrollo de programas paralelos (OpenMP, OpenACC, MPI, CUDA, X10 y otros)
- 3- Conocer los patrones de desarrollo de programas paralelos, aplicando los lenguajes, librerías y frameworks, que son utilizados en la actualidad.
- 4- Adquirir destrezas en la configuración de los ambientes de programación para el uso de Gpus y Aceleradores vectoriales, al igual que entornos multicore.

**V. PROPÓSITOS DE FORMACIÓN Y DE APRENDIZAJE (PFA) DEL ESPACIO ACADÉMICO**

Competencias	Dominio-Nivel	RA	Resultados de Aprendizaje
Generales	Se pretende con la asignatura que el estudiante desarrolle competencias genéricas instrumentales, tales como la de resolución de problemas y la de capacidad de análisis y síntesis, entendidas como la capacidad de identificar, analizar, definir y sintetizar los elementos significativos que constituyen un problema para resolverlo con criterio y de forma efectiva.		
Específicas	Elegir un lenguaje de programación para el desarrollo de un programa paralelo de acuerdo a los objetivos y restricciones que se presenten.		
	Utilizar y configurar de manera adecuada un entorno de programación paralela, para el desarrollo de software en arquitecturas heterogéneas.		
	Adquirir destreza en utilizar herramientas que permiten hacer seguimiento, profiling y debugging a software paralelo desarrollo.		

**VI. CONTENIDOS TEMÁTICOS**

La asignatura se divide en nueve (8) unidades didácticas:

1. INTRODUCCION A LA A LA PROGRAMACIÓN EN PARALELO
2. PATRONES PARA EL DESARROLLO DE SOFTWARE PARALELO
3. PROGRAMACION MULTITHILO CON PTHREADS
4. PROGRAMACION MEMORIA COMPARTIDA USANDO OPENMP, OPENACC
5. PROGRAMACION MEMORIA DISTRIBUIDA CON MPI
6. PROGRAMACION SIMD/SIMP con CUDA, OpenCL, CilkPlus, Chapel
7. PROGRAMACION VECTORIAL
8. APLICACIONES PRACTICAS

**VII. ESTRATEGIAS DE ENSEÑANZA QUE FAVORECEN EL APRENDIZAJE**

Tradicional		Basado en Proyectos		Basado en Tecnología	
Basado en Problemas		Colaborativo		Experimental	
Aprendizaje Activo		Autodirigido		Centrado en el estudiante	

**VIII. EVALUACIÓN**

Resultados de aprendizaje (RA) a ser evaluados:	Resultados de aprendizaje asociados a las evaluaciones					
	Actividades Entregables	Talleres	Parciales	Informes de proyecto final	Proyecto final	Exposiciones
RA01						

RA02						
RA03						
RA04						
RA05						
RA06						
RA07						
RA08						
RA09						
Tipo de evaluación**						
Porcentaje de evaluación (%)						
Trabajo Individual (I) o Grupal (G)						
Tipo de nota	0-5	0-5	0-5	0-5	0-5	0-5

**IX. MEDIOS Y RECURSOS EDUCATIVOS**

Medios y Ayudas: Espacio físico (aula); Recurso docente; Recursos bibliográficos (especializados); recursos tecnológicos (equipos)

**X. PRÁCTICAS ACADÉMICAS - SALIDAS DE CAMPO**

**XI. BIBLIOGRAFÍA**

**Básicas:**

ANON, High Performance Computing,  
DONGARRA, J., FOSTER, I., FOX, G. AND GROPP, W., 2003. Sourcebook of parallel computing,  
GROUP, F., 2008. Handbook of Parallel Computing,  
OLUKOTUN, K., HAMMOND, L. AND LAUDON, J., 2007. Chip Multiprocessor Architecture: Techniques to Improve Throughput and Latency,  
PASSING, M. AND FORUM, I., 2003. MPI-2 : Extensions to the Message-Passing Interface. , (21111).  
RAUBER, T. AND RÜNGER, G., 2010. Parallel programming: For multicore and cluster systems,  
YANG, L.T. AND GUO, M., 2005. High-Performance Computing L. T. Yang & M. Guo, eds., Hoboken, NJ, USA: John Wiley & Sons, Inc.  
GREGORY, K. AND MILLER, A., 2012. C++ AMP: Accelerated Massive Parallelism with Microsoft® Visual C++®.

**Complementarias:**

CHANDRA, R. (2001). Parallel programming in OpenMP. San Francisco, Calif, Morgan Kaufmann Publishers. <http://www.books24x7.com/marc.asp?bookid=7063>.  
GROPP, W., LUSK, E., & SKJELLUM, A. (1999). Using MPI portable parallel programming with the message-passing interface. Cambridge, Mass, MIT Press.  
<http://search.ebscohost.com/login.aspx?direct=true&scope=site&db=nlebk&db=nlabk&AN=18119>.  
Wagener, Jerrold L. "High Performance Fortran." Computer Standards and Interfaces 18 (1996): 371–377. Web.  
Saraswat, V A, V Sarkar, and C Von Praun. "X10: Concurrent Programming for Modern Architectures." Proceedings of the ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming, PPOPP. N.p., 2007. 271. Web.  
Odersky, Martin, Lex Spoon, and Bill Venners. "Programming in Scala." Theoretical Computer Science 410 (2008): 202–220. Web.

Cook, Shane. CUDA Programming. N.p., 2013. Web.

Nickolls, John et al. "Scalable Parallel Programming with CUDA." Queue 2008: 40. Web.

Nvidia Corporation. "OpenCL Programming Guide for the CUDA Architecture." NVIDIA 41 (2010): 371–379. Web.

**Revistas:**

<http://www.journals.elsevier.com/parallel-computing/> Parallel Computing, Systems & Applications

<http://www.springer.com/computer/theoretical+computer+science/journal/10766> International Journal of Parallel Programming

<http://hpc.sagepub.com/> International Journal of High Performance Computing Applications

Páginas web

Bases de datos: [www.elsevier.com](http://www.elsevier.com) -- [www.sciencedirect.com](http://www.sciencedirect.com) -- [www.ieeeexplore.ieee.org](http://www.ieeeexplore.ieee.org)

**XII. SEGUIMIENTO Y ACTUALIZACIÓN DEL SYLLABUS**

Fecha revisión por Consejo Curricular:

Fecha aprobación por Consejo Curricular:

Número de  
acta:

<b>**Tipo de Evaluación</b>	<b>Abreviatura</b>
1. Evaluación de habilidad	EHP
2. Evaluación basada en p	EBP
3. Evaluación oral o prese	EOP
4. Evaluación escrita	EE
5. Evaluación formativa	EF
6. Evaluación de desempe	ED