

Programación Científica I

IDENTIFICACIÓN DE LA ACTIVIDAD CURRICULAR

Nombre de la Actividad	:	COMPUTACIÓN CIENTÍFICA I					
Código	:	DIN-114					
Semestre lectivo	:	I					
Horas	:	Presencial:	72	Autónomas:	108	Total:	180
Créditos SCT	:	6					
Duración	:	Semestral:	X	Anual:			
Modalidad	:	Presencial:	X	Semi-presencial:			
Área de Formación		GENERAL					
Requisito		INGRESO					

DESCRIPCIÓN Y CARACTERIZACIÓN DE LA ACTIVIDAD CURRICULAR

Esta actividad se orienta a que el estudiante adquiera y utilice con soltura un lenguaje de programación de alto nivel para llevar a cabo la adecuación o el desarrollo de nuevos algoritmos aplicados a problemas científicos, por ejemplo, programando computacionalmente un problema matemático, simularlo numéricamente y representar los resultados en forma adecuada.

COMPETENCIAS DEL PERFIL DE GRADO ASOCIADAS A LA ACTIVIDAD CURRICULAR.

COMPETENCIA ESPECÍFICA	SUBCOMPETENCIA
2. Proponer sistemas tecnológicos o soluciones innovadoras, basados en la toma de datos en terreno, la experimentación y/o la modelación para la formulación; desde la Ingeniería, de soluciones eficientes y sostenibles.	2.1 Diseñar soluciones tecnológicas eficientes y sostenibles basadas en el análisis del estado del arte de un problema.

COMPETENCIAS GENÉRICAS	SUBCOMPETENCIA
5. Realizar investigaciones que contribuyan al desarrollo del conocimiento científico y aplicado, en el contexto propio de su proceso formativo	5.1 Analizar resultados de investigación en Ingeniería actualizados relacionados con las líneas de Sistemas Inteligentes y Reconocimiento de Patrones, Sistemas Silvoagrícolas Digitales, o Bioingeniería en Ambientes Naturales y Modificados, con el fin de construir el estado del arte de un problema científico y tecnológico.

RESULTADOS DE APRENDIZAJE.

RESULTADOS DE APRENDIZAJES (R.A.)
1. Conocer la estructura básica de un programa que resuelva un problema matemático sobre una plataforma de hardware y software orientado a cómputo científico.
2. Implementar programas para resolver problemas de cómputo científico mediante el uso de un lenguaje de programación.
3. Evaluar la eficiencia de soluciones algorítmicas involucradas en problemas de cómputo científico.

UNIDADES DE APRENDIZAJE Y EJES TEMÁTICOS

R.A.	UNIDAD	EJE(S) TEMÁTICO(S)
1, 2	Estructura de Programas Básicos y su implementación.	Introducción a la Programación y Sistemas Operativos basados en Unix (aspectos fundamentales del entorno de programación, conexión a S.O. remotos, comandos básicos). Programación Estructurada (bibliotecas, identificadores, palabras claves, operadores, declaración de variables globales y locales, asignación, inicialización, entrada y salida estándar, tipos de datos, estructuras). Estructuras de control y ciclos iterativos (expresiones lógicas, estructuras condicionales, estructuras iterativas). Funciones.
2	Manejo de Arreglos y Matrices n-dimensionales.	Arreglos y matrices n-dimensionales. Manipulación de arreglos y matrices en funciones.
2, 3	Análisis de Algoritmos y Ordenamiento	Características de un algoritmo (tiempo de ejecución, consumo de memoria, almacenamiento). Complejidad algorítmica. Introducción al Ordenamiento y Búsqueda de datos (Bubble Sort, Merge Sort, Insertion Sort, Quicksort, búsqueda exhaustiva).

ESTRATEGIAS DE ENSEÑANZA Y APRENDIZAJE

La metodología de trabajo para el desarrollo de la actividad curricular, se basa en un enfoque de resolución de ejercicios y problemas en el aula; esto implica entregar un rol protagónico al estudiante que es entendido como eje y centro de acción, y quién a través de su participación activa y con orientaciones y lineamientos que le entrega el docente va construyendo su propio aprendizaje. Para lograr este objetivo, las distintas clases consideran una serie de estrategias metodológicas, previamente seleccionadas por el docente:
--

- Método expositivo-participativo.
- Aprendizaje en base a resolución de ejercicios y problemas en el aula.
- Aprendizaje en base a resolución de ejercicios y problemas con uso software en laboratorio computación.

PROCEDIMIENTOS DE EVALUACION DE APRENDIZAJES.

Resultado de Aprendizajes	Indicadores de Evaluación	Estrategia de Enseñanza y Aprendizaje	Técnica/ Instrumento Evaluativo	Ponderación (%)
1.- Conocer la estructura básica de un programa que resuelva un problema matemático sobre una plataforma de hardware y software orientado a cómputo científico.	<p>Uso de un Sistema Operativo orientado a cómputo científico.</p> <p>Correcta implementación de un programa estructurado.</p> <p>Uso adecuado de estructuras de control iterativas y no iterativas.</p> <p>Implementar programas modulares en base a funciones.</p>	<p>Clase expositiva.</p> <p>Resolución de problemas de programación en clase.</p> <p>Talleres de Laboratorio.</p>	<p>Prueba escrita / Pauta.</p> <p>Tarea de Laboratorio / Pauta.</p>	35%
2. Implementar programas para resolver problemas de cómputo científico mediante el uso de un lenguaje de programación.	<p>Implementa adecuadamente arreglos y matrices n-dimensionales.</p> <p>Divide un problema en subproblemas.</p> <p>Reconoce cuando utilizar un arreglo o una matriz dependiendo del problema a resolver.</p>	<p>Clase expositiva.</p> <p>Resolución de problemas de programación en clase.</p> <p>Talleres de Laboratorio.</p>	<p>Prueba escrita / Pauta.</p> <p>Tarea de Laboratorio / Pauta.</p> <p>Exposición de la tarea / Pauta.</p>	40%
3. Evaluar la eficiencia de soluciones algorítmicas involucradas en problemas de cómputo científico.	<p>Estima la eficiencia de un algoritmo.</p> <p>Obtiene mediciones de rendimiento de un algoritmo.</p> <p>Utiliza método de ordenamiento</p>	<p>Clase expositiva.</p> <p>Resolución de problemas de programación en clase.</p> <p>Talleres de Laboratorio.</p>	<p>Estudio de casos / Pauta.</p> <p>Tarea de Laboratorio / Pauta.</p>	25%

	adecuado a la naturaleza del problema.		Exposición de la tarea / Pauta.	
--	--	--	---------------------------------	--

RECURSOS DE INFRAESTRUCTURA

Sala de clases, Servidores de cómputo LITRP, Proyector, Telones, Pizarras amplias, Compiladores e Intérpretes, Biblioteca, Sistemas LMS-UCM.

RECURSOS BIBLIOGRÁFICOS

	Autor, año; Título del trabajo, año edición, lugar de publicación, editorial. (De acuerdo a tipo de material consultado y de acuerdo a la normativa APA 6° Edición)
BÁSICA OBLIGATORIA	<ul style="list-style-type: none"> Joakim Sundnes (2020): Introduction to Scientific Programming with Python. Simula SpringerBriefs on Computing. https://doi.org/10.1007/978-3-030-50356-7 Chazallet, Sébastien (2016): Python 3 : Los fundamentos del lenguaje. Barcelona : Ediciones ENI. Luis Joyanes Aguilar (2017): FUNDAMENTOS DE PROGRAMACIÓN. Algoritmos, estructura de datos y objetos. Cuarta edición. McGraw Hill. Giancarlo Zaccone, Python Parallel Programming Cookbook, Packt 2019. Qingkai Kong, Timmy Siau, Alexandre Bayen. Python Programming And Numerical Methods: A Guide For Engineers And Scientists. Academic Press, 2020.
COMPLEMENTARIA	<ul style="list-style-type: none"> Joel Grus, Data Science from Scratch, First Principles with Python, O'Reilly, 2015. Jake VanderPlas, Python Data Science Handbook Essential Tools for Working with Data, O'Reilly, 2016. Sam Morley. Applying Math with Python. Packt, 2020.

OTROS RECURSOS

Nombre Recurso	Tipo de Recurso
Libros y revistas	Digital
Sitios Web	Digital

Sitio Web: Documentación Oficial de Python.	Digital, URL: https://docs.python.org
Sitio Web: Python Programming and Numerical Methods.	Digital, URL: https://pythonnumericalmethods.berkeley.edu/notebooks/Index.html
Repositorio de Código.	Digital, URL: https://github.com/PacktPublishing/Applying-Math-with-Python