

	Contenidos Programáticos Programas de Pregrado	Código	FGA-23 v.03
		Página	1 de 5

FACULTAD: Ciencias Básicas
PROGRAMA: Física
DEPARTAMENTO DE: Física y Geología

CURSO:	Física Computacional III	CÓDIGO:	157420
ÁREA:	Profundización		
REQUISITOS:	R-157219	CORREQUISITO:	
CRÉDITOS:	3	TIPO DE CURSO:	Teórico - Práctico
FECHA ÚLTIMA ACTUALIZACIÓN	JULIO 2020		

JUSTIFICACIÓN

Las herramientas numéricas y computacionales se han convertido en elementos básicos para el quehacer y desarrollo de la Física actual. En este sentido es imperioso que nuestras y nuestros egresados desarrollen competencias que les permitan desarrollar códigos y simulaciones computacionales que den respuesta a problemas físicos.

Ciertos fenómenos de la Física requieren de resolver ecuaciones que pueden resultar en soluciones analíticas muy complejas de resolver y/o modelar; por ejemplo, en electromagnetismo, difusión, etc. Por otra parte, existen fenómenos de la naturaleza que tienen un carácter aleatorio: decaimientos atómicos, movimiento Browniano, fondo de rayos cósmicos, etc. En este sentido es importante que nuestros y nuestras estudiantes desarrollen competencias que les permitan solucionar tanto este tipo de ecuaciones, como interacciones físicas aleatorias; sin olvidar la importancia de modelar, manipular, analizar e interpretar los datos que obtienen vía estas herramientas.

Un aspecto clave en el desarrollo de las competencias computacionales es el dominio de un lenguaje de programación; sin este no es posible pasar del algoritmo a la resolución de problemas. Por su estilo sencillo y práctico, el lenguaje de programación Python se ha convertido en una herramienta útil, capaz y principalmente de rápido aprendizaje. Estas características permiten que las y los estudiantes dediquen más tiempo a entender la Física detrás de sus códigos y a analizar de manera crítica los resultados que obtienen; competencias fundamentales en su desarrollo profesional.

OBJETIVO GENERAL

Aplicar soluciones numéricas para resolver problemas físicos que involucren sistemas de ecuaciones diferenciales parciales, procesos aleatorios y nociones básicas de análisis de datos; usando como lenguaje de programación Python.

	Contenidos Programáticos Programas de Pregrado	Código	FGA-23 v.03
		Página	2 de 5

OBJETIVOS ESPECÍFICOS

<ul style="list-style-type: none"> ✓ Aplicar elementos de programación en Python. ✓ Aplicar conceptos básicos de Git como herramienta para el desarrollo y manejo de versiones de códigos computacionales; vía la plataforma Web GitHub. ✓ Aprender a solucionar numéricamente ecuaciones diferenciales parciales. ✓ Aprender a solucionar numéricamente problemas que involucren generación de números y procesos. ✓ Aprender nociones básicas de análisis de datos.
--

COMPETENCIAS

<ul style="list-style-type: none"> ✓ Implementar algoritmos numéricos en el lenguaje de programación Python. ✓ Desarrollar códigos usando el control de versiones Git. ✓ Escribir códigos en Python que permitan solucionar ecuaciones diferenciales parciales. ✓ Escribir códigos en Python que permitan solucionar problemas que involucren generación de números y procesos. ✓ Representar datos a través de histogramas y asociarlos con distribuciones conocidas como Normal, Poisson y Gauss. ✓ Estimar probabilidades y niveles de confianza a partir de un conjunto de datos. ✓ Interpretar y analizar los resultados obtenidos a través de códigos computacionales.

Unidad 1: Soluciones numéricas de ecuaciones diferenciales parciales

TEMA	HORAS DE CONTACTO DIRECTO	HORAS DE TRABAJO INDEPENDIENTE DEL ESTUDIANTE
Ecuaciones diferenciales parciales I ✓ Introducción ✓ Problemas de valor inicial	7	5
Ecuaciones diferenciales parciales II ✓ Método FTCS ✓ Estabilidad numérica	7	5
Ecuaciones diferenciales parciales III ✓ Método implícito y de Crank-Nicolson ✓ Métodos espectrales	6	5

Unidad 2: Procesos aleatorios

TEMA	HORAS DE CONTACTO DIRECTO	HORAS DE TRABAJO INDEPENDIENTE DEL ESTUDIANTE
Procesos aleatorios ✓ Números aleatorios ✓ Generadores y semillas de números aleatorios	6	5

	Contenidos Programáticos Programas de Pregrado	Código	FGA-23 v.03
		Página	3 de 5

<ul style="list-style-type: none"> ✓ Distribuciones a partir de números aleatorios 		
Integración por método Monte Carlo <ul style="list-style-type: none"> ✓ Método del valor medio ✓ Integrales en varias dimensiones 	6	5
Simulaciones Monte Carlo <ul style="list-style-type: none"> ✓ Aplicaciones a la mecánica estadística ✓ Método de la cadena Markov 	6	5

Unidad 3: Nociones de análisis de datos

TEMA	HORAS DE CONTACTO DIRECTO	HORAS DE TRABAJO INDEPENDIENTE DEL ESTUDIANTE
Datos <ul style="list-style-type: none"> ✓ Tipos de datos ✓ Histogramas ✓ Valores medios 	6	5
Dispersión de datos <ul style="list-style-type: none"> ✓ Variancia ✓ Desviación estándar ✓ Medidas alternativas para medir la dispersión 	6	5
Momentos de una distribución <ul style="list-style-type: none"> ✓ <i>Skew</i> ✓ Momentos de orden superior ✓ Covariancia ✓ Correlación 	6	5
Propiedades generales de una distribución <ul style="list-style-type: none"> ✓ Ley de grandes números ✓ Expectación de valores ✓ Distribuciones de densidad de probabilidad 	6	5
Distribuciones <ul style="list-style-type: none"> ✓ Binomial ✓ Poisson ✓ Gaussiana 	6	5
Probabilidad <ul style="list-style-type: none"> ✓ Probabilidad matemática ✓ Límite de una frecuencia (empírico) ✓ Prospección objetiva ✓ Estadística Bayesiana 	6	5
Niveles de confianza <ul style="list-style-type: none"> ✓ Niveles de confianza en estadística descriptiva ✓ Intervalos de confianza en estimaciones ✓ Intervalos de confianza de Gaussianas 	6	4

	Contenidos Programáticos Programas de Pregrado	Código	FGA-23 v.03
		Página	4 de 5

METODOLOGÍA

Se implementarán herramientas basadas en tecnologías de la información y la comunicación para el desarrollo de las clases. Para esto se habilitará todo el material del curso (lecturas, videos, etc) en las plataformas tipo MOOC dispuestas por la universidad para este fin. Como herramienta adicional, se hará uso de las plataformas web y *open-access* Github y Github-classroom.

Durante la dinámica del curso se destinará una hora semanal al desarrollo de un entorno SOLE (Self Organized Learning Environment), donde las y los estudiantes, divididos en grupos, desarrollarán actividades propuestas por ellos mismos o por el docente, relacionada con los temas vistos en la semana:

- ✓ 5 minutos para el planteo de la pregunta o actividad a realizar (docente o estudiante).
- ✓ 35 minutos para el desarrollo del tema planteado (grupos de estudiantes).
- ✓ 15 minutos para la exposición de las conclusiones de cada grupo (representante de cada grupo).
- ✓ 5 minutos para desarrollar las conclusiones y el sumario (docente).

Se invitarán expertos para que realicen una charla donde expongan los últimos desarrollos relacionados las temáticas del curso. De esta manera, los y las estudiantes tendrán una visión más amplia actualizada del estado del arte de los contenidos aquí expuestos.

SISTEMA DE EVALUACIÓN

Se realizarán 3 evaluaciones, según el calendario académico, las cuales corresponden al 60% de la nota definitiva, más las actividades propuestas por el profesor (quices, trabajos, etc) correspondiente al 40% de la nota definitiva restante.

- ✓ Primera evaluación del 35%: 20% corresponderá a un proyecto a entregar durante la respectiva semana de examen; y 15% otras actividades tales como talleres y quices.
- ✓ Segunda evaluación del 35%: 20% corresponderá a un proyecto a entregar durante la respectiva semana de examen; y 15% otras actividades tales como talleres y quices.
- ✓ Tercera evaluación del 30%: 20% corresponderá a un proyecto a entregar durante la respectiva semana de examen; y 10% otras actividades tales como talleres y quices.

BIBLIOGRAFÍA DISPONIBLE EN UNIDAD DE RECURSOS BIBLIOGRÁFICOS DE LA UNIVERSIDAD DE PAMPLONA

Chapra, Steven, C. Métodos numéricos para ingenieros. Mc Graw-Hill, 2007.

	Contenidos Programáticos Programas de Pregrado	Código	FGA-23 v.03
		Página	5 de 5

BIBLIOGRAFÍA COMPLEMENTARIA

- ✓ Newman, Mark. Computational Physics. Univeristy of Michigan, 2013.
- ✓ Landau, Rubin H. Computational Problems for Physics. CRC Press, 2018.
- ✓ Sherer, Philipp O. J., Computational Physics, Simulation of Classical and Quantum Systems. Springer, 2010.

DIRECCIONES ELECTRÓNICAS DE APOYO AL CURSO

- ✓ <http://www.umich.edu/~mejn/cp>
- ✓ <http://sites.science.oregonstate.edu/~landaur/Books/CPbook/eBook/Lectures/>
- ✓ <https://www.python.org/>
- ✓ <https://git-scm.com/book/es/v2>

NOTA: EN CADA UNA DE LAS UNIDADES EL DOCENTE DEBERA PROPONER MÍNIMO UNA LECTURA EN LENGUA INGLESA Y SU MECANISMO DE CONTROL