

	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 II	CÓDIGO:	157219
ÁREA:	Profundización		
REQUISITOS:	R-157218	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 relacionadas al pensamiento computacional, esto es que les permitan desarrollar códigos y simulaciones computacionales que den respuesta a problemas físicos.

Sin excepción, los retos y fenómenos a los que se enfrenta la Física requieren de resolver una o un sistema de ecuaciones, lineales, no lineales, o diferenciales. Por esta razón es imperioso que nuestras y nuestros egresados desarrollen competencias que les permitan, no solo plantear soluciones en términos de algunas de estas ecuaciones, sino además resolverlas y modelarlas. Es en este escenario donde las herramientas y el pensamiento computacional son claves en su desarrollo profesional, lo que se traduce en egresados y egresadas competentes para el ejercicio y aplicación de su profesión.

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 lineales y no lineales, uso de la transformada de Fourier y ecuaciones diferenciales de primer orden; 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 sistemas de ecuaciones lineales. ✓ Aprender a solucionar numéricamente sistemas de ecuaciones no lineales. ✓ Aprender a solucionar numéricamente problemas que involucren la transformada de Fourier. ✓ Aprender a solucionar numéricamente ecuaciones diferenciales de primer orden.
--

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 problemas de la Física que involucren sistemas de ecuaciones lineales. ✓ Escribir códigos en Python que permitan solucionar problemas de la Física que involucren sistemas de ecuaciones no lineales. ✓ Escribir códigos en Python que permitan que permitan aplicar la transformada de Fourier. ✓ Escribir códigos en Python que permitan solucionar ecuaciones diferenciales de primer orden. ✓ Interpretar y analizar los resultados obtenidos a través de códigos computacionales.

Unidad 1: Soluciones numéricas de ecuaciones lineales

TEMA	HORAS DE CONTACTO DIRECTO	HORAS DE TRABAJO INDEPENDIENTE DEL ESTUDIANTE
Ecuaciones lineales simultaneas I ✓ Eliminación de Gauss ✓ Sustitución inversa ✓ Pivote	7	4
Ecuaciones lineales simultaneas II ✓ Descomposición LU ✓ Inversa de una matriz ✓ Matrices tridiagonales y de banda	7	5
Ecuaciones lineales simultaneas III ✓ Eingevalores y eingevectores	6	5

Unidad 2: Soluciones numéricas de ecuaciones no lineales

TEMA	HORAS DE CONTACTO DIRECTO	HORAS DE TRABAJO INDEPENDIENTE DEL ESTUDIANTE
✓ Ecuaciones no lineales I ✓ Método de relajación ✓ Convergencia del método de relajación	6	5

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

Ecuaciones no lineales II <ul style="list-style-type: none"> ✓ Método de relajación de dos o más variables ✓ Búsqueda binaria 	6	5
Ecuaciones no lineales III <ul style="list-style-type: none"> ✓ Método de Newton ✓ Método de la secante ✓ Método de Newton de dos o más variables 	6	5

Unidad 3: Máximo y mínimos de funciones

TEMA	HORAS DE CONTACTO DIRECTO	HORAS DE TRABAJO INDEPENDIENTE DEL ESTUDIANTE
Máximo y mínimos de funciones <ul style="list-style-type: none"> ✓ Búsqueda de la razón dorada ✓ Método de Gauss-Newton y gradiente descendente 	6	5

Unidad 4: Transformada de Fourier

TEMA	HORAS DE CONTACTO DIRECTO	HORAS DE TRABAJO INDEPENDIENTE DEL ESTUDIANTE
Transformada de Fourier I <ul style="list-style-type: none"> ✓ Series de Fourier ✓ Transformada discreta de Fourier 	6	5
Transformada de Fourier II <ul style="list-style-type: none"> ✓ Transformada de funciones seno y coseno ✓ Transformada rápida de Fourier 	6	5

Unidad 5: Soluciones numéricas de ecuaciones diferenciales

TEMA	HORAS DE CONTACTO DIRECTO	HORAS DE TRABAJO INDEPENDIENTE DEL ESTUDIANTE
Ecuaciones diferenciales de primer orden I <ul style="list-style-type: none"> ✓ Método de Euler ✓ Método de Runge-Kutta ✓ Solución sobre intervalos infinitos ✓ Solución por librería <i>Odeint</i> de Python 	6	5
Ecuaciones diferenciales de primer orden II <ul style="list-style-type: none"> ✓ Solución con más de una variable ✓ Ecuaciones diferenciales de segundo orden ✓ Solución con paso variable 	6	5

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

Valores de frontera ✓ Método de <i>shooting</i> ✓ Método de relajación ✓ Problemas de Eingevalores	6	5
Ecuaciones diferenciales parciales ✓ Problemas de valor de frontera y método de relajación ✓ Métodos rápidos para problemas de valor defrontera	6	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 EVALUACION

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.

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

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

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

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