

PROGRAMA DE LA ASIGNATURA		
1. Nombre de la actividad curricular Computación y Programación (FCN460)		
2. Nombre de la actividad curricular en inglés Computing and Programming		
3. Unidad Académica: Departamento de Física, Facultad de Ciencias Profesor Coordinador: José Rogan C. Profesores Colaboradores: No hay		
4. Ámbito: Eje disciplinar, Eje instrumental, Eje actitudinal Nivel: Cuarto semestre Carácter: Obligatorio Modalidad: Presencial Requisitos: Cálculo II y Álgebra y Geometría II		
4. Horas de trabajo Coordinador: Colaboradores:	presencial (directas) 4.5 h	no presencial (indirectas) 9.0 h
5. Tipo de créditos SCT	Horas semanales: Cátedra 3 h Taller 1.5 h Ayudantía 1.5 h	Horas semanales: 6 h
5. Número de créditos SCT – Chile 8		
6. Requisitos	Álgebra y Geometría II Cálculo II	

7. Propósito general del curso	Conocer y trabajar con herramientas básicas y avanzadas para el uso avanzado y administración básica de sistemas operativos tipo Unix, manejando a la vez elementos básicos de programación en un lenguaje interpretado (Python).
8. Competencias a las que contribuye el curso	D1 Domina los fundamentos de la disciplina y sus métodos de investigación, con el fin de comprender los modelos validados teórica y experimentalmente para describir nuestro entorno. D2 Manifiesta dominio del lenguaje matemático y del lenguaje técnico propio de la física, que permiten expresar el conocimiento científico en una forma universalmente comprensible para la disciplina. D3 Utiliza técnicas analíticas, experimentales o computacionales, para validar y desarrollar modelos físicos del entorno. D4 Utiliza adecuadamente los modelos existentes para la descripción de los fenómenos naturales, comprendiendo los límites de aplicabilidad de cada modelo disponible e interpretando adecuadamente el alcance de sus predicciones. I1 Demuestra gran capacidad de abstracción, análisis y pensamiento lógico. I2 Comunica, de manera escrita y oral, conocimientos y resultados relacionados con la disciplina. A4 Manifiesta un gran compromiso ético, comprendiendo que un alto estándar en

	este sentido es imprescindible para la construcción del conocimiento científico.
9. Subcompetencias	D1.4 Domina fundamentos disciplinares relativos a la formulación matemática de modelos físicos. D2.1 Domina el lenguaje matemático requerido para el ejercicio disciplinar. D3.3 Utiliza técnicas computacionales requeridas por los modelos físicos para describir el entorno. D4.4 Maneja apropiadamente la formulación matemática de los modelos físicos, en consistencia con los supuestos y aproximaciones de dichos modelos. I1.1 Demuestra gran capacidad de abstracción. I1.2 Demuestra gran capacidad de análisis. I1.3 Demuestra gran capacidad de pensamiento lógico. I2.1 Comunica adecuadamente conocimientos y resultados disciplinares, por medios escritos.
10. Resultados de Aprendizaje 1. Operar a nivel de usuario con herramientas básicas y avanzadas para realizar computación científica en computadores con sistema operativo tipo Unix. 2. Crear documentos científicos tales como tareas, informes, presentaciones, pósters y eventualmente publicaciones, que incluyan tablas, diagramas, imágenes y expresiones matemáticas complejas usando el sistema de preparación de documentos LaTeX. 3. Desarrollar y programar soluciones de problemas, tanto físicos como matemáticos, de forma algorítmica en el lenguaje de programación interpretado Python bajo diferentes paradigmas: funcional e imperativo estructurado, procedimental y orientado al objeto.	

11. Saberes / contenidos

1. Elementos de un sistema operativo tipo unix
 1. Introducción
 2. Algo de historia sobre Linux
 3. El proyecto Debian
 4. Ingresando al sistema
 5. Estructura de directorios
 6. Los archivos y directorios
 7. Los comandos básicos
 8. Permisos y propiedad de archivos
 9. Utilitarios
 10. Ayuda y documentación
 11. Shells
 12. Editores
 13. Filtros
 14. Procesos
 15. Internet
 16. Sistema X-windows
 17. Compresión
2. Gráfica en Linux
 1. Visualización de archivos gráficos
 2. Modificando imágenes
 3. Conversión entre formatos gráficos
 4. Captura de pantalla
 5. Creando imágenes
 6. Graficando funciones y datos
3. El sistema de preparación de documentos LaTeX

1. Introducción a TeX y LaTeX
 2. Archivos y proceso de compilación
 3. LaTeX básico, caracteres especiales, comandos y ambientes
 4. Estilos y paquetes, LaTeX en castellano
 5. Matemáticas en LaTeX
 6. LaTeX avanzado, clases de documento, paquetes adicionales
 7. Definición y redefiniendo comandos
 8. Alineación, listas, tablas, figuras, minipages y referencias cruzadas
 9. BiBTeX
 10. Errores y advertencias
4. Introducción a programación
 1. Conceptos básicos
 2. Lenguajes de programación
 3. Lenguajes naturales y formales
 4. Desarrollando programas
 5. La interfaz con el usuario
 6. Sacar los errores de un programa
5. Una breve introducción a Python
 1. Introducción a Python
 2. El primer programa
 3. Variables, entrada y salida
 4. Tipos numéricos
 5. Secuencias inmutables
 6. Secuencias mutables
 7. Tipos adicionales
 8. Control de flujo, bifurcaciones
 9. Control de flujo, iteraciones

- 10. Funciones
- 11. Funciones de módulos
- 12. Escribiendo funciones
- 13. Programación orientada al objeto
- 14. Programación funcional
- 15. Módulos
- 16. Archivos
- 17. Excepciones

12. Metodología

El curso consta de dos clases presenciales expositivas semanales más una clase de taller que incluye desarrollo de ejemplos que ilustran en los conceptos entregados en la cátedra y resolución de problemas.

Además de sesiones de práctica y resolución colaborativa de ejercicios (ayudantías: una por semana dividida en tres secciones).

13. Evaluación

Se controlan los resultados de los aprendizajes a través de dos pruebas, una tarea y un proyecto final:

Primera prueba (P_1) sobre la unidad uno. Resultado de Aprendizaje 1.

Tarea (T) sobre las unidades dos y tres. Resultado de Aprendizaje 2.

Segunda prueba (P_2) sobre las unidades cuatro y cinco. Resultado de Aprendizaje 3.

El proyecto final (N_{PF}) involucra todas las unidades. Resultados de Aprendizaje 1, 2 y 3.

Nota final:

La nota final del curso (N_f) se calculará de la siguiente forma:

$$N_f = 0.3 \times P_1 + 0.2 \times T_1 + 0.3 \times P_2 + 0.2 \times N_p$$

Fechas:

- Prueba 1 (P_1) sistema operativo Unix (03/10/25).
- Tarea (T) gráfica y LaTeX (14/10/25).
- Prueba 2 (P_2) programación (05/12/25).
- Proyecto Final, entrega del proyecto (21/11/25).
- Proyecto Final, entrega del *software* y un informe con los resultados del proyecto (09/12/25).
- Proyecto Final, presentaciones de los resultados del proyecto (11 y 12 /12/25).

14. Requisitos de aprobación

El estudiante deberá obtener una calificación mayor o igual a 4.0 en la nota final del curso (N_f).

15. Palabras Clave

Unix; Gráfica; LaTeX; Programación; Python.

16. Bibliografía Obligatoria (no más de 5 textos)

- Elementos del sistema operativo Unix, Revisión 250715-01, José Rogan y M^a Paulina Correa.
- Apuntes de Python (Jupyter notebook) versión: 9.0 (250808) José Rogan.

15. Bibliografía Complementaria

Grätzer. (2014). Practical LaTeX [electronic resource] (1st ed. 2014.). Springer International Publishing. <https://doi.org/10.1007/978-3-319-06425-3>

Datta. (2017). LaTeX in 24 Hours [electronic resource] : A Practical Guide for Scientific Writing (1st ed. 2017.). Springer International Publishing. <https://doi.org/10.1007/978-3-319-47831-9>

Mittelbach, Goossens, M., Rowley, C., & Braams, J. (2004). The LaTeX companion (2nd ed.). Addison-Wesley. ISBN 0-201-36299-6

Langtangen. (2016). A Primer on Scientific Programming with Python [electronic resource] (5th ed. 2016.). Springer Berlin Heidelberg. <https://doi.org/10.1007/978-3-662-49887-3>

16. Recursos web

- Documentación del portal de LaTeX en línea Overleaf para aprender LaTeX:
<https://www.overleaf.com/learn>
- Sitio especializado en Beamer, paquete para hacer presentaciones en LaTeX
<https://latex-beamer.com/>
- Página oficial de Python: <https://www.python.org/>
- Plataforma U-Cursos, donde se distribuye el material docente:
https://www.u-cursos.cl/ciencias/2023/2/FCN460/1/material_docente/