



Anexo 1. Programas de Asignaturas

Programa de Asignatura

I. IDENTIFICACIÓN				
Carrera o Programa: Ingeniería Civil en Computación e Informática Ingeniería Civil Industrial Ingeniería en Tecnologías de Información				
Unidad responsable: Escuela de Ingeniería				
Nombre de la asignatura: Programación Científica				
Código: ECIN-00520				
Semestre en la malla¹ : V				
Créditos SCT – Chile: 5				
Ciclo de Formación	Básico		Profesional	X
Tipo de Asignatura	Obligatoria		Electiva	X
Clasificación de área de Conocimiento²				
Área: Ingeniería y Tecnología		Sub área: Otras Ingenierías y Tecnologías		
Requisitos				
Pre - Requisitos: - Estructura de Datos - Estadística		Requisito para: Machine Learning		

II. ORGANIZACIÓN SEMESTRAL							
Horas Dedicación Semanal (Cronológicas)		Docencia Directa	4,5	Trabajo Autónomo	3,7	Total	8,2
Detalle Horas Directas	Cátedra	Ayudantía	Laboratorio	Taller	Terreno	Exp. Clínica	Supervisión
	3	1,5					

¹ Este campo sólo se completa en caso de carreras con programas semestrales.

² Clasificación del curso de acuerdo a la OCDE



III. APOORTE AL PERFIL DE EGRESO

La asignatura contribuye al perfil de egreso a través de los dominios I: Conocimiento Científico y Disciplinar, dominio II: Habilidades y Actitudes Personales y Profesionales y dominio III: Habilidades interpersonales. Al finalizar la asignatura, el y la estudiante es capaz de integrar conocimiento del área de ciencias básicas y ciencias de la computación en el contexto de la analítica avanzada de datos, concluyendo en el diseño e implementación de programas complejos en el ámbito científico.

IV. COMPETENCIAS

- 1.1 Aplicar conocimientos de matemáticas y ciencias naturales: física, química, fundamentos de la computación y análisis de señales, a la solución de problemas complejos de ingeniería.
- 1.3 Aplicar conocimientos, métodos y herramientas de la especialidad para resolver problemas complejos de Ingeniería de Software, Plataformas y Gestión de Tecnologías.
- 2.1 Identificar, formular, modelar y resolver problemas complejos de ingeniería considerando las interacciones y la dinámica de las variables.
- 3.2 Comunicar comprensivamente información técnica en español, en forma oral, escrita, y gráfica, a nivel avanzado.

CG1. Capacidad de Autoaprendizaje

V. RESULTADOS DE APRENDIZAJE

- 1. Aplicar técnicas algorítmicas resolviendo problemas científicos y de análisis de datos.
- 2. Aplicar bibliotecas y herramientas científicas populares en la resolución de problemas de sistemas inteligentes.
- 3. Analizar datos utilizando métodos estadísticos, visualizando y comunicando los resultados de manera efectiva.
- 4. Aplicar buenas prácticas de programación en la Implementación de los algoritmos de inteligencia artificial.
- 5. Construir instancias de autoaprendizaje, integrando saberes diversos, encontrados en las redes de información o generados por sí mismo desde la reflexión, para transferirlo a nuevas situaciones.



VI. ÁREAS TEMÁTICAS

1. Python
 - 1.1. Lenguaje Python
 - 1.2. Introducción a Conda y Jupyter
 - 1.3. Instalación de Python con Conda
 - 1.4. Línea de comandos (Shell)
2. Conceptos básicos del lenguaje Python
 - 2.1. Uso de shell Python
 - 2.2. Números, variables, comparaciones y lógica
 - 2.3. Objetos Python: Strings, listas, tuplas y ciclos
 - 2.4. Flujo de control
 - 2.5. Salida y entrada de archivos
 - 2.6. Funciones
 - 2.7. Gráficos simples, etiquetas, leyendas y personalización. Más ejemplos de gráficos avanzados
3. Conceptos avanzados del lenguaje Python
 - 3.1. Errores y excepciones
 - 3.2. Objetos Python: Diccionarios y conjuntos
 - 3.3. Azúcar sintáctico en Python
 - 3.4. Servicios de sistema operativo
 - 3.5. Módulos y paquetes
 - 3.6. Introducción a la programación orientada a objetos
4. IPython y Jupyter Notebook
 - 4.1. IPython
 - 4.2. Jupyter Notebook
5. Numpy
 - 5.1. Arreglos básicos
 - 5.2. Leyendo y escribiendo un archivo de arreglo
 - 5.3. Métodos estadísticos
 - 5.4. Polinomios
 - 5.5. Álgebra lineal
 - 5.6. Muestreo aleatorio
 - 5.7. Transformada discreta de Fourier
6. Matplotlib
 - 6.1. Gráficos de dispersión y de líneas
 - 6.2. Personalización de gráficos
 - 6.3. Gráficos de barra, torta y polares



- 6.4. Anotación en gráficos
- 6.5. Graficar contornos y heatmaps
- 6.6. Gráficos 3D
- 6.7. Animación
- 7. Scipy
 - 7.1. Constantes físicas y funciones especiales
 - 7.2. Integración y ecuaciones diferenciales ordinarias
 - 7.3. Interpolación
 - 7.4. Optimización, ajuste de datos y cálculo de raíz
- 8. Análisis de datos con Pandas
 - 8.1. Introducción a Pandas
 - 8.2. Leyendo y escribiendo Series y Dataframes
 - 8.3. Indexación avanzada
 - 8.4. Limpieza de datos y exploración
 - 8.5. Agrupación de datos y agregación
- 9. Programación científica general
 - 9.1. Aritmética de punto flotante
 - 9.2. Estabilidad y condicionalidad
 - 9.3. Técnicas de programación y desarrollo de software
- 10. Introducción a Julia
 - 10.1. Variables, expresiones y sentencias
 - 10.2. Funciones
 - 10.3. Caso de estudio



VII. ORIENTACIONES METODOLÓGICAS

1. La metodología a desarrollar en esta asignatura debe favorecer la interacción entre las y los estudiantes a través de trabajos prácticos colaborativos que permitan la solución a problemas específicos contextualizados a la asignatura.
 - Se sugiere el uso de clases expositivas y participativas con método combinado, es decir, clases expositivas con alternancia de trabajos en grupo de corta duración para responder preguntas.
 - Se sugiere la utilización de la metodología activa de análisis de casos para desarrollar experiencias que permitan incorporar los elementos teórico-prácticos asociados a los resultados de aprendizaje de la asignatura.
2. Las experiencias de cátedra/laboratorio/taller deben ser realizadas por medio de la utilización de software moderno aplicable a la asignatura.
3. Se recomienda que las y los estudiantes realicen presentaciones periódicas sobre el trabajo realizado que incluya: contextualización, desarrollo y conclusiones.
4. Actividades prácticas recomendadas: cápsulas teóricas, reuniones de trabajo, taller de trabajo en equipo y liderazgo, presentaciones e informes escritos de avance en español, revisión del estado del arte asociado al problema, lluvia de ideas, análisis de alternativas y descripción detallada de la solución.

VIII. ORIENTACIONES Y CRITERIOS PARA LA EVALUACIÓN

1. La asignatura contempla dos instancias de evaluación de los resultados de aprendizaje: Cátedra y Laboratorio. La asignatura se debe aprobar por separado: el porcentaje de ellas será de 60% para cátedra y 40% para laboratorio.
2. Se evaluará el conocimiento conceptual y procedimental mediante el desarrollo de dos pruebas sumativas de carácter presencial para cada una de las cátedras.
3. Se evaluará el conocimiento práctico mediante la evaluación de al menos 6 trabajos de laboratorio sumativos y de carácter presencial.
4. Semanalmente, se evaluará el contenido visto en las clases teóricas por medio de una prueba formativa corta online.
5. La asistencia y condiciones de aprobación de la asignatura debe ser acorde a la aplicación del Reglamento de Docencia de Pregrado.



IX. RECURSOS BIBLIOGRÁFICOS

Bibliografía complementaria

- Hill, C. (2020). *Learning Scientific Programming with Python*. Cambridge University Press.
- Zea, D. J. (2022). Interactive visualization and plotting with Julia: Create impressive data visualizations through Julia packages such as Plots, Makie, Gadfly, and more. Packt Publishing.
- Roy, K. (2022). Python for Data Analysis and Scientists: Jump start your career in Data Analysis and Data Science Field.

Software

- Conda
- Jupyter
- Interfaz de desarrollo para lenguaje Python
- Interfaz de desarrollo para lenguaje Julia