

Programa de Asignatura

2.5.16 Programación

I. IDENTIFICACIÓN				
Carrera o Programa: Ingeniería Civil Industrial				
Unidad responsable: Escuela de Ingeniería				
Nombre de la asignatura: Programación				
Código: ECIN 00200				
Semestre en la malla³² : 3				
Créditos SCT – Chile: 5				
Ciclo de Formación	Básico	X	Profesional	
Tipo de Asignatura	Obligatoria	X	Electiva	
Clasificación de área de Conocimiento³³				
Área: Ingeniería y Tecnología		Subárea: Ingeniería eléctrica, Ingeniería electrónica Informática		
Requisitos				
Pre - Requisitos: ▪ Proyecto Introducción a la Ingeniería ▪ Álgebra II		Requisito para: ▪ Ciencia de datos		

II. ORGANIZACIÓN SEMESTRAL							
Horas Dedicación Semanal (Cronológicas)		Docencia Directa	3,0	Trabajo Autónomo	4,5	Total	7,5
Detalle Horas Directas	Cátedra	Ayudantía	Laboratorio	Taller	Terreno	Exp. Clínica	Supervisión
	-	-	-	3,0	-	-	-

³² Este campo sólo se completa en caso de carreras con programas semestrales.

³³ Clasificación del curso de acuerdo a la OCDE

III. APOORTE AL PERFIL DE EGRESO

Al final del curso el estudiante podrá utilizar el computador como una herramienta de apoyo, a través del uso y manejo de aplicaciones y lenguajes que le permitan resolver problemas que se encuentran en el mundo de la ingeniería.

IV. HABILIDADES PERFIL DE EGRESO (Relación)

De acuerdo con las habilidades descritas en el perfil de egreso, Habilidades para las Prácticas de la Ingeniería, la habilidad a desarrollar es:

- 4.3 Concebir sistemas para gestionar las operaciones, la calidad y confiabilidad y la cadena de abastecimiento, generando valor a las organizaciones, orientado por el uso eficiente del capital humano y recursos.
- 4.4 Diseñar sistemas para gestionar las operaciones, la cadena de abastecimiento, la calidad y confiabilidad, orientado por el uso eficiente del capital humano y recursos.
- 4.5. Conducir procesos de implementación de mejoras de los sistemas de gestión de operaciones, la cadena de abastecimiento, la calidad y confiabilidad, orientado por el uso eficiente del capital humano y recursos

Competencias Genéricas:

CG6: Trabajo en equipo

V. RESULTADOS DE APRENDIZAJE

1. Identificar los conceptos básicos asociados a la computación.
2. Resolver problemas de ciencias o ingeniería construyendo programas computacionales haciendo uso de archivos, arreglos y subprogramas.
3. Construir programas que satisfagan las especificaciones, verificando el comportamiento esperado de las soluciones implementadas.

VI. ÁREAS TEMÁTICAS

1. Introducción (2 semanas)

- 1.1. Conceptos básicos: bit, byte, hardware, software, sistema operativo, redes de computadores, sistema computacional.
- 1.2. Arquitectura de un computador.
- 1.3. Proceso de traducción de un programa (programa fuente, programa objeto, compilador, intérprete).
- 1.4. Pruebas (Testing)
- 1.5. Depuración de programas.

2. Razonamiento Lógico y Descomposición Modular

- 2.1. Descomposición de un problema en subproblemas
- 2.2. Abstracción y encapsulación

3. Elementos básicos de programación (6 semanas)

- 3.1. Algoritmo, programa, lenguaje de programación.
- 3.2. Concepto de variable, constante, tipos de datos básicos y operadores. Alcance de las variables, el. Local, global, por bloque
- 3.3. Constructs
 - 3.3.1. Creación y asignatura de variable.
 - 3.3.2. Lectura y despliegue de datos desde y hacia la pantalla. Input, print.
 - 3.3.3. Condicionales: if, switch.
 - 3.3.4. Ciclos: for, while, do-while.
- 3.4. Patrón de programación: contador, acumulador, bandera, entre otros.
- 3.5. Validación de datos de entrada y casos de prueba.
- 3.6. Programas para resolución de problemas matemáticos, con operaciones básicas y álgebra
- 3.7. Programas para leer y grabar archivos secuenciales.

4. Programación con subprogramas (4 semanas)

- 4.1. Estrategia de construcción de programas: dividir para conquistar.
- 4.2. Concepto de subprograma: Función.
- 4.3. Funciones con diferentes tipos de retorno, ej. Void
- 4.4. Criterios para descomponer en subprogramas.
- 4.5. Traspaso de parámetros.

5. Programas que utilizan arreglos (4 semanas)

- 5.1. Concepto de arreglo: vector y matriz.
- 5.2. Operaciones básicas sobre un arreglo: declaración y creación del arreglo, ingreso de elementos, obtención de un elemento, despliegue de elementos.
- 5.3. Algoritmos en un arreglo: búsqueda, inserción ordenada, eliminación.

VII. ORIENTACIONES METODOLÓGICAS

- 1. Las metodologías sugeridas en la asignatura son clases de método combinado, es decir, clases expositiva con alternancia de trabajos de grupo de corta duración para responder preguntas.
- 2. Se utilizará la metodología activa de análisis de casos para desarrollar experiencias necesarias que permitan incorporar elementos prácticos/ teóricos con técnicas de algoritmo y estructuras de datos.
- 3. Se realizarán mini talleres complementarios que se relacionan directamente con la materia, estos son desarrollados en equipos para fortalecer la comunicación, expresión oral y escrita, y trabajo en equipo.
- 4. Se realizarán ayudantías prácticas con ejercicios de programación e implementación.
- 5. Se entregarán talleres grupales para trabajo autónomo.

VIII. ORIENTACIONES Y CRITERIOS PARA LA EVALUACIÓN

1. Se evaluará el conocimiento conceptual y metodológico mediante pruebas presenciales sumativas.
2. Se considera además la evaluación de talleres grupales autónomo.
3. Se realizarán evaluaciones de carácter formativo (procesual) lo que permitirá observar el proceso y progreso que están teniendo los y las estudiantes en relación con las unidades temáticas y sus respectivos aprendizajes esperados. De esta forma, el docente podrá introducir correcciones, añadir alternativas y reforzar ciertos aspectos para ayudar al estudiantado a mejorar su aprendizaje.
4. Asistencia y aprobación de la asignatura según reglamento de docencia de pregrado.

IX. RECURSOS BIBLIOGRÁFICOS

Bibliografía mínima

- Apuntes de clases

Bibliografía complementaria

- Raúl González Duque. (2011). *Python para todos*. <http://mundogeek.net/tutorial-python/>
- Zed A. Shaw. (2013). *Learn Python the Hard Way*. <http://learnpythonthehardway.org/book/>
- Allen B. Downey. (2016). *Think Python*. <http://www.greenteapress.com/thinkpython2/index.html>
- Ivan Idris. (2014). *Learning NumPy Array*. Packt Publishing. ISBN:978-1783983902.
- Jay Brockman. (2009). *Introduction to Engineering: Modeling and Problem Solving*. Wiley. ISBN: 978-0470419700
- Jaan Kiusalaas. (2013). *Numerical Methods in Engineering with Python 3*. Cambridge University Press. ISBN: 978-1107033856

Software

- Lenguaje de programación Python y NumPy.