



Programa de Asignatura

I. IDENTIFICACIÓN				
Carrera o Programa: Ingeniería en Información y Control de Gestión				
Unidad responsable: FACEA – DEPARTAMENTO DE ADMINISTRACIÓN				
Nombre de la asignatura: Programación				
Código: DAAD 00193				
Semestre en la malla¹: 1				
Créditos SCT – Chile: 6				
Ciclo de Formación	Básico	X	Profesional	
Tipo de Asignatura	Obligatoria	X	Electiva	
Clasificación de área de Conocimiento²				
Área: Computación y Ciencias de la Información			Sub área:	
Requisitos				
Pre - Requisitos: ▪ Admisión			Requisito para: ▪ Programación Avanzada	

II. ORGANIZACIÓN SEMESTRAL								
Horas Dedicación Semanal (Cronológicas)	Docencia Directa	4,5	Trabajo Autónomo	5	Total	9,5		
	Detalle Horas Directas	Cátedra	Ayudantía	Laboratorio	Taller	Terreno	Exp. Clínica	Supervisión
	3	1,5						

III. APORTE AL PERFIL DE EGRESO

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

² Clasificación del curso de acuerdo a la OCDE



Esta asignatura aporta a las competencias del Dominio 2: “Gestión de Sistemas de Información”. El profesional Ingeniero en Información y Control de Gestión considerando las necesidades organizacionales es capaz de proponer, analizar, diseñar e implementar sistemas de información administrativos, así como evaluar, seleccionar, administrar y utilizar tecnologías de información con el fin de generar información relevante para la toma de decisiones.

IV. COMPETENCIAS

La/s competencia(s) en las que aporta la asignatura es/son:

- *Competencia 3: Desarrollar Sistemas de Información Administrativos para dar soluciones desde la perspectiva estratégica de la organización.*

El/los nivel(es) a desarrollar de esta(s) competencia(s) es/son:

1. Nivel inicial: Identificar las etapas principales y herramientas básicas del proceso de desarrollo de un SIA.

La/s competencia(s) genéricas en las que aporta la asignatura corresponden al nivel básico y estas es/son:

- *Uso Eficiente de las Tecnologías de Información y Comunicación*

V. RESULTADOS DE APRENDIZAJE

Los resultados de aprendizaje a desarrollar son 5, que en detalle corresponden a:

- 1. Diferenciar la sintaxis y semántica en lenguajes de programación para la implementación de soluciones*
- 2. Aplicar los conceptos básicos de programación estructurada: datos primitivos, estructuras de datos, ciclos, funciones y manejo de archivos, en la programación de diversas soluciones de computación.*
- 3. Utilizar operaciones condicionales, ciclos y funciones en el desarrollo de soluciones básicas en el marco de programación.*
- 4. Manejar operaciones de entradas y salidas, para el desarrollo de soluciones algorítmicas básicas de programación.*
- 5. Desarrollar aplicaciones de programación y de desarrollo de software según los requerimientos de las organizaciones*

Los resultados de aprendizaje de competencia genéricas corresponden a:

- *Uso eficiente de las TIC: Aplicar herramientas TIC para seleccionar información válida y confiable, identificando fuentes de información pertinente en entornos digitales.*



VI. ÁREAS TEMÁTICAS

1. Algoritmos.

1.1 Definición de algoritmo.

1.1.1 Propiedades de un algoritmo.

1.1.2 Etapas de un algoritmo.

1.1.3 Ejemplos.

1.2 Historia.

1.2.1 Historia de los computadores.

1.2.2 Historia y clasificación en niveles de lenguajes de programación.

1.2.3 Conceptos de sintaxis y semántica.

1.3 Pseudocódigo y PSeInt.

1.3.1 Sintaxis y semántica de operaciones secuenciales en PSeInt.

1.3.2 Variables y operaciones de Entrada / Salida.

1.3.3 Operaciones condicionales y de repetición.

1.4 Algoritmos y SIA.

1.4.1 Definición de SIA.

1.4.2 Algoritmos como base de SIA.

2. Lenguaje Python.

2.1 Historia y Propiedades de Python.

2.2 Variables y tipos de datos primitivos de Python.

2.3 Operadores aritméticos.

1.1.2 Cálculos en línea de comandos.

1.1.3 Cálculos con variables en línea de comandos.

2.4 Estructuras de Control de Flujo.

2.4.1 Operadores lógicos.

2.4.2 if, elif, else

2.4.3 switch, case

2.5 Estructuras Repetitivas

2.5.1 Ciclos iterativos while y for.

2.6 Estructuras de control y flujo anidadas.

3. Tipos Estructurados.

3.1 Cadenas.



3.1.1 Longitud de cadena.

3.1.2 Índice de cadena.

3.1.3 Recorrer una cadena.

3.2 Listas.

3.2.1 Agregar elementos.

3.2.2 Eliminar elementos,

3.2.3 Modificar elementos,

3.2.4 Buscar y ordenar.

3.3 Matrices.

3.3.1 Dimensiones.

3.3.2 Operaciones.

4. Funciones.

4.1 Uso y definición de funciones.

4.1.1 Parámetros de entrada.

4.1.2 Parámetros de salida.

4.2 Funcionamiento de llamada a función.

4.2.1 Pila de llamada a funciones.

4.2.2 Variables locales y globales.

5. Recursividad.

5.1 Definición y componentes.

5.1.1 Casos base.

5.1.2 Llamadas recursivas.

5.2 Ejemplos base.

5.2.1 Factorial.

5.2.2 Números de Fibonacci.

5.2.3 Búsqueda Binaria.

5.2.4 Torres de Hanoi.

6. Archivos.

6.1 Memoria estática y dinámica.

6.2 Operaciones.

6.2.1 Abrir archivo.

6.2.2 Modo lectura: operaciones para obtener información desde archivo.



6.2.3 *Modo escritura: operaciones para escribir en archivo.*

6.2.4 *Modo agregar: agregar contenido en archivo existente.*

7. *Principios de Programación Orientada a Objetos.*

7.1 *Clase y objetos.*

7.1.1 *Atributos y métodos.*

7.1.2 *Herencia.*

VII. ORIENTACIONES METODOLÓGICAS

- *Clases teóricas*
- *Clases de laboratorio*
- *Clase invertida.*
- *Trabajo individual y grupal en computador.*

VIII. ORIENTACIONES Y CRITERIOS PARA LA EVALUACIÓN

- *Cátedras.*
- *Controles.*
- *Tareas.*
- *Presentaciones de clase invertida.*
- *Proyecto grupal.*

IX. RECURSOS BIBLIOGRÁFICOS



Bibliografía mínima

Zelle, John. “Python Programming – An Introduction to Computer Science”. Franklin, Beedle & Associates, 3era Edición, 2016.

Perkovic, Ljubomir. “Introduction to Computing Using Python”. Wiley, 2nd Edition, 2015.

Bibliografía Complementaria

Downey, Allen B. “Python for Software Design”. Cambridge University Press. 2009.

Manuscrito “Think Python, How to Think Like a Computer Scientist”. 2008.

Traducción “Guía básica para pensar como un informático: Aprender con Python”

<http://www.gulic.org/almacen/htlaclwp/index.htm>

Joyanes, L.. “Algoritmos, Programación y Estructuras de Datos”. McGraw Hill, 2005.

Joyanes, L. “Fundamentos De Programación”. McGraw Hill, 2008.

Marsal, A., García, I. “Introducción a la Programación con Python 3”. Universitat Jaume I, 2014.

Bibliografía de Laboratorio

Coursera - Introducción a la programación en Python I: Aprendiendo a

programar con Python: <https://www.coursera.org/learn/aprendiendo-programar-python>

Python.org – El tutorial de Python

<http://docs.python.org.ar/tutorial/pdfs/TutorialPython2.pdf>

Universidad Tecnológica Intercontinental – Python para Todos:

http://www.utic.edu.py/citil/images/Manuales/Python_para_todos.pdf