



PROGRAMA DE ASIGNATURA Año 2025

ANTECEDENTES GENERALES*

CARRERA	INGENIERIA CIVIL EN MINAS, INGENIERIA CIVIL EN PROCESOS DE MINERALES, INGENIERIA CIVIL ELECTRICA					
NOMBRE DE LA ASIGNATURA	Programación					
CÓDIGO DE LA ASIGNATURA	INIII23					
AÑO/SEMESTRE	Primer año/semestre II					
TIPO DE FORMACIÓN**	GENERAL (G)		BÁSICA (B)	X	PROFESIONAL (P)	
DURACIÓN	SEMESTRAL	X	ANUAL		OTRO (MODULAR)	
FLEXIBILIDAD	OBLIGATORIO (O)	X	ELECTIVO (E)			
CARÁCTER	TEÓRICO-PRÁCTICO (TP)		TEÓRICO Y PRÁCTICO (T/P)	X	PRÁCTICA (P)	
MODALIDAD	PRESENCIAL	X	VIRTUAL		MIXTA	
CRÉDITOS SCT	3					
HORAS DE DEDICACIÓN	HORAS PRESENCIALES DIRECTAS	1T 2P		HORAS DE TRABAJO AUTÓNOMO		3,75 C
APRENDIZAJES PREVIOS REQUERIDOS						

* Para el llenado de todos los elementos de esta dimensión, deberá considerar todo lo definido en el descriptor del plan de estudio decretado.

** En los puntos de Tipo de Formación deberá marcar con un X la opción referente a la asignatura.

DESCRIPCIÓN DE LA ASIGNATURA

En este apartado se deberá completar el siguiente recuadro de acuerdo con el Plan de estudio vigente decretado, donde se definen las competencias, niveles y resultados de aprendizaje que la asignatura o módulo desarrolla.

Competencia Específica y/o Genérica	6.1 Domina conceptos fundamentales de las ciencias de la Ingeniería necesarios para la solución de problemáticas propias del ingeniero de base científica.
Nivel de Desarrollo de la competencia	Nivel inicial: 6.1.1 Identifica los conceptos fundamentales de las ciencias de la Ingeniería necesarias para la solución de problemáticas propias del ingeniero de base científica.
Resultado/s de Aprendizaje	6.1.1.3 Reconoce los fundamentos de programación requeridos para razonar, en forma lógica, problemas del ámbito de la ingeniería de base científica. 6.1.1.4 Utiliza lenguajes de programación y herramientas computacionales para la resolución de problemas, utilizando el razonamiento lógico propio de las ingenieras de base científica.

UNIDADES DE APRENDIZAJE

Unidad 1: INTRODUCCION A LA PROGRAMACION

- 1.1.- Algoritmos
 - 1.1.1.- Qué es un Algoritmo
 - 1.1.2.- Ejemplo de Algoritmos
 - 1.1.3.- Representación de Algoritmos
- 1.2.- Programas
 - 1.2.1.- Qué es un Programa
 - 1.2.2.- Ejemplo de un Programa
 - 1.2.2.1.- En Diagrama de Flujo
 - 1.2.2.2.- En Pseudocódigo
 - 1.2.2.3.- En Python

Unidad 2: INTRODUCCION AL LENGUAJE DE PROGRAMACION PYTHON

- 2.1.- Entornos de programación
 - 2.1.1.- PYCHARM
 - 2.1.2.- VS CODE
 - 2.1.3.- Entorno del VS CODE
- 2.2.- Tipos de datos
 - 2.2.1.- Tipo STRING (cadena de texto)
 - 2.2.2.- Tipo INT (números enteros)
 - 2.2.3.- Tipo FLOAT (números con coma)
 - 2.2.4.- Tipo BOOL (valores booleanos)
 - 2.2.5.- Tipo LISTAS (datos compuestos)
- 2.3.- Constantes y Variables
 - 2.3.1.- Constantes
 - 2.3.2.- Variables numéricas
 - 2.3.3.- Variables de texto
- 2.4.- Operadores aritméticos
 - 2.4.1.- Suma, resta, multiplicación y división
 - 2.4.2.- Módulo, exponente, división baja
- 2.5.- Operadores de comparación
 - 2.5.1.- Igual, distinto, menor que, menor igual que, mayor que, mayor o igual que
- 2.6.- Operadores lógicos
 - 2.6.1.- AND, OR, NOT

Unidad 3: PROGRAMACION CON VARIABLES SIMPLES

3.1.- Entrada y salida de datos

3.1.1.- Sentencia INPUT

3.1.3.- Sentencia PRINT

3.2.- Sentencias Condicionales

3.2.1.- Sentencia IF ELSE

3.2.2.- Sentencia IF ELIF ELSE

3.3.- Sentencias de repetición

3.3.1.- Bucle FOR IN lista

3.3.2.- Bucle FOR IN ZIP lista

3.3.3.- Bucle FOR IN RANGE

3.3.4.- Bucle WHILE

Unidad 4: PROGRAMACION CON VARIABLES COMPUESTAS

4.1.- Estructuras de datos

4.1.1.- Listas

4.1.2.- Tuplas

4.1.3.- Métodos de Listas (len, append, insert, extend, pop, remove, clear, sort, reverse)

4.2.- Métodos de String (upper, lower, capitalize, find, index, isnumeric, isalpha, count, len, endswith, startswith, replace, split, substring).

4.3.- Funciones integradas

4.3.1.- max, min, round, bool, all, sum

4.4.- Construcción de Funciones

4.4.1.- Funciones simples sin argumentos

4.4.2.- Funciones con parámetros (argumentos)

4.4.3.- Funciones que retornen valores

4.4.4.- Funciones que retornen múltiples valores

4.4.5.- Llamada de funciones de otros módulos.

Unidad 5: MISCELANEOS

5.1.- Conjuntos

5.2.- Diccionarios

5.3.- Métodos de Diccionarios (keys, get, clear, pop, items)

ESTRATEGIAS DIDÁCTICAS Y PROCEDIMIENTOS DE EVALUACIÓN

RESULTADOS DE APRENDIZAJE	ESTRATEGIA DIDÁCTICA	ACTIVIDAD DE EVALUACIÓN***
6.1.1.3 (RA1) Reconoce los fundamentos de programación requeridos para razonar, en forma lógica, problemas del ámbito de la ingeniería de base científica.	Se sugiere: - Resolución guiada de problemas - Pruebas - Talleres evaluados - Portafolio - Exposiciones Se debe realizar evaluaciones parciales prácticas y una evaluación global práctica de los fundamentos asociados al RA.	Se sugiere: - Pruebas parciales prácticas - Tareas - Pruebas globales prácticas Se debe realizar: - Pruebas y/o tareas parciales prácticas para el RA1 - Prueba Global práctica para el RA1
6.1.1.4 (RA2) Utiliza lenguajes de programación y herramientas computacionales para la resolución de problemas, utilizando el razonamiento lógico propio de las ingenieras de base científica.	Se sugiere: - Aprendizaje basado en problemas - Pruebas - Talleres evaluados - Portafolio - Exposiciones Se debe realizar evaluaciones parciales prácticas y una evaluación global práctica de los fundamentos asociados al RA.	Se sugiere: - Pruebas parciales prácticas - Tareas - Pruebas globales prácticas Se debe realizar: - Pruebas y/o tareas parciales prácticas para el RA2 - Prueba Global práctica para el RA2

* Los "Se sugiere", serán entregadas por el comité de rediseño curricular para guiar la práctica docente, pero pueden ser cambiadas por el coordinador y su equipo según estimen pertinente.

**Los "Se debe", son consensuados por el comité de rediseño curricular y deben ser considerados y cumplidos por el coordinador y su equipo.

*** En el caso de alguna asignatura que requiera de una ponderación específica, indicarlo.

EXIGENCIAS DE LA ASIGNATURA (SI CORRESPONDE) *

Por ser una asignatura definida como Teórico y Práctica (T/P) requiere un 75% de asistencia mínimo.

BIBLIOGRAFÍA.

BIBLIOGRAFÍA BÁSICA

- Metodología de la Programación: Algoritmos, Diagramas de Flujo y Programas/Osvaldo Cairó Battistutti/2da. Edición/Ed. Alfaomega 2003/ISBN: 9701509404
- Metodología de la Programación a través de Pseudocódigo/Miguel Angel Rodríguez Almeida/Ed. McGraw Hill/1993/ISBN: 8476156332
- **Python.org:** El sitio oficial de Python proporciona documentación detallada, tutoriales y recursos para aprender sobre el lenguaje y sus bibliotecas estándar. Puedes acceder a la documentación en <https://www.python.org/doc/> y encontrar tutoriales en <https://docs.python.org/3/tutorial/index.html>.

BIBLIOGRAFÍA COMPLEMENTARIA

- **W3Schools - Python Tutorial:** W3Schools ofrece tutoriales en línea sobre varios temas, incluyendo Python. Puedes acceder a su tutorial en <https://www.w3schools.com/python/>.
- **Real Python:** Un sitio web con una variedad de artículos y tutoriales sobre Python, que cubren desde conceptos básicos hasta temas más avanzados. <https://realpython.com/>
- **GeeksforGeeks - Python Programming Language:** GeeksforGeeks ofrece tutoriales, ejemplos de código y desafíos de programación en Python. Puedes visitar su sección de Python en <https://www.geeksforgeeks.org/python-programming-language/>.