

1. IDENTIFICACIÓN DE LA GUÍA

Nombre de la guía:	Taller 1
Código de la guía (No.):	001
Taller(es) o Laboratorio(s) aplicable(s):	
Tiempo de trabajo práctico estimado:	6 horas
Asignatura(s) aplicable(s):	Programación Avanzada
Programa(s) Académico(s) / Facultad(es):	Ingeniería Electrónica / Ingeniería de Telecomunicaciones

COMPETENCIAS	CONTENIDO TEMÁTICO	INDICADOR DE LOGRO
C2: Capacidad para aplicar el diseño de ingeniería para generar soluciones que satisfagan necesidades específicas, considerando factores globales y económicos. C5: Capacidad para reconocer responsabilidades profesionales en situaciones de ingeniería y emitir juicios fundamentados al analizar e interpretar datos	Estructuras de control de flujo: Definición y uso de condicionales y bucles (ciclos). Estructuras de decisión: Implementación de condicionales simples y anidados para la validación de datos. Estructuras iterativas: Uso de ciclos para sumatorias, búsqueda de números primos y procesos iterativos de aproximación.	Controla el flujo de ejecución de un programa empleando condicionales y ciclos para resolver problemas de ingeniería. Identifica los requerimientos necesarios para el desarrollo de un diseño o algoritmo en ingeniería.

2. FUNDAMENTO TEÓRICO

El desarrollo de software y la automatización de procesos se han consolidado como competencias transversales e indispensables para los profesionales de Ingeniería Electrónica y de Ingeniería de Telecomunicaciones. En el entorno tecnológico actual, no basta con capturar datos; el ingeniero debe poseer la capacidad de manipular información y procesarla adecuadamente para la toma de decisiones informadas y el control de sistemas complejos. Para lograr esto, es imperativo dominar las estructuras de control de flujo, las cuales permiten que un algoritmo deje de ser una secuencia lineal de pasos y se convierta en una solución lógica capaz de reaccionar a diferentes escenarios. El uso de Python como lenguaje base responde a su versatilidad para la simulación de sistemas, el análisis de señales y la implementación de métodos numéricos.

En este taller, se busca que el estudiante utilice la lógica de programación para:

- Formular algoritmos que validen condiciones técnicas y geométricas específicas.
- Implementar procesos iterativos que permitan la resolución de problemas de programación científica, como la estimación de raíces cuadradas mediante métodos numéricos.
- Gestionar grandes volúmenes de datos de manera eficiente mediante el uso de bucles, preparando el terreno para el análisis de series de tiempo y señales de sensores que se abordará posteriormente en el curso.

La integración de estas estructuras en entornos como Google Colab o Visual Studio Code proporciona al estudiante un ecosistema profesional para el diseño de soluciones que satisfagan necesidades de ingeniería bajo criterios de eficiencia y precisión.

Código	FGL 029
Versión	03
Fecha	18-07-2023

3. OBJETIVO(S)

- Implementar estructuras condicionales para controlar el flujo de ejecución de un programa, permitiendo la toma de decisiones basada en la validación de datos y comparaciones lógicas.
- Desarrollar algoritmos iterativos mediante el uso de ciclos para la resolución de problemas de programación científica, tales como sumatorias complejas y procesos de aproximación numérica.
- Aplicar la lógica de programación en la simulación de escenarios reales.
- Traducir requerimientos técnicos en soluciones algorítmicas funcionales utilizando entornos de desarrollo profesional como Visual Studio Code o Google Colab.

4. RECURSOS REQUERIDOS

Los elementos requeridos para realizar el taller son:

- Computador personal con internet.
- Acceso a software tipo Visual Studio Code, Google Colab, Kagle Notebooks, entre otros.
- Material de estudio entregado en clase (Notebooks).

5. ASPECTOS DE SEGURIDAD

No aplica.

6. PROCEDIMIENTO O METODOLOGÍA PARA EL DESARROLLO

6.1. Resolver los siguientes ejercicios sobre la estructura condicional en Python:

- Escriba un programa que verifique si un número es divisible por 5 y 11, o no.
- Escriba un programa que verifique si un carácter es alfabético o no.
- Investigar para que sirven los métodos que empiezan con “is”, escribir “str.is” y observar la lista que se despliega. Escriba el programa anterior usando el método “isalpha”. <https://python-reference.readthedocs.io/en/latest/docs/str/isalpha.html>
- Diseñar un programa que reciba un carácter de parte de usuario y devuelva si el carácter es una vocal o una consonante.
- diseñar un programa que reciba un carácter de parte de usuario y devuelva si el carácter es del alfabeto, numérico o un carácter especial.
- Escriba un programa que reciba el número del mes y devuelva la cantidad de días que tiene el mes.
- Escriba un programa que reciba los ángulos de un triángulo y verifique si es válido o no.
- Escriba un programa que reciba todos los lados de un triángulo y verifique si es válido o no.
- Diseñe un programa que permita calcular la factura de consumo de energía eléctrica. Se recibe la cantidad de unidades (kWh) y la factura se calcula considerando las siguientes condiciones:
 - Para los primeros 50 unidades, se factura 0.50/unidad.
 - Para las siguientes 100 unidades, se factura 0.75/unidad.
 - Para las siguientes 100 unidades, se factura 1.20/unidad.
 - Para unidades mayores a 250, se factura 1.50/unidad.
 - En toda factura se aplica un sobrecargo adicional del 20%.

Ejemplo: Para 80 unidades se calcula: Factura = $(50 \times 0.5 + 30 \times 0.75) \times 1.20$

Código	FGL 029
Versión	03
Fecha	18-07-2023

6.2. Resolver los siguientes ejercicios sobre los ciclos en Python:

- a) Escriba un programa que recibe un número “inicial”, “final” y base. La idea es sumar todos los números que están entre el número “inicial” y “final” que sean múltiplos de “base”.
- b) Escriba un programa que determine si un número ingresado es primo o no.
- c) Escriba un programa que reciba un número “inicial”, “final”. Almacenar los números que sean primos.
- d) Escriba un programa que reciba un número entero e imprima los cubos desde el 1 hasta el numero ingresado.
- e) Escriba un programa que reciba números hasta que se ingrese un número negativo. El programa debe devolver cual fue el número menor y el mayor que se ingresaron (sin incluir el número negativo que termina el ciclo).
- f) Escriba un programa que calcule la factorial de un número.
- g) Escriba un algoritmo que estime la raíz cuadrada de cualquier número S por medio del siguiente proceso iterativo

$$x_{k+1} = \frac{1}{2} \left(x_k + \frac{S}{x_k} \right)$$

AYUDA:

- Debe definir un valor de partida para el algoritmo (es el primer valor de x_k y debe ser mayor que 0).
- Calcule sucesivamente el siguiente valor de la aproximación x_{k+1} .
- El programa debe terminar cuando $|x_{k+1} - x_k| \leq 10^{-3}$ (Disminuirlo si requiere mayor precisión).

- h) Escribir un programa que reciba un vector aleatorio de la librería (numpy) y un numero entre (0,1) como umbral por parte del usuario. Devolver dos listas, una con los números mayores al umbral y la otra lista con los números menores o iguales al umbral.

7. PARÁMETROS PARA ELABORACIÓN DEL INFORME

No aplica

8. DISPOSICIÓN DE RESIDUOS

No aplica

9. BIBLIOGRAFÍA

- [1] <https://github.com/cgl-itm/ProgramacionAvanzada-ITM>
- [2] <https://python-reference.readthedocs.io/en/latest/docs/str/isalpha.html>

Elaborado por:	<i>Cristian Guarnizo Lemus / Esteban Gonzalez Valencia</i>
Revisado por:	<i>Comité de Inteligencia Computacional</i>
Versión:	<i>1</i>
Fecha:	<i>Febrero de 2026</i>