

**TECNÓLOGICO NACIONAL DE MÉXICO**

**INSTITUTO TECNÓLOGICO DE HERMOSILLO**

**DEPARTAMENTO DE INGENIERÍA ELECTRÓNICA**

**MANUAL DE PRÁCTICAS**

**ASIGNATURA:** PROGRAMACIÓN ESTRUCTURADA

**CLAVE:** ETD-1024

**CARRERA:** INGENIERIA ELECTRONICA

JESÚS MANUEL SOLÍS SANTOSCOY

DICTAMEN NUMERO AS-1-063/2018

Junio de 2018

## AGRADECIMIENTOS

A Dios por todo lo que me ha dado, Gracias, Gracias, Gracias.

A la memoria de mis padres J. Guadalupe y Aurora por su apoyo en todo momento, por los valores que me inculcaron, y por haberme dado la oportunidad de tener la mejor educación en el transcurso de mi vida. Sobre todo por ser un excelente ejemplo de vida a seguir.

A Delia, mi compañera, por ser una parte muy importante de mi vida, por haberme apoyado en las buenas y en las malas, sobre todo por su paciencia (a veces perdida) y amor incondicional. Sin ella no hubiera ido ninguna parte, con ella vamos a donde sea... T.Q.

A mis hijos Jesús Manuel, Jesús Francisco y María Alejandra, los mayores dones que he podido recibir y que han sido el motor que me impulsa siempre.

Al Tecnológico de Chihuahua por la sólida formación que me dio a pesar de mis limitaciones y al Tecnológico de Hermosillo por recibirme y darme la oportunidad de desarrollarme como docente y con ello realizarme como ser humano.

A todos mis maestros desde la “seño” Chelito hasta mis compañeros del ITH...confieso que de todos, algo me he robado.

## INDICE DE PRÁCTICAS

PRACTICA 1	
DISEÑO DE ALGORITMOS	1
PRACTICA 2	
EL ENTORNO DE PROGRAMACIÓN	7
PRACTICA 3	
PROGRAMACIÓN BÁSICA I: Estructura básica de un programa	14
PRACTICA 4	
PROGRAMACIÓN BÁSICA II: Estructura de decisión.	20
PRACTICA 5	
PROGRAMACIÓN BÁSICA III: Estructura de repetición “for”	27
PRACTICA 6	
PROGRAMACIÓN BÁSICA IV: Estructura de repetición “while”	33
PRACTICA 7	
PROGRAMACIÓN BÁSICA V: Arreglos y matrices.	38
PRACTICA 8	
PROGRAMACIÓN ESTRUCTURADA I: Estructura función y librerías I	47
PRACTICA 9	
PROGRAMACIÓN ESTRUCTURADA II: Estructura función y librerías II	56
PRACTICA 10	
PROGRAMACIÓN ESTRUCTURADA III: Estructura función y librerías III	65
PRACTICA 11	
PROGRAMACIÓN ESTRUCTURADA IV: Estructura función y librerías IV	72
PRACTICA 12	
PROGRAMACIÓN ESTRUCTURADA V: Respuesta Transitoria	79
PRACTICA 13	
COMUNICACIÓN CON EL EXTERIOR I: Entrada-Salida binaria básica	91
PRACTICA 14	
COMUNICACIÓN CON EL EXTERIOR II: Entrada-Salida binaria aplicada	97
PRACTICA 15	
COMUNICACIÓN CON EL EXTERIOR III: Entrada-Salida para control de un sistema	103

## PRESENTACIÓN

El presente manual ha sido escrito con el propósito de servir de auxiliar en la impartición del curso “PROGRAMACIÓN ESTRUCTURADA ETD-1024” de la retícula de la carrera de INGENIERÍA ELECTRÓNICA, aunque debe resultar igualmente útil para estudiantes de las carreras de INGENIERÍA ELÉCTRICA (PROGRAMACIÓN ELC-1022), INGENIERÍA MECATRÓNICA (PROGRAMACIÓN BASICA MTD-1024) y otras.

Comúnmente las prácticas de un curso de programación suelen orientarse a aplicaciones generales, desatendiendo la oportunidad de fomentar el interés del estudiante al no enfocarlas hacia los conceptos, técnicas y procedimientos que son más específicos de la ingeniería que están cursando. No es infrecuente encontrar entonces desinterés por este tipo de cursos a los que “no se les ve aplicación”, perdiendo así la posibilidad de desarrollar habilidades como el pensamiento lógico, la búsqueda de alternativas de solución de problemas, la formulación de toma de decisiones y el desarrollo de la creatividad, entre otras al tomar el curso de programación como un curso “de relleno” en la retícula.

En este manual se busca que el estudiante se motive a usar la computadora como la herramienta poderosísima que puede ser, más allá del procesamiento de textos, la navegación por la Web o los juegos de entretenimiento.

Todas las prácticas se eligieron con el objetivo de que el estudiante desarrolle sus habilidades de programador mediante ejercicios enfocados a aplicaciones en el área de las ingenierías eléctrica y electrónica, sin olvidar que un curso de programación en estas carreras habitualmente se encuentra en los primeros semestres. El estudiante encontrará aquí ejercicios que tal vez en algunos casos involucren conocimientos todavía no alcanzados al momento de llevar este curso, pero el marco teórico incluido debe ser suficiente para desarrollar el programa requerido.

Los problemas planteados en cada práctica corresponden al nivel de avance del curso, de acuerdo al programa oficial vigente, esto se resalta explícitamente en el encabezado de cada práctica al anotar los elementos del lenguaje de programación que se utilizan.

La estructura de las prácticas en este Manual consta de un encabezado y 9 secciones. El encabezado contiene el nombre, el número de la práctica y la enumeración de los elementos del lenguaje utilizados. En la primera sección se especifican las competencias a desarrollar, tanto específicas como genéricas. En la segunda sección se hace una breve introducción o marco teórico del problema que se desea resolver con la práctica, con el propósito de establecer un entorno que intenta ser lo más completo, claro y breve que sea posible. Sin embargo, el estudiante que no esté familiarizado con el contenido expuesto deberá acudir a la bibliografía. En la tercera sección se especifica la relación que tiene la práctica actual con el programa de la asignatura. En la cuarta sección se enumeran las medidas de seguridad e higiene que son recomendables y/u obligatorias al desarrollar las prácticas. En la quinta sección se detallan los materiales y el equipo necesarios para el correcto desarrollo de la práctica. En la sexta sección se establece la metodología a desarrollar que incluye a) el planteamiento del problema, de la manera más breve y concisa posible. b) El algoritmo propuesto, que como todo algoritmo, es una propuesta que está abierta a ser modificada, corregida, ampliada, desechada o lo que el estudiante considere pertinente hacer. y c) el programa en lenguaje Python previamente probado en el Raspberry Pi y que implementa el algoritmo planteado. En la séptima sección se incluyen como sugerencias didácticas la realización de al menos 2 problemas similares al planteado en la práctica. (Estos problemas deberán ser resueltos por el estudiante si se quiere alcanzar el nivel pertinente de dominio del lenguaje). En la octava sección se proponen los criterios para calificar la práctica. En la última sección (9ª) se enumera la bibliografía mínima utilizada.

Porqué Python:

Python es un lenguaje de programación interpretado con sintaxis limpia (cerca al lenguaje natural) y código muy legible (se apoya en el uso de la indentación) por lo que resulta sencillo y fácil de aprender.

Al ser de código abierto, nos permite crear nuestras aplicaciones e involucrarnos en su desarrollo. Es multiplataforma y multiparadigma permitiendo la programación imperativa, orientada a objetos y funcional. Por si esto no fuera suficiente, tiene una gran comunidad dispuesta a ayudarnos con nuestro código. La plataforma tecnológica requerida puede ser de muy bajo costo, haciéndola accesible a casi cualquier estudiante.

*“Python es uno de los lenguajes más poderosos que existen, sobre todo en aplicaciones del lado del servidor, utilidades de inteligencia artificial, programación de juegos, cómputo científico y web, es de lo más usado. A tal punto que empresas como Google lo tienen entre sus requisitos para poder ingresar.”*

*“Python es un lenguaje orientado a objetos, así como JavaScript, C++, Ruby y otros lenguajes populares. Si estás pensando convertirte en desarrollador de software, es necesario que aprendas Python. Como se considera uno de los primeros lenguajes que debes aprender, te ayudará a familiarizarte con otros lenguajes de programación que tengan una estructura parecida, sin mencionar que te ayudará a comprender más sobre la programación. Python puede ser uno de “tus primeros peldaños” como programador y te ofrece una base sólida que puedes aplicar al momento de aprender otros lenguajes orientados a objetos en el futuro.”*

Y, para finalizar no es de menor importancia subrayar que Python puede correr en una computadora cualquiera, inclusive en una de muy bajo costo y por lo tanto al alcance de un mayor número de estudiantes: Raspberry Pi es una serie de pequeñas computadoras de bajo costo desarrolladas en Inglaterra por la Fundación Raspberry Pi con el objetivo de promover la enseñanza de la ciencia básica de computación en escuelas y en países en desarrollo. Por una fracción del costo de las computadoras más baratas, se puede tener una máquina con prácticamente todas las capacidades requeridas para aprender programación, además de usarla, si se quiere, como procesador de textos, navegar por internet o para juegos de entretenimiento.

## **OBJETIVO GENERAL**

El principal objetivo de este manual es que al final del curso de programación los estudiantes sean capaces de diseñar y escribir programas en el lenguaje Python que resuelvan problemas propios de su área. Adquiriendo en el transcurso una serie de habilidades y competencias importantes en su desarrollo como ingenieros. Las siguientes competencias deberán ser del dominio del estudiante al finalizar el curso:

- Diseñar algoritmos para el planteamiento de soluciones a problemas relacionados con las diferentes disciplinas del área.
- Desarrollar programas a partir del diseño de algoritmos, para su aplicación en la solución de problemas.
- Utilizar la metodología de la programación modular para desarrollar programas estructurados y simplificar el mantenimiento del código.
- Desarrollar programas que incluyan manejo de puertos para permitir la interacción con sistemas electrónicos externos a la computadora.

## ESQUEMA DE PRÁCTICAS POR UNIDAD

Unidad, Temas y Subtemas	Número, Nombre y Objetivo específico de cada Práctica
<p><b>UNIDAD 1. Fundamentos de programación.</b></p> <p>1.1. Importancia de la programación de computadoras</p> <p>1.2. Clasificación de los lenguajes de programación</p> <p>1.3. Diseño de algoritmos</p>	<p><b>PRÁCTICA # 1</b></p> <p><b>Diseño de Algoritmos</b></p> <p>Objetivo: Desarrollar algoritmos que planteen la solución a diferentes tipos de problemas relacionados con la ingeniería electrónica.</p>
<p><b>UNIDAD 2. Elementos del Lenguaje de Programación</b></p> <p>2.1. Introducción al entorno de programación</p> <p>2.2. Estructura básica de un programa</p> <p>2.3. Palabras reservadas</p> <p>2.4. Variables y constantes</p> <p>2.5. Tipos de datos</p> <p>2.6. Despliegue y formateo de datos</p> <p>2.7. Operadores aritméticos, lógicos y relacionales</p> <p>2.8. Control de flujo</p> <p>2.9. Ciclos</p>	<p><b>PRÁCTICA # 2</b></p> <p><b>El entorno de programación</b></p> <p>Objetivo: Describir los componentes y la función que desempeñan, de la plataforma tecnológica para el desarrollo de los programas.</p> <p><b>PRÁCTICA # 3</b></p> <p><b>Estructura básica de un programa</b></p> <p>Objetivo: Conocer la estructura básica de un programa en Python y demostrar la solución de varios ejercicios para programas secuenciales.</p> <p><b>PRÁCTICA # 4</b></p> <p><b>Estructura de decisión</b></p> <p>Objetivo: Conocer la estructura básica de un programa en Python que incluya una o más tomas de decisiones y demostrar la solución de varios ejercicios para programas secuenciales.</p> <p><b>PRÁCTICA # 5</b></p> <p><b>Estructura de repetición “for”</b></p> <p>Objetivo: Conocer la estructura básica de un programa en Python que incluya la repetición “for” y demostrar la solución de varios ejercicios.</p> <p><b>PRÁCTICA # 6</b></p> <p><b>Estructura de repetición “while”</b></p> <p>Objetivo: Conocer la estructura básica de un programa en Python que incluya la repetición “while” y demostrar la solución de varios ejercicios.</p>

	<p><b>PRÁCTICA # 7</b>  <b>Arreglos y Matrices</b>  Objetivo:  Conocer el manejo de arreglos y matrices (listas) mediante varios ejercicios.</p>
<p><b>UNIDAD 3. Programación modular</b>  3.1. Declaración de funciones  3.2. Uso de bibliotecas de funciones</p>	<p><b>PRÁCTICA # 8</b>  <b>Estructura función y librerías 1</b>  Objetivo:  Conocer la estructura de un programa estructurado en Python mediante funciones y demostrar la solución de varios ejercicios incluyendo el uso de librerías.</p> <p><b>PRÁCTICA # 9</b>  <b>Estructura función y librerías 2</b>  Objetivo:  Conocer la estructura de un programa estructurado en Python mediante funciones y demostrar la solución de varios ejercicios incluyendo el uso de librerías.</p> <p><b>PRÁCTICA # 10</b>  <b>Estructura función y librerías 3</b>  Objetivo:  Conocer la estructura de un programa estructurado en Python mediante funciones y demostrar la solución de varios ejercicios incluyendo el uso de librerías.</p> <p><b>PRÁCTICA # 11</b>  <b>Estructura función y librerías 4</b>  Objetivo:  Conocer la estructura de un programa estructurado en Python mediante funciones y demostrar la solución de varios ejercicios incluyendo el uso de librerías.</p> <p><b>PRÁCTICA # 12</b>  <b>Estructura función y librerías 5</b>  Objetivo:  Conocer la estructura de un programa estructurado en Python mediante funciones y demostrar la solución de varios ejercicios incluyendo el uso de librerías y las propiedades de graficación del lenguaje.</p>

<p><b>UNIDAD 4. Aplicación de puertos de comunicación.</b></p> <p>4.1. Puertos de comunicación 4.2. Envío y recepción de datos</p>	<p><b>PRÁCTICA # 13</b> <b>Entrada/Salida básica puertos binarios</b> Objetivo: Resolver problemas que involucren la entrada y salida de datos ON-OFF.</p> <p><b>PRÁCTICA # 14</b> <b>Entrada/Salida de puertos binarios</b> Objetivo: Resolver problemas que involucren la entrada y salida de datos ON-OFF, 2ª Parte.</p> <p><b>PRÁCTICA # 15</b> <b>Sistema de control secuencial</b> Objetivo: Resolver problemas que involucren control de sistemas secuenciales.</p>
--	--



## PRÁCTICA #1

### DISEÑO DE ALGORITMOS

#### COMPETENCIAS A DESARROLLAR

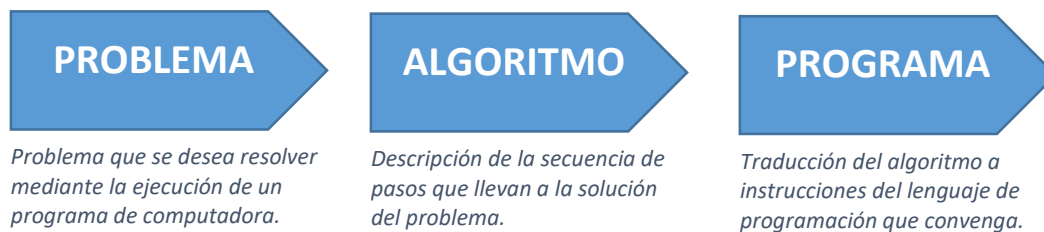
##### Específicas:

Diseña algoritmos para el planteamiento de soluciones a problemas relacionados con las diferentes disciplinas del área.

**Genéricas:** Capacidad creativa, Capacidad cognitiva para comprender y utilizar ideas, pensamientos e información, Capacidad para aprender, Capacidad crítica y autocrítica, Capacidad de investigación, Capacidad para tomar decisiones.

#### INTRODUCCIÓN

*Para llegar a la realización de un programa de computadora es necesario el diseño previo de un algoritmo, sin él nos puede existir el programa. Los algoritmos son independientes de la computadora donde se ejecutará el programa y del lenguaje de programación utilizado.*



*Un algoritmo debe ser:*

- *Preciso*
- *Definido*
- *Finito*

*Estructura:*

- *Cabecera*
- *Descripción*
- *Declaraciones*
- *Cuerpo*

*Ejemplo: Un algoritmo para calcular el área de una circunferencia se puede escribir de la siguiente manera:*

ALGORITMO: Área de una circunferencia

DESCRIPCIÓN: Se calcula el área de una circunferencia, dado el radio.

CONSTANTES:

*real* :  $\pi = 3.141592$

VARIABLES:

*real* : Área, radio

INICIO

1. Escribir ("Introduzca radio: ")
2. Leer (radio)
3.  $\text{Área} \leftarrow \pi \cdot \text{radio}^2$
4. Escribir ("El área de la circunferencia es :", Área)

FIN

**Nota:** Este algoritmo está escrito en pseudocódigo, forma más recomendable de escribir algoritmos.

### **CORRELACIÓN CON LOS TEMAS Y SUBTEMAS DEL PROGRAMA VIGENTE**

Esta práctica se relaciona directamente con los temas de la unidad 1. El propósito es que el estudiante empiece a desarrollar habilidades creativas, de comprensión y manejo de ideas, pensamientos e información, así como su capacidad para aprender con sentido crítico. También su capacidad de investigación y de toma de decisiones.

### **MEDIDAS DE SEGURIDAD E HIGIENE**

Por la naturaleza de esta práctica, no se utiliza un laboratorio. El estudiante realizará su práctica usando como herramienta únicamente "lápiz y papel" por lo que no se requieren de precauciones de seguridad e higiene.

### **MATERIAL Y EQUIPO NECESARIO**

Ninguno.

## METODOLOGÍA

### 1. Planteamiento del problema:

“Diseñar algoritmos para el planteamiento de soluciones a problemas relacionados con las diferentes disciplinas del área.” (Tomado textualmente del programa vigente PROGRAMACIÓN ESTRUCTURADA ETD-1024 de la carrera de INGENIERÍA ELECTRÓNICA).

#### Ejercicio 1.0

Diseñe un algoritmo para resolver el siguiente problema: El electrón y el protón de un átomo de hidrógeno están separados (en promedio) por una distancia de aproximadamente  $5.3 \times 10^{-11} m$ . Encuentre las magnitudes de la fuerza eléctrica y la fuerza gravitacional entre las dos partículas.

Algoritmo propuesto:

ALGORITMO: Fuerzas gravitacional y electrostática

DESCRIPCIÓN: Se calculan las fuerzas gravitacional y electrostática que se ejercen entre dos partículas con masa y cargadas eléctricamente, separadas cierta distancia.

CONSTANTES:

$$real : K = 8.987551788 \dots \times 10^9 N \cdot m^2 / C^2 \text{ (Ley de Coulomb)}$$

$$real : G = 6.67428 \times 10^{-11} N \cdot m^2 / kg^2 \text{ (Constante gravitacional)}$$

$$real : e = 1.602176487 \dots \times 10^{-19} C \text{ (Carga elemental)}$$

$$real : m_e = 9.10938215 (45) \times 10^{-31} kg \text{ (Masa del electrón)}$$

$$real : m_p = 1.672621637 (83) \times 10^{-27} kg \text{ (Masa del protón)}$$

$$real : Fuerza\_g \text{ Fuerza gravitacional}$$

VARIABLES:

Continua...

INICIO

1. Escribir ("introduzca distancia: ")
2. Leer ( $d$ )
3.  $Fuerza\_e \leftarrow K \cdot \frac{e \times e}{d^2}$
4.  $Fuerza\_g \leftarrow G \cdot \frac{m_e \times m_p}{d^2}$
5. Escribir ("Fuerza eléctrica es:",  $Fuerza\_e$ )
6. Escribir ("Fuerza gravitacional es:",  $Fuerza\_g$ )

FIN

**Programa propuesto:** NO APLICA

### SUGERENCIAS DIDÁCTICAS

Desarrollar algoritmos que resuelvan problemas similares al Ejercicio 1.0. Los siguientes son problemas sugeridos, pero cada estudiante podrá plantearse otros que sean de su interés.

#### Ejercicio 1.1

Calcular la magnitud de dos cargas puntuales separadas una distancia  $d$ , sobre las cuales se ejerce una fuerza eléctrica  $F$ . Considerando que la relación entre las magnitudes de las cargas es  $P$  ( $q_1 = P \cdot q_2$ ).

#### Ejercicio 1.2

Repetir el Ejercicio 1.1, considerando que las magnitudes calculadas siempre deben ser múltiplos de la carga elemental, de lo contrario se trata de resultados incongruentes.

#### Ejercicio 1.3

Repetir el Ejercicio 1.2, considerando que se trata de un conjunto de  $N$  pares de cargas.

#### Ejercicio 1.4

Calcular la ubicación entre dos cargas dadas positivas  $q_2$  y  $q_3$  ubicadas sobre el eje  $x$  y separadas una distancia  $d$ , de una carga puntual negativa  $q_1$ , de manera que la fuerza neta resultante en la carga  $q_1$  sea cero.

#### Ejercicio 1.5

Calcular la corriente y potencia consumida por una resistencia  $R$  a la que se le aplica un voltaje  $V$ . Considerando que se trata de un conjunto de tamaño indeterminado de valores de Resistencia-Voltaje aplicado.

**Ejercicio 1.6**

Calcular la magnitud y dirección del campo eléctrico en el origen de coordenadas, dadas 2 partículas puntuales cargadas ubicadas en los puntos p1 y p2.

**Ejercicio 1.7**

Calcular la resistencia equivalente de una combinación de N resistencias en:

- a) Serie
- b) Paralelo

**Ejercicio 1.8**

Calcular el punto de operación de un transistor FET de unión.

**Ejercicio 1.9**

Calcular el valor promedio, la desviación del valor promedio, el promedio de las desviaciones y la desviación estándar de un conjunto de mediciones eléctricas. (Ejemplo 2-4 de Guía para las Mediciones eléctricas y Electrónicas pags. 54...)

**REPORTE DEL ALUMNO**

Criterio para calificar	Si	No
El documento tiene portada	25%	0
Entregó en fecha programada	25%	0
El documento está completo	25%	0
Los algoritmos son correctos	25%	0

**BIBLIOGRAFÍA**

Profr. Miguel Rodríguez H.. (2017). *Recopilación de Algoritmos*. Febrero 28, 2018, de BUAP Sitio web:

[http://www.academia.edu/4562755/Algoritmos\\_50\\_Ejemplos](http://www.academia.edu/4562755/Algoritmos_50_Ejemplos)

Knuth, Donald E.. (2002). *EL ARTE DE PROGRAMAR ORDENADORES*. Barcelona: Editorial Reverté.

## PRÁCTICA #2

### EL ENTORNO DE PROGRAMACIÓN

#### COMPETENCIAS A DESARROLLAR

##### Específicas:

Comprender las diferencias entre lenguajes compilados, interpretados, estructurados y orientados a objetos.

Comprender los componentes y su función de una computadora adecuada a aplicaciones de ingeniería electrónica.

**Genéricas:** Capacidad creativa, Capacidad cognitiva para comprender y manipular ideas, pensamientos e información, Capacidad para aprender, Capacidad crítica y autocrítica, Capacidad de investigación, Capacidad para tomar decisiones.

#### INTRODUCCIÓN

##### 2.1 HARDWARE

**2.1.1 Raspberry Pi** es una pequeña computadora de bajo costo, diseñada para facilitar el aprendizaje de la programación y el uso de computadoras (figura 2.1) que corre el sistema operativo Linux. Posee enchufes para Bus Universal en Serie (USB) en donde se puede conectar un teclado y un ratón así como salida de video tipo Interface Multimedia de Alta Definición (HDMI) para conectar un monitor o una televisión y muchos otros dispositivos. También tiene 40 terminales de Entrada/Salida de Propósito General (GPIO).



Figura 2.1

Cuando se enciende el Raspberry Pi, se muestra un escritorio como el de la figura 2.2 , similar al escritorio de Windows.



Figura 2.2

- 2.1.2 **Raspberry Pi** sirve para casi todo lo que se puede hacer con una computadora de escritorio...con algunas limitaciones. No se tiene un disco duro, en su lugar existe una tarjeta micro SD (aunque se puede conectar un disco duro externo a uno de los puertos USB), pero se pueden editar documentos, navegar por internet y jugar juegos. Raspberry Pi es un candidato ideal para usarse como centro de medios.
- 2.1.3 Raspberry Pi está conformado por un Circuito Integrado (figura 2.3) “Sistema en un solo chip” de Broadcom ([www.broadcom.com](http://www.broadcom.com)) que incluye 1GB de memoria así como el procesador de propósito general y el de gráficas que controlan los demás componentes.



Figura 2.3

Se puede encontrar toda la información del hardware del Raspberry Pi en <https://www.raspberrypi.org/documentation/hardware/raspberrypi/>

2.1.4 Raspberry Pi necesita de unos cuantos componentes externos para funcionar (figura 2.4):

- a) Una fuente de poder que proporcione 5V @ 1A con conector micro-USB.
- b) Teclado y ratón que se conecten a puertos USB.
- c) Monitor con entrada HDMI.
- d) Una tarjeta microSD
- e) Cable HDMI (y un adaptador HDMI→VGA si el monitor (c) no tiene esa entrada).

**Se recomienda conectar la fuente de alimentación al final de todos los demás componentes:**

1. La tarjeta microSD debe estar en su ranura.
2. Conectar el teclado y el ratón.
3. Conectar el monitor.
4. Conectar la fuente de alimentación de 5V.



Figura 2.4

## 2.2 SOFTWARE

**2.2.1 Raspberry Pi** interactúa con el usuario mediante un sistema operativo que puede ser el estándar Raspbian u otro dependiendo de la experiencia o preferencias del mismo usuario. Raspbian contiene Python, Scrtach, Sonic PI, Java, Mathematica y más. Para instalar Raspbian se debe tener cargado NOOBS (New Out Of the Box Software, un sencillo instalador de sistema operativo que contiene Raspbian).

**2.2.2 El sistema** operativo de una computadora es un conjunto de programas vital para la computadora. Se encarga de la interface entre el usuario y la máquina y permite que funcionen todos los demás programas. Los sistemas operativos realizan tareas básicas, tales como reconocimiento de la conexión del teclado, enviar la información a la pantalla, administrar los



archivos y directorios en el disco, y controlar los dispositivos periféricos tales como impresoras, escáner, etc. Raspbian es un sistema operativo libre y gratuito basado en Debian y optimizado para el hardware de la Raspberry Pi. Para quienes aún no les haya quedado claro qué es un sistema de esta naturaleza, diremos que es el conjunto de programas básicos y utilitarios (programas de soporte, pues), que permiten que la Raspberry haga algo útil. Sin embargo, Raspbian es algo más que un sistema operativo, pues viene con unos 35 mil paquetes, precompilados, de forma tal que sea fácil instalar el que necesitemos en la Raspberry Pi.

### 2.2.3 Carga del sistema operativo en Raspberry Pi

La primera vez que se inserta la tarjeta microSD con el programa NOOBS y se energiza la tarjeta, procederá inmediatamente a cargar el sistema operativo Raspbian. (Esto se llevará un poco de tiempo.) Una vez terminado el proceso de carga, se mostrará el escritorio (figura 2), quedando listo el Raspberry Pi para empezar a trabajar.

### 2.2.4 Raspberry Pi interactivo

El programa IDLE (Ambiente de Aprendizaje y Desarrollo Integrado) tiene 2 tipos de ventana: La Ventana “Cáscara” (Shell) y la Ventana (puede ser múltiple) del Editor.

En la ventana “Shell” se pueden introducir instrucciones de Python y ejecutarse una a la vez. Esta modalidad es muy útil para probar instrucciones o partes de un programa.

Por ejemplo, podemos probar la instrucción *print*.

```
>>>print('Hola')  
  
Hola  
  
>>>
```

O usarla como calculadora:

```
>>>2 + 2  
  
4  
  
>>>
```

Nótese que **print** se escribe en color rojo, 'Hola' en color verde y 2+2 en color negro.

Esto es porque **print** es una instrucción (palabra reservada), 'Hola' es un **string** y 2 y + son **valores**.

### 2.2.5 Primer programa en Python.

Estando en la ventana Shell se abre un nuevo archivo (Editor) con el comando

**File → New File .**

En la nueva ventana (de editor) escribiremos:

```
#Este es mi primer programa en Python
a = 5 #Primer valor se almacena en la variable a
b = 6 #Segundo valor se almacena en la variable b
print('*****')
c = a + b      #Se calcula la suma y se almacena en c
print('El valor de c = ', c) #Se escribe el resultado
print('Fin del programa')
```

**File → Save As.**

Abriendo una ventana para seleccionar el lugar donde almacenar el archívito de nuestro primer programa, al cual deberemos ponerle nombre:

**Primerito.py**

(por ejemplo...además es muy importante agregarle la extensión py)

Por último, se corre el programa:

**Run → Run Module F5**

## CORRELACIÓN CON LOS TEMAS Y SUBTEMAS DEL PROGRAMA VIGENTE

Esta práctica se relaciona directamente con los temas de la unidad 1. El propósito es que el estudiante conozca las características del sistema de cómputo que se utilizará en el desarrollo de los programas.

### MEDIDAS DE SEGURIDAD E HIGIENE

Por la naturaleza de esta práctica, no se utiliza un laboratorio. El estudiante realizará su práctica usando como herramienta únicamente “*lápiz y papel*” por lo que no se requieren de precauciones de seguridad e higiene.

### MATERIAL Y EQUIPO NECESARIO

1. Paquete de Raspberry Pi (Tarjeta Raspberry Pi, Fuente de poder, Tarjeta micro SD con sistema operativo Raspbian, Teclado, ratón y monitor).
2. Computadora personal con Intérprete de Python.

### METODOLOGÍA

#### 1. Planteamiento del problema:

- a) Utilizar el “Shell” de Python

Pruebe en el “Shell” las siguientes instrucciones:

```
2**3          print('*'*20)          (20*9/5)+32          print('\t','***','\t','+++')
```

```
x = input()  Enter 45  Enter x          Enter (Se lee de la consola el valor 45 y se escribe)
```

- b) Utilizar el editor de Python

b.1 Con el Editor, escriba un programa y pruébelo para leer un valor y elevarlo al cuadrado, al cubo, y obtener su raíz cuadrada. Escriba los resultados en una sola línea, separando los valores con el tabulador.

b.2 Con el Editor, escriba un programa y pruébelo para leer 4 valores y escribir el promedio.

### SUGERENCIAS DIDÁCTICAS

Probar operaciones en el “Shell” para sumar, restar, multiplicar, dividir, elevar a potencias enteras o fraccionarias, evaluar expresiones con constantes y variables, etc.

Escribir y probar programas simples para leer datos y escribirlos en diferentes órdenes.

**REPORTE DEL ALUMNO**

Criterio para calificar	Si	No
El documento tiene portada	25%	0
Entregó en fecha programada	25%	0
El documento está completo	25%	0
Los gramas son correctos	25%	0

**BIBLIOGRAFÍA**

Raspberry Pi Foundation. Disponible en <https://www.raspberrypi.org/>

Lenguaje Python. Disponible en <https://www.python.org>

## PRACTICA #3

### ESTRUCTURA BÁSICA DE UN PROGRAMA EN PYTHON

Estructura básica de un programa en Python

Entrada / Salida de datos por consola

Operaciones : Suma aritmética

Tipos de datos: Enteros, cadenas de caracteres

### COMPETENCIAS A DESARROLLAR

#### Específicas:

Implementa programas a partir del diseño de algoritmos de aplicaciones básicas en la solución de problemas reales.

Utiliza adecuadamente las herramientas de diseño para el desarrollo de programas.

Desarrolla programas a partir del diseño de algoritmos, para su aplicación en la solución de problemas.

#### Genéricas:

Capacidad creativa, Capacidad cognitiva para comprender y manipular ideas, pensamientos e información, Capacidad para aprender, Capacidad crítica y autocrítica, Capacidad de investigación, Capacidad para tomar decisiones.

### INTRODUCCIÓN

En un circuito eléctrico puramente resistivo siempre es posible reemplazar su configuración de resistencias por una resistencia equivalente, según como estas estén conectadas hablamos de un sistema paralelo o un sistema serie, lo importante es que sea cual sea la configuración podemos establecer un cálculo de resistencias en paralelo o en serie de manera que nos quede una única resistencia equivalente al sistema.

Decimos que dos resistencias están en serie cuando la entrada de una está conectada a la salida de la otra, como en la figura 3.1

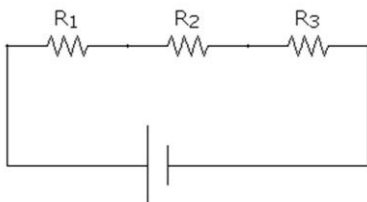


Figura 3.1

En un circuito serie puramente resistivo la corriente que circula por cada resistencia es la misma, eso produce que la resistencia total a la que se tiene que enfrentar la corriente para

circular es la suma de todas las resistencias, con lo cual la resistencia equivalente de un circuito serie es la suma de todas las resistencias.

Para calcular la resistencia en serie equivalente solo tenemos que realizar la suma de todos los resistores que forman parte de la rama serie.

$$R_{eq} = R_1 + R_2 + R_3 + \dots + R_n$$

Decimos que dos resistencias están en paralelo cuando los terminales de salida están unidos entre si al igual que los terminales de entrada, como se aprecia en la figura 3.2.

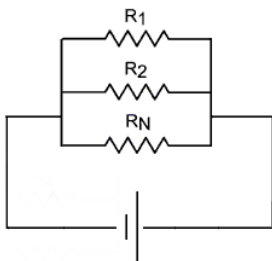


Figura 3.2

En un circuito paralelo puramente resistivo el voltaje presente en cada resistencia es el mismo, pero la corriente es diferente si la resistencia es diferente. Por lo tanto la corriente total ( ) se reparte entre cada resistencia, dando como resultado la siguiente ecuación:

$$I_{TOTAL} = I_1 + I_2 + I_3 + \dots + I_n \quad , \text{donde cada corriente será: } I_x = \frac{V}{R_x}$$

Sustituyendo para cada corriente en la ecuación anterior queda:

$$I_{TOTAL} = \frac{V}{R_1} + \frac{V}{R_2} + \frac{V}{R_3} + \dots + \frac{V}{R_n} \quad \text{y ya que la corriente total será } I_{TOTAL} = \frac{V}{R_{eq}}$$

...basta un poquito de álgebra para obtener: 
$$R_{eq} = \frac{1}{\frac{1}{R_1} + \frac{1}{R_2} + \frac{1}{R_3} + \dots + \frac{1}{R_n}}$$

Para calcular la resistencia paralelo equivalente solo tenemos que calcular el inverso de la suma de todos los inversos de los resistores que forman parte de la rama paralelo.

## **CORRELACIÓN CON LOS TEMAS Y SUBTEMAS DEL PROGRAMA VIGENTE**

Esta práctica se relaciona directamente con los temas de la unidad 2. El propósito es que el estudiante se familiarice con el entorno de programación por medio del diseño, edición y compilación de programas sencillos. Que investigue cuales son las palabras reservadas del lenguaje utilizado, que desarrolle programas que utilicen variables, constantes y los diferentes tipos de datos. También que desarrolle programas que utilicen los operadores aritméticos, lógicos y relacionales. Asimismo, que desarrolle programas que utilicen la mayoría de los puntos vistos en la estructura básica de un programa con la finalidad de dar solución a problemas cotidianos.

## **MEDIDAS DE SEGURIDAD E HIGIENE**

Son obligaciones de los usuarios del laboratorio:

- I. Para realizar la práctica el estudiante debe llevar el material necesario indicado por el Profesor;
- II. Minimizar el uso de celulares al interior del Laboratorio;
- III. Atender puntualmente las indicaciones que le dé su Profesor y el personal de laboratorio;
- IV. Mantener el orden y disciplina dentro de las instalaciones del laboratorio;
- V. Informar inmediatamente al Profesor y al Jefe de Laboratorio si ocurre un accidente;
- VI. Acatar las disposiciones de seguridad y preservar la higiene del laboratorio.

A los usuarios del laboratorio se les prohíbe:

- I. Introducir alimentos. Las bebidas permitidas como agua o refrescos no deben dejarse cerca de los circuitos o equipos eléctricos;
- II. Introducir mascotas, juguetes u otros objetos ajenos a las prácticas o al trabajo de laboratorio;
- III. Fumar, correr, gritar, jugar, empujar, tocar instrumentos musicales, usar radios o teléfonos, que alteren el orden;
- IV. Usar teléfonos móviles, radio localizadores, reproductores de música u otros aparatos ajenos a las prácticas de laboratorio;
- V. Usar las instalaciones de los laboratorios para otros propósitos, diferentes del desarrollo de prácticas programadas;
- VI. Incurrir en cualquier tipo de comportamiento que ponga en riesgo la integridad de las instalaciones, equipo de laboratorio, los usuarios, y jefe de laboratorio;
- VII. Introducir personas ajenas a las autorizadas para las prácticas.

## MATERIAL Y EQUIPO NECESARIO

1. Paquete de Raspberry Pi (Tarjeta Raspberry Pi, Fuente de poder, Tarjeta micro SD con sistema operativo Raspbian, Teclado, ratón y monitor), o...
2. Computadora personal con Intérprete de Python.

## METODOLOGÍA

### 1. Planteamiento del problema:

Diseñe un programa que calcule la resistencia equivalente de 2 resistencias en serie y muestre el resultado.

### 2. Algoritmo

*ALGORITMO: Resistencias serie*

*DESCRIPCIÓN: Se calcula el equivalente serie de 2 resistencias dadas.*

*CONSTANTES: -----*

*VARIABLES:*

*real : R1, R2 Resistencias*

*real : Serie Equivalente serie*

*INICIO*

1. *Escribir("Da valor de resistencia 1")*
2. *Leer (R1)*
3. *Escribir ("Da valor de valor de resistencia 2")*
4. *Leer (R2)*
5. *Suma  $\leftarrow R1 + R2$*
6. *Escribir (Suma)*

*FIN*



### 3. Programa desarrollado

```
#Programa 1  
#Calcula resistencias en serie  
R1 = int(input('Introduce valor de resistencia 1 '))  
R2 = int(input('Introduce valor de resistencia 2 '))  
Serie = R1 + R2  
print('Resistencia Total Serie = ', Serie)  
#Fin del programa
```

#### SUGERENCIAS DIDÁCTICAS

Desarrollar programas que resuelvan problemas similares al resuelto por el programa #1. Los siguientes son problemas sugeridos, pero cada estudiante podrá plantearse otros que sean de su interés.

Diseñe, Codifique y Pruebe programas para:

Ejercicio 3.1 (Resistencias en paralelo) Calcular el equivalente de 2 resistencias en paralelo.

Ejercicio 3.2 (Ley de Coulomb) Calcular la fuerza ejercida entre 2 cargas puntuales separadas una distancia  $d$ . (Ver Ejercicio 1.0)

Ejercicio 3.3 (Ley de Ohm) Calcular la corriente que pasa por una resistencia al aplicarle un potencial entre sus terminales. También calcular la potencia disipada en la resistencia.

**REPORTE DEL ALUMNO**

<b>Criterio para calificar</b>	<b>Si</b>	<b>No</b>
El documento tiene portada	10%	0
Entregó en fecha programada	10%	0
El documento está completo	20%	0
El programa funciona correctamente	20%	0
Incluye un ejercicio adicional (al menos)	20%	0
Incluye resultados y/o gráficas o tablas	20%	0

**BIBLIOGRAFÍA**

Raspberry Pi Foundation. Disponible en <https://www.raspberrypi.org>

Lenguaje Python. Disponible en <https://ww.python.org>

Young, Hugh D. y Freedman, Roger A. *Física Universitaria con Física Moderna*, Vol. 2. XII Edición. Pearson Educación.

## PRACTICA #4

### DECISIONES EN PYTHON

Estructura básica de un programa en Python

Entrada / Salida de datos por consola

Estructura de decisión if-else

Operaciones : Multiplicación aritmética, comparación

Tipos de datos: Enteros, cadenas de caracteres

### COMPETENCIAS A DESARROLLAR

#### Específicas:

Implementar programas a partir del diseño de algoritmos de aplicaciones básicas en la solución de problemas reales.

Utilizar adecuadamente las herramientas de diseño para el desarrollo de programas.

Desarrollar programas a partir del diseño de algoritmos, para su aplicación en la solución de problemas.

#### Genéricas:

Capacidad creativa, Capacidad cognitiva para comprender y manipular ideas, pensamientos e información, Capacidad para aprender, Capacidad crítica y autocrítica, Capacidad de investigación, Capacidad para tomar decisiones.

### INTRODUCCIÓN

La figura 4.1 representa un elemento de circuito con diferencia de potencial  $V_a - V_b = V_{ab}$  entre sus terminales y la corriente  $I$  que pasa a través del elemento en dirección de  $a$  hacia  $b$ .

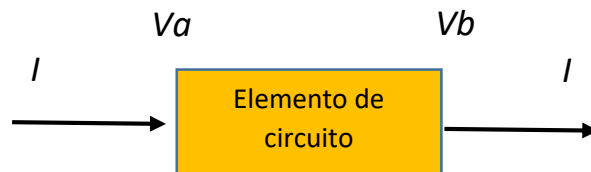


Fig. 4.1

Conforme la carga se desplaza a través del elemento de circuito, el campo eléctrico realiza trabajo sobre la carga. Conforme una cantidad de carga  $q$  pasa a través del elemento de circuito, hay un cambio en la energía potencial igual a  $qV_{ab}$ . La cantidad  $qV_{ab}$  representa energía eléctrica transferida hacia el elemento de circuito. Puede ocurrir que el potencial en  $b$  sea mayor que  $a$ . En este caso,  $V_{ab}$  es negativo, y hay una transferencia neta de energía hacia afuera del elemento de circuito. En los circuitos eléctricos es más frecuente que interese la **rapidez** con la que la energía se proporciona a un elemento de circuito o se extrae de él. Si la corriente a través del elemento es  $I$ , entonces en un intervalo de tiempo  $dt$  pasa una cantidad de carga  $dQ = Idt$  a través del elemento. El cambio en energía potencial para esta cantidad de carga es  $V_{ab}dQ = V_{ab}Idt$ . Si esta expresión se divide entre  $dt$ , se obtiene la rapidez a la que se transfiere la energía hacia fuera o hacia dentro del circuito. La relación de transferencia de energía por unidad de tiempo es la **potencia**, y se denota mediante  $P$ ; por lo tanto, escribimos

$$P = V_{ab}I$$

La unidad de  $V_{ab}$  es un voltio, o un joule por coulombio, y la unidad de  $I$  es un ampere, o un coulombio por segundo. Entonces la unidad de  $P = V_{ab}I$  es un watt.

$$(1 J/C)(1 C/s) = 1 J/s = 1 W$$

Si el elemento de circuito de la figura 5.1 es una resistencia, la diferencia de potencial es  $V_{ab} = IR$ , por lo tanto la potencia eléctrica entregada a la resistencia es

$$P = V_{ab}I = IR \cdot I = I^2R = \frac{V_{ab}^2}{R}$$

## CORRELACIÓN CON LOS TEMAS Y SUBTEMAS DEL PROGRAMA VIGENTE

Esta práctica se relaciona directamente con los temas de la unidad 2. El propósito es que el estudiante se familiarice con el entorno de programación por medio del diseño, edición y compilación de programas sencillos. Que investigue cuales son las palabras reservadas del lenguaje utilizado, que desarrolle programas que utilicen variables, constantes y los diferentes tipos de datos. También que desarrolle programas que utilicen los operadores

aritméticos, lógicos y relacionales. Asimismo, que desarrolle programas que utilicen la mayoría de los puntos vistos en la estructura básica de un programa con la finalidad de dar solución a problemas cotidianos.

## **MEDIDAS DE SEGURIDAD E HIGIENE**

Son obligaciones de los usuarios del laboratorio:

- I. Para realizar la práctica el estudiante debe llevar el material necesario indicado por el Profesor;
- II. Minimizar el uso de celulares al interior del Laboratorio;
- III. Atender puntualmente las indicaciones que le dé su Profesor y el personal de laboratorio;
- IV. Mantener el orden y disciplina dentro de las instalaciones del laboratorio;
- V. Informar inmediatamente al Profesor y al Jefe de Laboratorio si ocurre un accidente;
- VI. Acatar las disposiciones de seguridad y preservar la higiene del laboratorio.

A los usuarios del laboratorio se les prohíbe:

- I. Introducir alimentos. Las bebidas permitidas como agua o refrescos no deben dejarse cerca de los circuitos o equipos eléctricos;
- II. Introducir mascotas, juguetes u otros objetos ajenos a las prácticas o al trabajo de laboratorio;
- III. Fumar, correr, gritar, jugar, empujar, tocar instrumentos musicales, usar radios o teléfonos, que alteren el orden;
- IV. Usar teléfonos móviles, radio localizadores, reproductores de música u otros aparatos ajenos a las prácticas de laboratorio;
- V. Usar las instalaciones de los laboratorios para otros propósitos, diferentes del desarrollo de prácticas programadas;
- VI. Incurrir en cualquier tipo de comportamiento que ponga en riesgo la integridad de las instalaciones, equipo de laboratorio, los usuarios, y jefe de laboratorio;
- VII. Introducir personas ajenas a las autorizadas para las prácticas.

## MATERIAL Y EQUIPO NECESARIO

1. Paquete de Raspberry Pi (Tarjeta Raspberry Pi, Fuente de poder, Tarjeta micro SD con sistema operativo Raspbian, Teclado, ratón y monitor), o...
2. Computadora personal con Intérprete de Python.

## METODOLOGÍA

### 1. Planteamiento del problema:

Diseñe un programa que calcule la potencia eléctrica proporcionada por una fuente de voltaje, dados el voltaje y la corriente proporcionados por la fuente, siempre y cuando el valor calculado sea menor a un valor máximo dado. En caso contrario escribir un mensaje indicando que se excedió el límite establecido.

### 2. Algoritmo

*ALGORITMO: Potencia máxima*

*DESCRIPCIÓN: Se calcula la potencia proporcionada por una fuente, dada la corriente consumida y un valor máximo permitido de potencia. En caso de exceder el límite se emite una advertencia.*

*CONSTANTES:*

-----

*VARIABLES:*

*real : P max, P Potencia máxima permitida y Potencia proporcionada*

*real : V, I Valor de la fuente y la corriente*

*INICIO*

1. *Escribir ("Dar valor de Potencia\_maxima")*
2. *Leer( Pmax)*
3. *Escribir ("Dar valor de voltaje")*
4. *Leer (V)*
5. *Escribir ("Dar valor de corriente")*
6. *Leer (I)*
7.  *$P \leftarrow V * I$*
8. *Si  $P < Pmax$  entonces*

*Escribir (P)*

*otro*

*Escribir ("Valor límite alcanzado")*

*FIN*

### 3. Programa

```
#Programa 2  
#Calcula Potencia de Fuente  
Pmax = int(input('Introduce valor límite de la potencia (Watts) '))  
V = int(input('Introduce valor de voltaje (Voltios) '))  
I = int(input('Introduce valor de corriente (Amperes) '))  
P = V * I  
if P < Pmax :  
    print('Potencia entregada por la fuente = ', P, 'watts')  
else:  
    print('Se alcanzó o excedió el valor límite')  
#Fin del programa
```

#### **SUGERENCIAS DIDÁCTICAS**

Desarrollar programas que resuelvan problemas similares al programa #2. Los siguientes son problemas sugeridos, pero cada estudiante podrá plantearse otros que sean de su interés.

Diseñe, Codifique y Pruebe programas para:

Ejercicio 4.1 Calcular el valor de 2 cargas dadas la fuerza de atracción o repulsión entre

ellas, la distancia que las separa y la relación entre las dos magnitudes de las cargas (por ejemplo, carga 1 tres veces mayor que carga 2). Siempre y cuando los valores de las cargas sean múltiplos de la carga elemental. \*

Ejercicio 4.2 Calcular cual de 2 frecuencias dadas se aproxima más a la frecuencia resonante de un circuito LC serie. \*\*

Ejercicio 4.3 Dados los valores de los coeficientes de la ecuación característica de un sistema, determinar el tipo de respuesta temporal que tendrá el sistema a una entrada escalón. \*\*\*

Notas:

(\*) Ningún valor de carga eléctrica puede ser menor al de la carga del electrón (carga elemental,  $e = 1.602176487 \times 10^{-19} C$ ). Cuando se calcula el valor de una carga, siempre debe ser múltiplo de  $e$ . (Ver Ejercicios 1.1 y 1.2)

(\*\*) En un circuito LC serie, a cierta frecuencia las impedancias inductiva y capacitiva se hacen iguales. Esto es que la impedancia total resulta ser nula, y se dice entonces que el circuito entró en resonancia. Como  $X_L = j\omega L$  (impedancia inductiva) y  $X_C = \frac{1}{j\omega C}$  (impedancia capacitiva), entonces la frecuencia resonante se obtiene de igualar las impedancias (valor absoluto) dando entonces  $\omega_R = \frac{1}{2\pi\sqrt{LC}}$

(\*\*\*) La función de transferencia es un modelo matemático de un sistema físico que permite analizar el comportamiento del mismo.

Función de Transferencia de un sistema de 2º orden:

$$T(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} = \frac{K_1}{s^2 + K_2 s + K_1}$$

El valor del parámetro  $\zeta$  (factor de amortiguamiento) determina la forma de la respuesta temporal del sistema, que puede ser clasificada como subamortiguada ( $\zeta < 1$ ), sobreamortiguada ( $\zeta > 1$ ) o críticamente amortiguada ( $\zeta = 1$ ).



**REPORTE DEL ALUMNO**

<b>Criterio para calificar</b>	<b>Si</b>	<b>No</b>
El documento tiene portada	10%	0
Entregó en fecha programada	10%	0
El documento está completo	20%	0
El programa funciona correctamente	20%	0
Incluye un ejercicio adicional (al menos)	20%	0
Incluye resultados y/o gráficas o tablas	20%	0

**BIBLIOGRAFÍA**

Raspberry Pi Foundation. Disponible en [www.raspberrypi.org](http://www.raspberrypi.org)

Lenguaje Python. Disponible en [www.python.org](http://www.python.org)

Potencia consumida por una resistencia. Young, Hugh D. y Freedman, Roger A. *Física Universitaria con Física Moderna*, Vol. 2. XII Edición. Pearson Educación.

## PRACTICA #5

### ESTRUCTURA DE REPETICIÓN

Estructura básica de un programa en Python

Entrada / Salida de datos por consola

Estructura if-else, Estructura for

Operaciones : multiplicación y división aritméticas, comparación

Tipos de datos: Enteros, Punto flotante, cadenas de caracteres

### COMPETENCIAS A DESARROLLAR

#### **Específicas:**

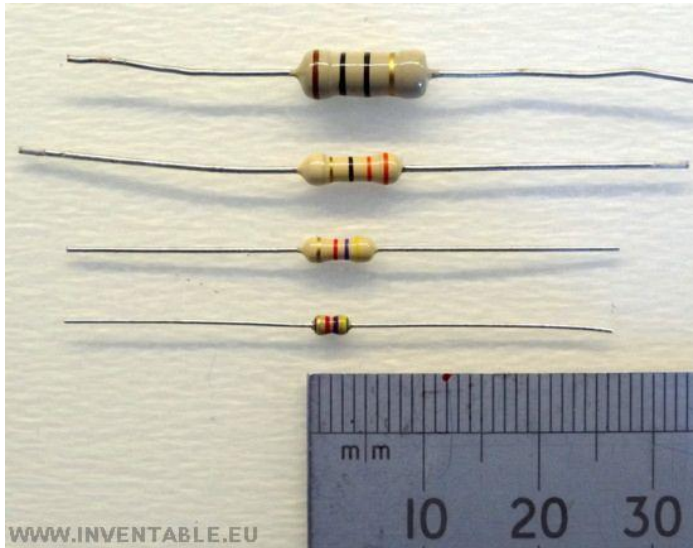
Utiliza adecuadamente las herramientas de diseño para el desarrollo de programas. Implementa programas a partir del diseño de algoritmos de aplicaciones básicas en la solución de problemas reales.

**Genéricas:** Capacidad creativa, Capacidad cognitiva para comprender y manipular ideas, pensamientos e información, Capacidad para aprender, Capacidad crítica y autocrítica, Capacidad de investigación, Capacidad para tomar decisiones.

### INTRODUCCIÓN

La potencia máxima que puede soportar una resistencia es aquella potencia para la cual el valor de la resistencia se mantiene inalterado a lo largo del tiempo. Cuanto más se calienta una resistencia, más peligro corre que su valor se vea modificado a lo largo de su vida útil.

La idea básica debe ser calcular la potencia de disipación de una determinada resistencia y sobredimensionarla lo suficiente para que su valor permanezca invariable con el paso de los años. Los fabricantes ya han tenido en cuenta esto e indican dicha potencia de disipación para que esta nunca se sobrepase.



La potencia de trabajo de una resistencia común generalmente no se encuentra indicada en el cuerpo de esta, pero podemos conocerla con una cierta aproximación en base a sus dimensiones.

En las resistencias con dos terminales y bandas de colores, las potencias disponibles van desde  $1/8W$  (1 octavo de watt o también  $0,125W$ ) hasta varios watts. Las más usadas son las de  $1/4 W$  ( $0,25W$ ).

### **CORRELACIÓN CON LOS TEMAS Y SUBTEMAS DEL PROGRAMA VIGENTE**

Esta práctica se relaciona directamente con los temas de la unidad 2. El propósito es que el estudiante se familiarice con el entorno de programación por medio del diseño, edición y compilación de programas sencillos. Que investigue cuales son las palabras reservadas del lenguaje utilizado, que desarrolle programas que utilicen variables, constantes y los diferentes tipos de datos. También que desarrolle programas que utilicen los operadores aritméticos, lógicos y relacionales. Asimismo, que desarrolle programas que utilicen la mayoría de los puntos vistos en la estructura básica de un programa con la finalidad de dar solución a problemas cotidianos.

### **MEDIDAS DE SEGURIDAD E HIGIENE**

Son obligaciones de los usuarios del laboratorio:

- I. Para realizar la práctica el estudiante debe llevar el material necesario indicado por el Profesor;
- II. Minimizar el uso de celulares al interior del Laboratorio;

- III. Atender puntualmente las indicaciones que le dé su Profesor y el personal de laboratorio;
- IV. Mantener el orden y disciplina dentro de las instalaciones del laboratorio;
- V. Informar inmediatamente al Profesor y al Jefe de Laboratorio si ocurre un accidente;
- VI. Acatar las disposiciones de seguridad y preservar la higiene del laboratorio.

A los usuarios del laboratorio se les prohíbe:

- I. Introducir alimentos. Las bebidas permitidas como agua o refrescos no deben dejarse cerca de los circuitos o equipos eléctricos;
- II. Introducir mascotas, juguetes u otros objetos ajenos a las prácticas o al trabajo de laboratorio;
- III. Fumar, correr, gritar, jugar, empujar, tocar instrumentos musicales, usar radios o teléfonos, que alteren el orden;
- IV. Usar teléfonos móviles, radio localizadores, reproductores de música u otros aparatos ajenos a las prácticas de laboratorio;
- V. Usar las instalaciones de los laboratorios para otros propósitos, diferentes del desarrollo de prácticas programadas;
- VI. Incurrir en cualquier tipo de comportamiento que ponga en riesgo la integridad de las instalaciones, equipo de laboratorio, los usuarios, y jefe de laboratorio;
- VII. Introducir personas ajenas a las autorizadas para las prácticas.

## **MATERIAL Y EQUIPO NECESARIO**

1. Paquete de Raspberry Pi (Tarjeta Raspberry Pi, Fuente de poder, Tarjeta micro SD con sistema operativo Raspbian, Teclado, ratón y monitor).
2. Computadora personal con Intérprete de Python.

## **METODOLOGÍA**

### **1. Planteamiento del problema:**

Diseñe un programa que calcule N valores de corriente y potencia disipada por un voltaje aplicado a una resistencia. Al introducir los datos deberá evitarse la introducción de un divisor cero, en cuyo caso se escribirá un mensaje de error y se pasará al siguiente par de valores.

## 2. Algoritmo

ALGORITMO: Corriente y potencia disipados por 10 resistencias

DESCRIPCIÓN: Se calcula la corriente y la potencia dados un conjunto de pares de resistencia-voltaje, evitando la división por cero.

CONSTANTES: -----

VARIABLES:

*real* :  $V$  Valor de voltaje

*entero* :  $R$  Valor de resistencia

*real* :  $I, P$  Valores de corriente y potencia

INICIO

1. Para  $i = 1$  hasta  $i = 10$ ,  $i = i + 1$

1.1 Escribir("Dar valor de voltaje")

1.2 Leer ( $V$ )

1.3 Escribir("Dar valor de resistencia")

1.4 Leer ( $R$ )

1.5 Si  $R = 0$  entonces

Escribir("Valor inaceptable")

Otro

1.51  $I \leftarrow \frac{V}{R}$

1.52  $P \leftarrow V \cdot I$

1.53 Escribir( $I, P$ )

Fin Para

FIN

### 3. Programa

```
#Programa 3  
#Calcula 10 valores de corriente y potencia, evita división por cero.  
for i in range(1, 11):  
    voltaje = float(input('Introduce valor del voltaje aplicado (Voltios)'))  
    resistencia = float(input('Introduce valor de la resistencia (Ohmios)'))  
    if resistencia == 0.0 :  
        print('No es valor aceptable de la resistencia ')  
    else:  
        corriente = voltaje / resistencia  
        potencia = voltaje * corriente  
print('Corriente(', i, ') = ', corriente, 'Amperes')  
print('Potencia(', i, ') = ', potencia, 'Watts')  
#Fin del programa
```

## SUGERENCIAS DIDÁCTICAS

Desarrollar programas que resuelvan problemas similares al programa #3. Los siguientes son problemas sugeridos, pero cada estudiante podrá plantearse otros que sean de su interés.

Diseñe, Codifique y Pruebe programas para:

Ejercicio 5.1 El problema planteado en el ejercicio 1.3

Ejercicio 5.2 El problema planteado en el ejercicio 1.6

Ejercicio 5.3 El problema planteado en el ejercicio 1.7

## REPORTE DEL ALUMNO

Criterio para calificar	Si	No
El documento tiene portada	10%	0
Entregó en fecha programada	10%	0
El documento está completo	20%	0
El programa funciona correctamente	20%	0
Incluye un ejercicio adicional (al menos)	20%	0
Incluye resultados y/o gráficas o tablas	20%	0

## BIBLIOGRAFÍA

Raspberry Pi Foundation. Disponible en [www.raspberrypi.org](http://www.raspberrypi.org)

Lenguaje Python. Disponible en [www.python.org](http://www.python.org)

Potencia en resistencias. Young, Hugh D. y Freedman, Roger A. *Física Universitaria con Física Moderna*, Vol. 2. XII Edición. Pearson Educación.

## PRACTICA #6

### ESTRUCTURA “WHILE”

Estructura básica de un programa en Python

Entrada / Salida de datos por consola

Estructura if-else, Estructura while

Operaciones : incremento, suma y división aritméticas, comparación

Tipos de datos: Enteros, Punto flotante, cadenas de caracteres

### COMPETENCIAS A DESARROLLAR

#### Específicas:

Utilizar adecuadamente las herramientas de diseño para el desarrollo de programas. Implementa programas a partir del diseño de algoritmos de aplicaciones básicas en la solución de problemas reales.

**Genéricas:** Capacidad creativa, Capacidad cognitiva para comprender y manipular ideas, pensamientos e información, Capacidad para aprender, Capacidad crítica y autocrítica, Capacidad de investigación, Capacidad para tomar decisiones.

### INTRODUCCIÓN

Las medidas de resistencia de un lote de resistencias son:

100.1Ω 101.0Ω 100.6Ω 99.5Ω 99.0Ω  
100.0Ω 99.5Ω 100.4Ω 99.8Ω 100.1Ω

Asumiendo que solamente hay errores aleatorios, ¿a)Cuál es la **media aritmética**, b) la **desviación estándar** y c) la **probabilidad de error**?

Respuesta:

La media aritmética es:  $\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i = \frac{1000.0}{10} = 100.0$

La desviación estándar es:  $\sigma = \sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} = 3.08$

La probabilidad de error es:  $\sqrt{\frac{\sigma}{n-1}} = 0.58$



## **CORRELACIÓN CON LOS TEMAS Y SUBTEMAS DEL PROGRAMA VIGENTE**

Esta práctica se relaciona directamente con los temas de la unidad 2. El propósito es que el estudiante se familiarice con el entorno de programación por medio del diseño, edición y compilación de programas sencillos. Que investigue cuales son las palabras reservadas del lenguaje utilizado, que desarrolle programas que utilicen variables, constantes y los diferentes tipos de datos. También que desarrolle programas que utilicen los operadores aritméticos, lógicos y relacionales. Asimismo, que desarrolle programas que utilicen la mayoría de los puntos vistos en la estructura básica de un programa con la finalidad de dar solución a problemas cotidianos.

## **MEDIDAS DE SEGURIDAD E HIGIENE**

Son obligaciones de los usuarios del laboratorio:

- I. Para realizar la práctica el estudiante debe llevar el material necesario indicado por el Profesor;
- II. Minimizar el uso de celulares al interior del Laboratorio;
- III. Atender puntualmente las indicaciones que le dé su Profesor y el personal de laboratorio;
- IV. Mantener el orden y disciplina dentro de las instalaciones del laboratorio;
- V. Informar inmediatamente al Profesor y al Jefe de Laboratorio si ocurre un accidente;
- VI. Acatar las disposiciones de seguridad y preservar la higiene del laboratorio.

A los usuarios del laboratorio se les prohíbe:

- I. Introducir alimentos. Las bebidas permitidas como agua o refrescos no deben dejarse cerca de los circuitos o equipos eléctricos;
- II. Introducir mascotas, juguetes u otros objetos ajenos a las prácticas o al trabajo de laboratorio;
- III. Fumar, correr, gritar, jugar, empujar, tocar instrumentos musicales, usar radios o teléfonos, que alteren el orden;
- IV. Usar teléfonos móviles, radio localizadores, reproductores de música u otros aparatos ajenos a las prácticas de laboratorio;
- V. Usar las instalaciones de los laboratorios para otros propósitos, diferentes del desarrollo de prácticas programadas;
- VI. Incurrir en cualquier tipo de comportamiento que ponga en riesgo la integridad de las instalaciones, equipo de laboratorio, los usuarios, y jefe de laboratorio;
- VII. Introducir personas ajenas a las autorizadas para las prácticas.

## MATERIAL Y EQUIPO NECESARIO

1. Paquete de Raspberry Pi (Tarjeta Raspberry Pi, Fuente de poder, Tarjeta micro SD con sistema operativo Raspbian, Teclado, ratón y monitor).
2. Computadora personal con Intérprete de Python.

## METODOLOGÍA

### 1. Planteamiento del problema:

Calcular las características de un conjunto de mediciones de un lote de resistencias (tamaño indeterminado, la recolección de datos termina cuando se lee un valor de resistencia cero). Se deberá reportar el tamaño del lote, el promedio de los valores, el valor máximo y el valor mínimo encontrados.

### 2. Algoritmo

*ALGORITMO: Mínimo, máximo y promedio de N valores*

*DESCRIPCIÓN: Se calculan los valores mínimo, máximo y promedio de N resistencias.*

*CONSTANTES: -----*

*VARIABLES:*

*real : R, Min, Max, Total, Prom Valor de resistencia, Valores mínimo, máximo , Total  
(acumulador)y promedio*

*entero: Contador Contador de valores leídos*

*INICIO*

1.  $Contador \leftarrow 0$ ,  $Total \leftarrow 0$ ,  $max \leftarrow 0$ ,  $min \leftarrow 1e10$ ,  $Prom \leftarrow 0$
2. Leer("Dar valor de resistencia ", R)
3. Mientras  $R \neq 0$  repetir:
  - 3.1  $Contador \leftarrow Contador + 1$
  - 3.2  $Total \leftarrow Total + R$
  - 3.3 Si  $Max < R$  entonces  $max \leftarrow R$
  - 3.4 Si  $Min > R$  entonces  $min \leftarrow R$

*Fin mientras*

4.  $Prom \leftarrow Total / Contador$
5. Escribe (Contador)
6. Escribe (Prom)
7. Escribe (Max)
8. Escribe (Min)

*FIN*

### 3. Programa

```
#Programa 4
#Calcula minimo, máximo y promedio de N valores de #resistencias.
Contador = 0
Minimo = 1e10
Maximo = 0
Total = 0
Resistencia = int(input('Ingrese valor de resistencia '))
while Resistencia != 0 :
    Contador = Contador +1
    Total = Total + Resistencia
    if Maximo < Resistencia:
        Maximo = Resistencia
    if Minimo > Resistencia:
        Minimo = Resistencia
    Resistencia = int(input('Ingrese valor de resistencia '))
#Fin while
Media = Total / Contador
print('Numero total de piezas = ',Contador)
print('Promedio de las mediciones = ', Media)
print('Valor máximo de resistencia = ',Maximo)
print('Valor mínimo de resistencia = ',Minimo)
#Fin del programa
```

## SUGERENCIAS DIDÁCTICAS

Desarrollar programas que resuelvan problemas similares al programa #4. Los siguientes son problemas sugeridos, pero cada estudiante podrá plantearse otros que sean de su interés.

Diseñe, Codifique y Pruebe programas para:

Ejercicio 6.1 Resolver completo el ejemplo descrito en la Introducción de esta práctica.

Ejercicio 6.2 El problema planteado en el Ejercicio 1.8

Ejercicio 6.3 Calcule las corrientes en un circuito resistivo como el de la figura 6.1, introduciendo los valores de los parámetros uno por uno. Se sugiere utilizar el método de Matrices para resolver el sistema de ecuaciones lineales.

## REPORTE DEL ALUMNO

Criterio para calificar	Si	No
El documento tiene portada	10%	0
Entregó en fecha programada	10%	0
El documento está completo	20%	0
El programa funciona correctamente	20%	0
Incluye un ejercicio adicional (al menos)	20%	0
Incluye resultados y/o gráficas o tablas	20%	0

## BIBLIOGRAFÍA

Raspberry Pi Foundation. Disponible en [www.raspberrypi.org](http://www.raspberrypi.org)

Lenguaje Python. Disponible en [www.python.org](http://www.python.org)

Wolf, Stanley y Smith, Richard M. Guía para las Mediciones Electrónicas y Prácticas de Laboratorio. Prentice Hall Hispanoamericana. (problema)

Chapra, Steven C. y Canale, Raymond P. Métodos Numéricos para Ingenieros. 5ª Edición. McGrawHill. (Ejercicio 6.3)

## PRACTICA #7

### ARREGLOS Y MATRICES

Estructura básica de un programa en Python

Entrada / Salida de datos por consola

Estructura if-else, Estructura while, Arreglos y Matrices

Operaciones : incremento, suma y división aritméticas, comparación

Tipos de datos: Enteros, Punto flotante, cadenas de caracteres

### COMPETENCIAS A DESARROLLAR

#### Específicas:

Utilizar adecuadamente las herramientas de diseño para el desarrollo de programas. Implementa programas a partir del diseño de algoritmos de aplicaciones básicas en la solución de problemas reales.

**Genéricas:** Capacidad creativa, Capacidad cognitiva para comprender y manipular ideas, pensamientos e información, Capacidad para aprender, Capacidad crítica y autocrítica, Capacidad de investigación, Capacidad para tomar decisiones.

### INTRODUCCIÓN

Los fundamentos del análisis de circuitos eléctricos se deben a Gustav Kirchoff, un profesor universitario alemán de hace aproximadamente 2 siglos. Gracias a sus experimentos desarrolló las leyes que llevan su nombre. En lo que sigue se llamará **rama** a una parte de un circuito con dos terminales, **nodo** al punto donde dos o más ramas se conectan y **lazo** a una trayectoria cerrada formada por varias ramas.

Ley de corrientes de Kirchoff: Si se miden las corrientes que fluyen hacia o desde un nodo, la suma de las corrientes que entran será siempre igual a la suma de las corrientes que salen. Si se asigna por convención signos opuestos a las corrientes que entran y salen, la suma algebraica será entonces siempre cero.

$$\sum i = 0 = i_1 + i_2 + i_3 + \dots + i_n$$

Ley de voltajes de Kirchoff: Si se miden las caídas de potencial a lo largo de un lazo, la suma será siempre cero; si se asigna por convención signos opuestos a los potenciales producidos por corrientes opuestas en la trayectoria seguida a lo largo del lazo.

$$\sum v = 0 = v_1 + v_2 + v_3 + \dots + v_m$$

En un circuito donde existan múltiples lazos, se podrá establecer una ecuación por cada lazo aplicando la ley de voltajes de Kirchoff y otras ecuaciones aplicando a cada nodo la ley de corrientes. Una vez obtenidas tantas ecuaciones independientes, como incógnitas existan, se podrán resolver utilizando alguno de los diversos métodos disponibles.

El método de Determinantes es conveniente ya que es sistemático y general; puede ser usado para resolver grandes y complicados circuitos.

Ejemplo:

En el circuito de la figura 7.1 las ecuaciones de malla son:

$$\begin{aligned} -i_{54}R_{54} - i_{43}R_{43} - i_{32}R_{32} - i_{52}R_{52} &= 0 \\ -i_{65}R_{65} - i_{52}R_{52} - i_{12}R_{12} - 200 &= 0 \end{aligned}$$

Las ecuaciones de nodos son:

$$\begin{aligned} i_{12} + i_{52} + i_{32} &= 0 \\ i_{65} - i_{52} - i_{54} &= 0 \\ i_{43} - i_{32} &= 0 \\ i_{54} - i_{43} &= 0 \end{aligned}$$

Re-escribiendo estas ecuaciones en forma matricial, queda:

$$\begin{bmatrix} +01 & +01 & +01 & +00 & +00 & +00 \\ +00 & -01 & +00 & +01 & -01 & +00 \\ +00 & +00 & -01 & +00 & +00 & +01 \\ +00 & +00 & +00 & +00 & +01 & -01 \\ +00 & +10 & -10 & +00 & -15 & -05 \\ +05 & -10 & +00 & -20 & +00 & +00 \end{bmatrix} \cdot \begin{bmatrix} i_{12} \\ i_{52} \\ i_{32} \\ i_{65} \\ i_{54} \\ i_{43} \end{bmatrix} = \begin{bmatrix} 000 \\ 000 \\ 000 \\ 000 \\ 000 \\ 200 \end{bmatrix}$$

Para resolver y encontrar los valores de las corrientes, basta aplicar algebra elemental de matrices, considerando  $A = \text{Matriz de coeficientes}$ ,  $I = \text{Matriz de corrientes}$  y  $X = \text{Matriz de voltajes}$ :

$$A \cdot I = X, \text{ por lo tanto: } I = \frac{X}{A} = X \cdot A^{-1}$$

El conjunto de ecuaciones también se puede resolver por otros métodos, como el de eliminación, aunque resulta poco práctico. En cambio, el apoyo computacional nos proporciona una solución rápida y eficaz.

## **CORRELACIÓN CON LOS TEMAS Y SUBTEMAS DEL PROGRAMA VIGENTE**

Esta práctica se relaciona directamente con los temas de la unidad 2. El propósito es que el estudiante se familiarice con el entorno de programación por medio del diseño, edición y compilación de programas sencillos. Que investigue cuales son las palabras reservadas del lenguaje utilizado, que desarrolle programas que utilicen variables, constantes y los diferentes tipos de datos. También que desarrolle programas que utilicen los operadores aritméticos, lógicos y relacionales. Asimismo, que desarrolle programas que utilicen la mayoría de los puntos vistos en la estructura básica de un programa con la finalidad de dar solución a problemas cotidianos.

## **MEDIDAS DE SEGURIDAD E HIGIENE**

Son obligaciones de los usuarios del laboratorio:

- I. Para realizar la práctica el estudiante debe llevar el material necesario indicado por el Profesor;
- II. Minimizar el uso de celulares al interior del Laboratorio;
- III. Atender puntualmente las indicaciones que le dé su Profesor y el personal de laboratorio;
- IV. Mantener el orden y disciplina dentro de las instalaciones del laboratorio;
- V. Informar inmediatamente al Profesor y al Jefe de Laboratorio si ocurre un accidente;
- VI. Acatar las disposiciones de seguridad y preservar la higiene del laboratorio.

A los usuarios del laboratorio se les prohíbe:

- I. Introducir alimentos. Las bebidas permitidas como agua o refrescos no deben dejarse cerca de los circuitos o equipos eléctricos;
- II. Introducir mascotas, juguetes u otros objetos ajenos a las prácticas o al trabajo de laboratorio;
- III. Fumar, correr, gritar, jugar, empujar, tocar instrumentos musicales, usar radios o teléfonos, que alteren el orden;
- IV. Usar teléfonos móviles, radio localizadores, reproductores de música u otros aparatos ajenos a las prácticas de laboratorio;
- V. Usar las instalaciones de los laboratorios para otros propósitos, diferentes del desarrollo de prácticas programadas;
- VI. Incurrir en cualquier tipo de comportamiento que ponga en riesgo la integridad de las instalaciones, equipo de laboratorio, los usuarios, y jefe de laboratorio;

Introducir personas ajenas a las autorizadas para las prácticas.

## MATERIAL Y EQUIPO NECESARIO

1. Paquete de Raspberry Pi (Tarjeta Raspberry Pi, Fuente de poder, Tarjeta micro SD con sistema operativo Raspbian, Teclado, ratón y monitor).
2. Computadora personal con Intérprete de Python.

## METODOLOGÍA

### 1. Planteamiento del problema:

Diseñe un programa que calcule las corrientes en un circuito resistivo como el de la figura 7.1, introduciendo los valores de los parámetros uno por uno. Se sugiere utilizar el método de Matrices para resolver el sistema de ecuaciones lineales.

### 2. Algoritmo

*ALGORITMO: Análisis de redes eléctricas.*

*DESCRIPCIÓN:*

*CONSTANTES:*

-----

*VARIABLES:*

*INICIO*

1. Leer número de ecuaciones
2. Leer matriz de coeficientes
3. Repetir
  - 3.1 Leer matriz de voltajes
  - 3.2 Calcular inversa de matriz A
  - 3.3 Calcular matriz de corrientes
  - 3.4 Escribir resultados
  - 3.5 Leer continuar?

*Hasta que continuar = 'no'*

*FIN*



### 3. Programa

```
#Ecuaciones Algebraicas Lineales
#Matriz de 6X6
import numpy as np
from numpy.linalg import inv

def dato(ecuacion):
    pos=ecuacion.index(',')
    valor=ecuacion[:pos+1]
    ec2=ecuacion.replace(valor,"1")
    valor2=ecuacion[:pos]
return ec2,valor2
```

```
num_ec = int(input('Numero de ecuaciones = '))
A = []
e = []
V=[]
for i in range(num_ec):
    e.append(None)
    V.append(None)

for i in range(num_ec):
    e.append(None)
for i in range(num_ec):
    A.append([])
    e[i]=input('Ecuación ')
    ecnueva=e[i]
    for j in range(num_ec):
        ecnueva,valor = dato(ecnueva)
        A[i].append(float(valor))

print(A)
rep='s'
while rep=='s':
    disc='n'
    while disc=='n':
        ecV=input('Voltajes = ')
        print('Es correcto (Si = s)? -----> ',ecV)
        disc=input()
```

**Continúa...**

```
for k in range(num_ec):
    ecnueva,numero=dato(ecV)
    V[k]=float(numero)
    ecV=ecnueva

A2=inv(A)
print('*'*40)
I=A2.dot(V)

print('Soluciones del sistema de ecuaciones ')
print(I)
print('*'*40)
print('Otro conjunto de valores de voltajes? ')
rep=input()
```

**SUGERENCIAS DIDÁCTICAS**

Desarrollar programas que resuelvan problemas similares al programa #5. Los siguientes son problemas sugeridos, pero cada estudiante podrá plantearse otros que sean de su interés.

Diseñe, Codifique y Pruebe programas para:

Ejercicio 7.1 Crear un catálogo de resistencias que contenga el valor en ohmios, la potencia en watts y la tolerancia en %.

Ejercicio 7.2 La Pila es una zona de los registros de segmento de memoria que la unidad aritmética y lógica utiliza para almacenar temporalmente los datos que está manipulando. Cuando la cantidad de datos a manejar es demasiado grande u otras necesidades del proceso impiden que estos datos puedan almacenarse en los registros creados para ello se envían a la pila, donde se almacenan hasta que la unidad de control recupera la información para que la procese la unidad aritmética y lógica.

La ventaja de manejar una pila como almacén de información es que la información que se guarda en ella tiene que entrar y salir, obligatoriamente, por una sola dirección de memoria. Esto permite que la unidad de control no necesite conocer más que esa dirección para poder manejar los datos almacenados en la pila. Escriba un programa que simule la operación de una pila de datos.

### REPORTE DEL ALUMNO

Criterio para calificar	Si	No
El documento tiene portada	10%	0
Entregó en fecha programada	10%	0
El documento está completo	20%	0
El programa funciona correctamente	20%	0
Incluye un ejercicio adicional (al menos)	20%	0
Incluye resultados y/o gráficas o tablas	20%	0

### BIBLIOGRAFÍA

Raspberry Pi Foundation. Disponible en [www.raspberrypi.org](http://www.raspberrypi.org)

Lenguaje Python. Disponible en [www.python.org](http://www.python.org)

Análisis de circuitos.

1. Smith, Ralph J., Circuits, Devices And Systems, A First Course in ]Electrical Engineering, John Wiley & Sons, 1973.
2. Floyd, Thomas L., Electric Circuits Fundamentals, Pearson Education, 2004.

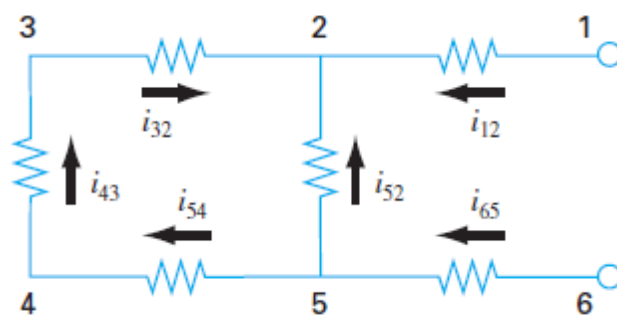
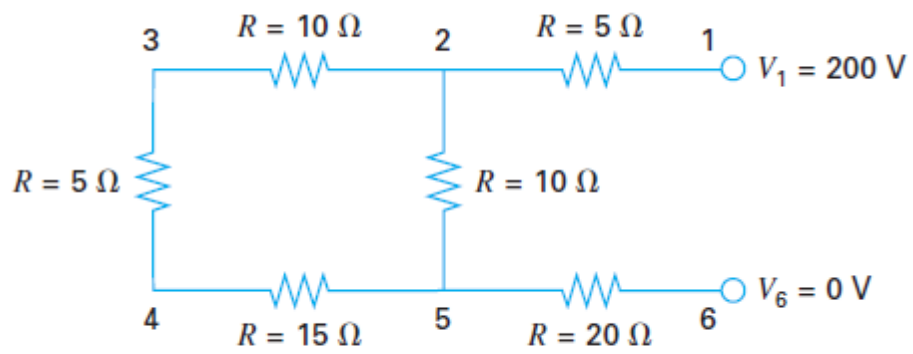


Figura 7.1

## PRACTICA #8

### IMPEDANCIAS RLC

Programa estructurado en Python

Entrada / Salida de datos por consola

Estructura Función, if, while

Librerías

Operaciones : suma, resta, multiplicación y división aritméticas, elevación a potencia, comparación

Tipos de datos: Enteros, Punto flotante, cadenas de caracteres

### COMPETENCIAS A DESARROLLAR

#### Específicas:

Utilizar adecuadamente las herramientas de diseño para el desarrollo de programas.

Utilizar la metodología de la programación modular para desarrollar programas estructurados y simplificar el mantenimiento del código.

Desarrollar programas a partir del diseño de algoritmos, para su aplicación en la solución de problemas.

**Genéricas:** Capacidad creativa, Capacidad cognitiva para comprender y manipular ideas, pensamientos e información, Capacidad para aprender, Capacidad crítica y autocrítica, Capacidad de investigación, Capacidad para tomar decisiones.

### INTRODUCCIÓN

En un circuito RLC serie al que se aplica un voltaje senoidal  $v(t) = V_{\max} \text{sen} \omega t$ , la corriente será también senoidal pero con un ángulo de desfaseamiento  $i(t) = I_{\max} \text{sen}(\omega t + \phi)$

El valor de  $I_{\max}$  y de  $\phi$  se calculan fácilmente usando fasores:

Impedancia de la resistencia =  $Z_R = R$

Impedancia del capacitor =  $Z_C = \frac{1}{j\omega C} = -\frac{j}{\omega C}$

Impedancia de la inductancia =  $Z_L = j\omega L$



Impedancia total =  $Z$

$$Z = Z_R + Z_L + Z_C$$

$$I_{\max} = \frac{V_{\max}}{Z}$$

$$\phi = \text{tg}^{-1} \frac{Z_L + Z_C}{Z_R}$$

## **CORRELACIÓN CON LOS TEMAS Y SUBTEMAS DEL PROGRAMA VIGENTE**

Esta práctica se relaciona directamente con los temas de la unidad 3. El propósito es que el estudiante reconozca las ventajas de la programación estructurada y se acostumbre al pensamiento “arriba-abajo” (top down design), aplicando estas técnicas en la solución de problemas cada vez más grandes.

## **MEDIDAS DE SEGURIDAD E HIGIENE**

Son obligaciones de los usuarios del laboratorio:

- I. Para realizar la práctica el estudiante debe llevar el material necesario indicado por el Profesor;
- II. Minimizar el uso de celulares al interior del Laboratorio;
- III. Atender puntualmente las indicaciones que le dé su Profesor y el personal de laboratorio;
- IV. Mantener el orden y disciplina dentro de las instalaciones del laboratorio;
- V. Informar inmediatamente al Profesor y al Jefe de Laboratorio si ocurre un accidente;
- VI. Acatar las disposiciones de seguridad y preservar la higiene del laboratorio.

A los usuarios del laboratorio se les prohíbe:

- I. Introducir alimentos. Las bebidas permitidas como agua o refrescos no deben dejarse cerca de los circuitos o equipos eléctricos;
- II. Introducir mascotas, juguetes u otros objetos ajenos a las prácticas o al trabajo de laboratorio;
- III. Fumar, correr, gritar, jugar, empujar, tocar instrumentos musicales, usar radios o teléfonos, que alteren el orden;
- IV. Usar teléfonos móviles, radio localizadores, reproductores de música u otros aparatos ajenos a las prácticas de laboratorio;
- V. Usar las instalaciones de los laboratorios para otros propósitos, diferentes del desarrollo de prácticas programadas;
- VI. Incurrir en cualquier tipo de comportamiento que ponga en riesgo la integridad de las instalaciones, equipo de laboratorio, los usuarios, y jefe de laboratorio;
- VII. Introducir personas ajenas a las autorizadas para las prácticas;

## MATERIAL Y EQUIPO NECESARIO

1. Paquete de Raspberry Pi (Tarjeta Raspberry Pi, Fuente de poder, Tarjeta micro SD con sistema operativo Raspbian, Teclado, ratón y monitor), o...
2. Computadora personal con Intérprete de Python.

## METODOLOGÍA

### 1. Planteamiento del problema:

Diseñe un programa estructurado con las funciones pertinentes para que dado un circuito RLC serie, calcule:

- a) Las impedancias de cada componente, b) La impedancia total del circuito.
- c) La corriente instantánea y RMS, d) El ángulo de fase entre el voltaje y la corriente

### 2. Algoritmo

ALGORITMO: Impedancias de un circuito RLC

DESCRIPCIÓN: Se calculan las impedancias de los componentes de un circuito RLC. También la corriente instantánea y RMS así como el ángulo de fase entre el voltaje y la corriente.

CONSTANTES: -----

VARIABLES:

*real* :  $R, L, C$  Valores de Resistencia, Inductancia y Capacitancia

*entero* :  $V, Frec$  Valores de voltaje y frecuencia

*real* :  $Z_R, Z_L, Z_C, Z_T$  Impedancias

*real* :  $I_{max}, Fase$  Corriente y Fase



## Funciones

*Función Lectura\_de\_datos:*

1.1 Leer("Valor de resistencia", R)

1.2 Leer("Valor de inductancia", L)

1.3 Leer ("Valor de capacitancia", C)

1.4 Leer ("Valor de voltaje", V)

1.5 Leer ("Valor de frecuencia", Frec)

*Fin función*

*Función impedancia (valor, dispositivo)*

2.1 Si dispositivo =" R" entonces

*resultado* ← *valor*

2.2 Si dispositivo =" L" entonces

*resultado* ←  $\text{valor} * 2 * \pi * \text{Frec}$

2.3 Si dispositivo =" C" entonces

$\text{resultado} \leftarrow \frac{1}{\text{valor} * 2 * \pi * \text{Frec}}$

*Fin función*

*Función impedancia total serie*

$$3.1 \text{ resultado} \leftarrow \sqrt{((ZR)^2 + (ZL - ZC)^2)}$$

*Fin función*

*Función corriente(V, ZT)*

$$4.1 \text{ resultado} \leftarrow \frac{V}{ZT}$$

*Fin función*

*Función fase (ZL, ZC, ZR)*

$$5.1 \text{ resultado} \leftarrow \text{tg}^{-1} \frac{ZL - ZC}{ZR}$$

*Fin función*

*Parte principal del algoritmo**INICIO*

1. *Continua* ← "si"
2. *Hasta que Continua* ≠ "si" repetir :
  - 2.1 *Lectura\_de\_datos*
  - 2.2  $ZR \leftarrow \text{fimpedancia}(\text{resistencia}, "R")$
  - 2.3  $ZL \leftarrow \text{fimpedancia}(\text{bobina}, "L")$
  - 2.4  $ZC \leftarrow \text{fimpedancia}(\text{capacitor}, "C")$
  - 2.5  $ZT \leftarrow \text{fimpedancia\_total\_serie}(ZR, ZL, ZC)$
  - 2.6  $\text{Fase} \leftarrow \text{ffase}(ZR, ZL, ZC)$
  - 2.7  $I\_max \leftarrow \text{fcorriente}(V, ZT)$
  - 2.8 *Escribir*(*ZR*, *ZL*, *ZC*, *ZT*, *Fase*)
  - 2.9 *Escribir*("Continuar ?")
  - 2.10 *Leer* (*Continua*)

Fin Hasta

*FIN*

### 3. Programa

```
#Programa 6
#Programa para resolver un circuito RLC serie
#Usando funciones
import math
def lectura_datos():
    Res=int(input('Resistencia (Ohms) = '))
    Ind=int(input('Inductancia (miliHenrios) = '))
    Cap=float(input('Capacitancia (microFaradios) = '))
    Volt=int(input('Voltaje de alimentacion (Voltios) = '))
    Frec=int(input('Frecuencia (Hertz) = '))
    return Res,Ind,Cap,Volt,Frec

def impedancia(valor,disp):
    if disp=='R':
        resultado=valor
    if disp=='L':
        resultado=valor*w*1e-3
    if disp=='C':
        resultado=1/(w*valor*1e-6)
    return resultado
```

Continua...

```
def impedancia_serie(R, L, C):  
    resultado=(R**2+(L-C)**2)**0.5  
    return resultado  
  
def corriente_fase(Vol, ZR, ZL, ZC):  
    resultado1=1000*Vol/ZT  
    resultado2=math.degrees(math.atan((ZL-ZC)/Zr))  
    return resultado1,resultado2  
  
preg='si'  
while preg=='si':  
    Res,Ind,Cap,Volt,Frec=lectura_datos()  
    w=2*math.pi*Frec  
    Zr=impedancia(Res,'R')  
    ZL=impedancia(Ind,'L')  
    ZC=impedancia(Cap,'C')  
    ZT=impedancia_serie(Zr,ZL,ZC)  
    Imax,Fase=corriente_fase(Volt,Zr,ZL,ZC)  
    print('Impedancia total = ',ZT,'Ohms')  
    print('Corriente maxima = ',Imax,'miliAmperes')  
    print('Angulo de fase = ',Fase,'grados')  
    preg=input('Continuar? (si/no) ')  
print('Fin de programa')  
#Fin del programa
```

## SUGERENCIAS DIDÁCTICAS

Desarrollar programas que resuelvan problemas similares al programa #5. Los siguientes son problemas sugeridos, pero cada estudiante podrá plantearse otros que sean de su interés.

Diseñe, Codifique y Pruebe programas para:

Ejercicio 8.1 Dados un conjunto de resistencias calcular:

- a) El equivalente serie    b) El equivalente paralelo
- c) La corriente y voltajes en cada resistencia serie para un voltaje dado.
- d) El voltaje y las corrientes en cada resistencia paralelo para un voltaje dado.
- e) Otra(s) opciones

Ejercicio 8.2 Diseñe un amperímetro con escalas de medición de 5, 10, 20 y 50 mA en

base a un Galvanómetro de 1mA a plena escala y 20 ohms de resistencia interna. Repita el diseño para convertir el galvanómetro anterior a un voltímetro con escalas de 10, 20 y 50 voltios.

## REPORTE DEL ALUMNO

Criterio para calificar	Si	No
El documento tiene portada	10%	0
Entregó en fecha programada	10%	0
El documento está completo	20%	0
El programa funciona correctamente	20%	0
Incluye un ejercicio adicional (al menos)	20%	0
Incluye resultados y/o gráficas o tablas	20%	0

## BIBLIOGRAFÍA

Raspberry Pi Foundation. Disponible en [www.raspberrypi.org](http://www.raspberrypi.org)

Lenguaje Python. Disponible en [www.python.org](http://www.python.org)

Fasores y Corrientes alternas. Young, Hugh D. y Freedman, Roger A. *Física Universitaria con Física Moderna*, Vol. 2. XII Edición. Pearson Educación.

Instrumentos de medición eléctrica. Wolf, Stanley y Smith, Richard M. Guía para las Mediciones Electrónicas y Prácticas de Laboratorio. Prentice Hall Hispanoamericana.

## PRACTICA #9

### PORCENTAJE DE ENERGÍA DISIPADA RLC

Programa estructurado en Python

Entrada / Salida de datos por consola

Estructura Función, if, while

Librerías

Operaciones : suma, resta, multiplicación y división aritméticas, elevación a potencia, comparación

Tipos de datos: Enteros, Punto flotante, cadenas de caracteres

### COMPETENCIAS A DESARROLLAR

#### Específicas:

Utilizar adecuadamente las herramientas de diseño para el desarrollo de programas.

Utilizar la metodología de la programación modular para desarrollar programas estructurados y simplificar el mantenimiento del código.

Desarrollar programas a partir del diseño de algoritmos, para su aplicación en la solución de problemas.

**Genéricas:** Capacidad creativa, Capacidad cognitiva para comprender y manipular ideas, pensamientos e información, Capacidad para aprender, Capacidad crítica y autocrítica, Capacidad de investigación, Capacidad para tomar decisiones.

### INTRODUCCIÓN

En un circuito RLC serie como el de la figura, el capacitor se carga inicialmente al voltaje de la batería ( $V_0$ ) y al mover el interruptor se establece un circuito serie RLC con una respuesta transitoria que desaparece después de cierto tiempo mientras la energía oscila entre la capacitancia y la inductancia, siendo la función de la resistencia encargarse de la disipación de la energía. La segunda ley de Kirchoff permite establecer la ecuación que describe el comportamiento anterior:

$$L \frac{di}{dt} + Ri + \frac{1}{C} \int i dt = 0 \quad \text{Como } i = \frac{dq}{dt}, \text{ la ecuación anterior se convierte en:}$$

$$L \frac{d^2q}{dt^2} + R \frac{dq}{dt} + \frac{1}{C} q = 0$$

La solución a la ecuación anterior está dada por (\*):

$$q(t) = q_0 e^{-Rt/(2L)} \cos \left[ \sqrt{\frac{1}{LC} - \left(\frac{R}{2L}\right)^2} t \right] \quad \text{por lo tanto:}$$

$$e^{-Rt/(2L)} \cos \left[ \sqrt{\frac{1}{LC} - \left(\frac{R}{2L}\right)^2} t \right] - \frac{q(t)}{q_0} = 0$$

Si en la ecuación anterior se sustituyen los valores dados de inductancia, capacitancia, tiempo de evaluación y valor disipado de energía, se requiere de algún método de aproximación numérica para calcular el valor de la resistencia.

Es importante notar que se debe cumplir la relación  $\frac{1}{LC} > \left(\frac{R}{2L}\right)^2$ , de lo contrario no existirá una solución real. Por lo tanto se tendrá un punto de partida para calcular el valor de la resistencia, es decir  $R_{\max} > R > 0$ , donde el valor máximo se obtiene despejando.

### **CORRELACIÓN CON LOS TEMAS Y SUBTEMAS DEL PROGRAMA VIGENTE**

Esta práctica se relaciona directamente con los temas de la unidad 3. El propósito es que el estudiante reconozca las ventajas de la programación estructurada y se acostumbre al pensamiento “arriba-abajo” (top down design), aplicando estas técnicas en la solución de problemas cada vez más grandes.

### **MEDIDAS DE SEGURIDAD E HIGIENE**

Son obligaciones de los usuarios del laboratorio:

- I. Para realizar la práctica el estudiante debe llevar el material necesario indicado por el Profesor;
- II. Minimizar el uso de celulares al interior del Laboratorio;
- III. Atender puntualmente las indicaciones que le dé su Profesor y el personal de laboratorio;
- IV. Mantener el orden y disciplina dentro de las instalaciones del laboratorio;
- V. Informar inmediatamente al Profesor y al Jefe de Laboratorio si ocurre un accidente;
- VI. Acatar las disposiciones de seguridad y preservar la higiene del laboratorio.



A los usuarios del laboratorio se les prohíbe:

- I. Introducir alimentos. Las bebidas permitidas como agua o refrescos no deben dejarse cerca de los circuitos o equipos eléctricos;
- II. Introducir mascotas, juguetes u otros objetos ajenos a las prácticas o al trabajo de laboratorio;
- III. Fumar, correr, gritar, jugar, empujar, tocar instrumentos musicales, usar radios o teléfonos, que alteren el orden;
- IV. Usar teléfonos móviles, radio localizadores, reproductores de música u otros aparatos ajenos a las prácticas de laboratorio;
- V. Usar las instalaciones de los laboratorios para otros propósitos, diferentes del desarrollo de prácticas programadas;
- VI. Incurrir en cualquier tipo de comportamiento que ponga en riesgo la integridad de las instalaciones, equipo de laboratorio, los usuarios, y jefe de laboratorio;
- VII. Introducir personas ajenas a las autorizadas para las prácticas.

## MATERIAL Y EQUIPO NECESARIO

1. Paquete de Raspberry Pi (Tarjeta Raspberry Pi, Fuente de poder, Tarjeta micro SD con sistema operativo Raspbian, Teclado, ratón y monitor), o...
2. Computadora personal con Intérprete de Python.

## METODOLOGÍA

### 1. Planteamiento del problema:

Diseñe un programa estructurado con las funciones pertinentes para que dado un circuito RLC serie, calcular el valor apropiado de la resistencia dados los valores de capacitancia e inductancia, así como el valor del voltaje de alimentación, suponiendo que se desea que la energía disipada alcance un cierto porcentaje del valor inicial en un tiempo determinado:

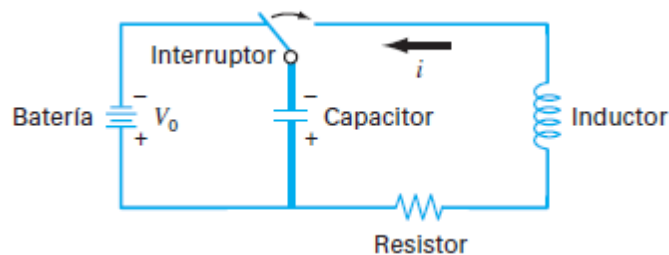


Figura 9.1

## 2. Algoritmo

ALGORITMO: Energía disipada en circuito RLC

DESCRIPCIÓN: Se calcula el valor de la resistencia R en un circuito RLC de manera que la energía disipada llegue a cierto porcentaje de su valor inicial en un determinado tiempo.

CONSTANTES:

-----

VARIABLES:

*real* :  $R, L, C$  Valores de Resistencia, Inductancia y Capacitancia

*entero* :  $V, Frec$  Valores de voltaje y frecuencia

*real* :  $ZR, ZL, ZC, ZT$  Impedancias

*real* :  $I\_max, Fase$  Corriente y Fase

FUNCIONES:

*Función Evalua\_ecuación:*

$$Valor = e^{-Rt/(2L)} \cos \left[ \sqrt{\frac{1}{LC} - \left(\frac{R}{2L}\right)^2} t \right] - \frac{q(t)}{q_0}$$

*Fin función*

*Función Introduce\_datos:*

2.1 Leer("Valor de inductancia", L)

2.2 Leer("Valor de capacitancia, C")

2.2 Leer("Valor de intervalo de tiempo", t)

2.2 Leer("Porcentaje deseado", energia)

*Fin función*

*INICIO*

1. *Escribir("Introduce número de iteraciones")*
  2. *Leer (N)*
  3. *Introduce\_datos (L, C, t, energía)*
  4. *Para i = 1 hasta i = N, i = i + 1:*
    - 4.1  $R_{max} \leftarrow R_{max}/2$
    - 4.2 *Valor 1  $\leftarrow$  Verdadero Si Evalúa\_ecuación para  $R_{max} > 0$*
    - 4.3 *Valor 2  $\leftarrow$  Verdadero Si Evalúa\_ecuación para  $R_{min} > 0$*
    - 4.4 *Si Valor 1  $\oplus$  Valor 2 entonces*
      - 4.41  $Aux \leftarrow R_{min}$
      - 4.42  $R_{min} \leftarrow R_{max}$
      - 4.43  $R_{max} \leftarrow (R_{max}-aux)/2+R_{min}$
    - Otro*
    - $R_{max} \leftarrow (R_{max}-R_{min})/2+R_{min}$
- Fin Para*
5. *Escribe ("Valor de resistencia requerido = "  $R_{max}$ )*

*FIN*

### 3. Programa

```
#Programa 7
#Programa para calcular Valor mediante bisección
#Circuito RLC respuesta temporal
#Datos L, C y porcentaje de reducción, calcular R
import math
def evalua(Res,L,C,t,porc):
    resultado=math.exp(-(Res*t)/(2*L))*math.cos((1/(L*C)-(Res/(2*L))**2)**0.5*t)porc
    return resultado
#Lectura de datos
Ind=float(input('Valor de la inductancia L (Henrios)= '))
Cap=float(input('Valor de la capacitancia C (Faradios)= '))
time=float(input('Tiempo para evaluar = '))
Pcien=float(input('Porcentaje de reduccion requerido = '))
#Calculo de Valores limite
Rmax=(4*Ind/Cap)**0.5/2
Rmin=0
#Iteraciones
for i in range(50):
    val1=evalua(Rmin,Ind,Cap,time,Pcien)
    val2=evalua(Rmax,Ind,Cap,time,Pcien)
    sign1=evalua(Rmin,Ind,Cap,time,Pcien)>0.0
    sign2=evalua(Rmax,Ind,Cap,time,Pcien)>0.0
```

Continúa...

```

if not(not sign1 and sign2 or sign1 and not sign2):

    aux=Rmin

    Rmin=Rmax

    Rmax=(Rmax-aux)/2+Rmax

else:

    Rmax=(Rmax-Rmin)/2+Rmin

print('Valor de resistencia requerido = ',Rmax)

#Fin del programa

```

### SUGERENCIAS DIDÁCTICAS

Desarrollar programas que resuelvan problemas similares al programa #6. Los siguientes son problemas sugeridos, pero cada estudiante podrá plantearse otros que sean de su interés.

Diseñe, Codifique y Pruebe programas para:

Ejercicio 9.1 Un filtro es un circuito que deja pasar señales de ciertas frecuencias mientras aquellas que sean de otras frecuencias. Un filtro pasa bajas deja pasar señales de bajas frecuencias desde la entrada hasta la salida mientras rechaza las frecuencias altas. Calcule las frecuencias de corte y los valores de el voltaje de salida para N valores de frecuencia, de un conjunto de circuitos RC pasa-bajos.

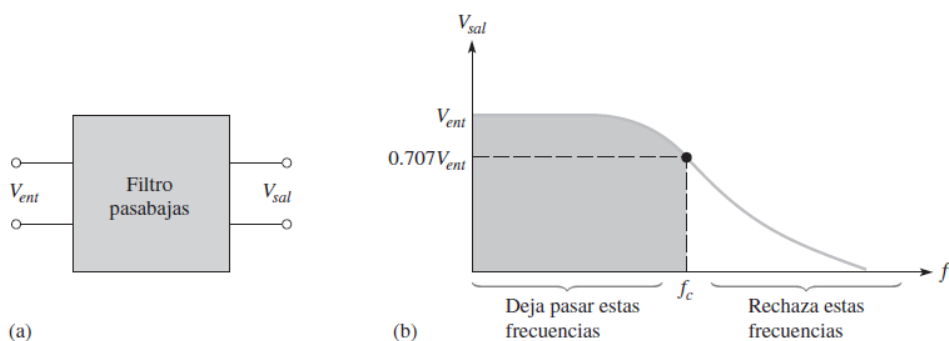


Figura 9.2

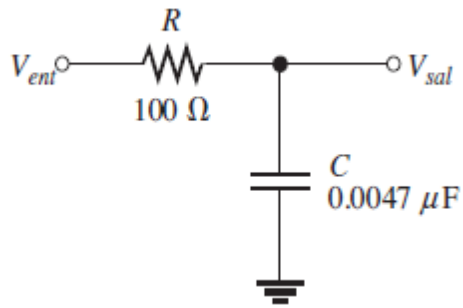


Figura 9.3

$$f_c = \frac{1}{2\pi RC}$$

$$V_{sal} = \frac{X_C}{\sqrt{R^2 + X_C^2}} \cdot V_{ent}$$

Ejercicio 9.2 Un filtro **pasa banda** deja pasar cierta banda de frecuencias y atenúa o rechaza todas las frecuencias por debajo y por encima de la banda de paso. El ancho de banda de un filtro pasa banda es el intervalo de frecuencias dentro del cual la corriente, y por tanto el voltaje de salida, es igual o mayor que el 70.7% de su valor en la frecuencia de resonancia.

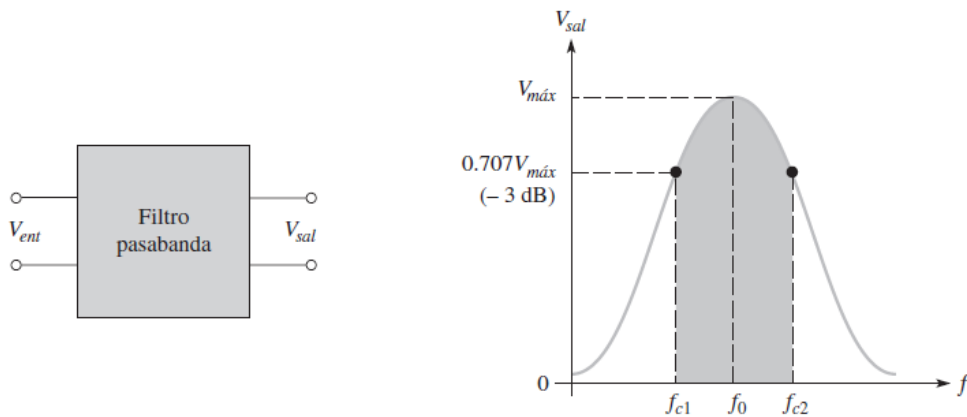


Figura 9.4

En el circuito de la figura 9.5 el voltaje de salida y la frecuencia central son:

$$V_{sal} = \frac{R}{R + R_w} \cdot V_{ent}$$

$$f_o = \frac{1}{2\pi\sqrt{LC}}$$

$$X_L = 2\pi f L \quad Q = \frac{X_L}{R_{TOT}}$$

$$R_{TOT} = R + R_w \quad AB = \frac{f_o}{Q}$$

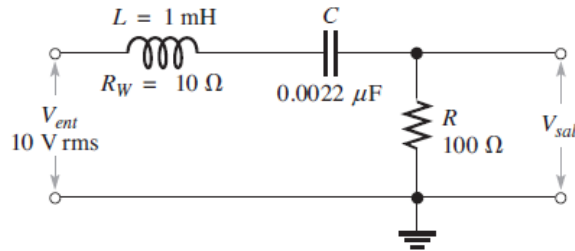


Figura 9.5

Calcule las frecuencias de corte y los valores de el voltaje de salida para N valores de frecuencia, de un conjunto de circuitos RC pasa-bajos.

### REPORTE DEL ALUMNO

Criterio para calificar	Si	No
El documento tiene portada	10%	0
Entregó en fecha programada	10%	0
El documento está completo	20%	0
El programa funciona correctamente	20%	0
Incluye un ejercicio adicional (al menos)	20%	0
Incluye resultados y/o gráficas o tablas	20%	0

### BIBLIOGRAFÍA

Raspberry Pi Foundation. Disponible en [www.raspberrypi.org](http://www.raspberrypi.org)

Lenguaje Python. Disponible en [www.python.org](http://www.python.org)

Diseño de un circuito eléctrico. Chapa, Steven C., Canale, Raymond P. Métodos Numéricos para Ingenieros. 5ª Ed. McGrawHill

## PRACTICA #10

### MAXIMA TRANSFERENCIA DE POTENCIA

Programa estructurado en Python

Entrada / Salida de datos por consola

Estructura Función, if, while

Librerías

Operaciones : suma, resta, multiplicación y división aritméticas, elevación a potencia, comparación

Tipos de datos: Enteros,Punto flotante, cadenas de caracteres

### COMPETENCIAS A DESARROLLAR

#### Específicas:

Utilizar adecuadamente las herramientas de diseño para el desarrollo de programas.

Utilizar la metodología de la programación modular para desarrollar programas estructurados y simplificar el mantenimiento del código.

Desarrollar programas a partir del diseño de algoritmos, para su aplicación en la solución de problemas.

**Genéricas:** Capacidad creativa, Capacidad cognitiva para comprender y manipular ideas, pensamientos e información, Capacidad para aprender, Capacidad crítica y autocrítica, Capacidad de investigación, Capacidad para tomar decisiones.

### INTRODUCCIÓN

En un circuito como el de la figura 10.1, la potencia disipada por la resistencia  $R_a$  el capacitor se carga inicialmente al voltaje de la batería está dada por la ecuación 9.1 Si esta ecuación se grafica para valores de  $R_a$  mayores que cero, se obtiene una gráfica como la mostrada en la figura 9.2, donde se observa que la potencia aumenta al aumentar el valor de  $R_a$ , hasta llegar a un máximo para entonces empezar a disminuir. Se puede entonces establecer un proceso repetitivo que calcule la potencia para valores de  $R_a$  que se incrementen hasta que se obtenga un nuevo valor de potencia menor al inmediato anterior.



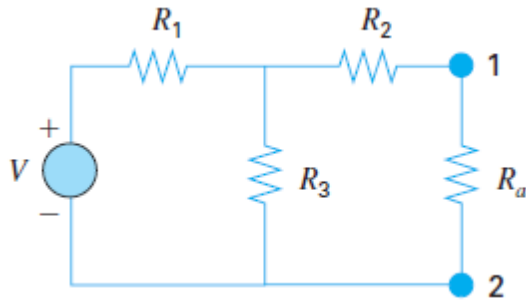


Figura 10.1

$$\text{Ecuación 10.1: } P(R_a) = \frac{\left[ \frac{VR_a R_3}{R_1(R_a + R_2 + R_3) + R_a R_3 + R_2 R_3} \right]^2}{R_a}$$

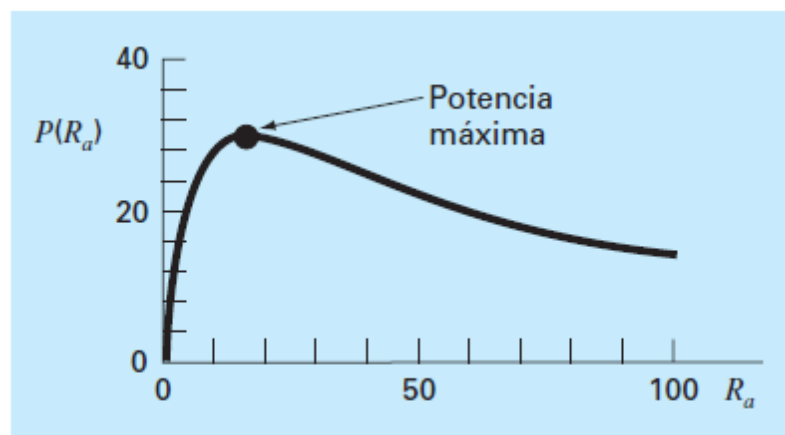


Figura 10.2

### CORRELACIÓN CON LOS TEMAS Y SUBTEMAS DEL PROGRAMA VIGENTE

Esta práctica se relaciona directamente con los temas de la unidad 3. El propósito es que el estudiante reconozca las ventajas de la programación estructurada y se acostumbre al pensamiento “arriba-abajo” (top down design), aplicando estas técnicas en la solución de problemas cada vez más grandes.

## **MEDIDAS DE SEGURIDAD E HIGIENE**

Son obligaciones de los usuarios del laboratorio:

- i. Para realizar la práctica el estudiante debe llevar el material necesario indicado por el Profesor;
- II. Minimizar el uso de celulares al interior del Laboratorio;
- III. Atender puntualmente las indicaciones que le dé su Profesor y el personal de laboratorio;
- IV. Mantener el orden y disciplina dentro de las instalaciones del laboratorio;
- V. Informar inmediatamente al Profesor y al Jefe de Laboratorio si ocurre un accidente;
- VI. Acatar las disposiciones de seguridad y preservar la higiene del laboratorio.

A los usuarios del laboratorio se les prohíbe:

- I. Introducir alimentos. Las bebidas permitidas como agua o refrescos no deben dejarse cerca de los circuitos o equipos eléctricos;
- II. Introducir mascotas, juguetes u otros objetos ajenos a las prácticas o al trabajo de laboratorio;
- III. Fumar, correr, gritar, jugar, empujar, tocar instrumentos musicales, usar radios o teléfonos, que alteren el orden;
- IV. Usar teléfonos móviles, radio localizadores, reproductores de música u otros aparatos ajenos a las prácticas de laboratorio;
- V. Usar las instalaciones de los laboratorios para otros propósitos, diferentes del desarrollo de prácticas programadas;
- VI. Incurrir en cualquier tipo de comportamiento que ponga en riesgo la integridad de las instalaciones, equipo de laboratorio, los usuarios, y jefe de laboratorio;
- VII. Introducir personas ajenas a las autorizadas para las prácticas;

## **MATERIAL Y EQUIPO NECESARIO**

1. Paquete de Raspberry Pi (Tarjeta Raspberry Pi, Fuente de poder, Tarjeta micro SD con sistema operativo Raspbian, Teclado, ratón y monitor), o...
2. Computadora personal con Intérprete de Python.

## METODOLOGÍA

### 1. Planteamiento del problema:

Diseñe un programa estructurado con las funciones pertinentes para que dado un circuito (figura 10.1) resistivo calcule el valor apropiado de la resistencia  $R_a$  que permita una “máxima transferencia de potencia” desde la fuente de alimentación.

### 2. Algoritmo

ALGORITMO: Máxima Transferencia de Potencia

*DESCRIPCIÓN: Se calcula el valor de resistencia ajustable en un circuito para una máxima transferencia de potencia.*

CONSTANTES:

-----

VARIABLES:

*real* :  $V, R_1, R_2, R_3$  Parámetros fijos del circuito

*real* :  $R_a$  Valor de resistencia ajustable y resistencia final

*real* :  $P_1, P_2$  Potencias consecutivas

FUNCIONES:

*Función Potencia:*

$$\text{Valor} = \frac{\left[ \frac{VR_a R_3}{R_1(R_a + R_2 + R_3) + R_a R_3 + R_2 R_3} \right]^2}{R_a}$$

*Fin función*

*Función Introduce\_datos:*

2.1 Leer("Valor de voltaje")

2.1 Leer("Valor de resistencia 1")

2.1 Leer("Valor de resistencia 2")

2.1 Leer("Valor de resistencia 3")

*Fin función*

*INICIO*

1. *Introduce datos (V, R1, R2, R3)*
2. *Calcula Potencia 1 = Potencia(V, R1, R2, R3, 1)*
3. *Calcula Potencia 2 = Potencia(V, R1, R2, R3, 1.01)*
4. *Rnueva ← 1.01*
5. *Mientras Potencia 2 > Potencia 1 repetir:*
  - 5.1 *Potencia 1 ← Potencia 2*
  - 5.2 *Rnueva ← Rnueva + 0.01*
  - 5.3 *Potencia 2 ← "Potencia (V, R1, R2, R3, Rnueva)"*
- Fin mientras*
6. *Escribe ("Valor de resistencia requerido = " Rnueva)*
7. *Escribe ("Potencia transferida = " Potencia 2)*

*FIN*

### 3. Programa

```

#Programa 8
#Maxima Transferencia de Potencia
#Problema 16.3 (Chapra)
def potencia(V,R1,R2,R3,Ra):
    resultado=(Ra*R3*V/(R1*R2+R1*R3+R1*Ra+R2*R3+Ra*R3))**2/(Ra)
    return resultado

Vol=float(input('Valor de voltaje de alimentación = '))
Res1=int(input('Valor de Resistencia 1 = '))
Res2=int(input('Valor de Resistencia 2 = '))
Res3=int(input('Valor de Resistencia 3 = '))
P1=potencia(Vol,Res1,Res2,Res3,1.0)
P2=potencia(Vol,Res1,Res2,Res3,1.01)
Rsig=1.01
while P2>P1:
    P1=P2
    Rsig=Rsig+0.01
    P2=potencia(Vol,Res1,Res2,Res3,Rsig)
print('Valor requerido de Ra = ',Rsig,'Ohms')
print('Potencia transferida a Ra = ',P2,'watts')
print('+++++')

```

## SUGERENCIAS DIDÁCTICAS

Desarrollar programas que resuelvan problemas similares al programa #8. Los siguientes son problemas sugeridos, pero cada estudiante podrá plantearse otros que sean de su interés.

Diseñe, Codifique y Pruebe programas para:

Ejercicio 10.1 Convertir circuitos Delta a Estrella y Estrella a Delta (FLOYD, Cap. 8.8)

Ejercicio 10.2 Calcular las raíces de un polinomio por el método de “punto fijo” (CHAPRA Cap. 6.1.2)

## REPORTE DEL ALUMNO

<b>Criterio para calificar</b>	<b>Si</b>	<b>No</b>
El documento tiene portada	10%	0
Entregó en fecha programada	10%	0
El documento está completo	20%	0
El programa funciona correctamente	20%	0
Incluye un ejercicio adicional (al menos)	20%	0
Incluye resultados y/o gráficas o tablas	20%	0

## BIBLIOGRAFÍA

Raspberry Pi Foundation. Disponible en [www.raspberrypi.org](http://www.raspberrypi.org)

Lenguaje Python. Disponible en [www.python.org](http://www.python.org)

Máxima transferencia de potencia en un circuito. Chapra, Steven C. y Canale, Raymond P. Métodos Numéricos para Ingenieros. 5ª Edición. McGrawHill

## PRACTICA #11

### POLARIZACIÓN DE JFET

Programa estructurado en Python

Entrada / Salida de datos por consola

Estructura función, while

Librerías

Operaciones : suma, resta, multiplicación y división aritméticas, elevación a potencia, comparación

Tipos de datos: Enteros,Punto flotante, cadenas de caracteres

### COMPETENCIAS A DESARROLLAR

#### Específicas:

Utilizar adecuadamente las herramientas de diseño para el desarrollo de programas.

Utilizar la metodología de la programación modular para desarrollar programas estructurados y simplificar el mantenimiento del código.

Desarrollar programas a partir del diseño de algoritmos, para su aplicación en la solución de problemas.

**Genéricas:** Capacidad creativa, Capacidad cognitiva para comprender y manipular ideas, pensamientos e información, Capacidad para aprender, Capacidad crítica y autocrítica, Capacidad de investigación, Capacidad para tomar decisiones.

### INTRODUCCIÓN

Los amplificadores JFET tienen que tener un punto de operación Q en la región activa. A causa del amplio rango de valores de los parámetros del JFET, se requiere de un circuito de polarización que reduzca la dependencia del punto Q a variaciones de esos parámetros.

En un circuito de polarización por **divisor de voltaje** el divisor genera un voltaje de compuerta que es una fracción del voltaje de alimentación. Restando el voltaje compuerta-fuente se obtiene el voltaje en la resistencia de fuente:

$$V_S = V_G - V_{GS}$$

Dado que  $V_{GS}$  es negativa, el voltaje de fuente será ligeramente mayor que el voltaje de compuerta. Si se divide este voltaje de fuente entre la resistencia de fuente, se obtiene la corriente de drenaje:

$$I_D = \frac{V_G - V_{GS}}{R_S} = \frac{V_{DD} \cdot \frac{R_2}{R_1 + R_2} - V_{GS}}{R_S}$$

Mientras que la curva de transconductancia (ecuación de Shockley) también relaciona  $I_D$  con  $V_{GS}$ .

$$I_D = I_{DSS} \cdot \left(1 - \frac{V_{GS}}{V_P}\right)^2$$

Ambas ecuaciones se pueden combinar gráficamente y el punto de intersección nos dará los valores de polarización o punto Q del transistor, como se observa en la figura 11.1.

Resolver las incógnitas analíticamente es un proceso no tan simple, por lo que una buena opción es un método iterativo con ayuda computacional.

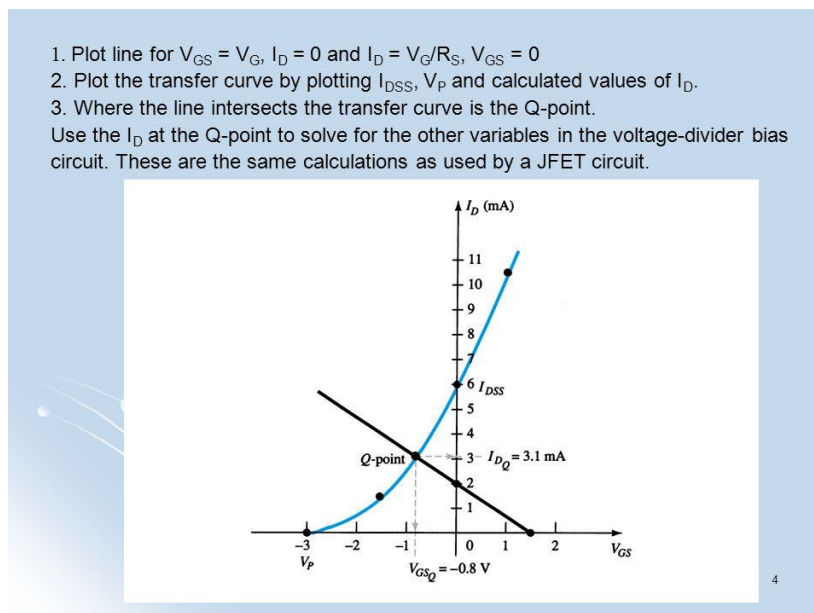


Figura 11.1



## **CORRELACIÓN CON LOS TEMAS Y SUBTEMAS DEL PROGRAMA VIGENTE**

Esta práctica se relaciona directamente con los temas de la unidad 3. El propósito es que el estudiante reconozca las ventajas de la programación estructurada y se acostumbre al pensamiento “arriba-abajo” (top down design), aplicando estas técnicas en la solución de problemas cada vez más grandes.

### **MEDIDAS DE SEGURIDAD E HIGIENE**

Son obligaciones de los usuarios del laboratorio:

- I. Para realizar la práctica el estudiante debe llevar el material necesario indicado por el Profesor;
- II. Minimizar el uso de celulares al interior del Laboratorio;
- III. Atender puntualmente las indicaciones que le dé su Profesor y el personal de laboratorio;
- IV. Mantener el orden y disciplina dentro de las instalaciones del laboratorio;
- V. Informar inmediatamente al Profesor y al Jefe de Laboratorio si ocurre un accidente;
- VI. Acatar las disposiciones de seguridad y preservar la higiene del laboratorio.

A los usuarios del laboratorio se les prohíbe:

- I. Introducir alimentos. Las bebidas permitidas como agua o refrescos no deben dejarse cerca de los circuitos o equipos eléctricos;
- II. Introducir mascotas, juguetes u otros objetos ajenos a las prácticas o al trabajo de laboratorio;
- III. Fumar, correr, gritar, jugar, empujar, tocar instrumentos musicales, usar radios o teléfonos, que alteren el orden;
- IV. Usar teléfonos móviles, radio localizadores, reproductores de música u otros aparatos ajenos a las prácticas de laboratorio;
- V. Usar las instalaciones de los laboratorios para otros propósitos, diferentes del desarrollo de prácticas programadas;
- VI. Incurrir en cualquier tipo de comportamiento que ponga en riesgo la integridad de las instalaciones, equipo de laboratorio, los usuarios, y jefe de laboratorio;
- VII. Introducir personas ajenas a las autorizadas para las prácticas;

## MATERIAL Y EQUIPO NECESARIO

1. Paquete de Raspberry Pi (Tarjeta Raspberry Pi, Fuente de poder, Tarjeta micro SD con sistema operativo Raspbian, Teclado, ratón y monitor), o...
2. Computadora personal con Intérprete de Python.

## METODOLOGÍA

### 1. Planteamiento del problema:

Diseñe un programa estructurado con las funciones pertinentes para calcular el punto de operación (Q) de un transistor JFET polarizado por divisor de voltaje.

### 2. Algoritmo

ALGORITMO: Polarización de JFET

DESCRIPCIÓN: Se calcula el valor del punto de operación  $I_{DQ}$ ,  $V_{GSQ}$

CONSTANTES:

-----

VARIABLES:

*real*:  $I_{DSS}$ ,  $V_P$ ,  $V_{DD}$  Parámetros del JFET y Alimentación

*entero*:  $R_S$ ,  $R_1$ ,  $R_2$ ,  $D_t$  Parámetros del circuito, Numero de intervalos de prueba

FUNCIONES:

*Función Lectura de Datos*

1. Leer corriente máxima de drenado  $I_{DSS}$
2. Leer voltaje de estrangulamiento  $V_P$
3. Leer voltaje de alimentación  $V_{DD}$
4. Leer Resistencia de fuente  $R_S$
5. Leer resistencia de polarización  $R_1$
6. Leer resistencia de polarización  $R_2$
7. Leer intervalos de prueba  $D_t$

*Fin función*

*Algoritmo principal:*

*Mientras continuar = si:*

1. *Lectura de Datos*
2. *Inicializa corrientes IDA e IDB*
3. *Inicializa voltaje de polarización compuerta-fuente (VGS)*
4. *Calcula voltaje de polarización de divisor de voltaje*
5. *Mientras IDA > IDB:*
  - 5.1 *Calcula nueva IDB (según ecuación de Kirchoff)*
  - 5.2 *Calcula nueva IDB (según ecuación de Shockley)*
  - 5.3 *Calcula nueva VGS*
6. *Escribir valores de punto de operación VGSQ y IDQ*
7. *Leer continuar*

### 3. Programa

```
# Programa para calcular punto de operacion JFET polarizacion divisor de voltaje

import time

def lectura_datos():
    IDSS=float(input("Corriente maxima de Drenador (mA)="))
    VP=float(input("Voltaje de estrangulamiento (voltios)="))
    VDD=int(input("Voltaje de Alimentacion (voltios)="))
    RS=int(input("Resistencia de Fuente (ohmios)="))
    R1=int(input("Resistencia R1 (ohmios)="))
    R2=int(input("Resistencia R2 (ohmios)="))
    Dt=int(input("Intervalos de prueba ="))
    return IDSS, VP, VDD, RS, R1, R2, Dt

dis=int(input("teclea 0"))

while dis == 0:
    IDSS, VP, VDD, RS, R1, R2, Dt= lectura_datos()

    IDta=IDSS
    IDtb=0
    VGS=0
    VG=VDD*(R2/(R1+R2))
    print("VG =",VG)
    print("IDta=", IDta)
    print("IDtb=", IDtb)
    while IDta>IDtb:
        IDtb=((VG+VGS)/RS)*1000
        IDta=IDSS*((1-VGS/VP)**2)
        VGS=VGS+VP/Dt
        print("IDta=", IDta)
        print("IDtb=", IDtb)
    print("Punto de Operacion =", VGS, "voltios")
    dis=int(input("continuar? teclea 0"))
```

## SUGERENCIAS DIDÁCTICAS

Desarrollar programas que resuelvan problemas similares al programa #7. Los siguientes son problemas sugeridos, pero cada estudiante podrá plantearse otros que sean de su interés.

Diseñe, Codifique y Pruebe programas para:

Ejercicio 11.1 Calcular el punto de operación de un JFET (Transistor de Efecto de Campo de Unión) para varios circuitos de polarización.

Ejercicio 11.2 Calcular el punto de operación de un BJT (Transistor Bipolar de Unión) para varios circuitos de polarización.

## REPORTE DEL ALUMNO

Criterio para calificar	Si	No
El documento tiene portada	10%	0
Entregó en fecha programada	10%	0
El documento está completo	20%	0
El programa funciona correctamente	20%	0
Incluye un ejercicio adicional (al menos)	20%	0
Incluye resultados y/o gráficas o tablas	20%	0

## BIBLIOGRAFÍA

Raspberry Pi Foundation. Disponible en [www.raspberrypi.org](http://www.raspberrypi.org)

Lenguaje Python. Disponible en [www.python.org](http://www.python.org)

Transistor JFET. Malvino, A. & Bates, David J, *Principios de Electrónica*. 7ª Edición, 2007, McGrawHill

## PRACTICA #12

### RESPUESTA TRANSITORIA

Programa estructurado en Python

Entrada / Salida de datos por consola

Estructura Función, if, while

Librerías

Operaciones : suma, resta, multiplicación y división aritméticas, elevación a potencia, comparación

Tipos de datos: Enteros,Punto flotante, cadenas de caracteres

### COMPETENCIAS A DESARROLLAR

#### Específicas:

Utilizar adecuadamente las herramientas de diseño para el desarrollo de programas.

Utilizar la metodología de la programación modular para desarrollar programas estructurados y simplificar el mantenimiento del código.

Desarrollar programas a partir del diseño de algoritmos, para su aplicación en la solución de problemas.

**Genéricas:** Capacidad creativa, Capacidad cognitiva para comprender y manipular ideas, pensamientos e información, Capacidad para aprender, Capacidad crítica y autocrítica, Capacidad de investigación, Capacidad para tomar decisiones.

### INTRODUCCIÓN

En el análisis y diseño de los sistemas de control, se debe tener una base de comparación del comportamiento de los sistemas para entradas particulares. Una señal de entrada de prueba típica es la función escalón unitario y la respuesta se puede escribir como

$$c(t) = c_{rt} + c_{ee}$$

Donde el primer término del lado derecho de la ecuación es la respuesta transitoria mientras que el segundo de esos términos es la respuesta de estado estacionario.

La característica más importante del comportamiento de un sistema es la estabilidad absoluta, esto es que la respuesta del sistema a una entrada limitada (como es la entrada escalón) sea también limitada. Las siguientes características importantes de la respuesta

son, para la parte transitoria son las oscilaciones amortiguadas que frecuentemente se presentan y para la parte estacionaria la diferencia entre la salida estacionaria y la entrada aplicada, o error estacionario.

Para un sistema de primer orden, la función de transferencia tiene la forma general de la figura 12.1

$$\frac{C(s)}{R(s)} = \frac{1}{Ts + 1}$$



Figura 12.1

Si consideramos que la entrada sea un escalón unitario  $R(s) = \frac{1}{s}$ , entonces la respuesta estará dada por

$$C(s) = \frac{1}{Ts + 1} \cdot \frac{1}{s} = \frac{1}{s} - \frac{1}{s + (1/T)}$$

Aplicando la transformada inversa de Laplace, se obtiene la respuesta en el tiempo de la figura 12.2:

$$c(t) = 1 - e^{-\frac{t}{T}} \quad \text{para } t \geq 0$$

Las características principales de esta respuesta son a) La constante de tiempo ( $t$  para que  $e^{-\frac{t}{T}} = e^{-1}$ )  $t = T$  y el tiempo requerido para alcanzar el valor final (en realidad el 98 o 99% del valor final) que viene siendo en  $t = 4T$  y en  $t = 5T$ , respectivamente.

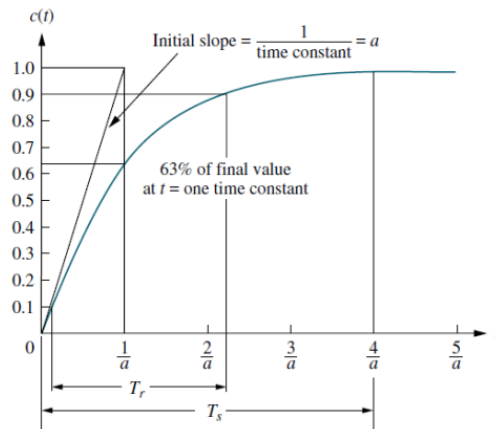


Figura 12.2

Para un sistema de segundo orden, la función de transferencia tiene la forma general

$$\frac{C(s)}{R(s)} = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

Si consideramos que la entrada sea un escalón unitario  $R(s) = \frac{1}{s}$ , entonces la respuesta estará dada por

$$C(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \cdot \frac{1}{s}$$

Consultando una Tabla de transformadas de Laplace, se obtiene la respuesta en el tiempo:

$$c(t) = 1 - \frac{e^{-\zeta\omega_n t}}{\sqrt{1-\zeta^2}} \operatorname{sen} \left( \omega_n \sqrt{1-\zeta^2} t + \operatorname{tg}^{-1} \frac{\sqrt{1-\zeta^2}}{\zeta} \right) \quad \text{para } t \geq 0 \text{ y } \zeta < 1$$

Las características principales de esta respuesta son a) El tiempo de pico ( $t_p$  para que se alcance el valor máximo de la respuesta). b) El sobre-nivel (Valor del pico máximo de respuesta). c) El tiempo de subida (tiempo para el cual se alcanza por primera vez el valor final). d) El tiempo de estabilización (tiempo para el cual la respuesta ya no excede del 2% del valor final).

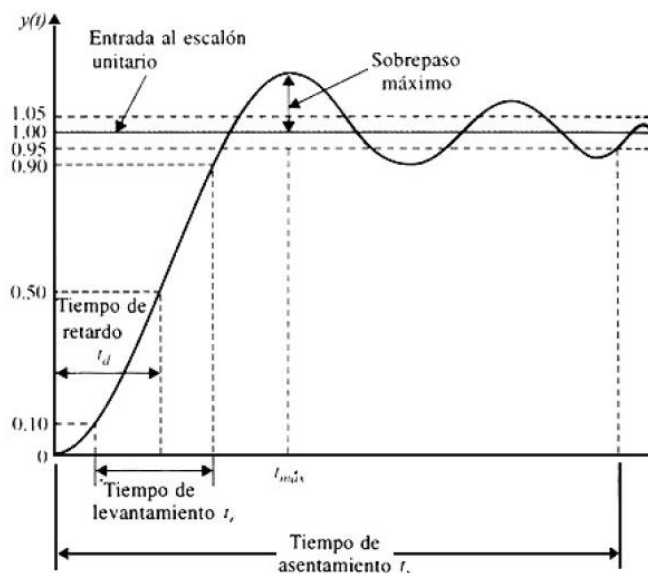


Figura 12.3



## **CORRELACIÓN CON LOS TEMAS Y SUBTEMAS DEL PROGRAMA VIGENTE**

Esta práctica se relaciona directamente con los temas de la unidad 3. El propósito es que el estudiante reconozca las ventajas de la programación estructurada y se acostumbre al pensamiento “arriba-abajo” (top down design), aplicando estas técnicas en la solución de problemas cada vez más grandes.

## **MEDIDAS DE SEGURIDAD E HIGIENE**

Son obligaciones de los usuarios del laboratorio:

- I. Para realizar la práctica el estudiante debe llevar el material necesario indicado por el Profesor;
- II. Minimizar el uso de celulares al interior del Laboratorio;
- III. Atender puntualmente las indicaciones que le dé su Profesor y el personal de laboratorio;
- IV. Mantener el orden y disciplina dentro de las instalaciones del laboratorio;
- V. Informar inmediatamente al Profesor y al Jefe de Laboratorio si ocurre un accidente;
- VI. Acatar las disposiciones de seguridad y preservar la higiene del laboratorio.

A los usuarios del laboratorio se les prohíbe:

- I. Introducir alimentos. Las bebidas permitidas como agua o refrescos no deben dejarse cerca de los circuitos o equipos eléctricos;
- II. Introducir mascotas, juguetes u otros objetos ajenos a las prácticas o al trabajo de laboratorio;
- III. Fumar, correr, gritar, jugar, empujar, tocar instrumentos musicales, usar radios o teléfonos, que alteren el orden;
- IV. Usar teléfonos móviles, radio localizadores, reproductores de música u otros aparatos ajenos a las prácticas de laboratorio;
- V. Usar las instalaciones de los laboratorios para otros propósitos, diferentes del desarrollo de prácticas programadas;
- VI. Incurrir en cualquier tipo de comportamiento que ponga en riesgo la integridad de las instalaciones, equipo de laboratorio, los usuarios, y jefe de laboratorio;
- VII. Introducir personas ajenas a las autorizadas para las prácticas;

## MATERIAL Y EQUIPO NECESARIO

1. Paquete de Raspberry Pi (Tarjeta Raspberry Pi, Fuente de poder, Tarjeta micro SD con sistema operativo Raspbian, Teclado, ratón y monitor), o...
2. Computadora personal con Intérprete de Python.

## METODOLOGÍA

### 1. Planteamiento del problema:

Diseñe un programa estructurado con las funciones pertinentes para que dados los coeficientes de la función de transferencia de un sistema, se calcule la respuesta del sistema a una entrada escalón unitario. Deberá graficarse la forma de la respuesta y calcularse el valor de todos los parámetros importantes de dicha respuesta.

### 2. Algoritmo

ALGORITMO: Respuesta Transitoria

*DESCRIPCIÓN: Se calcula el valor de los principales parámetros de respuesta transitoria de un sistema definido por su función de transferencia, también se grafica la forma de la respuesta.*

CONSTANTES:

-----

VARIABLES:

*real*: Coeficientes de función de transferencia primer orden

*real*: Coeficientes de función de transferencia segundo orden

*real*: Valor de la respuesta, parámetros de interés

## FUNCIONES:

*Función Primer Orden:*

*INICIO*

1. *Lectura de parámetros*
2. *Calcula límites de cálculos (5 constantes de tiempo)*
3. *Genera Tabla de respuestas*
4. *Guarda Constante de tiempo, Valor de respuesta en 1, 2 y 3 constantes de tiempo.*

*FIN*

$$r = 1 - e^{-\frac{t}{T}}$$

Ecuación que define la respuesta  
transitoria de primer orden

*Función Segundo Orden:*

*INICIO*

5. *Lectura de parámetros*
6. *Calcula límites de cálculos ( $\frac{5}{\sigma}$ )*
7. *Genera Tabla de respuestas*
8. *Guarda Tiempo de pico, Sobre Nivel, Tiempo de estabilización, etc...*

*FIN*

$$r = 1 - \frac{e^{-\zeta\omega_n t}}{\sqrt{1-\zeta^2}} \cdot \text{sen} \left( \omega_n \sqrt{1-\zeta^2} t + \text{tg}^{-1} \frac{\sqrt{1-\zeta^2}}{\zeta} \right)$$

Ecuación

que define la respuesta transitoria de segundo orden

*Función Grafica:*

*INICIO*

1. *Grafica Tabla de valores*

*FIN*

*INICIO*

1. *Repita hasta que preg <> 's'*
  - 1.1 *¿Tipo de sistema?*
    - 1.1.1 *Primer Orden*
    - 1.1.2 *Segundo Orden*
  - 1.2 *Grafica*
  - 1.3 *Parámetros importantes*
  - 1.4 *¿Continuar?*

*FIN*

### 3. Programa

```
#Programa Num. 10
#Respuesta Transitoria con graficación
#Sistemas de 1er Orden y 2º Orden

import math
import time
import matplotlib.pyplot as plt
import scipy as sp

def Kaptura(t):
    return input(t)

def Menu(menu0):
    print('*** Menu ***')
    for x in range(len(menu0)):
        print(str(x+1)+'.- '+menu0[x])
```

Continúa...

```
def primer_orden():  
    A=float(input('Parámetro A = '))  
    B=float(input('Parámetro B = '))  
    T=1/B  
    print('Constante de tiempo = ',T)  
    C=int(input('Intervalos de graficación = '))  
    t=0  
    It=5*T/C  
    Tabla=[]  
    while t < 5*T:  
        r=(A/B)*(1-math.e**(-t/T))  
        Tabla.append(r)  
        t=t+It  
    r1 = (A/B)*(1-math.e**(-1))  
    r2 = (A/B)*(1-math.e**(-2))  
    r3 = (A/B)*(1-math.e**(-3))  
    return Tabla, r1, r2, r3, It
```

Continúa...

```

def segundo_orden():
    W=float(input('Parámetro wn = '))
    F=float(input('Parámetro famor = '))
    C=int(input('Intervalos de graficación = '))
    t=0 / Sigma = F*W / Ts = 5/Sigma #En líneas diferentes en el original
    It=Ts/C / Tabla=[] / raiz = (1-F**2)**0.5
    rmax = 0 / tp = 0.0 / tsub = 0 / prim = 0
    while t < Ts:
        r = 1 -(math.e**(-F*W*t))/(raiz))*math.sin(W*raiz*t+math.atan(raiz/F))
        if r > rmax:
            rmax = r
            tp = t
        if ((r - 1) < 0.0 and prim == 0):
            tsub = t
        else:
            prim = 1
        if math.fabs(r-1.0)> 0.020:
            ts = t
        Tabla.append(r)
        t=t+It
    return Tabla, rmax, tp, tsub, ts, It

```

Continúa....

```
def graficador_con():  
    plt.figure()  
    plt.plot(Tabla)  
    plt.suptitle('Escalador de tiempo = '  
    plt.title(str(It))  
    plt.show()  
  
#Programa principal  
menu0 = ['Respuesta Transitoria Primer Orden', 'Respuesta Transitoria Segundo Orden', 'Salir']  
#Opciones del menu1  
  
opcion = 0 #Variable para controlar las acciones  
while opcion != len(menu0):  
    Menu(menu0)  
    opcion = int(Kaptura('\t Tu Opción: '))  
    if opcion == 1:  
        Tabla, r1, r2, r3, It = primer_orden()  
        graficador_con()  
        print('Respuesta para 1 T = ',r1)  
        print('Respuesta para 2 T = ',r2)  
        print('Respuesta para 3 T = ',r3)  
        print('*'*30)
```



## SUGERENCIAS DIDÁCTICAS

Desarrollar programas que resuelvan problemas similares al programa #7. Los siguientes son problemas sugeridos, pero cada estudiante podrá plantearse otros que sean de su interés.

Diseñe, Codifique y Pruebe programas para:

Ejercicio 12.1 Repita el ejercicio 9.1 agregando la graficación de los valores calculados de salida con respecto a la frecuencia de la señal de entrada.

Ejercicio 12.2 Repita el ejercicio 9.2 agregando la graficación de los valores calculados de salida con respecto a la frecuencia de la señal de entrada.

## REPORTE DEL ALUMNO

<b>Criterio para calificar</b>	<b>Si</b>	<b>No</b>
El documento tiene portada	10%	0
Entregó en fecha programada	10%	0
El documento está completo	20%	0
El programa funciona correctamente	20%	0
Incluye un ejercicio adicional (al menos)	20%	0
Incluye resultados y/o gráficas o tablas	20%	0

## BIBLIOGRAFÍA

Raspberry Pi Foundation. Disponible en [www.raspberrypi.org](http://www.raspberrypi.org)

Lenguaje Python. Disponible en [www.python.org](http://www.python.org)

Respuesta Transitoria. Ogata, Katsuhiko. Ingeniería de Control Moderna. 5ª Edición. Pearson Educación.

## PRACTICA #13

### ENTRADA-SALIDA BINARIA

Programa estructurado en Python

Entrada / Salida de datos por GPIO

Estructura while

Librerías

Operaciones : booleanas

Tipos de datos: Enteros

### COMPETENCIAS A DESARROLLAR

#### **Específicas:**

Desarrollar programas que incluyan el manejo de puertos para permitir la interacción con sistemas electrónicos externos a la computadora.

**Genéricas:** Capacidad creativa, Capacidad cognitiva para comprender y manipular ideas, pensamientos e información, Capacidad para aprender, Capacidad crítica y autocrítica, Capacidad de investigación, Capacidad para tomar decisiones.

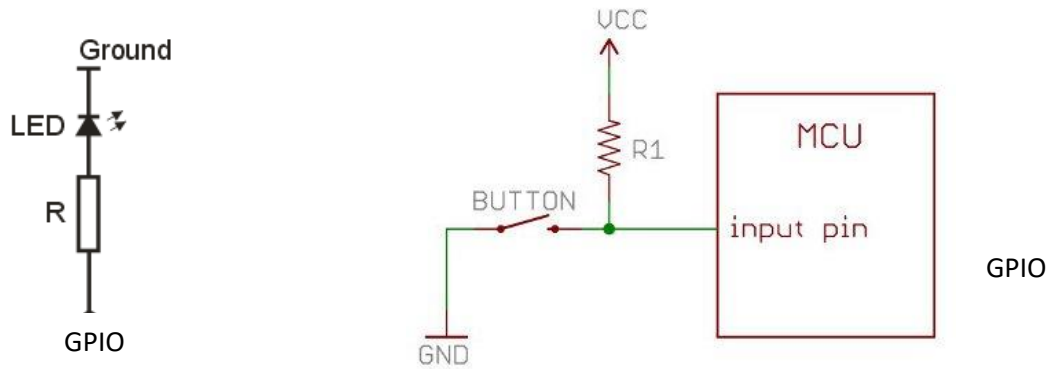
### INTRODUCCIÓN

Los dispositivos de Entrada-Salida son aquellos que permiten la comunicación entre la computadora y el usuario. Más allá del teclado, la pantalla (monitor), la impresora y el ratón existen una gran cantidad de dispositivos periféricos que se pueden conectar a una computadora con el objetivo de introducir datos o sacar resultados. Entre estos dispositivos se encuentran los que únicamente pueden tener dos estados: Encendido y Apagado, Falso y Verdadero o ON-OFF.

Sensores inductivos magnéticos, ópticos, pulsadores, termopares, termoresistencias, codificadores, entre otros son dispositivos de entrada ON-OFF. Los contactores de motor, las electroválvulas, los indicadores luminosos, los relevadores electromagnéticos, entre otros son dispositivos de salida ON-OFF.

Los diodos de unión emisores de luz (LED), generalmente construidos de materiales semiconductores GaAs o AlGaAs emiten luz cuando se hace pasar por ellos una corriente de polarización. Las salidas GPIO proporcionan un nivel de voltaje suficiente para encender un LED a través de una resistencia limitadora R de  $470\Omega$ .

Como entrada ON-OFF se puede usar un interruptor momentáneo de dos posiciones conectado a la fuente de alimentación mediante una resistencia R de  $470\Omega$ .



### CORRELACIÓN CON LOS TEMAS Y SUBTEMAS DEL PROGRAMA VIGENTE

Esta práctica se relaciona directamente con los temas de la unidad 4. El propósito es que el estudiante haga interactuar su computadora con sistemas electrónicos externos, desde los más sencillos sistemas Encendido-Apagado hasta la lectura y procesamiento de señales analógicas.

### MEDIDAS DE SEGURIDAD E HIGIENE

Son obligaciones de los usuarios del laboratorio:

- I. Para realizar la práctica el estudiante debe llevar el material necesario indicado por el Profesor;
- II. Minimizar el uso de celulares al interior del Laboratorio;
- III. Atender puntualmente las indicaciones que le dé su Profesor y el personal de laboratorio;
- IV. Mantener el orden y disciplina dentro de las instalaciones del laboratorio;
- V. Informar inmediatamente al Profesor y al Jefe de Laboratorio si ocurre un accidente;
- VI. Acatar las disposiciones de seguridad y preservar la higiene del laboratorio.

A los usuarios del laboratorio se les prohíbe:

- I. Introducir alimentos. Las bebidas permitidas como agua o refrescos no deben dejarse cerca de los circuitos o equipos eléctricos;
- II. Introducir mascotas, juguetes u otros objetos ajenos a las prácticas o al trabajo de laboratorio;
- III. Fumar, correr, gritar, jugar, empujar, tocar instrumentos musicales, usar radios o teléfonos, que alteren el orden;

- IV. Usar teléfonos móviles, radio localizadores, reproductores de música u otros aparatos ajenos a las prácticas de laboratorio;
- V. Usar las instalaciones de los laboratorios para otros propósitos, diferentes del desarrollo de prácticas programadas;
- VI. Incurrir en cualquier tipo de comportamiento que ponga en riesgo la integridad de las instalaciones, equipo de laboratorio, los usuarios, y jefe de laboratorio;
- VII. Introducir personas ajenas a las autorizadas para las prácticas;

### **MATERIAL Y EQUIPO NECESARIO**

- 1. Paquete de Raspberry Pi (Tarjeta Raspberry Pi, Fuente de poder, Tarjeta micro SD con sistema operativo Raspbian, Teclado, ratón y monitor).
- 2. Tarjeta de experimentación
- 3. Diodos LED
- 4. Interruptores miniatura de 2 posiciones.
- 5. Resistencias de  $470\Omega$  o  $1k\Omega$  .

### **METODOLOGÍA**

#### **1. Planteamiento del problema:**

Diseñe un programa para encender y apagar 2 diodos LED mediante un interruptor de encendido y un interruptor de apagado.

## 2. Algoritmo

ALGORITMO: Entrada Salida Binaria

*DESCRIPCIÓN: Se genera una salida ON-OFF conectada a terminales del GPIO, controlada por 2 interruptores (A y B) de 2 posiciones o momentáneos. El interruptor A enciende el LED y el interruptor B lo apaga.*

CONSTANTES: -----

VARIABLES:

*Booleano: IA, IB, Lo* Interruptores de encendido/apagado, Lámpara LED

INICIO

1. Repite indefinidamente:

1.1  $Lo \leftarrow 0$  Apaga LED

1.2  $IA \leftarrow 0$  Inicializa posición de interruptor

1.3 Mientras  $IA = 0$  repite:

1.3.1 Lee IA

Hasta que  $IA = 1$

1.4.  $Lo \leftarrow 1$  Enciende LED

1.5  $IB \leftarrow 0$

1.6 Mientras  $IB = 0$  repite:

1.6.1 Lee IB

Hasta que  $IB = 1$

Fin (Repite)

FIN

### 3. Programa

```
#Programa para controlar semáforo
#Salidas Luz 18
#Entradas IA (23), IB(25)
from gpiozero import LED
from gpiozero import Button

Luz = LED(18)
IA = Button(23)
IB = Button(25)

while True :

    Luz = 0

    ON = IA.is_pressed

    while ON==1:

        ON = IA.is_pressed

    Luz = 1

    ON = IB.is_pressed

    while ON==1:

        ON = IB.is_pressed
```

## SUGERENCIAS DIDÁCTICAS

Desarrollar programas que resuelvan problemas similares al programa #13. Los siguientes son problemas sugeridos, pero cada estudiante podrá plantearse otros que sean de su interés.

Diseñe, Codifique y Pruebe programas para:

Ejercicio 13.1 Controlar una luz LED con un solo interruptor momentáneo de encendido-apagado.

Ejercicio 13.2 Controlar 4 luces LED para que se enciendan en una de 4 secuencias posibles, seleccionadas con 2 interruptores de 2 posiciones.

## REPORTE DEL ALUMNO

Criterio para calificar	Si	No
El documento tiene portada	10%	0
Entregó en fecha programada	10%	0
El documento está completo	20%	0
El programa funciona correctamente	20%	0
Incluye un ejercicio adicional (al menos)	20%	0
Incluye resultados y/o gráficas o tablas	20%	0

## BIBLIOGRAFÍA

Raspberry Pi Foundation. Disponible en [www.raspberrypi.org](http://www.raspberrypi.org)

Lenguaje Python. Disponible en [www.python.org](http://www.python.org)

[Monk, Simon. Programming the Raspberry Pi. Second Edition. McGraHill Education.](#)

## PRACTICA #14

### SEMÁFORO SIMPLE

Programa estructurado en Python

Entrada / Salida de datos por GPIO

Estructura Función, if, while

Librerías

Operaciones : suma, resta, multiplicación y división aritméticas, elevación a potencia, comparación

Tipos de datos: Enteros,Punto flotante, cadenas de caracteres

### COMPETENCIAS A DESARROLLAR

#### Específicas:

Desarrollar programas que incluyan el manejo de puertos para permitir la interacción con sistemas electrónicos externos a la computadora.

**Genéricas:** Capacidad creativa, Capacidad cognitiva para comprender y manipular ideas, pensamientos e información, Capacidad para aprender, Capacidad crítica y autocrítica, Capacidad de investigación, Capacidad para tomar decisiones.

### INTRODUCCIÓN

La secuencia “normal” de un semáforo es Luz Verde (LV) enciende T1 segundos, Luz Amarilla (LA) enciende durante T2 segundos, Luz Roja (LR) enciende T3 segundos. Se puede establecer una Tabla de funcionamiento como la siguiente:

Interruptor	Secuencia	Descripción
0	Normal	Se repite continuamente:  Enciende LV T1 segundos  Enciende LA T2 segundos  Enciende LR T3 segundos
1	Ninguna	Todas las luces apagadas.



## **CORRELACIÓN CON LOS TEMAS Y SUBTEMAS DEL PROGRAMA VIGENTE**

Esta práctica se relaciona directamente con los temas de la unidad 4. El propósito es que el estudiante haga interactuar su computadora con sistemas electrónicos externos, desde los más sencillos sistemas Encendido-Apagado hasta la lectura y procesamiento de señales analógicas.

## **MEDIDAS DE SEGURIDAD E HIGIENE**

Son obligaciones de los usuarios del laboratorio:

- I. Para realizar la práctica el estudiante debe llevar el material necesario indicado por el Profesor;
- II. Minimizar el uso de celulares al interior del Laboratorio;
- III. Atender puntualmente las indicaciones que le dé su Profesor y el personal de laboratorio;
- IV. Mantener el orden y disciplina dentro de las instalaciones del laboratorio;
- V. Informar inmediatamente al Profesor y al Jefe de Laboratorio si ocurre un accidente;
- VI. Acatar las disposiciones de seguridad y preservar la higiene del laboratorio.

A los usuarios del laboratorio se les prohíbe:

- I. Introducir alimentos. Las bebidas permitidas como agua o refrescos no deben dejarse cerca de los circuitos o equipos eléctricos;
- II. Introducir mascotas, juguetes u otros objetos ajenos a las prácticas o al trabajo de laboratorio;
- III. Fumar, correr, gritar, jugar, empujar, tocar instrumentos musicales, usar radios o teléfonos, que alteren el orden;
- IV. Usar teléfonos móviles, radio localizadores, reproductores de música u otros aparatos ajenos a las prácticas de laboratorio;
- V. Usar las instalaciones de los laboratorios para otros propósitos, diferentes del desarrollo de prácticas programadas;
- VI. Incurrir en cualquier tipo de comportamiento que ponga en riesgo la integridad de las instalaciones, equipo de laboratorio, los usuarios, y jefe de laboratorio;
- VII. Introducir personas ajenas a las autorizadas para las prácticas;

## MATERIAL Y EQUIPO NECESARIO

1. Paquete de Raspberry Pi (Tarjeta Raspberry Pi, Fuente de poder, Tarjeta micro SD con sistema operativo Raspbian, Teclado, ratón y monitor).
2. Tarjeta de experimentación
3. Diodos LED
4. Interruptores miniatura de 2 posiciones.
5. Resistencias de  $470\Omega$  o  $1k\Omega$  .

## METODOLOGÍA

### 1. Planteamiento del problema:

Diseñe un programa para controlar la operación de un semáforo, mediante un interruptor de encendido/apagado.

### 2. Algoritmo

#### **Funciones**

*Función Secuencia:*

- 1.1  $L_V \leftarrow ON$
- 1.2 *Tiempo(T1)*
- 1.3  $L_V \leftarrow OFF$
- 1.4  $L_A \leftarrow ON$
- 1.5 *Tiempo(T2)*
- 1.6  $L_A \leftarrow OFF$
- 1.7  $L_R \leftarrow ON$
- 1.8 *Tiempo(T3)*
- 1.9  $L_R \leftarrow OFF$

*Fin función*

ALGORITMO: Semáforo simple

*DESCRIPCIÓN: Se generan 3 salidas ON-OFF representando las luces de un semáforo. La secuencia encendido-apagado puede activarse o detenerse mediante la posición de un interruptor ON-OFF.*

CONSTANTES:

-----

VARIABLES:

*Booleano:  $I_e$*  Interruptor de encendido.

*Booleano:  $L_V, L_A, L_R$*  Lámparas para cada color del semáforo

INICIO

1. *Mientras  $I_e = 1$  repite :*

1.1 *Lee ( $I_e$ )*

1.2 *Si ( $I_e = ON$ ) entonces*

1.21 *Secuencia*

*Fin Mientras*

FIN

### 3. Programa

```
#Programa para controlar semáforo

#Salidas Luces Verde = 18, Amarillo = 23, Rojo = 25

import time

from gpiozero import LED

from gpiozero import Button

Verde=LED(18)

Amarillo=LED(23)

Rojo=LED(25)

boton=Button(5)

def secuencia():

    Verde.on()

    time.sleep(05)

    Verde.off()

    Amarillo.on()

    time.sleep(02)

    Amarillo.off()

    Rojo.on()

    time.sleep(05)

    Rojo.off()

    return

while True :

    ON=boton.is_pressed

    If ON==1:

        Secuencia()
```

```

#Continua...

    time.sleep(05)

    Rojo.off()

return

while True :

    ON=boton.is_pressed

    If ON==1:

        Secuencia()

```

### SUGERENCIAS DIDÁCTICAS

Desarrollar programas que resuelvan problemas similares al programa #14. Los siguientes son problemas sugeridos, pero cada estudiante podrá plantearse otros que sean de su interés.

Diseñe, Codifique y Pruebe programas para:

Ejercicio 14.1 Controlar un semáforo en 4 secuencias diferentes, mediante un interruptor de encendido/apagado y 2 interruptores selectores de secuencia.

Ejercicio 14.2 Manejo de PWM, para controlar la intensidad de un LED o la velocidad de un motor de CD.

### REPORTE DEL ALUMNO

Criterio para calificar	Si	No
El documento tiene portada	10%	0
Entregó en fecha programada	10%	0
El documento está completo	20%	0
El programa funciona correctamente	20%	0
Incluye un ejercicio adicional (al menos)	20%	0
Incluye resultados y/o gráficas o tablas	20%	0

### BIBLIOGRAFÍA

Raspberry Pi Foundation. Disponible en [www.raspberrypi.org](http://www.raspberrypi.org)

Lenguaje Python. Disponible en [www.python.org](http://www.python.org)

[Monk, Simon. Programming the Raspberry Pi. Second Edition. McGraHill Education.](#)

## PRACTICA #15

### CONTROL SECUENCIAL

Programa estructurado en Python

Entrada / Salida de datos por GPIO

Estructura Función, if, while

Librerías

Operaciones : incremento

Tipos de datos: Enteros

### COMPETENCIAS A DESARROLLAR

#### Específicas:

Desarrollar programas que incluyan el manejo de puertos para permitir la interacción con sistemas electrónicos externos a la computadora.

**Genéricas:** Capacidad creativa, Capacidad cognitiva para comprender y manipular ideas, pensamientos e información, Capacidad para aprender, Capacidad crítica y autocrítica, Capacidad de investigación, Capacidad para tomar decisiones.

### INTRODUCCIÓN

Una **cinta transportadora** o **banda transportadora** o

**transportadora de banda** o **cintas francas** es un sistema de transporte continuo formado por una banda continua que se mueve entre dos tambores.

Por lo general, la banda es arrastrada por la fricción de sus tambores, que a la vez este es accionado por su motor. Esta fricción es la resultante de la aplicación de una tensión a la banda transportadora, habitualmente mediante un mecanismo tensor por husillo o tornillo tensor. El otro tambor suele girar libre, sin ningún tipo de accionamiento, y su función es servir de retorno a la banda. La banda es soportada por rodillos entre los dos tambores. Denominados rodillos de soporte.

Debido al movimiento de la banda el material depositado sobre la banda es transportado hacia el tambor de accionamiento donde la banda gira y da la vuelta en sentido contrario. En esta zona el material depositado sobre la banda es vertido fuera de la misma debido a la acción de la gravedad y/o de la inercia.

Las cintas transportadoras se usan principalmente para transportar materiales granulados, agrícolas e industriales, tales como cereales, carbón, minerales, etcétera, aunque también

se pueden usar para transportar personas en recintos cerrados (por ejemplo, en grandes hospitales y ciudades sanitarias).

### **CORRELACIÓN CON LOS TEMAS Y SUBTEMAS DEL PROGRAMA VIGENTE**

Esta práctica se relaciona directamente con los temas de la unidad 4. El propósito es que el estudiante haga interactuar su computadora con sistemas electrónicos externos, desde los más sencillos sistemas Encendido-Apagado hasta la lectura y procesamiento de señales analógicas.

### **MEDIDAS DE SEGURIDAD E HIGIENE**

Son obligaciones de los usuarios del laboratorio:

- I. Para realizar la práctica el estudiante debe llevar el material necesario indicado por el Profesor;
- II. Minimizar el uso de celulares al interior del Laboratorio;
- III. Atender puntualmente las indicaciones que le dé su Profesor y el personal de laboratorio;
- IV. Mantener el orden y disciplina dentro de las instalaciones del laboratorio;
- V. Informar inmediatamente al Profesor y al Jefe de Laboratorio si ocurre un accidente;
- VI. Acatar las disposiciones de seguridad y preservar la higiene del laboratorio.

A los usuarios del laboratorio se les prohíbe:

- I. Introducir alimentos. Las bebidas permitidas como agua o refrescos no deben dejarse cerca de los circuitos o equipos eléctricos;
- II. Introducir mascotas, juguetes u otros objetos ajenos a las prácticas o al trabajo de laboratorio;
- III. Fumar, correr, gritar, jugar, empujar, tocar instrumentos musicales, usar radios o teléfonos, que alteren el orden;
- IV. Usar teléfonos móviles, radio localizadores, reproductores de música u otros aparatos ajenos a las prácticas de laboratorio;
- V. Usar las instalaciones de los laboratorios para otros propósitos, diferentes del desarrollo de prácticas programadas;
- VI. Incurrir en cualquier tipo de comportamiento que ponga en riesgo la integridad de las instalaciones, equipo de laboratorio, los usuarios, y jefe de laboratorio;
- VII. Introducir personas ajenas a las autorizadas para las prácticas;

## MATERIAL Y EQUIPO NECESARIO

1. Paquete de Raspberry Pi (Tarjeta Raspberry Pi, Fuente de poder, Tarjeta micro SD con sistema operativo Raspbian, Teclado, ratón y monitor).
2. Tarjeta de experimentación
3. Diodos LED
4. Interruptores miniatura de 2 posiciones.
5. Resistencias de  $470\Omega$  o  $1k\Omega$ .

## METODOLOGÍA

### 1. Planteamiento del problema:

Diseñe un programa para controlar la operación de las bandas transportadoras para la distribución del equipaje a cargar en un avión, con una proporción de 60% en la parte delantera y 40% en la parte trasera. Existen tres bandas, la principal y las dos secundarias distribuidoras del equipaje. Se tiene un botón de encendido/apagado del proceso y un sensor de maletas en la banda principal.

### 2. Algoritmo

ALGORITMO: Control De Banda Transportadora

DESCRIPCIÓN: Se controlan tres bandas distribuidoras del equipaje que se va a cargar en un avión. Una banda principal y 2 secundarias para la parte delantera y trasera del avión.

CONSTANTES: -----

VARIABLES:

*Booleano:*  $I_e, L_V, L_A, L_R$  Interruptor de encendido y lámparas del semáforo

INICIO

1. Inicia con bandas detenidas y contador de equipaje en cero.
2. Espera a botón de arranque ( $bA$ )
3. Mientras  $bA = 1$  repite :
  - 3.1 Arranca banda principal y banda delantera
  - 3.2 Mientras contador  $< 4$  repite:
    - 3.2.1 Espera maleta
    - 3.2.2 Incrementa contador
  - 3.3 Pone contador en 1
  - 3.4 Arranca banda trasera y detiene banda delantera
  - 3.5 Mientras contador  $< 3$  repite:
    - 3.5.1 Espera maleta
    - 3.5.2 Incrementa contador
  - 3.6 Pone contador en 1

Fin Mientras

FIN



### 3. Programa

```
#Programa Num 15

#Control de Banda Transportadora Distribuidora de Equipaje

from gpiozero import LED
from gpiozero import Button

bA = Button(5)
s1 = Button(19)
banda1 = LED(18)
banda2 = LED(23)
banda3 = LED(25)

def sensor():
    Si = s1.is_pressed
    while Si == 0:
        Si = s1.is_pressed
    while Si == 1:
        Si = s1.is_pressed
    return

#Inicio programa principal
print('Arranca programa, bandas detenidas')
cont = 0 #Inicializa contador de maletas
print('En espera de boton de arranque')
ON = bA.is_pressed
while ON == 0: #Espera a que se arranque el sistema
    ON = bA.is_pressed
banda1 = 1 #Arranca banda principal
print('Arrancan banda principal y banda delantera')
```

**Continúa...**

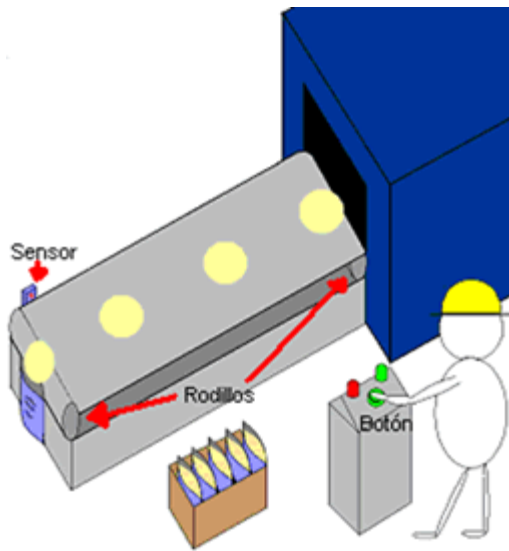
```
while ON == 1:
    banda2 = 1 #Arranca banda delantera
    print('Banda delantera encendida')
    banda3 = 0 #Banda trasera detenida
    while cont < 4:
        print('Espera maleta para banda 2')
        sensor()
        cont = cont +1
        print('contador = ',cont)
    cont = 1
    print('Arranca banda trasera')
    banda2 = 0 #Detiene banda delantera
    banda3 = 1 #Arranca banda trasera
    print('Banda trasera encendida ')
    while cont < 3:
        print('Espera maleta para banda 3')
        sensor()
        cont = cont +1
        print('contador = ',cont)
#fin while ba == 1
```

## SUGERENCIAS DIDÁCTICAS

Desarrollar programas que resuelvan problemas similares al programa #15. Los siguientes son problemas sugeridos, pero cada estudiante podrá plantearse otros que sean de su interés.

Diseñe, Codifique y Pruebe programas para:

### Ejercicio 15.1



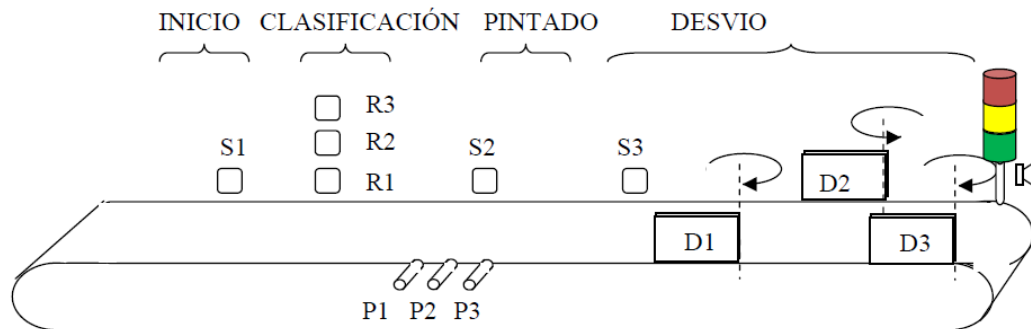
Se requiere transportar tortillas desde el horno de cocción hasta el área de embolsado en la cual se empaquetan 10 tortillas en cada bolsa, por ello la necesidad de utilizar un contador automatizado pues el operador puede tener una distracción y contar una tortilla de más o de menos.

Como se puede observar en el bosquejo para comenzar el proceso el operador pulsara un botón que es el que dará inicio y arrancara la producción de tortillas, indicando a la vez con el encendido de una lámpara de color verde que el proceso está en funcionamiento. Cada tortilla será detectada por el sensor situado al inicio de la banda transportadora, dicho sensor envía una señal a una entrada GPIO con lo que se incrementara el conteo del registro correspondiente (contador C0).

Cuando el contador llegue a 10 unidades originara una señal con la cual se detendrá la producción, esto es, cesara el movimiento de la banda transportadora y con ello se apagará la lámpara verde, encendiéndose una lámpara roja que indica el fin del proceso.

Una vez que el proceso contabilizo 10 unidades, el operador podrá colocar una bolsa nueva ya que la anterior acaba de ser llenada, las bolsas llenas son colocadas en una caja para su futura repartición, y nuevamente el operador tendrá que presionar el botón de inicio para que se llene automáticamente una bolsa con exactamente 10 unidades.

## Ejercicio 15.2 Descripción de Proceso.



El sistema se divide en cuatro zonas: Inicio, Clasificación, Pintura, Desviación. El proceso inicia al pulsar el interruptor ON, momento en que se activa la banda transportadora hacia adelante. Si después de 30 segundos el sistema no detecta presencia de piezas en la zona de inicio (S1 inactivo), la banda se desactiva y se enciende una alarma sonora (CH) durante 3 segundos, pasados estos 3 segundos el sistema va a estado de reposo. Si por el contrario se detecta la presencia de piezas en zona de inicio (S1) la banda permanece activa.

Una vez la pieza alcance la zona de clasificación (R1 o R2 o R3), el sistema debe detenerse durante 8 segundos, tiempo en el cual se realizará la clasificación de la pieza como: Pequeña, Mediana o Grande. Esta clasificación se realiza con los sensores R1, R2 y R3 los cuales han sido dispuestos de manera apropiada para tal fin. Después de los 8 segundos de espera la pieza clasificada debe ser trasladada hacia la zona de pintura. Cuando la pieza entre en la zona de pintura (S2), la banda se detiene. La pieza procederá a ser pintada de un color elegido con base en la clasificación de altura realizada. Para ello, el sistema contará con 3 dosificadores de pintura (P1-pequeño, P2-mediano y P3-grande). Las piezas grandes se pintarán durante 30 segundos, las medianas durante 20 segundos y las pequeñas durante 15 segundos. Habiendo terminado el proceso de pintado, la banda volverá a ser activada trasladando la pieza hacia la zona de Desviación.

Cuando se alcance la zona de desviación (S3) la banda se detiene. El operario evaluará la calidad de pintado y determinará:

En caso de encontrar que la pieza ha sido bien pintada, presionará el botón PINT\_OK provocando que la pieza avance en la zona de desviadores.

En caso de encontrar errores en el pintado, presionará el botón REPINT, esto provocará que la pieza en desviación retroceda hacia la zona de pintura y sea pintada nuevamente.

Esta rectificación de pintado solo se puede repetir en 3 ocasiones, la cuarta repetición tratará la caja como si estuviese bien pintada.

Habiendo pasado la evaluación de pintado se activará uno de tres desviadores (D1, D2, D3) que conducen la pieza a los contenedores de piezas grandes, pequeñas o medianas. El desvío va a depender de la clasificación realizada en la zona de clasificación: D1 para piezas pequeñas, D2 para medianas y D3 para grandes. Este estado de desviación durará hasta que el operador confirme el almacenamiento de la pieza pulsando el botón PIEZA\_ALMAC.

Una vez se presione el botón PIEZA\_ALMAC, los desviadores pasarán a reposo y el sistema podrá procesar una nueva pieza

**REPORTE DEL ALUMNO**

<b>Criterio para calificar</b>	<b>Si</b>	<b>No</b>
El documento tiene portada	10%	0
Entregó en fecha programada	10%	0
El documento está completo	20%	0
El programa funciona correctamente	20%	0
Incluye un ejercicio adicional (al menos)	20%	0
Incluye resultados y/o gráficas o tablas	20%	0

**BIBLIOGRAFÍA**

Raspberry Pi Foundation. Disponible en [www.raspberrypi.org](http://www.raspberrypi.org)

Lenguaje Python. Disponible en [www.python.org](http://www.python.org)

<ftp://ftp.unicauca.edu.co/Facultades/FIET/DEIC/Materias/LaboratoriodecontroldeProcesos/Planta/Banda/banda.pdf>

<https://instrumentacionycontrol.net/automatizacion-de-un-faja-transportadora-con-un-plc/>