

# ALGORITMO: CUESTIONARIO DE FÍSICA

$E0 = (1,2,3,4,5,6,7)$

GRAVEDAD = 9.81

## Calificación

funcion calificacion (respuesta\_t1n1\_1, respuesta\_t1n1\_2, respuesta\_t1n1\_3):

    cont = 0

    si respuesta\_t1n1\_1 == "a":

        cont= cont+1

        print ("Correcto")

    else:

        print ("Incorrecto")

    si respuesta\_t1n1\_2 == notacion\_cientifica(3890000):

        cont= cont+1

        print ("Correcto")

    else:

        print ("Incorrecto")

    si respuesta\_t1n1\_3 == notacion\_decimal ("01.7E+05"):

        cont= cont +1

        print ("Correcto")

    else:

        print ("Incorrecto")

```
calif = (cont*100/3)

print ("Tu calificación final es:", calif)
```

""" Se hará uso de condicionales donde se comprarán las respuestas del usuario con los resultados obtenidos en python, determinando si el resultado es correcto o incorrecto (le mostrará esto al usuario, y se le sumará ciertos puntos a la calificación final)"""

#Solo un ejemplo de cómo serán usadas las condicionales, agregar el uso de listas para guardar las respuestas dadas por el usuario

#Modificar cuando se encuentre una manera de hacerlo más corto y eficiente

## Tema 1\_Operaciones

# Notación científica

```
funcion notacion_cientifica(numero_nd):

    return format(numero_nd, ".1E")
```

```
funcion notacion_decimal(numero_nc):

    return float(numero_nc)
```

## Tema 1\_Preguntas

Notación científica

```
funcion tema1_dificultad1(tema_e, dificultad_e):
```

```
    print ("Tema 1: " + tema_e + "\n\nNivel " + dificultad_e)
```

```
respuesta_t1n1_1 = input("""
```

¿Para qué se utiliza la notación científica?

a) Para expresar cantidades muy grandes o muy pequeñas

b) Solo para expresar cantidades muy grandes

c) Solo para expresar cantidades muy pequeñas

```
""")
```

```
respuesta_t1n1_2 = input("""
```

¿Cuál es la notación científica de 3890000?

favor de expresar el valor como en el siguiente ejemplo:

1400 sería 1.4E+03

→ Solo un decimal

```
""")
```

Llamar función `notacion_cientifica(3890000)`

```
respuesta_t1n1_3 = float(input("""
```

Para el número en notación científica a notación decimal:

1.7E+05

""))

Llamar funcion notacion\_decimal ("0.17E+05")

funcion tema1\_dificultad2(tema\_e, dificultad\_e):

print ("Tema 1: " + tema\_e + "\n\nNivel " + dificultad\_e)

print ("La misma temática de nivel uno pero más difícil")

funcion tema1\_dificultad3(tema\_e, dificultad\_e):

print ("Tema 1: " + tema\_e + "\n\nNivel " + dificultad\_e)

print ("La misma temática de nivel uno, pero más difícil que uno y dos")

## Tema 2\_Operaciones

### MRU

funcion velocidad\_mru(distancia\_mru, tiempo\_mru):

return "{:.2f}".format(distancia\_mru/tiempo\_mru))

funcion distancia\_mru(velocidad\_mru, tiempo\_mru):

return "{:.2f}".format(velocidad\_mru\*tiempo\_mru))

```
funcion tiempo_mru(distancia_mru, velocidad_mru):  
    return "{:.2f}".format(distancia_mru/velocidad_mru))
```

## MRUA

"""Tiempo"""

```
funcion tiempo_mrva_vfvia (velocidad_f_mrva, velocidad_i_mrva, aceleracion_mrva):  
    return "{:.2f}".format((velocidad_f_mrva-  
velocidad_i_mrva)/aceleracion_mrva))
```

```
funcion tiempo_mrva_vfvid (velocidad_f_mrva, velocidad_i_mrva, distancia_mrva):  
    return "{:.2f}".format(((distancia_mrva/((velocidad_f_mrva+  
velocidad_i_mrva)/2))))
```

```
funcion tiempo_mrva_da (distancia_mrva, aceleracion_mrva):  
    return "{:.2f}".format(((2* distancia_mrva)/aceleracion_mrva)**(1/2)))
```

"""Distancia"""

```
funcion distancia_mrva_vita (velocidad_i_mrva, tiempo_mrva, aceleracion_mrva):  
    return "{:.2f}".format((velocidad_i_mrva * tiempo_mrva + ( (1/2)*  
aceleracion_mrva* tiempo_mrva**2))))
```

```
funcion distancia_mrva_vfvit (velocidad_f_mrva, velocidad_i_mrva, tiempo_mrva):  
    return "{:.2f}".format((((velocidad_f_mrva + velocidad_i_mrva) / 2) *  
tiempo_mrva)))
```

```
funcion distancia_mrva_vfvia (velocidad_f_mrva,  
velocidad_i_mrva,aceleracion_mrva):  
    return "{:.2f}".format((velocidad_f_mrva**2 - velocidad_i_mrva**2) /  
(2*aceleracion_mrva)))
```

#### ""Velocidad Inicial""

```
funcion velocidad_i_mrúa_vfat (velocidad_f_mrúa, aceleración_mrúa, tiempo_mrúa):  
    return "{:.2f}".format((velocidad_f_mrúa-(aceleración_mrúa* tiempo_mrúa))))
```

```
funcion velocidad_i_mrúa_vfda (velocidad_f_mrúa, distancia_mrúa,  
aceleración_mrúa):  
    return "{:.2f}".format((velocidad_f_mrúa**2 - 2*distancia_mrúa *  
aceleración_mrúa)**(1/2)))
```

#### ""Velocidad Final""

```
funcion velocidad_f_mrúa_viat (velocidad_i_mrúa, aceleración_mrúa, tiempo_mrúa):  
    return "{:.2f}".format((aceleración_mrúa*tiempo_mrúa + velocidad_i_mrúa)))
```

```
funcion velocidad_f_mrúa_viad (velocidad_i_mrúa, aceleración_mrúa, distancia_mrúa):  
    return "{:.2f}".format((2*distancia_mrúa * aceleración_mrúa +  
velocidad_i_mrúa**2)**(1/2)))
```

#### ""Aceleración""

```
funcion aceleración_mrúa_v0vft (velocidad_f_mrúa, velocidad_i_mrúa, tiempo_mrúa):  
    return "{:.2f}".format((velocidad_f_mrúa-velocidad_i_mrúa)/tiempo_mrúa))
```

```
funcion aceleración_mrúa_v0vfd (velocidad_f_mrúa, velocidad_i_mrúa,  
distancia_mrúa):  
    return "{:.2f}".format((velocidad_f_mrúa**2-velocidad_i_mrúa**2)/  
(2*distancia_mrúa)))
```

## Tema 2\_Preguntas

### MRU y MRUA

funcion tema2\_dificultad1(tema\_e, dificultad\_e):

```
print ("Tema 1: " + tema_e + "\n\nNivel " + dificultad_e)
```

```
respuesta_t2n1_1 = input("""
```

¿Cuál es la fórmula para calcular la velocidad en MRU?

a)  $v = d/t$

b)  $v = d*t$

c)  $v = d+t$

```
""")
```

```
respuesta_t2n1_2 = float(input("""
```

Calcula la velocidad de un auto que se mueve a 450 m en 120 segundos

(solo ingrese el número con dos decimales)

```
"""))
```

Llamar funcion velocidad\_mru(450.0, 120.0)

```
respuesta_t2n1_3 = float(input("""
```

¿Cuál es la aceleración de un objeto que de tener una velocidad de 0 m/s  
pasó a tener una velocidad de 45 m/s en 89 segundos?

(solo ingrese el número con dos decimales)

```
"""))
```

Llamar funcion aceleracion\_mrva\_v0vft (45.0, 0.0, 89.0)

```
funcion tema2_dificultad2(tema_e, dificultad_e):
```

```
    print ("Tema 1: " + tema_e + "\n\nNivel " + dificultad_e)
```

```
    print ("La misma tematica de nivel uno pero más difícil")
```

```
funcion tema2_dificultad3(tema_e, dificultad_e):
```

```
    print ("Tema 1: " + tema_e + "\n\nNivel " + dificultad_e)
```

```
    print ("La misma tematica de nivel uno pero más difícil que uno y dos")
```

Tema 3\_Operaciones

Caída Libre



""" Altura """

funcion altura\_cl\_t (tiempo\_cl):

return (" {:.2f} ".format((1/2)\* GRAVEDAD \*tiempo\_cl\*\*2))

funcion altura\_cl\_vit (velocidad\_i\_cl, tiempo\_cl):

return (" {:.2f} ".format((velocidad\_i\_cl\*tiempo\_cl)+(1/2)\* GRAVEDAD \*  
tiempo\_cl\*\*2))

funcion altura\_cl\_vivft (velocidad\_i\_cl, velocidad\_f\_cl, tiempo\_cl):

return (" {:.2f} ".format(((velocidad\_i\_cl + velocidad\_f\_cl)/2)\*tiempo\_cl))

""" Tiempo """

funcion tiempo\_cl\_a (altura\_cl):

return (" {:.2f} ".format(((2\*altura\_cl)/ GRAVEDAD)\*\*(1/2)))

funcion tiempo\_cl\_vivf (velocidad\_i\_cl, velocidad\_f\_cl):

return (" {:.2f} ".format((velocidad\_f\_cl-velocidad\_i\_cl)/ GRAVEDAD))

funcion tiempo\_cl\_vivfa (velocidad\_i\_cl, velocidad\_f\_cl, altura\_cl):

return (" {:.2f} ".format(altura\_cl/((velocidad\_i\_cl+velocidad\_f\_cl)/2)))

""" Velocidad Inicial """

funcion velocidad\_i\_cl\_vft (velocidad\_f\_cl, tiempo\_cl):

return (" {:.2f} ".format(velocidad\_f\_cl- GRAVEDAD \*tiempo\_cl))

```
funcion velocidad_i_cl_vfa (velocidad_f_cl, altura_cl):  
    return "{:.2f}".format((velocidad_f_cl**2-2* GRAVEDAD  
*altura_cl)**(1/2)))  
  
funcion velocidad_i_cl_at (altura_cl, tiempo_cl):  
    return "{:.2f}".format((altura_cl-(1/2)* GRAVEDAD  
*(tiempo_cl**2))/tiempo_cl))  
  
funcion velocidad_i_cl_vfat (velocidad_f_cl, altura_cl, tiempo_cl):  
    return "{:.2f}".format(((2*altura_cl)/tiempo_cl)-velocidad_f_cl))
```

""" Velocidad Final """

```
funcion velocidad_f_cl_vit (velocidad_i_cl, tiempo_cl):  
    return "{:.2f}".format(velocidad_i_cl+ GRAVEDAD *tiempo_cl))  
  
funcion velocidad_f_cl_via (velocidad_i_cl, altura_cl):  
    return "{:.2f}".format((velocidad_i_cl**2+2* GRAVEDAD  
*altura_cl)**(1/2)))  
  
funcion velocidad_f_cl_viat (velocidad_i_cl, altura_cl, tiempo_cl):  
    return "{:.2f}".format(((2*altura_cl)/tiempo_cl)-velocidad_i_cl))
```

### Tema 3\_Preguntas

#### Caída Libre

```
funcion tema3_dificultad1(tema_e, dificultad_e):  
  
    print ("Tema 1: " + tema_e + "\n\nNivel " + dificultad_e)
```

```
respuesta_t3n1_1 = input("""
```

"¿Cuál sería la mejor definición para caída libre?

a) Un objeto que cae

b) Cualquier objeto bajo la acción de la gravedad en un lugar donde  
la resistencia del aire es despreciable

c) Un objeto en caída libre

```
""")
```

```
respuesta_t3n1_2 = float(input("""
```

Calcula la altura de un objeto que cae en un tiempo de 67 segundos

[considera que tu velocidad inicial es 0 m/s y tu gravedad de 9.81 m/s<sup>2</sup>]

(solo ingrese el número con dos decimales)

```
"""))
```

Llamar funcion altura\_cl\_t (67.0)

```
respuesta_t3n1_3 = float(input("""
```

¿Cuál es la velocidad final de un objeto que recorre 450 m durante 20 segundos  
y que tiene una velocidad inicial de 13 m/s?

(solo ingrese el número con dos decimales)

""))

Llamar funcion velocidad\_f\_cl\_viat(13.0, 450.0, 20.0)

funcion tema3\_dificultad2(tema\_e, dificultad\_e):

print ("Tema 1: " + tema\_e + "\n\nNivel " + dificultad\_e)

print ("La misma temática de nivel uno pero más difícil")

funcion tema3\_dificultad3(tema\_e, dificultad\_e):

print ("Tema 1: " + tema\_e + "\n\nNivel " + dificultad\_e)

print ("La misma temática de nivel uno pero más difícil que uno y dos")

## Tema 4\_Operaciones

### Masa y Peso

funcion masa\_mp (peso):

return (" {:.2f} ".format(peso/ GRAVEDAD))

funcion peso\_mp (masa):

return (" {:.2f} ".format(masa\* GRAVEDAD))

## Tema 4\_Preguntas

### Masa y Peso

funcion tema4\_dificultad1(tema\_e, dificultad\_e):

```
print ("Tema 1: " + tema_e + "\n\nNivel " + dificultad_e)
```

```
respuesta_t4n1_1 = input("""
```

La unidad de medida de la masa en el SI es:

a) gramos

b) onzas

c) Kilogramos

```
""")
```

```
respuesta_t4n1_2 = float(input("""
```

Calcula la masa de un automovil que pesa 17500 N

(solo ingrese el número con dos decimales)

```
"""))
```

Llamar funcion masa\_mp (17500.0)

```
respuesta_t4n1_3 = float(input("""
```

¿Cuál es el peso de una caja, cuando su masa es igual a 75 Kg?

(solo ingrese el número con dos decimales)

```
"""))
```

Llamar funcion peso\_mp (75.0)

```
funcion tema4_dificultad2(tema_e, dificultad_e):
```

```
    print ("Tema 1: " + tema_e + "\n\nNivel " + dificultad_e)
```

```
    print ("La misma temática de nivel uno pero más difícil")
```

```
funcion tema4_dificultad3(tema_e, dificultad_e):
```

```
    print ("Tema 1: " + tema_e + "\n\nNivel " + dificultad_e)
```

```
    print ("La misma temática de nivel uno pero más difícil que uno y dos")
```

## Tema 5\_Operaciones

### Leyes de Newton

```
funcion fuerza_neta (masa_n, aceleracion_n):
```

```
    return (" {:.2f} ".format(masa_n *aceleracion_n))
```

```
funcion masa_2n (fuerza_n, aceleracion_n):
```

```
return "{:.2f}".format(fuerza_n/aceleracion_n))
```

```
funcion aceleracion_2n (fuerza_n, masa_n):
```

```
    return "{:.2f}".format(fuerza_n/masa_n))
```

## Tema 5\_Preguntas

### Leyes de Newton

```
funcion tema5_dificultad1(tema_e, dificultad_e):
```

```
    print ("Tema 1: " + tema_e + "\n\nNivel " + dificultad_e)
```

```
    respuesta_t5n1_1 = input("""
```

"A toda acción corresponde una reacción de igual magnitud,  
pero en sentido contrario"

¿De qué ley estamos hablando?

a) 1ra Ley de Newton

b) 2da Ley de Newton

c) 3ra Ley de Newton

```
""")
```

```
    respuesta_t5n1_2 = float(input("""
```

¿Cuál es la Fuerza Neta de un objeto con una masa= 40 Kg y una aceleración = 3.5 m/s<sup>2</sup>?

(solo ingrese el número con dos decimales)

""))

Llamar funcion fuerza\_neta (40.0,3.5)

respuesta\_t5n1\_3 = input("""

¿Qué otro nombre se le da a la primera Ley de Newton?

a) Karma

b) Ley de la Inercia

c) Ley de los objetos estáticos

""")

funcion tema5\_dificultad2(tema\_e, dificultad\_e):

print ("Tema 1: " + tema\_e + "\n\nNivel " + dificultad\_e)

print ("La misma temática de nivel uno pero más difícil")

funcion tema5\_dificultad3(tema\_e, dificultad\_e):

print ("Tema 1: " + tema\_e + "\n\nNivel " + dificultad\_e)

print ("La misma temática de nivel uno pero más difícil que uno y dos")



funcion tema\_y\_nivel ():

```
tema = int(input("""
```

1.- Notación científica

2.- MRU y MRUA

3.- Caída Libre

4.- Masa y Peso

5.- Leyes de Newton

```
"""))
```

si tema < 1 or tema > 5:

```
tema = int(input("""
```

Lo siento el número seleccionado no está en los parámetros,  
por favor ingrese un número que este dentro de los parámetros:

1.- Notación científica

2.- MRU y MRUA

3.- Caída Libre

4.- Masa y Peso

5.- Leyes de Newton

```
"""))
```

```
dificultad = int(input("""
```

¿Qué nivel deseas completar? Escribe el número:

1.- Sencillo

2.- Normal

3.- Avanzado

```
"""))
```

si dificultad < 0 or dificultad > 3:

```
    dificultad = int(input("""
```

Lo siento el número seleccionado no está en los parámetros,

por favor ingrese un número que este dentro de los parámetros:

1.- Sencillo

2.- Normal

3.- Avanzado

```
"""))
```

si tema == 1 and dificultad == 1:

```
    tema1_dificultad1("Notación Cinética", "1")
```

sino tema == 1 and dificultad == 2:

```
    tema1_dificultad2("Notación Cinética", "2")
```

sino tema == 1 and dificultad == 3:

```
    tema1_dificultad3 ("Notación Cinética", "3")
```

sino tema == 2 and dificultad == 1:

tema2\_dificultad1("MRU y MRUA", "1")

sino tema == 2 and dificultad == 2:

tema2\_dificultad2("MRU y MRUA", "2")

sino tema == 2 and dificultad == 3:

tema2\_dificultad3("MRU y MRUA", "3")

sino tema == 3 and dificultad == 1:

tema3\_dificultad1("Caída Libre", "1")

sino tema == 3 and dificultad == 2:

tema3\_dificultad2("Caída Libre", "2")

sino tema == 3 and dificultad == 3:

tema3\_dificultad3("Caída Libre", "3")

sino tema == 4 and dificultad == 1:

tema4\_dificultad1("Masa y Peso", "1")

sino tema == 4 and dificultad == 2:

tema4\_dificultad2("Masa y Peso", "2")

sino tema == 4 and dificultad == 3:

tema4\_dificultad3("Masa y Peso", "3")

sino tema == 5 and dificultad == 1:

tema5\_dificultad1("Leyes de Newton", "1")

sino tema == 5 and dificultad == 2:

tema5\_dificultad2("Leyes de Newton", "2")

sino tema == 5 and dificultad == 3:

tema5\_dificultad3("Leyes de Newton", "3")

EF = (" Tú calificación es: " + (lo que haya en funcion calificacion))

##### Algoritmo principal #####

print ("Hola usuario! ¿Qué tema deseas repasar? Por favor ingresa el número")

tema\_y\_nivel ()

continuar\_p = input("¿Deseas repasar otro tema?\n a)Sí\n b)No\n")

while continuar\_p == "a" or continuar\_p == "A":

llamar funcion tema\_y\_nivel ()

continuar\_p = input("¿Deseas repasar otro tema?\n a)Sí\n b)No\n")

EF ("¡Gracias por repasar en este programa!")