

# ALGORITMO: CUESTIONARIO DE FÍSICA

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

global nivel

global calificacionfinal

calificacionfinal = 0

global buscar

global velocidad\_mru

global distancia\_mru

global tiempo\_mru

global velocidad\_i\_mrue

global velocidad\_f\_mrue

global tiempo\_mrue

global distancia\_mrue

global aceleración\_mrue

global gravedad\_cl

global tiempo\_cltv

global velocidad\_i\_cltv

global velocidad\_f\_cltv

global altura\_cltv

global masa

global peso

global aceleracion\_2ln

global fuerza

buscar = “ ”

velocidad\_mru = 0.0

distancia\_mru = 0.0

tiempo\_mru = 0.0

velocidad\_i\_mrui = 0.0

velocidad\_f\_mrui = 0.0

tiempo\_mrui = 0.0

distancia\_mrui = 0.0

aceleración\_mrui = 0.0

gravedad\_cl = 9.81

tiempo\_cltv = 0.0

velocidad\_i\_cltv = 0.0

velocidad\_f\_cltv = 0.0

altura\_cltv = 0.0

masa = 0.0

peso = 0.0

aceleracion\_2ln = 0.0

fuerza = 0.0

tema = int(input (““““Hola usuario! Bienvenido a “(*Nombre del proyecto*)”, ¿Qué tema deseas repasar? Por favor ingresa el número:

- 1.- Notación científica
- 2.- MRU y MRUA
- 3.- Caída Libre
- 4.- Masa y Peso
- 5.- Leyes de Newton"""))

mientras tema != (1,2,3,4,5,6,7):

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"""))

si tema == 1:

llamar función nivel\_seleccionado()  
llamar función tema1 ()

si tema == 2:

llamar función nivel\_seleccionado()  
llamar función tema2 ()

si tema == 3:

llamar función nivel\_seleccionado()  
llamar función tema3 ()

si tema == 4:

llamar función nivel\_seleccionado()

llamar función tema4 ()

si tema == 5:

llamar función nivel\_seleccionado()

llamar función tema5 ()

función nivel\_seleccionado ():

nivel = int(input("¿Qué nivel deseas completar? Escribe el número:

1.- Sencillo

2.- Normal

3.- Avanzado"))))

mientras nivel != (1,2,3):

nivel = 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"))))

función tema1 ():

si nivel ==1:

respuestat1n1\_1 = input("Pregunta \n")

llamar función respuesta\_Cadena\_(*Número de tema/temas*)\_(*Número de  
nivel/niveles*)\_(*Número de pregunta/preguntas*) ()

respuestat1n1\_1 = float(input("Pregunta \n"))

Depende del problema  
que se haya planteado

$\left\{ \begin{array}{l} \texttt{guardar\_nnd} = \texttt{float}(\textit{"Guardar el número decimal"}) \\ \texttt{guardar\_nnc\_cadena} = (\textit{"Guardar el número en notación científica como un string"}) \end{array} \right.$

llamar función `notacion_cientifica_p()` o `notacion_cientifica_d()` → Según corresponda

si `nivel == 2`:

`respuestat1n2_1 = input("Pregunta \n")`

llamar función `respuesta_Cadena_(Número de tema/temas)_(Número de nivel/niveles)_(Número de pregunta/preguntas)()`

`respuestat1n2_1 = float(input("Pregunta \n"))`

Depende del problema  
que se haya planteado

$\left\{ \begin{array}{l} \texttt{guardar\_nnd} = \texttt{float}(\textit{"Guardar el número decimal"}) \\ \texttt{guardar\_nnc\_cadena} = (\textit{"Guardar el número en notación científica como un string"}) \end{array} \right.$

llamar función `notacion_cientifica_p()` o `notacion_cientifica_d()` → Según corresponda

si `nivel == 3`:

`respuestat1n3_1 = input("Pregunta \n")`

llamar función `respuesta_Cadena_(Número de tema/temas)_(Número de nivel/niveles)_(Número de pregunta/preguntas)()`

`respuestat1n3_1 = float(input("Pregunta \n"))`

Depende del problema  
que se haya planteado

$\left\{ \begin{array}{l} \texttt{guardar\_nnd} = \texttt{float}(\textit{"Guardar el número decimal"}) \end{array} \right.$

*guardar\_nnc\_cadena = ("Guardar el número en notación científica como un string")*

llamar función `notacion_cientifica_p ()` o `notacion_cientifica_d ()` → Según corresponda

función `tema2 ()`:

si `nivel ==1`:

`respuestat2n1_1 = input("Pregunta \n")`

llamar función `respuesta_Cadena_(Número de tema/temas)_(Número de nivel/niveles)_(Número de pregunta/preguntas) ()`

`respuestat2n1_2 = float(input("Pregunta \n"))`

`buscar = ("distancia / tiempo / velocidad inicial / velocidad final / aceleración")` ← Según corresponda

*Se modifican los valores de las variables que aparecen en la pregunta*

llamar función `mru ()` o `funcion mrua ()` → Según corresponda

si `nivel ==2`:

`respuestat2n2_1 = input("Pregunta \n")`

llamar función `respuesta_Cadena_(Número de tema/temas)_(Número de nivel/niveles)_(Número de pregunta/preguntas) ()`

`respuestat2n2_2 = float(input("Pregunta \n"))`

`buscar = ("distancia / tiempo / velocidad inicial / velocidad final / aceleración")` ← Según corresponda

*Se modifican los valores de las variables que aparecen en la pregunta*

llamar función `mru ()` o `funcion mrua ()` → Según corresponda

si `nivel ==3`:

respuestat2n3\_1 = input("Pregunta \n")

llamar función respuesta\_Cadena\_(Número de tema/temas)\_(Número de nivel/niveles)\_(Número de pregunta/preguntas) ()

respuestat2n3\_2 = float(input("Pregunta \n"))

buscar = ("distancia / tiempo / velocidad inicial / velocidad final / aceleración") ← Según corresponda

Se modifican los valores de las variables que aparecen en la pregunta

llamar funcion mru () o funcion mrua () → Según corresponda

función tema3 ():

si nivel ==1:

respuestat3n1\_1 = input("Pregunta \n")

llamar función respuesta\_Cadena\_(Número de tema/temas)\_(Número de nivel/niveles)\_(Número de pregunta/preguntas) ()

respuestat3n1\_2 = float(input("Pregunta \n"))

buscar = ("altura / tiempo / velocidad inicial / velocidad final") ← Según corresponda

Se modifican los valores de las variables que aparecen en la pregunta

llamar funcion caida\_libre ()

si nivel ==2:

respuestat3n2\_1 = input("Pregunta \n")

llamar función respuesta\_Cadena\_(Número de tema/temas)\_(Número de nivel/niveles)\_(Número de pregunta/preguntas) ()

respuestat3n2\_2 = float(input("Pregunta \n"))

buscar = (“altura / tiempo / velocidad inicial / velocidad final”) ← Según corresponda

Se modifican los valores de las variables que aparecen en la pregunta

llamar funcion caida\_libre ()

si nivel ==3:

respuestat3n3\_1 = input(“Pregunta \n”)

llamar función respuesta\_Cadena\_(Número de tema/temas)\_(Número de nivel/niveles)\_(Número de pregunta/preguntas) ()

respuestat3n3\_2 = float(input(“Pregunta \n”))

buscar = (“altura / tiempo / velocidad inicial / velocidad final”) ← Según corresponda

Se modifican los valores de las variables que aparecen en la pregunta

llamar funcion caida\_libre ()

función tema4 ():

si nivel ==1:

respuestat4n1\_1 = input(“Pregunta \n”)

llamar función respuesta\_Cadena\_(Número de tema/temas)\_(Número de nivel/niveles)\_(Número de pregunta/preguntas) ()

respuestat4n1\_2 = float(input(“Pregunta \n”))

buscar = (“masa / peso”) ← Según corresponda

Se modifican el valor de masa o peso, según sea el caso

llamar función masa\_y\_peso ()

si nivel ==2:



respuestat4n2\_1 = input("Pregunta \n")

llamar función respuesta\_Cadena\_(Número de tema/temas)\_(Número de nivel/niveles)\_(Número de pregunta/preguntas) ()

respuestat4n2\_2 = float(input("Pregunta \n"))

buscar = ("masa / peso") ← Según corresponda

Se modifican el valor de masa o peso, según sea el caso

llamar función masa\_y\_peso ()

si nivel ==3:

respuestat4n3\_1 = input("Pregunta \n")

llamar función respuesta\_Cadena\_(Número de tema/temas)\_(Número de nivel/niveles)\_(Número de pregunta/preguntas) ()

respuestat4n3\_2 = float(input("Pregunta \n"))

buscar = ("masa / peso") ← Según corresponda

Se modifican el valor de masa o peso, según sea el caso

llamar función masa\_y\_peso ()

función tema5 ():

si nivel ==1:

respuestat5n1\_1 = input("Pregunta \n")

llamar función respuesta\_Cadena\_(Número de tema/temas)\_(Número de nivel/niveles)\_(Número de pregunta/preguntas) ()

respuestat5n1\_2 = float(input("Pregunta \n"))

buscar = ("fuerza / aceleración / masa") ← Según corresponda

Se modifican el valor de fuerza, masa o aceleración 2ln, según sea el caso

llamar función 2\_ley\_newton ()

si nivel ==2:

respuestat5n2\_1 = input("Pregunta \n")

llamar función respuesta\_Cadena\_(Número de tema/temas)\_(Número de nivel/niveles)\_(Número de pregunta/preguntas) ()

respuestat5n2\_2 = float(input("Pregunta \n"))

buscar = ("fuerza / aceleración / masa") ← Según corresponda

Se modifican el valor de fuerza, masa o aceleración 2ln, según sea el caso

llamar función 2\_ley\_newton ()

si nivel ==3:

respuestat5n3\_1 = input("Pregunta \n")

llamar función respuesta\_Cadena\_(Número de tema/temas)\_(Número de nivel/niveles)\_(Número de pregunta/preguntas) ()

respuestat5n3\_2 = float(input("Pregunta \n"))

buscar = ("fuerza / aceleración / masa") ← Según corresponda

Se modifican el valor de fuerza, masa o aceleración 2ln, según sea el caso

llamar función 2\_ley\_newton ()

**Nota\_1:** Todas las variables marcadas al principio, ya que habrán sido modificada, en el mismo apartado en el que se hizo la pregunta, por el nuevo valor correspondiente necesario para la realizar la formula.

funcion notacion\_cientifica\_p ():

```
num_nc = format( guardar_nnd, “.1E”)
```

```
si respuesta(Número de tema)(Número de nivel)_(Número de pregunta) ==  
num_nc:
```

```
    imprimir (“Respuesta correcta, ¡Felicidades!”)
```

```
    calificacionfinal = calificacionfinal + int(Valor que tenga esa pregunta)
```

```
else:
```

```
    imprimir (“Respuesta incorrecta, la respuesta correcta sería”, Num_nc,  
    “”“”“porque:
```

```
    (Explicación corta del por qué es incorrecta) ”””””)
```

funcion notacion\_cientifica\_d ()

```
num_nd = float (guardar_nnc_cadena)
```

```
si respuesta(Número de tema)(Número de nivel)_(Número de pregunta) ==  
num_nd:
```

```
    imprimir (“Respuesta correcta, ¡Felicidades!”)
```

```
    calificacionfinal = calificacionfinal + int(Valor que tenga esa pregunta)
```

```
else:
```

```
    imprimir (“Respuesta incorrecta, la respuesta correcta sería”, Num_nc,  
    “”“”“porque:
```

```
    (Explicación corta del por qué es incorrecta) ”””””)
```

funcion mru ():

si distancia\_mru > 0 and tiempo\_mru > 0:

velocidad\_mru = float (distancia\_mru/tiempo\_mru)

llamar funcion calificación\_f ()

llamar funcion clear ()

si tiempo\_mru > 0 and distancia\_mru == 0:

distancia\_mru = float (velocidad\_mru\* tiempo\_mru)

llamar funcion calificación\_f ()

llamar funcion clear ()

si distancia\_mru > 0 and tiempo\_mru == 0:

tiempo\_mru = float (distancia\_mru / velocidad\_mru)

llamar funcion calificación\_f ()

llamar funcion clear ()

funcion mrua ():

si buscar == "distancia":

si aceleración\_mrua != 0 and tiempo\_mrua >= 0:

distancia\_mrua = (velocidad\_i\_mrua \* tiempo\_mrua + ( (1/2)\*  
aceleración\_mrua\* tiempo\_mrua\*\*2))

llamar funcion calificación\_f ()

llamar funcion clear ()

si tiempo\_mrúa  $\geq$  0 and aceleracion\_mrúa ==0:

distancia\_mrúa = (((velocidad\_f\_mrúa + velocidad\_i\_mrúa) / 2) \*  
tiempo\_mrúa)

llamar funcion calificación\_f ()

llamar funcion clear ()

si aceleración\_mrúa  $\neq$  0 and tiempo\_mrúa == 0:

distancia\_mrúa = (velocidad\_f\_mrúa\*\*2 - velocidad\_i\_mrúa\*\*2) /  
(2\*aceleracion\_mrúa)

llamar funcion calificación\_f ()

llamar funcion clear ()

si buscar == “tiempo”:

si aceleración\_mrúa  $\neq$  0 and distancia\_mrúa == 0:

tiempo\_mrúa = (velocidad\_f\_mrúa - velocidad\_i\_mrúa) /  
aceleracion\_mrúa

llamar funcion calificación\_f ()

llamar funcion clear ()

si distancia\_mrúa  $\geq$  0 and aceleracion\_mrúa ==0:

tiempo\_mrúa = (distancia\_mrúa / ((velocidad\_f\_mrúa +  
velocidad\_i\_mrúa) / 2))

llamar funcion calificación\_f ()

llamar funcion clear ()

si aceleración\_mrúa != 0 and distancia\_mrúa >= 0 and velocidad\_i\_mrúa == 0:

tiempo\_mrúa = ((2\* distancia\_mrúa) / aceleración\_mrúa)\*\*(1/2)

llamar función calificación\_f ()

llamar función clear ()

si buscar == “velocidad inicial”:

si aceleración\_mrúa != 0 and tiempo\_mrúa >= 0:

velocidad\_i\_mrúa = (velocidad\_f\_mrúa - (aceleración\_mrúa \* tiempo\_mrúa))

llamar función calificación\_f ()

llamar función clear ()

si aceleración\_mrúa != 0 and distancia\_mrúa >= 0:

velocidad\_i\_mrúa = (velocidad\_f\_mrúa\*\*2 - 2\*distancia\_mrúa \* aceleración\_mrúa)\*\*(1/2)

llamar función calificación\_f ()

llamar función clear ()

si buscar == “velocidad final”:

si aceleración\_mrúa != 0 and tiempo\_mrúa >= 0:

velocidad\_f\_mrúa = aceleración\_mrúa\*tiempo\_mrúa + velocidad\_i\_mrúa

llamar función calificación\_f ()

llamar función clear ()

si aceleración\_mrúa != 0 and distancia\_mrúa >= 0:

velocidad\_f\_mrúa = (2\*distancia\_mrúa\* aceleración\_mrúa + velocidad\_i\_mrúa\*\*2) \*\* (1/2)

llamar funcion calificación\_f ()

llamar funcion clear ()

si buscar == “aceleración”:

si tiempo\_mrúa >= 0 and distancia == 0:

$$\text{aceleración\_mrúa} = (\text{velocidad\_f\_mrúa} - \text{velocidad\_i\_mrúa}) / \text{tiempo\_mrúa}$$

llamar funcion calificación\_f ()

llamar funcion clear ()

si distancia\_mrúa >= 0 and tiempo\_mrúa == 0:

$$\text{aceleración\_mrúa} = (\text{velocidad\_f\_mrúa}^2 - \text{velocidad\_i\_mrúa}^2) / (2 * \text{distancia\_mrúa})$$

llamar funcion calificación\_f ()

llamar funcion clear ()

funcion caida\_libre ():

si buscar == “altura”:

si tiempo\_cltv >= 0 and velocidad\_i\_cltv != 0 or velocidad\_f\_cltv != 0:

$$\text{altura\_cltv} = ((\text{velocidad\_i\_cltv} + \text{velocidad\_f\_cltv}) / 2) * \text{tiempo\_cltv}$$

llamar funcion calificación\_f ()

llamar funcion clear ()

si tiempo\_cltv >= 0 and velocidad\_i\_cltv != 0 or velocidad\_f\_cltv == 0:

$$\text{altura\_cltv} = (\text{velocidad\_i\_cltv} * \text{tiempo\_cltv}) + (1/2) * \text{gravedad\_cl} * \text{tiempo\_cltv}^2$$

llamar funcion calificación\_f ()

llamar funcion clear ()

si tiempo\_cltv >= 0 and or velocidad\_i\_cltv == 0:

altura\_cltv = (1/2) \* gravedad\_cl \* tiempo\_cltv\*\*2

llamar funcion calificación\_f ()

llamar funcion clear ()

si tiempo\_cltv == 0:

altura\_cltv = (velocidad\_f\_cltv\*\*2 – velocidad\_i\_cltv\*\*2) /  
(2\*gravedad\_cl)

llamar funcion calificación\_f ()

llamar funcion clear ()

si buscar == “tiempo”:

si altura\_cltv >= 0 and velocidad\_i\_cltv == 0:

tiempo\_cltv = ((2\*altura\_cltv) / gravedad\_cl)\*\*(1/2)

llamar funcion calificación\_f ()

llamar funcion clear ()

si altura\_cltv >= 0:

tiempo\_cltv = altura\_cltv / ((velocidad\_i\_cltv + velocidad\_f\_cltv) / 2)

llamar funcion calificación\_f ()

llamar funcion clear ()



si velocidad\_i\_cltv != 0 or velocidad\_f\_cltv != 0:

tiempo\_cltv = ( velocidad\_f\_cltv - velocidad\_i\_cltv ) / gravedad\_cl

llamar funcion calificación\_f ()

llamar funcion clear ()

si buscar == “velocidad inicial”:

si tiempo\_cltv >= 0 and altura\_cltv == 0:

velocidad\_i\_cltv = velocidad\_f\_cltv - gravedad\_cl \* tiempo\_cltv

llamar funcion calificación\_f ()

llamar funcion clear ()

si tiempo\_cltv == 0 and altura\_cltv >= 0:

velocidad\_i\_cltv = (velocidad\_f\_cltv\*\*2 - 2 \* gravedad\_cl \*  
altura\_cltv) \*\*(1/2)

llamar funcion calificación\_f ()

llamar funcion clear ()

si tiempo\_cltv >= and velocidad\_f\_cltv != 0 and altura\_cltv >= 0:

velocidad\_i\_cltv = (altura\_cltv - (1/2)\*gravedad\_cl \*  
(tiempo\_cltv\*\*2) ) / tiempo\_cltv

llamar funcion calificación\_f ()

llamar funcion clear ()

si tiempo\_cltv >= and altura\_cltv >= 0:

velocidad\_i\_cltv = ((2\*altura\_cltv) / tiempo\_cltv) - velocidad\_f\_cltv

llamar funcion calificación\_f ()

llamar funcion clear ()

si buscar == “velocidad final”:

si tiempo\_cltv >= 0 and altura\_cltv == 0:

velocidad\_f\_cltv = velocidad\_i\_cltv + gravedad\_cl \* tiempo\_cltv

llamar funcion calificación\_f ()

llamar funcion clear ()

si tiempo\_cltv == 0 and altura\_cltv >= 0:

velocidad\_f\_cltv = (velocidad\_i\_cltv \*\*2 + 2 \* gravedad\_cl \*  
altura\_cltv) \*\* (1/2)

llamar funcion calificación\_f ()

llamar funcion clear ()

si tiempo\_cltv >= and altura\_cltv >= 0:

velocidad\_f\_cltv = ((2\*altura\_cltv) / tiempo\_cltv) – velocidad\_i\_cltv

llamar funcion calificación\_f ()

llamar funcion clear ()

función masa\_y\_peso ():

si buscar == “masa”:

masa = peso / gravedad\_cl

llamar funcion calificación\_f ()

llamar funcion clear ()

si buscar == “peso”:

peso = masa \* gravedad\_cl

llamar funcion calificación\_f ()

llamar funcion clear ()

función 2\_ley\_newton ():

si buscar == “fuerza”:

fuerza = masa \* aceleracion\_2ln

llamar funcion calificación\_f ()

llamar funcion clear ()

si buscar == “aceleración”:

aceleracion\_2ln = fuerza / masa

llamar funcion calificación\_f ()

llamar funcion clear ()

si buscar == “masa”:

masa = fuerza / aceleracion\_2ln

llamar funcion calificación\_f ()

llamar funcion clear ()

funcion clear ():

buscar = “ ”

velocidad\_mru = 0.0

distancia\_mru = 0.0

tiempo\_mru = 0.0

velocidad\_i\_mrui = 0.0

velocidad\_f\_mrui = 0.0

tiempo\_mrui = 0.0

distancia\_mrui = 0.0

aceleración\_mrui = 0.0

```
tiempo_cltv = 0.0  
velocidad_i_cltv = 0.0  
velocidad_f_cltv = 0.0  
altura_cltv = 0.0  
  
masa = 0.0  
peso = 0.0  
  
aceleracion_2ln = 0.0  
fuerza = 0.0
```

funcion calificación\_f ():

```
si respuesta(Número de tema)(Número de nivel) _(Número de pregunta) ==  
(Resultado de la operación Correspondiente):  
    imprimir ("Respuesta correcta, ¡Felicidades!")  
    calificacionfinal = calificacionfinal + int(Valor que tenga esa pregunta)  
  
else:  
    imprimir ("Respuesta incorrecta, la respuesta correcta sería", operacion,  
    """"porque:  
    (Explicación corta del por qué es incorrecta) """"")
```

EF = print ("¡Hola de nuevo usuario! Tu calificación en esta prueba es:", calificacionfinal)

**Nota\_2: El nombre del proyecto está ampliamente sujeto al cambio**

**Las partes entre paréntesis y cursiva se modificarán de acuerdo con el nombre que reciba la función, dependiendo de que pregunta o preguntas responde, el tema y nivel en que están dichas preguntas**

respuesta(*Número de tema*)(*Número de nivel*)\_(*Número de pregunta/preguntas*) →  
respuestat5n1\_1