```
In[ ]:= (* Exercício 01 *)
        (* A tendência do sentido do movimento considerando as forças resistivas *)
        (* A partir da derivada de segunda ordem para a posição x,
        então obtemos a aceleração *)
        (* Para as forças resultantes em m1 e em m2 e suas
         respectivas acelerações a partir dos posições x1 e x2 *)
        (* O sistema é unico então podemos considerar a massa
           notação O
         reduzida e isso pode ser escrito a partir de Momento *)
        (* Se m1 e m2 são iguais e as molas são idênticas isso é
         ótimo porém vamos considerar que m1 e m2 são diferentes *)
        (* O referencial galileano O único que está na superfície vertical
           notação O                  notação O
         do desenho para facilitar na hora de escrever as equações *)

        (* Defining Variables and Functions *)
                        variáveis
        k
        m1
        m2
        ω1
        ω2
        Clear[k, m1, m2, ω1, ω2]
        apaga
        ω1 = sqrt (k / m1);
        ω2 = sqrt (k / m2);

        eq11 = (m1) * (x1''[t]) == - (k * x1[t]) + (k * x1[t])
        eq21 = (m2) * (x2''[t]) == - (k * x2 [t])
        sol1 = DSolve[{eq11, eq21, x1[0] == N[0], x2[0] == N[A], x1'[0] == N[0], x2'[0] == N[0]},
               resolve equação diferencial       valor numérico  valor numérico  valor numérico  valor numéri
            {x1[t], x2[t]}, t] // FullSimplify
                                  simplifica completamente
        eqf1 = sol1


        (* Defining ω as a function variable *)
        eq12 = (x1''[t]) == - ((ω1^(2)) * (x1[t])) + ((ω1^(2)) * (x1[t]))
        eq22 = (x2''[t]) == - ((ω2^(2)) * (x2[t]))
        sol2 = DSolve[{eq12, eq22, x1[0] == N[0], x2[0] == N[A], x1'[0] == N[0], x2'[0] == N[0]},
               resolve equação diferencial       valor numérico  valor numérico  valor numérico  valor numéri
            {x1[t], x2[t]}, t] // FullSimplify
                                  simplifica completamente
        eqf2 = sol2

        fEq1[ω1_, x1_] := - ((ω1^(2)) * (x1[t])) + ((ω1^(2)) * (x1[t]))
        fEq2[ω2_, x2_] := - ((ω2^(2)) * (x2[t]))
        sol3 = DSolve[{fEq1, fEq2, x1[0] == 0, x2[0] == N[A], x1'[0] == N[0], x1'[0] == N[0]},
               resolve equação diferencial               valor numérico  valor numérico  valor numérico
            {x1[t], x2[t]}, t] // FullSimplify
                                  simplifica completamente
```

```
                          simplifica completamente
Plot[sol3, {ω1, 1, 100}]
gráfico

Plot[fEq1[ω1], {ω1, 1, 100}, PlotLegends → Automatic,
gráfico                      legenda do gráficc automático
 PlotLabel → Style["TÍTULO DO GRÁFICO", FontSize → 20],
             estilo                     tamanho da fonte
 AxesLabel → {"Título Horizontal", "Título Vertical"},
 legenda dos eixos
 LabelStyle → Directive[Blue, Bold], Background → LightYellow]
 estilo de etiqueta diretiva  azul  negrito  imagem de fu···  amarelo claro
Plot[fEq2[ω2], {ω2, 1, 100}, PlotLegends → Automatic,
                             legenda do gráficc automático
 PlotLabel → Style["TÍTULO DO GRÁFICO", FontSize → 20],
 etiqueta de gr·· estilo                tamanho da fonte
 AxesLabel → {"Título Horizontal", "Título Vertical"},
 legenda dos eixos
 LabelStyle → Directive[Blue, Bold], Background → LightYellow]
              diretiva  azul  negrito  imagem de fu···  amarelo claro

(* Setting Variables and Graph Plot *)
   ajuste  variáveis     grafo gráfico
(* case 1 // ω1 = ω2 *)

(* case 2 // ω1<<ω2 *)

(* case 3 // ω1>>ω2 *)

(* Exercício 02 *)
(* Defining Variables and Functions *)
            variáveis
a
b
T
Clear[a, b, T]
apaga
σ[a_, b_, T_] := a + bT
SetAttributes[{a, b}, Constant]
                      constante
listamin = {{-8, 77.00}, {-5, 76.40}, {0, 75.60}, {5, 74.90},
  {10, 74.22}, {15, 73.49}, {18, 73.05}, {20, 72.75}, {30, 71.18}, {40, 69.56},
  {50, 67.91}, {60, 66.18}, {70, 64.40}, {80, 62.60}, {100, 58.90}}

ajuste = Fit[listamin, {1, T}, T]
         ajusta
Chop[ajuste]
substitui números pequenos por 0
graficoAjuste1 = Plot[ajuste, {T, -100, 100}, PlotLegends → Automatic,
                 gráfico                      legenda do gráficc automático
  PlotLabel → Style["σ x T", FontSize → 20], AxesLabel → {"T", "σ"},
              estilo         tamanho da fonte legenda dos eixos
```

```
                  estilo               tamanho da fonte     legenda dos eixos
  LabelStyle → Directive[Blue, Bold],  Background → LightYellow]
                  diretiva       azul   negrito   imagem de fu···  amarelo claro
Show[graficoAjuste1]
mostra
subs = {a → N[75.86552344833568], b → N[0.16462443654646322]}
                valor numérico                     valor numérico
meuGrafico = Plot[σ[t] /. subs, {T, -100, 100}, PlotLegends → Automatic,
                gráfico                          legenda do gráficc automático
  PlotLabel → Style["σ x T", FontSize → 20], AxesLabel → {"T", "σ"},
                  estilo               tamanho da fonte     legenda dos eixos
  LabelStyle → Directive[Blue, Bold],  Background → LightYellow]
                  diretiva       azul   negrito   imagem de fu···  amarelo claro
Show[meuGrafico]
mostra


(* Exercício 03 *)
(* Defining Variables and Functions *)
                variáveis
x
y
m
k
β
Clear[x, y, m, k , β]
apaga
eq31 =
 (m) * (x''[t]) == ((-1) * (k * x[t])) / (((((x) ^ (2)) + ((y) ^ (2))) ^ (1 / 2)) ^ (1 + β))
eq32 = (m) * (y''[t]) ==
  ((-1) * (k * x[t])) / (((((x) ^ (2)) + ((y) ^ (2))) ^ (1 / 2)) ^ (1 + β))
sol3 = DSolve[{eq31[t], eq31[t], x[0] == N[0], y[0] == N[0], x'[0] == N[0], y'[0] == N[0]},
          resolve equação diferencial      valor numérico  valor numérico  valor numérico  valor nun
   {x[β], y[β]}, β] // FullSimplify
                      simplifica completamente
subs1 = {m → N[10], k → N[10], a → N[10], β → 2}
              valor numérico valor numérico valor numérico
subs2 = {m → N[10], k → N[10], a → N[10], β → 3}
              valor numérico valor numérico valor numérico


(* Velocidade Circular *)
Vcirc = (((G * M) / (r)) ^ (1 / 2))


(* Velocidade de Escape *)
Vescape = (((N[2] * G * M) / (r)) ^ (1 / 2))
              valor numérico
(*Para Órbita Circular*)
v0 = Vcirc


(* Para Órbita Elíptica *)
(* Para Órbita Parabólica *)
(* Para Órbita Hiperbólica *)
```

```
graficoAjuste2 = ParametricPlot[Evaluate[{eq31[t], eq31'[t]} /. subs1],
                 gráfico paramétrico  calcula
  {t, 0, 100}, Evaluate[optam], PlotRange → All]
               calcula            intervalo do g·· tudo
Graphics[graficoAjuste2]
representação gráfica
```

Out[ ]= k

Out[ ]= m1

Out[ ]= m2

Out[ ]= $\frac{k\, sqrt}{m1}$

Out[ ]= $\frac{k\, sqrt}{m2}$

Out[ ]= $m1\, x1''[t] == 0$

Out[ ]= $m2\, x2''[t] == -k\, x2[t]$

Out[ ]= $\left\{\left\{x1[t] \to 0,\ x2[t] \to A \,Cos\left[\frac{\sqrt{k}\ t}{\sqrt{m2}}\right]\right\}\right\}$

Out[ ]= $\left\{\left\{x1[t] \to 0,\ x2[t] \to A \,Cos\left[\frac{\sqrt{k}\ t}{\sqrt{m2}}\right]\right\}\right\}$

Out[ ]= $x1''[t] == 0$

Out[ ]= $x2''[t] == -\frac{k^2\, sqrt^2\, x2[t]}{m2^2}$

Out[ ]= $\left\{\left\{x1[t] \to 0,\ x2[t] \to A \,Cos\left[\frac{k\, sqrt\, t}{m2}\right]\right\}\right\}$

Out[ ]= $\left\{\left\{x1[t] \to 0,\ x2[t] \to A \,Cos\left[\frac{k\, sqrt\, t}{m2}\right]\right\}\right\}$

DSolve: Equation or list of equations expected instead of fEq1 in the first argument {fEq1, fEq2, x1[0] == 0, x2[0] == A, x1′[0] == 0., x1′[0] == 0.}.

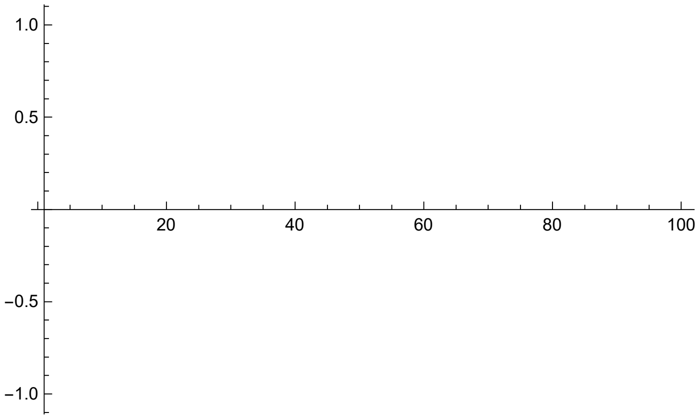
Out[ ]= DSolve[{fEq1, fEq2, x1[0] == 0, A == x2[0], x1′[0] == 0, x1′[0] == 0}, {x1[t], x2[t]}, t]

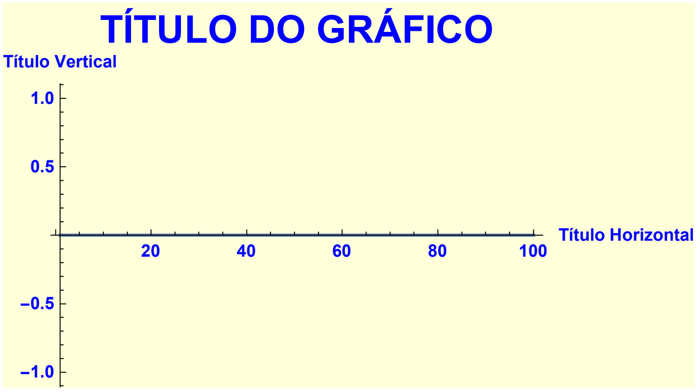
DSolve: Equation or list of equations expected instead of fEq1 in the first argument {fEq1, fEq2, x1[0.] == 0., A == x2[0.], x1′[0.] == 0., x1′[0.] == 0.}.

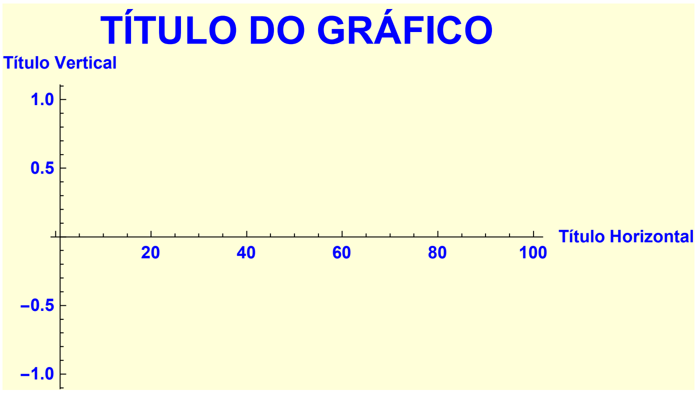
DSolve: Equation or list of equations expected instead of fEq1 in the first argument {fEq1, fEq2, x1[0.] == 0., A == x2[0.], x1′[0.] == 0., x1′[0.] == 0.}.

DSolve: Equation or list of equations expected instead of fEq1 in the first argument {fEq1, fEq2, x1[0.] == 0., A == x2[0.], x1′[0.] == 0., x1′[0.] == 0.}.

General: Further output of DSolve::deqn will be suppressed during this calculation.

Out[ ]=


Out[ ]=


Out[ ]=


Out[ ]= a

Out[ ]= b

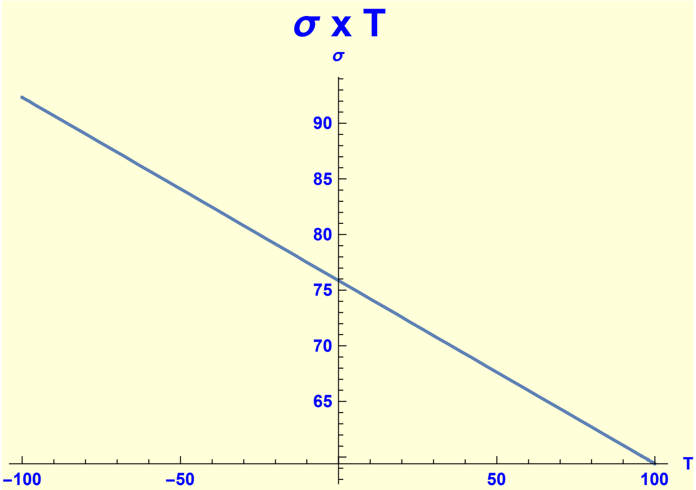
Out[ ]= T

Out[ ]= {{-8, 77.}, {-5, 76.4}, {0, 75.6}, {5, 74.9}, {10, 74.22}, {15, 73.49}, {18, 73.05}, {20, 72.75}, {30, 71.18}, {40, 69.56}, {50, 67.91}, {60, 66.18}, {70, 64.4}, {80, 62.6}, {100, 58.9}}

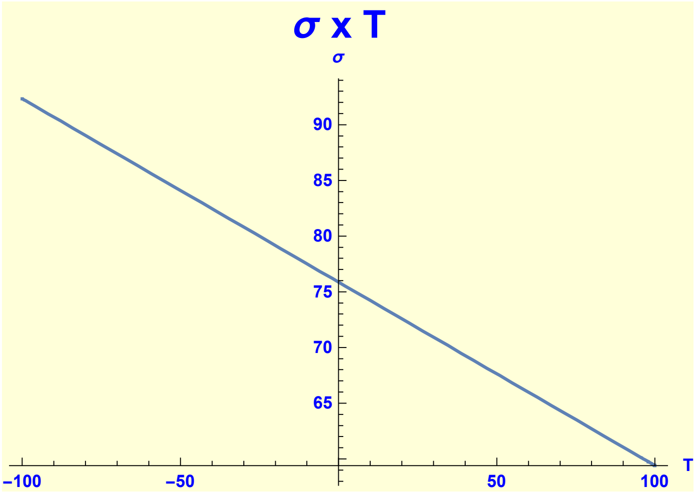
Out[ ]= 75.8655 - 0.164624 T

Out[ ]= 75.8655 - 0.164624 T

Out[ ]=

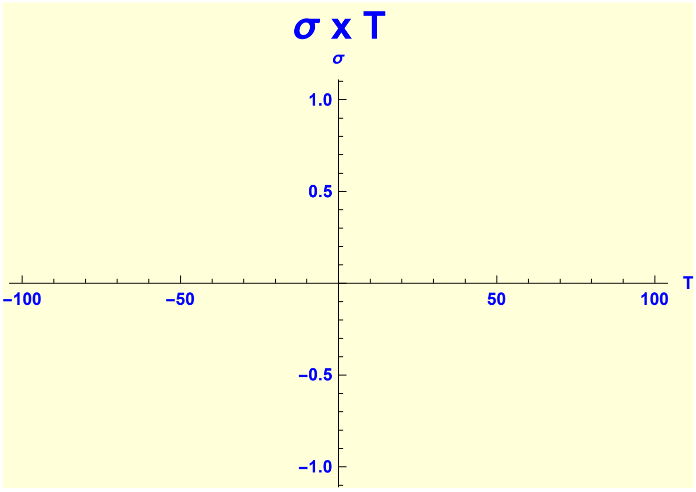


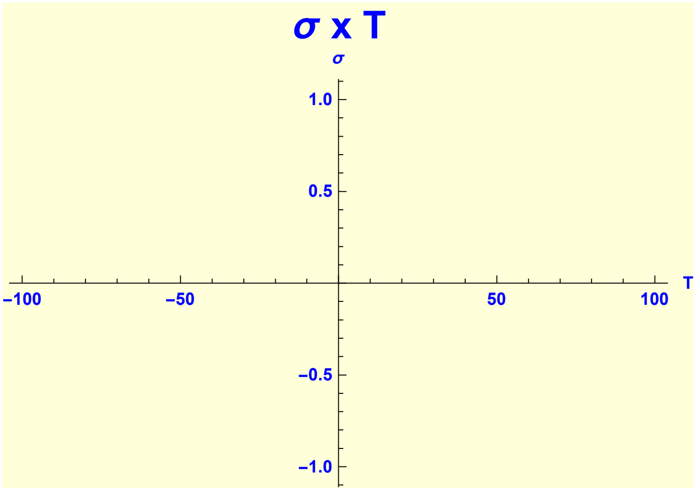
Out[ ]=



Out[ ]= {a → 75.8655, b → 0.164624}

Out[ ]=



Out[ ]= 

Out[ ]= x

Out[ ]= y

Out[ ]= m

Out[ ]= k

Out[ ]= β

Out[ ]= $m\,x''[t] == -k\,(x^2+y^2)^{\frac{1}{2}(-1-\beta)}\,x[t]$

Out[ ]= $m\,y''[t] == -k\,(x^2+y^2)^{\frac{1}{2}(-1-\beta)}\,x[t]$

••• DSolve: Equation or list of equations expected instead of $\left(m\,x''[t] == -k\,(x^2+y^2)^{\frac{1}{2}(-1-\beta)}\,x[t]\right)[t]$ in the first argument $\left\{\left(m\,x''[t] == -k\,(x^2+y^2)^{\frac{1}{2}(-1-\beta)}\,x[t]\right)[t],\ \left(m\,x''[t] == -k\,(x^2+y^2)^{\frac{1}{2}(-1-\beta)}\,x[t]\right)[t],\ x[0] == 0.,\ y[0] == 0.,\ x'[0] == 0.,\ y'[0] == 0.\right\}$.

Out[ ]= $\text{DSolve}\left[\left\{\left(k\,(x^2+y^2)^{-\frac{1}{2}-\frac{\beta}{2}}\,x[t]+m\,x''[t] == 0\right)[t],\ \left(k\,(x^2+y^2)^{-\frac{1}{2}-\frac{\beta}{2}}\,x[t]+m\,x''[t] == 0\right)[t],\ x[0] == 0,\ y[0] == 0,\ x'[0] == 0,\ y'[0] == 0\right\},\ \{x[\beta],\ y[\beta]\},\ \beta\right]$

Out[ ]= $\{m \to 10.,\ k \to 10.,\ a \to 10.,\ \beta \to 2\}$

Out[ ]= $\{m \to 10.,\ k \to 10.,\ a \to 10.,\ \beta \to 3\}$

Out[ ]= $\sqrt{\frac{G\,M}{r}}$

Out[ ]= $1.41421\sqrt{\frac{G\,M}{r}}$

Out[ ]= $\sqrt{\frac{G\,M}{r}}$

Out[ ]= System`SampledPlotsDump`iParametricPlot[

$$\left\{\left(10.\, x''[t] == -\frac{10.\, x[t]}{\left(x^2+y^2\right)^{3/2}}\right)[t],\ \left(10.\, x''[t] == -\frac{10.\, x[t]}{\left(x^2+y^2\right)^{3/2}}\right)'[t]\right\},$$

{t, 0, 100}, optam, PlotRange → All]

Out[ ]=