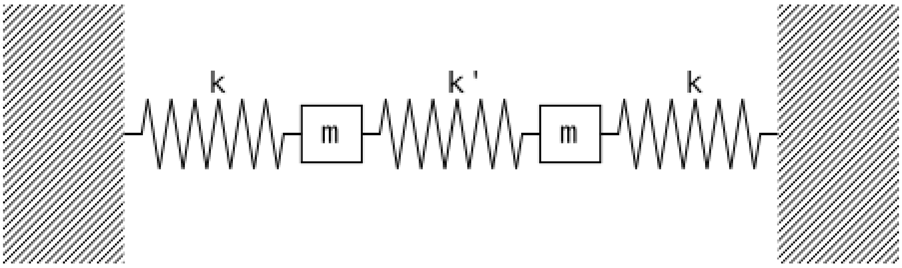
Modelação de Sistemas Físicos - Aula Prática nº12

Realização e resolução de problemas sobre Cap. 6:

- Osciladores acoplados

Exercício 1: Oscilador Acoplado

Considere dois corpos A e B acoplados através de uma mola de constante elástica k', e ligados a um ponto fixo através de molas de constante elástica k.



Considere:

$$k = 1\,\mathrm{N/m}$$

$$k' = 0.5\,\mathrm{N/m}$$

$$m = 1\,\mathrm{kg}$$

$$x_{\mathrm{A,eq}} = 1\,\mathrm{m}$$

$$x_{\mathrm{B,eq}} = 2\,\mathrm{m}$$

a) Encontre a lei de movimento dos dois corpos numericamente, para as seguintes condições iniciais:

i) $x_{\mathrm{A},t=0} = x_{\mathrm{A,eq}} + 0.3\,\mathrm{m}$, $x_{\{\mathrm{B}\},\mathrm{t=0}} = x_{\mathrm{B,eq}} + 0.3\,\mathrm{m}$, $v_{\mathrm{A},t=0} = v_{\mathrm{B,eq}} = 0\,\mathrm{m/s}$.

ii) $x_{\mathrm{A},t=0} = x_{\mathrm{A,eq}} + 0.3\,\mathrm{m}$, $x_{\{\mathrm{B}\},\mathrm{t=0}} = x_{\mathrm{B,eq}} - 0.3\,\mathrm{m}$, $v_{\mathrm{A},t=0} = v_{\mathrm{B,eq}} = 0\,\mathrm{m/s}$.

iii) $x_{\mathrm{A},t=0} = x_{\mathrm{A,eq}} + 0.3\,\mathrm{m}$, $x_{\{\mathrm{B}\},\mathrm{t=0}} = x_{\mathrm{B,eq}} - 0.1\,\mathrm{m}$, $v_{\mathrm{A},t=0} = v_{\mathrm{B,eq}} = 0\,\mathrm{m/s}$.

Solução do caso i):

$x_{\mathrm{A},t=0} = x_{\mathrm{A,eq}} + 0.3\,\mathrm{m}$, $x_{\{\mathrm{B}\},\mathrm{t=0}} = x_{\mathrm{B,eq}} + 0.3\,\mathrm{m}$, $v_{\mathrm{A},t=0} = v_{\mathrm{B},t=0} = 0\,\mathrm{m/s}$.

A força total na massa A que resulta do seu posicionamento arbitrario das massas nas coordendadas x_{A} e x_{B}, é dada por:

$$F_{\mathrm{A}} = -k(x_{\mathrm{A}} - x_{\mathrm{A,eq}}) - k'(x_{\mathrm{A}} - x_{\mathrm{A,eq}}) + k'(x_{\mathrm{B}} - x_{\mathrm{B,eq}}),$$

que pode ser simplificado:

$$F_\mathrm{A} = -(k+k')(x_\mathrm{A} - x_\mathrm{A,eq}) + k'(x_\mathrm{B} - x_\mathrm{B,eq}).$$

De forma analoga, para a massa B temos:

$$F_\mathrm{B} = -(k+k')(x_\mathrm{B} - x_\mathrm{B,eq}) + k'(x_\mathrm{A} - x_\mathrm{A,eq}).$$

Assim sendo, as acelerações das massas são:

$$a_\mathrm{A} = [k'(x_\mathrm{B} - x_\mathrm{B,eq}) - (k+k')(x_\mathrm{A} - x_\mathrm{A,eq})]/m$$

$$a_\mathrm{B} = [k'(x_\mathrm{A} - x_\mathrm{A,eq}) - (k+k')(x_\mathrm{B} - x_\mathrm{B,eq})]/m$$

É útil enfatizar as dependencias temporais:

$$a_{\mathrm{A},t} = [k'(x_{\mathrm{B},t} - x_\mathrm{B,eq}) - (k+k')(x_{\mathrm{A},t} - x_\mathrm{A,eq})]/m$$

$$a_{\mathrm{B},t} = [k'(x_{\mathrm{A},t} - x_\mathrm{A,eq}) - (k+k')(x_{\mathrm{B},t} - x_\mathrm{B,eq})]/m$$

ou de forma mais compacta:

$$a_{\mathrm{A},t} = [k'\Delta x_\mathrm{B} - (k+k')\Delta x_\mathrm{A}]/m$$

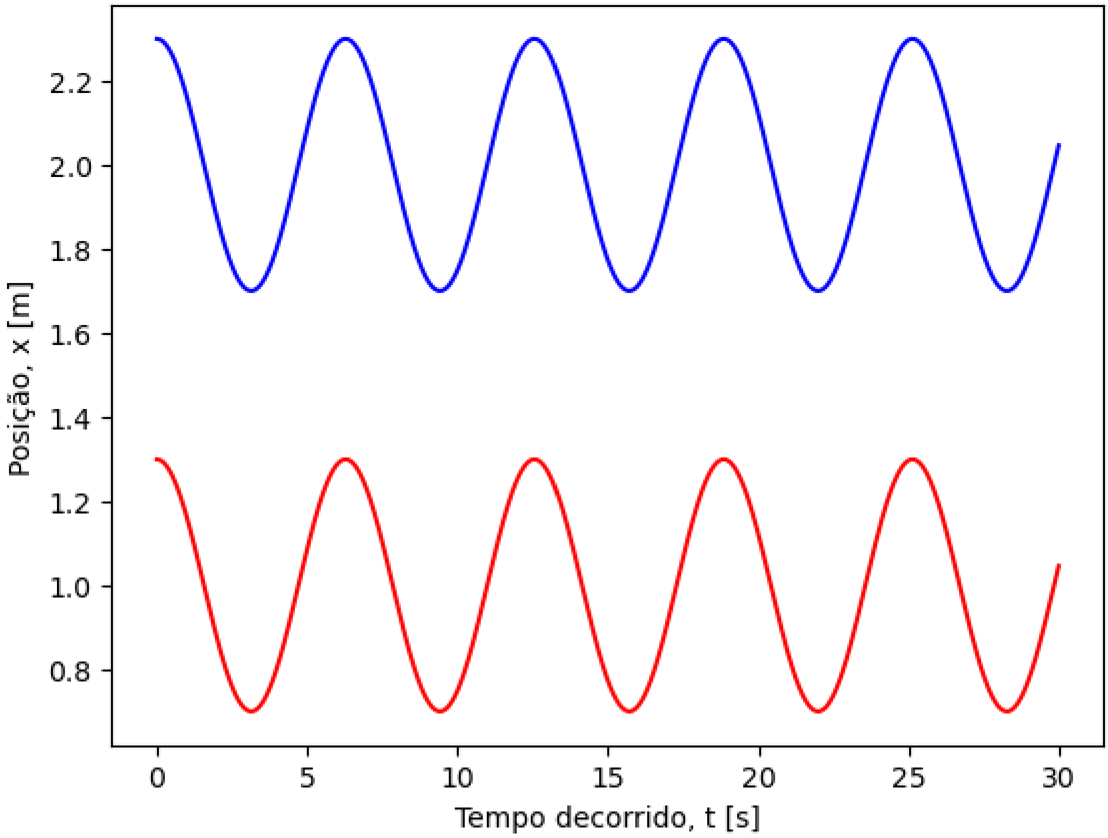
$$a_{\mathrm{B},t} = [k'\Delta x_\mathrm{A} - (k+k')\Delta x_\mathrm{B}]/m$$

em que $\Delta x_i = x_{\mathrm{i},t} - x_\mathrm{i,eq}$.

Podemos então usar o método de Euler-Cromer para obter a lei de movimento dos dois corpos:

In [1]:
```
import numpy as np
import matplotlib.pyplot as plt

# Domínio e amostragem temporal
t0 = 0.0                            # condição inicial, tempo [s]
tf = 200.0                          # limite do domínio, tempo final [s]
dt = 0.001                          # passo [s]
t = np.arange(t0, tf, dt)           # sequencia de tempo [s]

# Constantes physicas
k = 1.0                             # constante da mola entre massa e ponto fixo [N/
kp = 0.5                            # constante da mola entre massas [N/m]
m = 1.0                             # massa [kg]

# inicializar solução
N = 2                               # número de massas
xeq = np.zeros(N)                   # posição de equilíbrio das massas [m]
x = np.zeros((N, np.size(t)))       # posição [m]
v = np.zeros((N, np.size(t)))       # velocidade [m/s]
a = np.zeros((N, np.size(t)))       # aceleração [m/s2]

x1 = np.zeros((N, np.size(t)))      # posição das massas no problema (i) [m]
x2 = np.zeros((N, np.size(t)))      # posição das massas no problema (ii) [m]
x3 = np.zeros((N, np.size(t)))      # posição das massas no problema (iii) [m]

# condições iniciais
xeq = np.array([ 1.0, 2.0 ])        # posição de equilíbrio das massas [m]
x[:, 0] = xeq                       # condição inicial, posição inicial [m]
x[:, 0] += np.array([ +0.3, +0.3 ])
v[:, 0]  = np.array([ 0, 0 ])       # condição inicial, velocidade inicial [m/s]

# método de Euler-Cromer
for i in range(np.size(t) - 1):
```

```
    a[0, i] = ( kp * (x[1, i] - xeq[1]) - (k + kp) * (x[0, i] - xeq[0]) ) / m
    a[1, i] = ( kp * (x[0, i] - xeq[0]) - (k + kp) * (x[1, i] - xeq[1]) ) / m
    v[:, i + 1] = v[:, i] + a[:, i] * dt
    x[:, i + 1] = x[:, i] + v[:, i+1] * dt

# Reservamos as posições para mais tarde (Exercício 2)
x1 = x.copy()

# Encontrar o índice para t = 30 s
it=np.size(t[t<30])

plt.plot(t[0:it], x1[0, 0:it], 'r-', t[0:it], x1[1, 0:it], 'b-')
plt.xlabel("Tempo decorrido, t [s]")
plt.ylabel("Posição, x [m]")
plt.show()
```

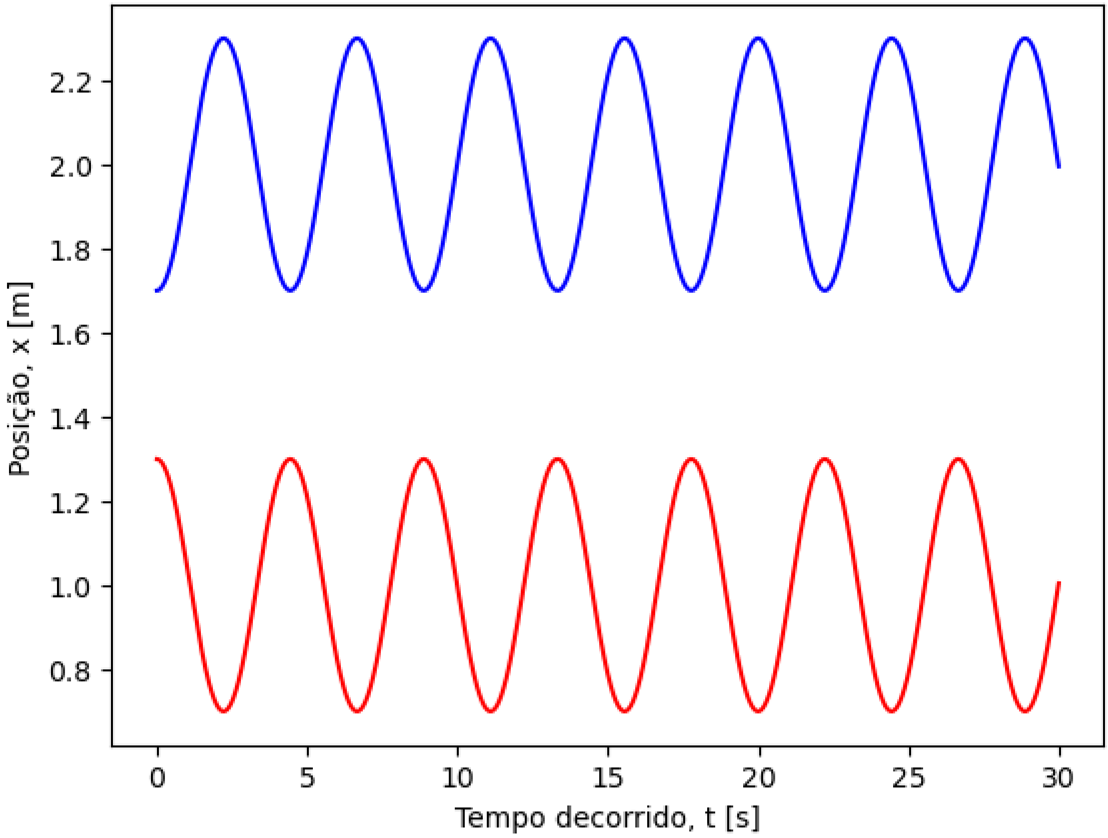


Solução do caso ii):

$x_{\mathrm{A},t=0} = x_{\mathrm{A,eq}} + 0.3\,\mathrm{m}$, $x_{\{\mathrm{B}\},\mathrm{t}=0} = x_{\mathrm{B,eq}} - 0.3\,\mathrm{m}$, $v_{\mathrm{A},t=0} = v_{\mathrm{B},t=0} = 0\,\mathrm{m/s}$.

In [2]:
```
# condições iniciais
xeq = np.array([ 1.0, 2.0 ])         # posição de equilíbrio das massas [m]
x[:, 0] = xeq                        # condição inicial, posição inicial [m]
x[:, 0] += np.array([ +0.3, -0.3 ])
v[:, 0]  = np.array([ 0, 0 ])        # condição inicial, velocidade inicial [m/s]

# método de Euler-Cromer
for i in range(np.size(t) - 1):
    a[0, i] = ( kp * (x[1, i] - xeq[1]) - (k + kp) * (x[0, i] - xeq[0]) ) / m
    a[1, i] = ( kp * (x[0, i] - xeq[0]) - (k + kp) * (x[1, i] - xeq[1]) ) / m
    v[:, i + 1] = v[:, i] + a[:, i] * dt
    x[:, i + 1] = x[:, i] + v[:, i+1] * dt

# Reservamos as posições para mais tarde (Exercício 2)
x2 = x.copy()

# Encontrar o índice para t = 30 s
```

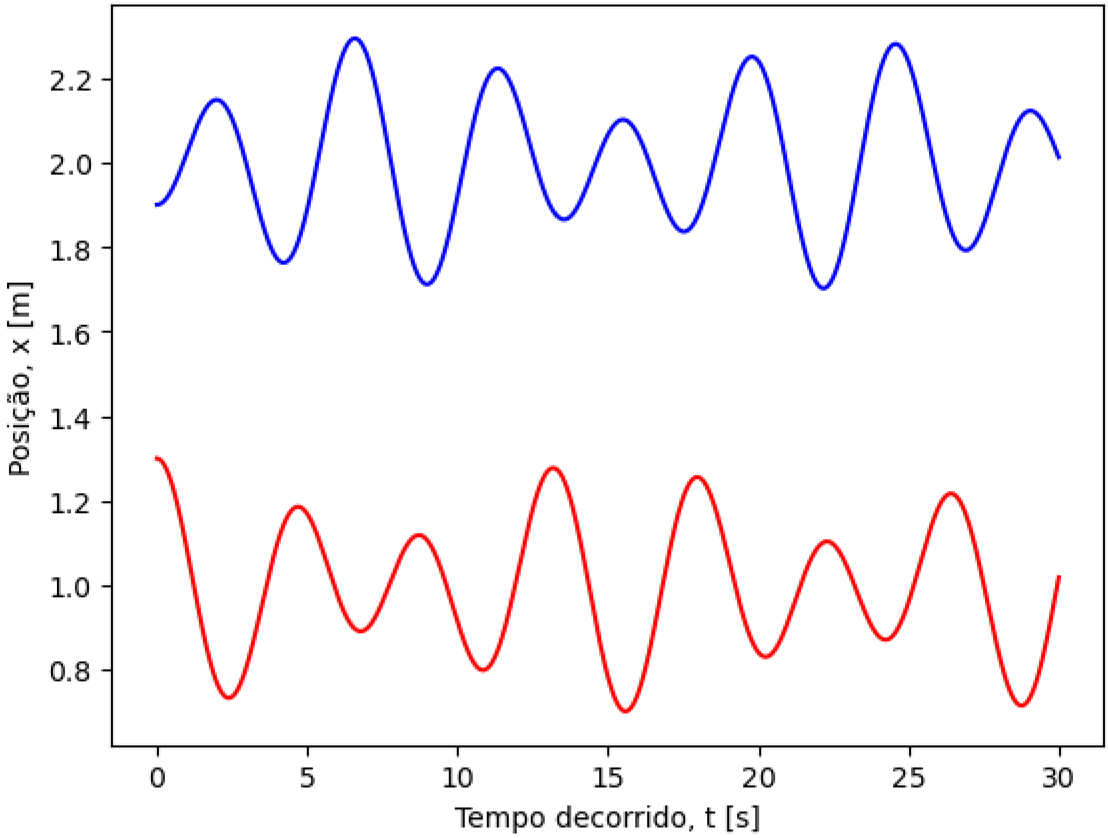
```
it=np.size(t[t<30])

plt.plot(t[0:it], x2[0, 0:it], 'r-', t[0:it], x2[1, 0:it], 'b-')
plt.xlabel("Tempo decorrido, t [s]")
plt.ylabel("Posição, x [m]")
plt.show()
```



Solução do caso iii):

$x_{\mathrm{A},t=0} = x_{\mathrm{A,eq}} + 0.3\,\mathrm{m}$, $x_{\{\mathrm{B}\},\mathrm{t=0}} = x_{\mathrm{B,eq}} - 0.1\,\mathrm{m}$, $v_{\mathrm{A},t=0} = v_{\mathrm{B},t=0} = 0\,\mathrm{m/s}$.

In [3]:
```
# condições iniciais
xeq = np.array([ 1.0, 2.0 ])         # posição de equilíbrio das massas [m]
x[:, 0] = xeq                        # condição inicial, posição inicial [m]
x[:, 0] += np.array([ +0.3, -0.1 ])
v[:, 0]  = np.array([ 0, 0 ])        # condição inicial, velocidade inicial [m/s]

# método de Euler-Cromer
for i in range(np.size(t) - 1):
    a[0, i] = ( kp * (x[1, i] - xeq[1]) - (k + kp) * (x[0, i] - xeq[0]) ) / m
    a[1, i] = ( kp * (x[0, i] - xeq[0]) - (k + kp) * (x[1, i] - xeq[1]) ) / m
    v[:, i + 1] = v[:, i] + a[:, i] * dt
    x[:, i + 1] = x[:, i] + v[:, i+1] * dt

# Reservamos as posições para mais tarde (Exercício 2)
x3 = x.copy()

# Encontrar o índice para t = 30 s
it=np.size(t[t<30])

plt.plot(t[0:it], x3[0, 0:it], 'r-', t[0:it], x3[1, 0:it], 'b-')
plt.xlabel("Tempo decorrido, t [s]")
plt.ylabel("Posição, x [m]")
plt.show()
```



b) Como carateriza o movimento dos corpos em cada um dos três casos?

- O primeiro caso (i) envolve o movimento em fase das duas massas, resultando na ação das duas molas ligadas ao ponto fixo com constante de força $k = 1.0\ \mathrm{N/m}$ (a mola central não atua);
- O segundo caso (ii) envolve o movimento em oposição de fase das dias massas, resultanda na ação das três molas;
- O terceiro caso (iii) é mais complexo e envolve a ação das três molas. Aparenta ser uma onda com a frequência do caso ii) modulada por uma aplitude de baixa frequência.

Pergunta 1:

Como seria o movimento das massas se k' fosse igual a zero?

Exercício 2: Oscilador Acoplado: Solução Analítica

O movimento do oscilador acoplado do exercício anterior pode ser descrito como uma sobreposição de dois modos normais, cada um sendo um movimento harmónico simples:

$$\begin{cases} x_{\mathrm{A}} = x_{\mathrm{A,eq}} + A_1 \cos(\omega_1 t + \phi_1) + A_2 \cos(\omega_2 t + \phi_2) \\ x_{\mathrm{B}} = x_{\mathrm{B,eq}} + A_1 \cos(\omega_1 t + \phi_1) - A_2 \cos(\omega_2 t + \phi_2) \end{cases}$$

As frequências angulares dos modos normais são:

$$\omega_1 = \sqrt{\frac{k}{m}} \text{ e } \omega_2 = \sqrt{\frac{k + 2k'}{m}}$$

a) Encontre os valores de A_1, A_2, ϕ_1 e ϕ_2 que correspondem a cada um dos três casos de exercício 1. Faça gráficos que comparam a solução teórica com a solução numérica para cada caso.

Solução:

Reescrevemos a equação das posições em termos de deslocamentos:

$$\begin{cases} \Delta x_\mathrm{A} = A_1 \cos(\omega_1 t + \phi_1) + A_2 \cos(\omega_2 t + \phi_2) \\ \Delta x_\mathrm{B} = A_1 \cos(\omega_1 t + \phi_1) - A_2 \cos(\omega_2 t + \phi_2) \end{cases}$$

onde $\Delta x_\mathrm{i} = x_\mathrm{i} - x_\mathrm{i,eq}$, e escrevemos as equações das velocidades respetivas (derivando no tempo):

$$\begin{cases} v_\mathrm{A} = -A_1\omega_1 \sin(\omega_1 t + \phi_1) - A_2\omega_2 \sin(\omega_2 t + \phi_2) \\ v_\mathrm{B} = -A_1\omega_1 \sin(\omega_1 t + \phi_1) + A_2\omega_2 \sin(\omega_2 t + \phi_2) \end{cases}$$

Em cada um dos sistemas, adicionamos e subtraimos ambos os membros das equações, para obter:

$$\begin{cases} \Delta x_\mathrm{A} + \Delta x_\mathrm{B} = 2A_1 \cos(\omega_1 t + \phi_1) \\ \Delta x_\mathrm{A} - \Delta x_\mathrm{B} = 2A_2 \cos(\omega_2 t + \phi_2) \end{cases}$$

e para as velocidades:

$$\begin{cases} v_\mathrm{A} + v_\mathrm{B} = -2A_1\omega_1 \sin(\omega_1 t + \phi_1) \\ v_\mathrm{B} - v_\mathrm{B} = -2A_2\omega_2 \sin(\omega_2 t + \phi_2) \end{cases}$$

É conveniente converter o sistema de equações das velocidades para unidades de distância, usando $\omega_i = 2\pi/T_i$ (poderemos assim, quando necessário, realizar operações algébricas diretas entre os membros):

$$\boxed{\begin{cases} \Delta x_\mathrm{A} + \Delta x_\mathrm{B} = 2A_1 \cos(\omega_1 t + \phi_1) \\ \Delta x_\mathrm{A} - \Delta x_\mathrm{B} = 2A_2 \cos(\omega_2 t + \phi_2) \\ (v_\mathrm{A} + v_\mathrm{B})T_1 = -4\pi A_1 \sin(\omega_1 t + \phi_1) \\ (v_\mathrm{B} - v_\mathrm{B})T_2 = -4\pi A_2 \sin(\omega_2 t + \phi_2) \end{cases}}$$

Para o caso (i), temos as seguintes condições iniciais:

$$\begin{cases} t = 0 \\ \Delta x_\mathrm{A} = \Delta x_\mathrm{B} = 0.3\,\mathrm{m} \\ v_\mathrm{A} = v_\mathrm{B} = 0\,\mathrm{m/s} \end{cases}$$

O sistema de equações consiste em:

$$\begin{cases} 0.6 = 2A_1 \cos\phi_1 \\ 0 = 2A_2 \cos\phi_2 \\ 0 = \sin\phi_1 \\ 0 = \sin\phi_2 \end{cases}$$

que se traduz nas seguintes soluções:

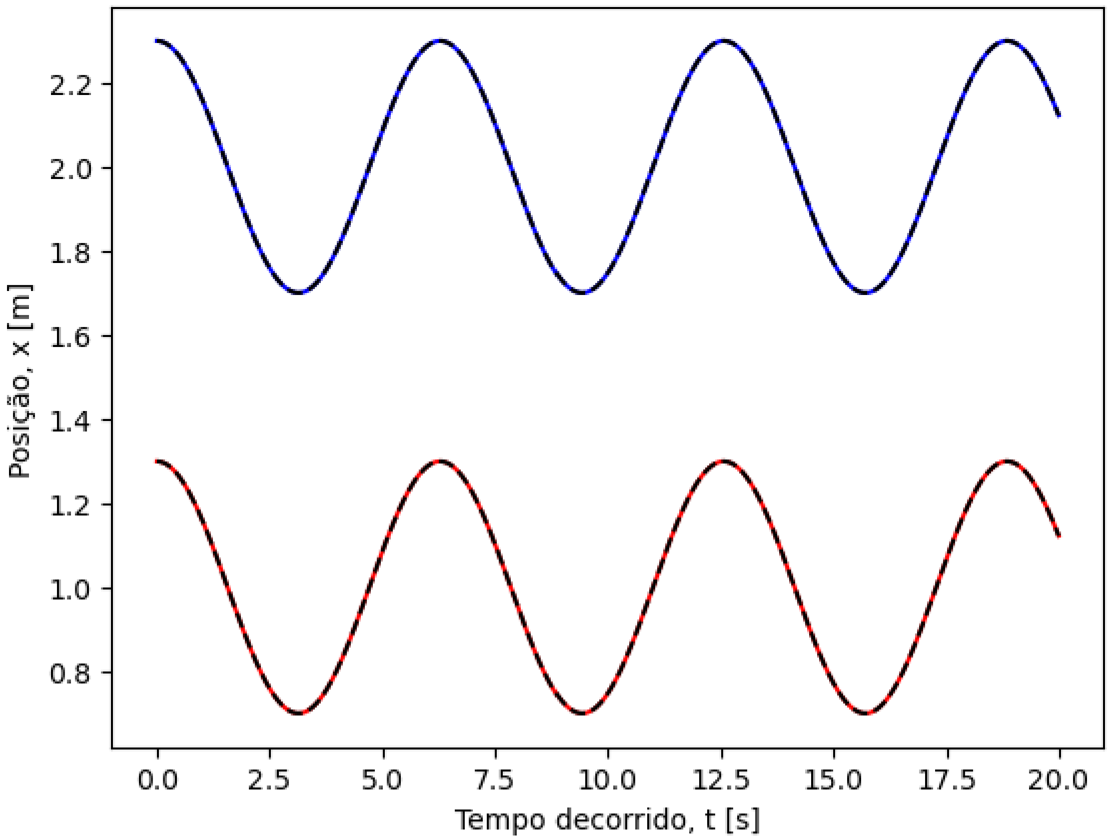
$$\begin{cases} A_1 = 0.3\,\mathrm{m} \\ A_2 = 0\,\mathrm{m} \\ \phi_1 = 0\,\mathrm{rad} \\ \phi_2 = 0\,\mathrm{rad} \end{cases}$$

Podemos agora codificar em Python, começando pela definição das frequências, assim como das posições (cáculo teórico):

In [4]:
```
ω1 = np.sqrt(k / m)
ω2 = np.sqrt((k + 2 * kp) / m)
xA_teorico = np.zeros(np.size(t))
xB_teorico = np.zeros(np.size(t))
```

Seguimos agora para a solução das equações de movimento (cálculo teórico):

In [5]:
```
A1 = 0.3
A2 = 0
φ1 = 0
φ2 = 0
xA_teorico = xeq[0] + A1 * np.cos(ω1 * t + φ1) + A2 * np.cos(ω2 * t + φ2)
xB_teorico = xeq[1] + A1 * np.cos(ω1 * t + φ1) - A2 * np.cos(ω2 * t + φ2)

# Encontrar o índice para t = 20 s
it=np.size(t[t<20])

plt.plot(t[0:it], x1[0, 0:it], 'r-', t[0:it], x1[1, 0:it], 'b-')
plt.plot(t[0:it], xA_teorico[0:it], 'k--', t[0:it], xB_teorico[0:it], 'k--')
plt.xlabel("Tempo decorrido, t [s]")
plt.ylabel("Posição, x [m]")
plt.show()
```



Para o caso (ii), temos as seguintes condições iniciais:

$$\begin{cases} t = 0 \\ \Delta x_\mathrm{A} = -\Delta x_\mathrm{B} = 0.3\,\mathrm{m} \\ v_\mathrm{A} = v_\mathrm{B} = 0\,\mathrm{m/s} \end{cases}$$

O sistema de equações consiste em:

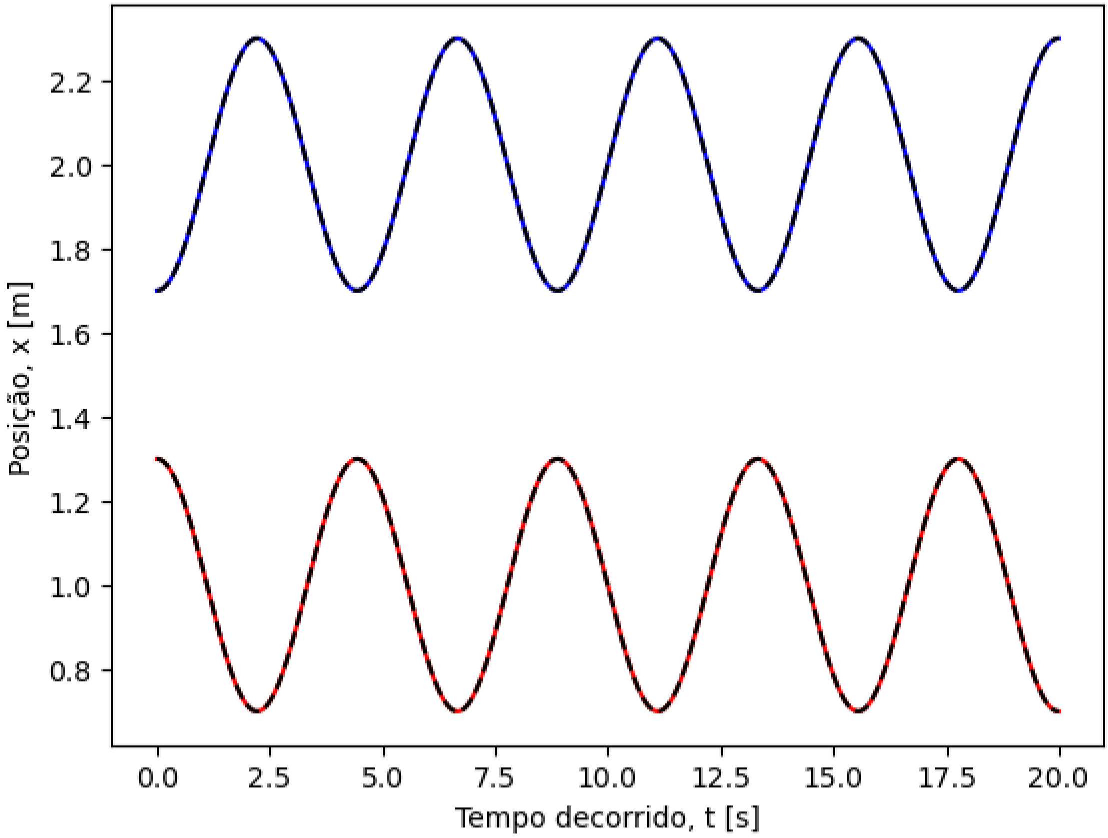
$$
\begin{cases}
0 = 2A_1 \cos\phi_1 \\
0.6 = 2A_2 \cos\phi_2 \\
0 = \sin\phi_1 \\
0 = \sin\phi_2
\end{cases}
$$

que se traduz nas seguintes soluções:

$$
\begin{cases}
A_1 = 0\,\mathrm{m} \\
A_2 = 0.3\,\mathrm{m} \\
\phi_1 = 0\,\mathrm{rad} \\
\phi_2 = 0\,\mathrm{rad}
\end{cases}
$$

In [6]:

```
A1 = 0.0
A2 = 0.3
φ1 = 0
φ2 = 0
xA_teorico = xeq[0] + A1 * np.cos(ω1 * t + φ1) + A2 * np.cos(ω2 * t + φ2)
xB_teorico = xeq[1] + A1 * np.cos(ω1 * t + φ1) - A2 * np.cos(ω2 * t + φ2)

# Encontrar o índice para t = 20 s
it=np.size(t[t<20])

plt.plot(t[0:it], x2[0, 0:it], 'r-', t[0:it], x2[1, 0:it], 'b-')
plt.plot(t[0:it], xA_teorico[0:it], 'k--', t[0:it], xB_teorico[0:it], 'k--')
plt.xlabel("Tempo decorrido, t [s]")
plt.ylabel("Posição, x [m]")
plt.show()
```



Para o caso (iii), temos as seguintes condições iniciais:

$$
\begin{cases}
t = 0 \\
\Delta x_{\mathrm{A}} = 0.3\,\mathrm{m} \\
\Delta x_{\mathrm{B}} = -0.1\,\mathrm{m} \\
v_{\mathrm{A}} = v_{\mathrm{B}} = 0\,\mathrm{m/s}
\end{cases}
$$

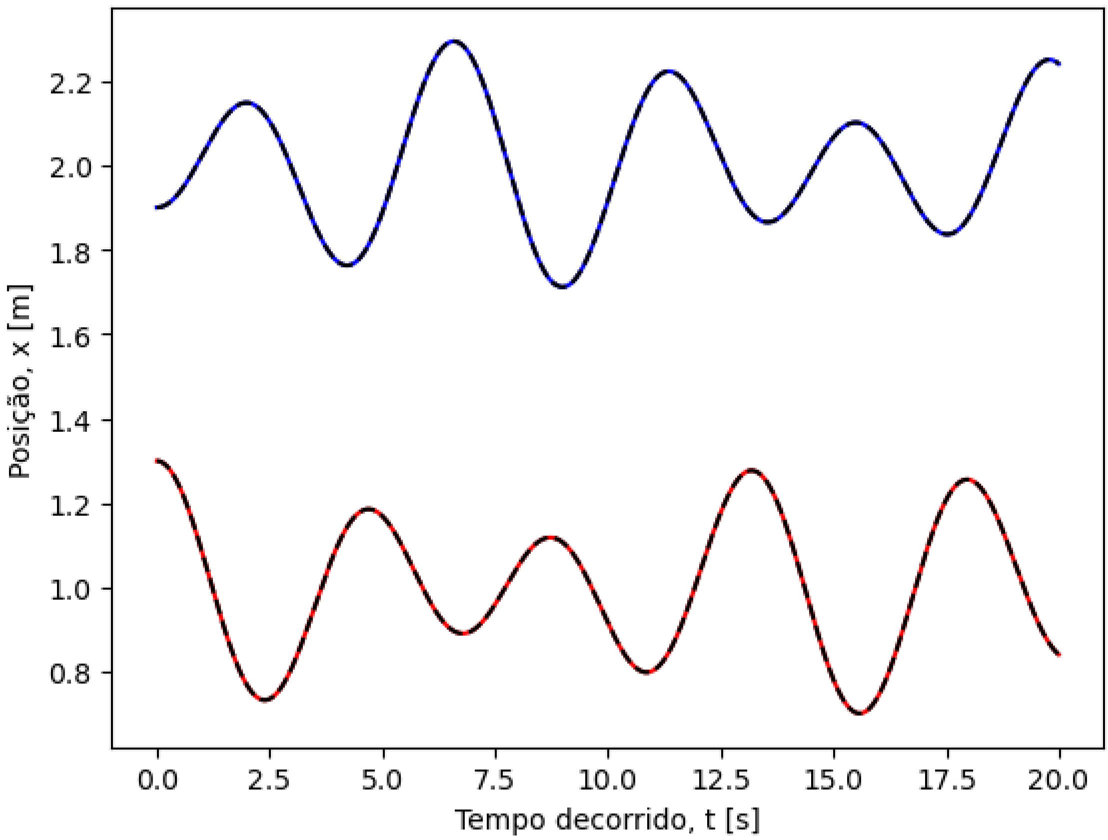
O sistema de equações consiste em:

$$\begin{cases} 0.2 = 2A_1 \cos\phi_1 \\ 0.4 = 2A_2 \cos\phi_2 \\ 0 = \sin\phi_1 \\ 0 = \sin\phi_2 \end{cases}$$

que se traduz nas seguintes soluções:

$$\begin{cases} A_1 = 0.1\,\mathrm{m} \\ A_2 = 0.2\,\mathrm{m} \\ \phi_1 = 0\,\mathrm{rad} \\ \phi_2 = 0\,\mathrm{rad} \end{cases}$$

In [7]:
```
A1 = 0.1
A2 = 0.2
φ1 = 0
φ2 = 0
xA_teorico = xeq[0] + A1 * np.cos(ω1 * t + φ1) + A2 * np.cos(ω2 * t + φ2)
xB_teorico = xeq[1] + A1 * np.cos(ω1 * t + φ1) - A2 * np.cos(ω2 * t + φ2)

# Encontrar o índice para t = 20 s
it=np.size(t[t<20])

plt.plot(t[0:it], x3[0, 0:it], 'r-', t[0:it], x3[1, 0:it], 'b-')
plt.plot(t[0:it], xA_teorico[0:it], 'k--', t[0:it], xB_teorico[0:it], 'k--')
plt.xlabel("Tempo decorrido, t [s]")
plt.ylabel("Posição, x [m]")
plt.show()
```



Exercício 3: Análise com Série de Fourier

Considere outra vez o sistema de exercício 1, e os três casos de condições iniciais:

i) $x_{\mathrm{A},t=0} = x_{\mathrm{A,eq}} + 0.3\ \mathrm{m}$, $x_{\{\mathrm{B}\},\mathrm{t}=0} = x_{\mathrm{B,eq}} + 0.3\ \mathrm{m}$, $v_{\mathrm{A}}(t=0) = v_{\mathrm{B,eq}} = 0\ \mathrm{m/s}$.

ii) $x_{\mathrm{A},t=0} = x_{\mathrm{A,eq}} + 0.3\ \mathrm{m}$, $x_{\{\mathrm{B}\},\mathrm{t}=0} = x_{\mathrm{B,eq}} - 0.3\ \mathrm{m}$, $v_{\mathrm{A}}(t=0) = v_{\mathrm{B,eq}} = 0\ \mathrm{m/s}$.

iii) $x_{\mathrm{A},t=0} = x_{\mathrm{A,eq}} + 0.3\ \mathrm{m}$, $x_{\{\mathrm{B}\},\mathrm{t}=0} = x_{\mathrm{B,eq}} - 0.1\ \mathrm{m}$, $v_{\mathrm{A}}(t=0) = v_{\mathrm{B,eq}} = 0\ \mathrm{m/s}$.

Solução:

Suponhamos que existe uma função arbitrária $x(t)$, para a qual dispomos unicamente de uma sequência de pares:

$$(x_p,\, t_p),\ \text{em que}\ x_p = x(t_p).$$

A função $x(t)$ pode ser descrita por uma série de Furier,

$$x(t) = \frac{a_0}{2} + \sum_{n>0} \left(a_n \cos \omega_n t + b_n \sin \omega_n t\right),$$

em que as frequências $\omega_n = 2\pi n/T$ são harmónicos baseados no período da função x, e os coeficientes de Fourier optêm-se por integração,

$$\begin{cases} a_n = \frac{2}{T}\int_0^T x(t)\cos(\omega_n t)\mathrm{d}t \\ b_n = \frac{2}{T}\int_0^T x(t)\sin(\omega_n t)\mathrm{d}t, \end{cases}$$

integração essa que pode ser numéria (aproximação trapezoidal). Esse método é implementado na função `abfourier(tp, xp, it0, it1, nf)`, cujos parâmetros são:

- `tp`, coordenadas no tempo, t_p
- `xp`, ordenadas x_p, correspondentes a $x_p = x(t_p)$
- `it0` e `it1`, limites inferior e superior de integração tal que $T = t_{\mathtt{it1}} - t_{\mathtt{it0}}$
- `nf`, índice dos coeficientes de Fourier a calcular,

A função devolve:

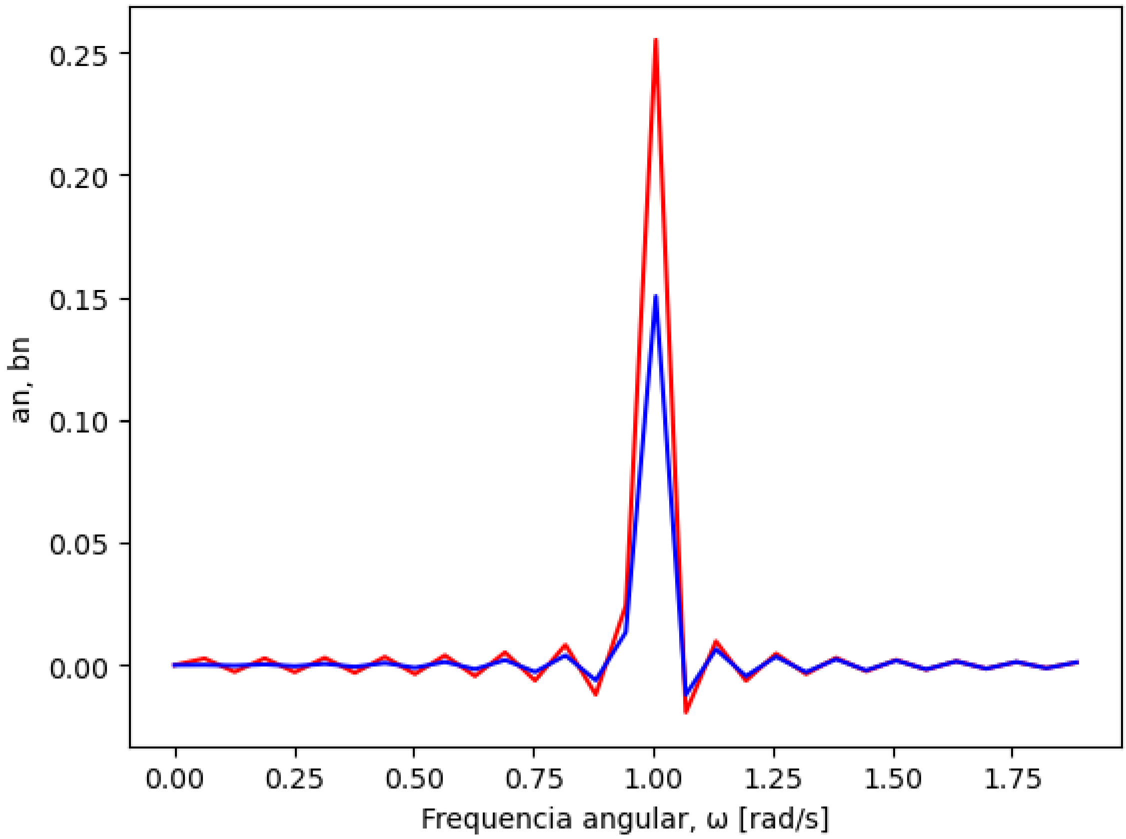
- `af` e `bf`, coeficientes $(a_{\mathtt{nf}},\, b_{\mathtt{nf}})$ correspondentes ao índice `nf`

In [8]:
```
def abfourier(tp,xp,it0,it1,nf):
    # cálculo dos coeficientes de Fourier a_nf e b_nf
    # a_nf = 2/T integral ( xp cos( nf w) ) dt entre tp(it0) e tp(it1)
    # b_nf = 2/T integral ( xp sin( nf w) ) dt entre tp(it0) e tp(it1)
    # integracao numerica pela aproximação trapezoidal
    # input: matrizes tempo tp (abcissas)
    # posição xp (ordenadas)
    # indices inicial it0
    # final it1 (ao fim de um período)
    # nf índice de Fourier
    # output: af_bf e bf_nf
    dt=tp[1]-tp[0]
    per=tp[it1]-tp[it0]
    ome=2*np.pi/per
    s1=xp[it0]*np.cos(nf*ome*tp[it0])
    s2=xp[it1]*np.cos(nf*ome*tp[it1])
    st=xp[it0+1:it1]*np.cos(nf*ome*tp[it0+1:it1])
    soma=np.sum(st)
    q1=xp[it0]*np.sin(nf*ome*tp[it0])
    q2=xp[it1]*np.sin(nf*ome*tp[it1])
    qt=xp[it0+1:it1]*np.sin(nf*ome*tp[it0+1:it1])
```

```
    somq=np.sum(qt)
    intega=((s1+s2)/2+soma)*dt
    af=2/per*intega
    integq=((q1+q2)/2+somq)*dt
    bf=2/per*integq
    return af,bf
```

Para o caso (i)

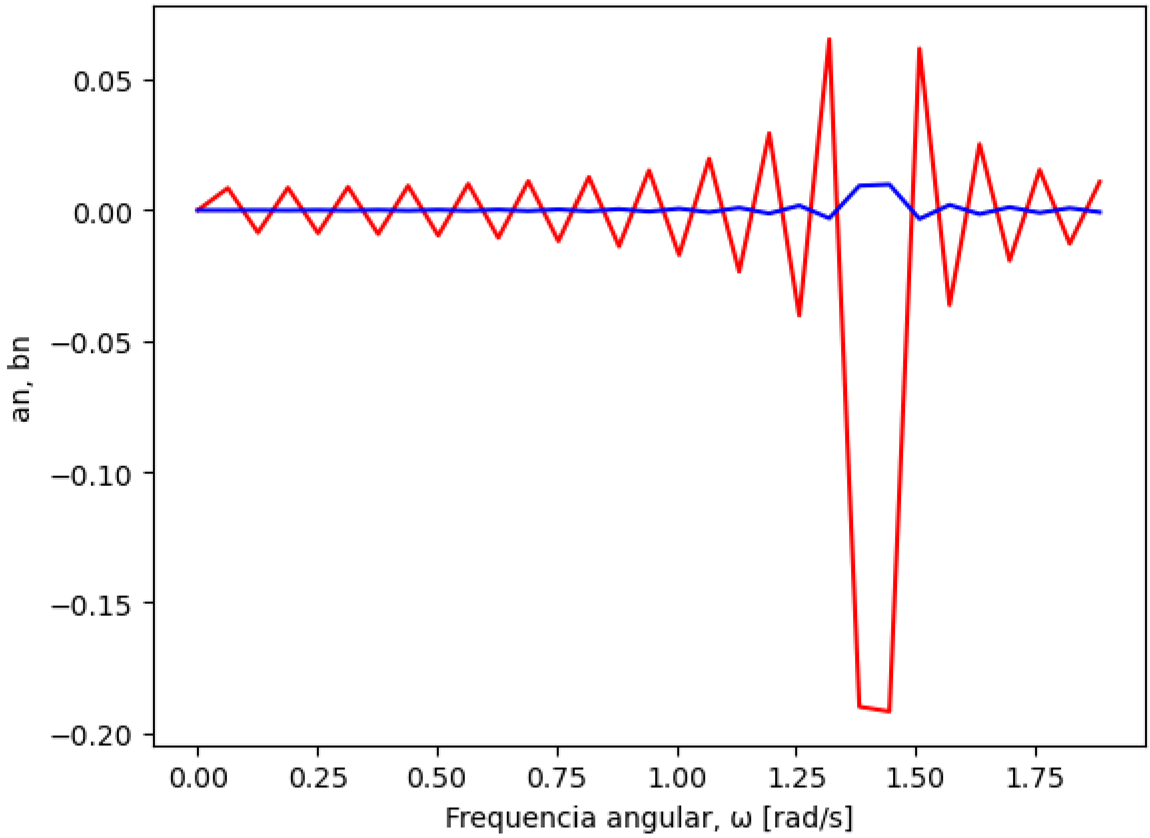
In [9]:
```
# Número de coeficientes de Fourier a considerar (excl n=0)
N = 30

# Vamos analisar a função no intervalo 50 s < t < 150 s
it0=np.size(t[t<50])
it1=np.size(t[t<150])

# Cálculo do período
T = t[it1] - t[it0]

# Inicializar coeficientes de Fourier
ωn = np.zeros(N+1)
an = np.zeros(N+1)
bn = np.zeros(N+1)

# Loop nos coeficientes de Fourier
for n in range(1, N + 1):
    an[n], bn[n] = abfourier(t, x1[0,:], it0, it1, n)
    ωn[n] = 2 * np.pi * n / T


plt.plot(ωn, an, 'r-', ωn, bn, 'b-')
plt.xlabel("Frequencia angular, ω [rad/s]")
plt.ylabel("an, bn")
plt.show()
```



Verificamos um *pico* em $\omega = 1\,\mathrm{rad/s}$, que corresponde a $\omega = \sqrt{k/m}$, ou seja, resulta do modo normal de vibração em que as massas vibram em fase.

Para o caso (ii)

In [10]:
```
# Número de coeficientes de Fourier a considerar (excl n=0)
N = 30

# Vamos analisar a função no intervalo 50 s < t < 150 s
it0=np.size(t[t<50])
it1=np.size(t[t<150])

# Cálculo do período
T = t[it1] - t[it0]

# Inicializar coeficientes de Fourier
ωn = np.zeros(N+1)
an = np.zeros(N+1)
bn = np.zeros(N+1)

# Loop nos coeficientes de Fourier
for n in range(1, N + 1):
    an[n], bn[n] = abfourier(t, x2[0,:], it0, it1, n)
    ωn[n] = 2 * np.pi * n / T


plt.plot(ωn, an, 'r-', ωn, bn, 'b-')
plt.xlabel("Frequencia angular, ω [rad/s]")
plt.ylabel("an, bn")
plt.show()
```



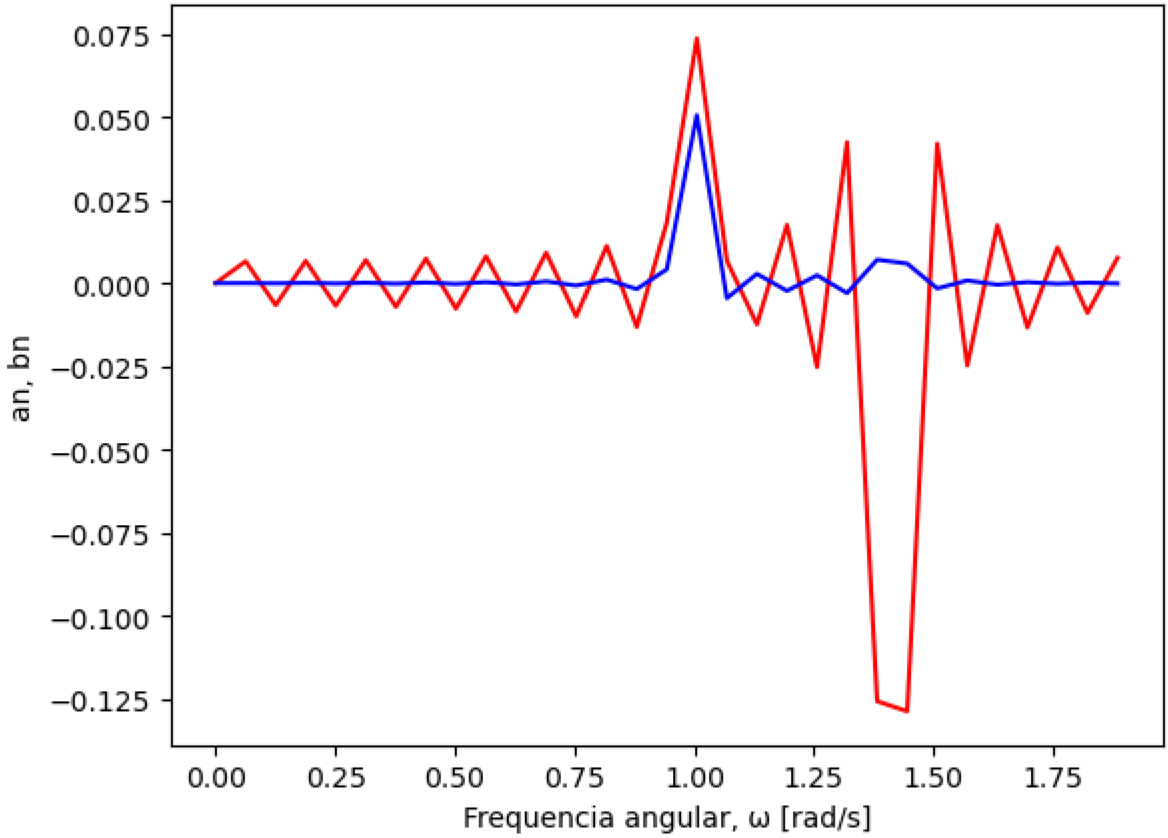
Verificamos um *pico* em $\omega \approx 1.4\,\mathrm{rad/s}$, que corresponde a $\omega = \sqrt{(k+2k')/m}$, ou seja, resulta do modo normal de vibração em que as massas vibram em oposição de fase.

In [11]:
```
print("w = sqrt((k+2kp)/m) = ", np.sqrt((k + 2 * kp) / m))
```

```
w = sqrt((k+2kp)/m) =  1.4142135623730951
```

Para o caso (iii)

In [12]:
```python
# Número de coeficientes de Fourier a considerar (excl n=0)
N = 30

# Vamos analisar a função no intervalo 50 s < t < 150 s
it0=np.size(t[t<50])
it1=np.size(t[t<150])

# Cálculo do período
T = t[it1] - t[it0]

# Inicializar coeficientes de Fourier
ωn = np.zeros(N+1)
an = np.zeros(N+1)
bn = np.zeros(N+1)

# Loop nos coeficientes de Fourier
for n in range(1, N + 1):
    an[n], bn[n] = abfourier(t, x3[0,:], it0, it1, n)
    ωn[n] = 2 * np.pi * n / T


plt.plot(ωn, an, 'r-', ωn, bn, 'b-')
plt.xlabel("Frequencia angular, ω [rad/s]")
plt.ylabel("an, bn")
plt.show()
```



Verificamos dois *picos* em $\omega = \sqrt{k/m} = 1\,\mathrm{rad/s}$ e em $\omega = \sqrt{(k+2k')/m} \approx 1.4\,\mathrm{rad/s}$. Desta forma concluimos que, pese embora as oscilações do caso (iii) não apresentem *ordem* temporal, a análise de Fourier mostra que o movimento das massas é composto pelos dois modos normais de vibração identificados.

Pergunta 2:

O que se observa no gráfico dos coeficientes Fourier? Comente na relação com a solução teórica.