In [71]:

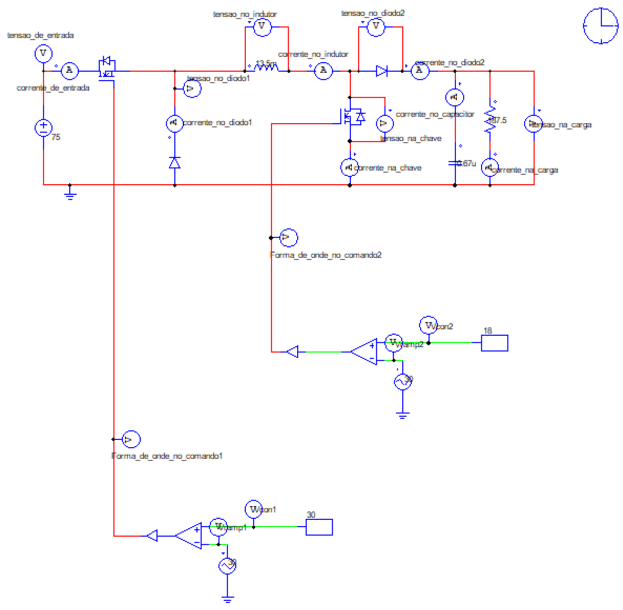
```python
import ipywidgets as widgets
from IPython.display import display, HTML

javascript_functions = {False: "hide()", True: "show()"}
button_descriptions  = {False: "Show code", True: "Hide code"}


def toggle_code(state):

    """
    Toggles the JavaScript show()/hide() function on the div.input element.
    """

    output_string = "<script>$(\"div.input\").{}</script>"
    output_args   = (javascript_functions[state],)
    output        = output_string.format(*output_args)

    display(HTML(output))


def button_action(value):

    """
    Calls the toggle_code function and updates the button description.
    """

    state = value.new

    toggle_code(state)

    value.owner.description = button_descriptions[state]


state = False
toggle_code(state)

button = widgets.ToggleButton(state, description = button_descriptions[state])
button.observe(button_action, "value")

display(button)
```

Buck Conversor Buck-Boost não inversor

Um conversor buck-boost não inversor é essencialmente uma combinação em cascata de um conversor de buck seguido por um conversor boost, onde um único indutor-capacitor é usado para ambos. Este conversor não inverte as polaridades da tensão de saída em relação às polaridades da entrada.

Basicamente, requer o uso de duas chaves e é projetado combinando um conversor buck e um conversor de boost, ambos projetados na mesma topologia de forma que este conversor possa funcionar como Conversor Buck, Boost ou Buck-Boost. O MOSFET 1 é conectado entre a fonte de tensão de entrada e o diodo 1. O indutor está conectado entre o diodo 1 e o MOSFET 2, enquanto o diodo 2 está conectado entre MOSFET 2 e a saída ou capacitor de carga.

Esquemático:



Etapas do projeto para cada modo de operação (buck-boost, buck ou boost):

1. Definir parâmetros de projeto (Vin, Vout, Po, fs, variação de IL e de Vc)
2. Calcular a razão cíclica (D)
3. Calcular indutância (L)
4. Calcular a capacitância (C)
5. Calcular os esforços nos semicondutores (Ismd, Isef, Ismax, Idmds, Idef, Idmax, Vsmax e Vdmax)

Valores que serão definidos:

- Vin = 75V
- fs = 50KHz
- D = 0.4
- Ro = 50 ohms
- Lo = 3.6 mH
- Co = 16 uC

Atuando como conversor Buck-Boost:

Nesta topologia, as chaves MOSFET 1 e MOSFET 2 estão simultaneamente ligadas durante o ciclo de comutação ou on-time, enquanto D1 e D2 estão simultaneamente ligados durante o oposto ciclo de comutação ou off-time. Isso significa que quando os MOSFETs 1 e 2 estão ligados, o indutor é carregado, assim D1 e D2 estão desligados. Quando D1 e D2 estão ligados, o indutor está carregando o capacitor, logo as duas chaves MOSFETs estão desligadas.

Tensão de Entrada:

In [72]:

```python
Vin = 75
print("A tensão de entrada é", Vin, "V")
```

```
A tensão de entrada é 75 V
```

Tensão de Saída:

In [73]:

```python
Vout = 50
print("A tensão de saída é", Vout, "V")
```

```
A tensão de saída é 50 V
```

Potência máxima de saída:

In [74]:

```python
Po = 50
print("A potência máxima de saída é", Po, "W")
```

```
A potência máxima de saída é 50 W
```

Frequência de comutação:

In [75]:

```python
fs = 50000
print("A frequência de comutação é", fs, "Hz")
```

```
A frequência de comutação é 50000 Hz
```

Razão cíclica:

$$D = \frac{Vout}{(Vout + Vin)}$$

In [76]:

```python
D = Vout/(Vout+Vin)
print("A razão ciclica é", D)
```

```
A razão ciclica é 0.4
```

Corrente de saída:

$$Io = \frac{Po}{Vout}$$

In [77]:

```python
Io = Po/Vout
print("A corrente de saída é", "%.3f" % Io, "A")
```

```
A corrente de saída é 1.000 A
```

Ondulação de corrente no indutor:

Ondulação de 10%:

$$\Delta_{Il} = 0.1(Io + \frac{Po}{Vin})$$

In [78]:

```
delta_Il = 0.1*(Io + Po/Vin)
print("A ondulação de corrente no indutor é", "%.3f" % delta_Il, "A")
```

```
A ondulação de corrente no indutor é 0.167 A
```

Ondulação de tensão no capacitor:

Ondulação de 1%:

$$\Delta_{Vc} = 0.01Vout$$

In [79]:

```
delta_Vc = 0.01*Vout
print("A ondulação de tensão no capacitor é", delta_Vc, "V")
```

```
A ondulação de tensão no capacitor é 0.5 V
```

Resistência de carga:

$$Ro = \frac{Vout^2}{Po}$$

In [80]:

```
Ro = (Vout**2)/Po
print("A resistência de carga é", Ro, "ohms")
```

```
A resistência de carga é 50.0 ohms
```

Indutor de Saída:

$$Lo = \frac{VinD}{fs\Delta_{Il}}$$

In [81]:

```
Lo = Vin*D/(fs*delta_Il)
print("O indutor de saída é", "%.4f" % Lo, "H")
```

```
O indutor de saída é 0.0036 H
```

Capacitor de Saída:

$$Co = \frac{IoD}{fs\Delta_{Vc}}$$

In [82]:

```
Co = Io*D/(fs*delta_Vc)
print("O capacitor de saída é", Co, "F")
```

```
O capacitor de saída é 1.6e-05 F
```

Esforços na chave:

Valor médio da corrente na chave:

$$Is_{md} = DIo$$

In [83]:

```
Is_md = D*Io
print("O valor médio da corrente na chave é", "%.3f" % Is_md, "A")
```

```
O valor médio da corrente na chave é 0.400 A
```

Valor eficaz da corrente na chave:

$$Is_{ef} = \sqrt{D}Io$$

In [84]:

```
Is_ef = (D**0.5)*Io
print("O valor eficaz da corrente na chave é", "%.3f" % Is_ef, "A")
```

```
O valor eficaz da corrente na chave é 0.632 A
```

Valor máximo da corrente na chave:

$$Is_{max} = Io + \frac{\Delta_{Il}}{2}$$

In [85]:

```
Is_max = Io + delta_Il/2
print("O valor máximo da corrente na chave é", Is_max, "A")
```

```
O valor máximo da corrente na chave é 1.0833333333333333 A
```

Valor máximo da tensão na chave:

$$Vs_{max} = Vin$$

In [86]:

```
Vs_max = Vin
print("O valor máximo da tensão na chave é", Vs_max, "V")
```

```
O valor máximo da tensão na chave é 75 V
```

Esforços no diodo:

Valor médio da corrente no diodo:

$$Id_{md} = (1 - D)Io$$

In [87]:

```
Id_md = (1-D)*Io
print("O valor médio da corrente no diodo é", "%.3f" % Id_md, "A")
```

```
O valor médio da corrente no diodo é 0.600 A
```

Valor eficaz da corrente no diodo:

$$Id_{ef} = \sqrt{(1 - D)}Io$$

In [88]:

```
Id_ef = ((1-D)**0.5)*Io
print("O valor eficaz da corrente no diodo é", "%.3f" % Id_ef, "A")
```

```
O valor eficaz da corrente no diodo é 0.775 A
```

Valor máximo da corrente no diodo:

$$Id_{max} = Io + \frac{\Delta_{Il}}{s}$$

In [89]:

```
Id_max = Io + delta_Il/2
print("O valor máximo da corrente no diodo é", Id_max, "A")
```

```
O valor máximo da corrente no diodo é 1.0833333333333333 A
```

Valor máximo da tensão no diodo:

$$Vd_{max} = Vin$$

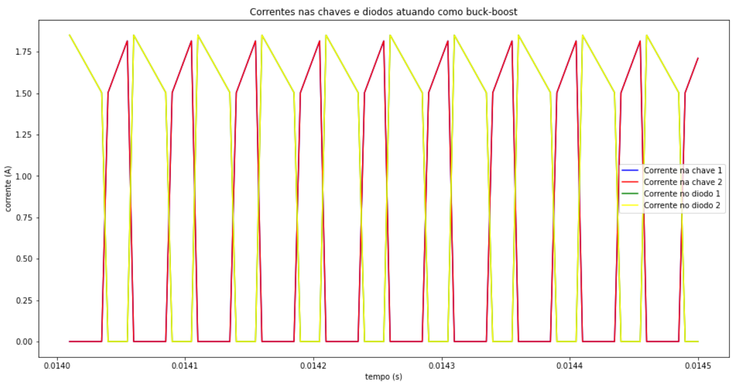
In [90]:

```
Vd_max = Vin
print("O valor máximo da tensão no diodo é", Vd_max, "V")
```

```
O valor máximo da tensão no diodo é 75 V
```

In [91]:

```
import numpy  as np
import matplotlib.pyplot as plt
data = np.loadtxt('dados/3em1/buck-boost/corrente_chave_diodo.txt')


x = data[:, 0]#tempo
y = data[:, 1]#chave1
z = data[:, 2]#chave2
g = data[:, 3]#diodo1
h = data[:, 4]#diodo2
plt.figure(figsize=(16,8))#altera as dimensões do gráfico
plt.plot(x,y,'blue')
plt.plot(x,z,'red')
plt.plot(x,g,'green')
plt.plot(x,h,'yellow')

plt.title('Correntes nas chaves e diodos atuando como buck-boost')
plt.xlabel('tempo (s)')
plt.ylabel('corrente (A)')
plt.legend(['Corrente na chave 1','Corrente na chave 2','Corrente no diodo 1','Corrent
plt.show()
```



Atuando como conversor Buck:

Nesta topologia, a chave MOSFET 2 está sempre desligada e o diodo D2 está sempre ligado.

Valores iguais:

In [92]:

```
print("A tensão de entrada é", Vin, "V")
print("A frequência de comutação é", fs, "Hz")
print("A razão ciclíca é", D)
print("A resistência de carga é", Ro, "ohms")
print("O indutor de saída é", "%.4f" % Lo, "H")
print("O capacitor de saída é", Co, "F")
```

```
A tensão de entrada é 75 V
A frequência de comutação é 50000 Hz
A razão ciclíca é 0.4
A resistência de carga é 50.0 ohms
O indutor de saída é 0.0036 H
O capacitor de saída é 1.6e-05 F
```

Tensão de saída:

$$Vout = VinD$$

In [93]:

```
Vout2 = Vin*D
print("A tensão de saída é", Vout2, "V")
```

```
A tensão de saída é 30.0 V
```

Potência máxima de saída:*

*mesma fórmula anterior

In [94]:

```
Po2 = (Vout2**2)/Ro
print("A potência máxima de saída é", Po2, "W")
```

```
A potência máxima de saída é 18.0 W
```

Corrente de saída:*

*mesma fórmula anterior

In [95]:

```
Io2 = Po2/Vout2
print("A corrente de saída é", "%.3f" % Io2, "A")
```

```
A corrente de saída é 0.600 A
```

Corrente de ondulção do indutor e tensão de ondulação do capacitor:

$$\Delta_{Il} = \frac{(Vin - Vout)D}{fsLo}$$

$$\Delta_{Vc} = \frac{\Delta_{Il}}{8fsCo}$$

In [96]:

```
delta_Il2 = (Vin-Vout2)*D/(fs*Lo)
delta_Vc2 = delta_Il2/(8*fs*Co)
print("A ondulação de corrente no indutor é", "%.3f" % delta_Il2, "A")
print("A ondulação de tensão no capacitor é", "%.3f" % delta_Vc2, "V")
```

```
A ondulação de corrente no indutor é 0.100 A
A ondulação de tensão no capacitor é 0.016 V
```

Esforços:*

*mesmas fórmulas anteriores

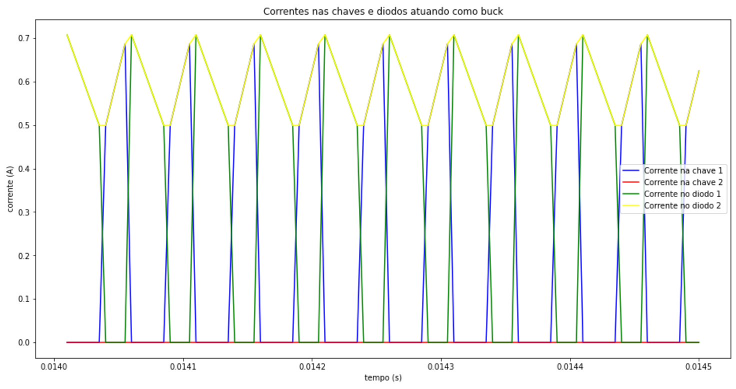
In [97]:

```
Is_md2 = D*Io2
print("O valor médio da corrente na chave é", "%.3f" % Is_md2, "A")
Is_ef2 = (D**0.5)*Io2
print("O valor eficaz da corrente na chave é", "%.3f" % Is_ef2, "A")
Is_max2 = Io2 + delta_Il2/2
print("O valor máximo da corrente na chave é", "%.3f" % Is_max2, "A")
Vs_max2 = Vin
print("O valor máximo da tensão na chave é", Vs_max2, "V")
Id_md2 = (1-D)*Io2
print("O valor médio da corrente no diodo é", "%.3f" % Id_md2, "A")
Id_ef2 = ((1-D)**0.5)*Io2
print("O valor eficaz da corrente no diodo é", "%.3f" % Id_ef2, "A")
Id_max2 = Io2 + delta_Il2/2
print("O valor máximo da corrente no diodo é", "%.3f" % Id_max2, "A")
Vd_max2 = Vin
print("O valor máximo da tensão no diodo é", Vd_max2, "V")
```

```
O valor médio da corrente na chave é 0.240 A
O valor eficaz da corrente na chave é 0.379 A
O valor máximo da corrente na chave é 0.650 A
O valor máximo da tensão na chave é 75 V
O valor médio da corrente no diodo é 0.360 A
O valor eficaz da corrente no diodo é 0.465 A
O valor máximo da corrente no diodo é 0.650 A
O valor máximo da tensão no diodo é 75 V
```

In [98]:

```
import numpy  as np
import matplotlib.pyplot as plt
data = np.loadtxt('dados/3em1/buck/corrente_chave_diodo.txt')


x = data[:, 0]#tempo
y = data[:, 1]#chave1
z = data[:, 2]#chave2
g = data[:, 3]#diodo1
h = data[:, 4]#diodo2
plt.figure(figsize=(16,8))#altera as dimensões do gráfico
plt.plot(x,y,'blue')
plt.plot(x,z,'red')
plt.plot(x,g,'green')
plt.plot(x,h,'yellow')

plt.title('Correntes nas chaves e diodos atuando como buck')
plt.xlabel('tempo (s)')
plt.ylabel('corrente (A)')
plt.legend(['Corrente na chave 1','Corrente na chave 2','Corrente no diodo 1','Corrent
plt.show()
```



Ao se analisar esses gráficos nota-se que a chave 2 está em aberto, não apresentando corrente. E, por isso, a corrente no diodo 2 é a própria corrente do indutor. O que já era esperado para o funcionamento como conversor Buck.

Atuando como conversor Boost:

Nesta topologia, a chave MOSFET 1 está sempre ligada e o diodo D1 está sempre desligado.

Valores iguais:

In [99]:

```
print("A tensão de entrada é", Vin, "V")
print("A frequência de comutação é", fs, "Hz")
print("A razão ciclíca é", D)
print("A resistência de carga é", Ro, "ohms")
print("O indutor de saída é", "%.4f" % Lo, "H")
print("O capacitor de saída é", Co, "F")
```

```
A tensão de entrada é 75 V
A frequência de comutação é 50000 Hz
A razão ciclíca é 0.4
A resistência de carga é 50.0 ohms
O indutor de saída é 0.0036 H
O capacitor de saída é 1.6e-05 F
```

Tensão de saída:

$$Vout = \frac{Vin}{(1-D)}$$

In [100…

```
Vout3 = Vin/(1-D)
print("A tensão de saída é", Vout3, "V")
```

```
A tensão de saída é 125.0 V
```

Potência máxima de saída:*

*mesma fórmula anterior

In [101…

```
Po3 = (Vout3**2)/Ro
print("A potência máxima de saída é", Po3, "W")
```

```
A potência máxima de saída é 312.5 W
```

Corrente de saída:*

*mesma fórmula anterior

In [102…

```
Io3 = Po3/Vout3
print("A corrente de saída é", "%.3f" % Io3, "A")
```

```
A corrente de saída é 2.500 A
```

Corrente de ondulção do indutor e tensão de ondulação do capacitor:

$$\Delta_{Il} = \frac{VinD}{fsLo}$$

$$\Delta_{Vc} = \frac{IoD}{fsCo}$$

In [103…

```
delta_Il3 = Vin*D/(fs*Lo)
delta_Vc3 = D*Io3/(fs*Co)
print("A ondulação de corrente no indutor é", "%.3f" % delta_Il3, "A")
print("A ondulação de tensão no capacitor é", "%.3f" % delta_Vc3, "V")
```

```
A ondulação de corrente no indutor é 0.167 A
A ondulação de tensão no capacitor é 1.250 V
```

Esforços:*

*mesmas fórmulas anteriores

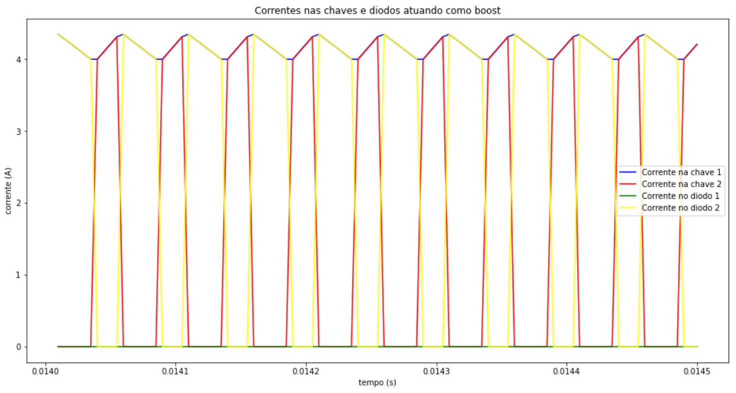
In [104…

```
Is_md3 = D*Io3
print("O valor médio da corrente na chave é", "%.3f" % Is_md3, "A")
Is_ef3 = (D**0.5)*Io3
print("O valor eficaz da corrente na chave é", "%.3f" % Is_ef3, "A")
Is_max3 = Io3 + delta_Il3/2
print("O valor máximo da corrente na chave é", "%.3f" % Is_max3, "A")
Vs_max3 = Vin
print("O valor máximo da tensão na chave é", Vs_max3, "V")
Id_md3 = (1-D)*Io3
print("O valor médio da corrente no diodo é", "%.3f" % Id_md3, "A")
Id_ef3 = ((1-D)**0.5)*Io3
print("O valor eficaz da corrente no diodo é", "%.3f" % Id_ef3, "A")
Id_max3 = Io3 + delta_Il3/2
print("O valor máximo da corrente no diodo é", "%.3f" % Id_max3, "A")
Vd_max3 = Vin
print("O valor máximo da tensão no diodo é", Vd_max3, "V")
```

```
O valor médio da corrente na chave é 1.000 A
O valor eficaz da corrente na chave é 1.581 A
O valor máximo da corrente na chave é 2.583 A
O valor máximo da tensão na chave é 75 V
O valor médio da corrente no diodo é 1.500 A
O valor eficaz da corrente no diodo é 1.936 A
O valor máximo da corrente no diodo é 2.583 A
O valor máximo da tensão no diodo é 75 V
```

In [105…

```
import numpy  as np
import matplotlib.pyplot as plt
data = np.loadtxt('dados/3em1/boost/corrente_chave_diodo.txt')


x = data[:, 0]#tempo
y = data[:, 1]#chave1
z = data[:, 2]#chave2
g = data[:, 3]#diodo1
h = data[:, 4]#diodo2
plt.figure(figsize=(16,8))#altera as dimensões do gráfico
plt.plot(x,y,'blue')
plt.plot(x,z,'red')
plt.plot(x,g,'green')
plt.plot(x,h,'yellow')

plt.title('Correntes nas chaves e diodos atuando como boost')
plt.xlabel('tempo (s)')
plt.ylabel('corrente (A)')
plt.legend(['Corrente na chave 1','Corrente na chave 2','Corrente no diodo 1','Corren
plt.show()
```

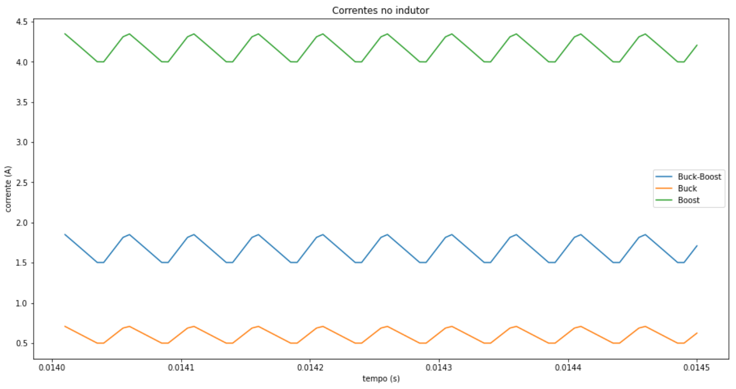


Ao se analisar esses gráficos nota-se que a chave 1 está fechada sempre, por isso, a corrente no diodo 1 é zero, pois o diodo está em OFF. Além disso, é possível notar que a corrente na chave 1 é a própria corrente no indutor. O que já era esperado para o funcionamento como conversor Boost.

Comparações entre corrente e tensão no indutor, e tensão na carga para os três modos de operação:

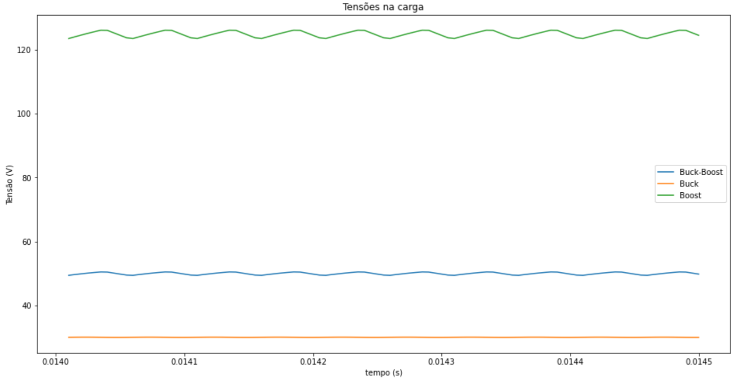
In [106…

```
import numpy  as np
import matplotlib.pyplot as plt
data = np.loadtxt('dados/3em1/corrente_no_indutor.txt')


x = data[:, 0]#tempo
y = data[:, 1]#buck-boost
z = data[:, 2]#buck
g = data[:, 3]#boost
plt.figure(figsize=(16,8))#altera as dimensões do gráfico
plt.plot(x,y)
plt.plot(x,z)
plt.plot(x,g)

plt.title('Correntes no indutor')
plt.xlabel('tempo (s)')
plt.ylabel('corrente (A)')
plt.legend(['Buck-Boost','Buck','Boost'], loc = 'best', fontsize=10)
plt.show()
```



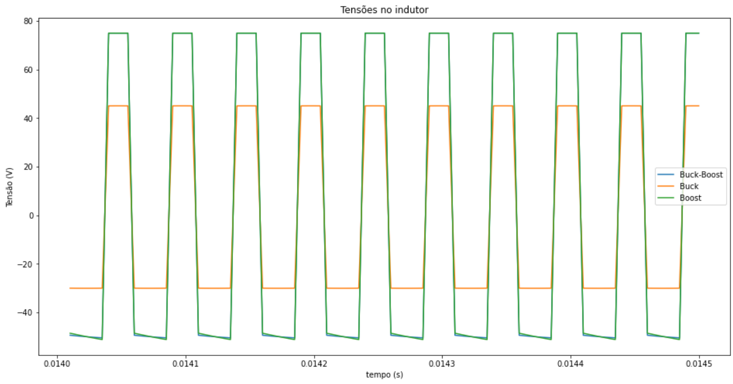
In [107…

```
import numpy  as np
import matplotlib.pyplot as plt
data = np.loadtxt('dados/3em1/tensao_na_carga.txt')


x = data[:, 0]#tempo
y = data[:, 1]#buck-boost
z = data[:, 2]#buck
g = data[:, 3]#boost
plt.figure(figsize=(16,8))#altera as dimensões do gráfico
plt.plot(x,y)
plt.plot(x,z)
plt.plot(x,g)

plt.title('Tensões na carga')
plt.xlabel('tempo (s)')
plt.ylabel('Tensão (V)')
plt.legend(['Buck-Boost','Buck','Boost'], loc = 'best', fontsize=10)
plt.show()
```



In [108…

```
import numpy  as np
import matplotlib.pyplot as plt
data = np.loadtxt('dados/3em1/tensao_no_indutor.txt')


x = data[:, 0]#tempo
y = data[:, 1]#buck-boost
z = data[:, 2]#buck
g = data[:, 3]#boost
plt.figure(figsize=(16,8))#altera as dimensões do gráfico
plt.plot(x,y)
plt.plot(x,z)
plt.plot(x,g)

plt.title('Tensões no indutor')
plt.xlabel('tempo (s)')
plt.ylabel('Tensão (V)')
plt.legend(['Buck-Boost','Buck','Boost'], loc = 'best', fontsize=10)
plt.show()
```



Projeto do Indutor:

- Selecionar o núcleo magnético mais adequado;
- Calcular o número de espiras;
- Calcular tamanho do entreferro;
- Escolher o condutor;
- Verificar a possibilidade da execução do projeto.

Selecionar o núcleo magnético adequado

Ferrita:

In [109…

```
B_max = 0.3
print("O valor da densidade de fluxo magnético máximo é", B_max, "T")
Kw = 0.6
print("O valor do fator de utilização da área de enrolamento é", Kw)
J = 450
print("O valor da densidade de corrente no condutor é", J, "A/cm^2")
```

```
O valor da densidade de fluxo magnético máximo é 0.3 T
O valor do fator de utilização da área de enrolamento é 0.6
O valor da densidade de corrente no condutor é 450 A/cm^2
```

In [110…

```
Il_max = 4.348
Il_min = 4
Il_rms = 4.148
print("O valor máximo da corrente no indutor é", Il_max, "A")
print("O valor mínimo da corrente no indutor é", Il_min, "A")
print("O valor RMS da corrente no indutor é", Il_rms, "A")
print("Cosiderando os maiores valores - atuando como conversor boost")
```

```
O valor máximo da corrente no indutor é 4.348 A
O valor mínimo da corrente no indutor é 4 A
O valor RMS da corrente no indutor é 4.148 A
Cosiderando os maiores valores - atuando como conversor boost
```

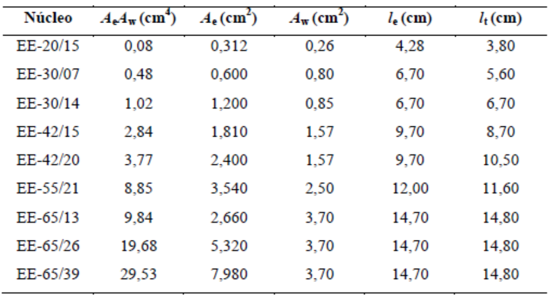
$$AeAw = \frac{LoI_{lmax}I_{lrms}10^4}{B_{max}K_wJ}$$

In [111…

```
A_eA_w = Lo*Il_max*Il_rms*10**4/(B_max*Kw*J)
print("A multiplicação entre a área efetiva e a área da janela do núcleo é", A_eA_w,"
```

```
A multiplicação entre a área efetiva e a área da janela do núcleo é 8.015779555555557
cm^4
```

Tabela: Núcleos Magnéticos de ferrite com geométrica EE



Núcleo	A_eA_w (cm^4)	A_e (cm^2)	A_w (cm^2)	l_e (cm)	l_t (cm)
EE-20/15	0,08	0,312	0,26	4,28	3,80
EE-30/07	0,48	0,600	0,80	6,70	5,60
EE-30/14	1,02	1,200	0,85	6,70	6,70
EE-42/15	2,84	1,810	1,57	9,70	8,70
EE-42/20	3,77	2,400	1,57	9,70	10,50
EE-55/21	8,85	3,540	2,50	12,00	11,60
EE-65/13	9,84	2,660	3,70	14,70	14,80
EE-65/26	19,68	5,320	3,70	14,70	14,80
EE-65/39	29,53	7,980	3,70	14,70	14,80

In [112…

```
A_eA_w = 9.84
```

```
A_e = 2.66
A_w = 3.7
le = 14.7
lt = 14.8
print("A multiplicação entre a área efetiva e a área da janela do núcleo será", A_eA_v
print("A área efetiva será", A_e,"cm^2 - valor escolhido pela tabela")
print("A área da janela do núcleo será", A_w,"cm^2 - valor escolhido pela tabela")
print("O comprimento do caminho magnético será", le,"cm - valor escolhido pela tabela'
print("O comprimento médio de uma espira será", lt,"cm - valor escolhido pela tabela")
```

```
A multiplicação entre a área efetiva e a área da janela do núcleo será 9.84 cm^4 - val
or mais próximo do valor calculado
A área efetiva será 2.66 cm^2 - valor escolhido pela tabela
A área da janela do núcleo será 3.7 cm^2 - valor escolhido pela tabela
O comprimento do caminho magnético será 14.7 cm - valor escolhido pela tabela
O comprimento médio de uma espira será 14.8 cm - valor escolhido pela tabela
```

Número de espiras

$$N = \frac{LoI_{lmax}10^4}{B_{max}A_e}$$

In [113…

```
N = Lo*Il_max*10**4/(B_max*A_e)
N = int(N)
print("O número de espiras é", N)
```

```
O número de espiras é 196
```

Tamanho do entreferro

$$l_g = \frac{N2u_oA_e}{Lo10^4}$$

In [114…

```
uo = 4*3.14159265358979331*10**(-7)
lg = N**2*uo*A_e/(Lo*10**4)
print("O tamanho do entreferro é", "%.3f" % (lg*10**3), "mm")
```

```
O tamanho do entreferro é 3.567 mm
```

Dimensionamento dos condutores

$$I_{tot} = 1.1Nl_t$$

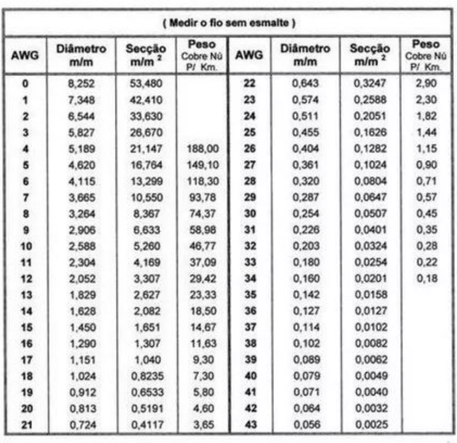
$$S_{fio} = \frac{I_{lrms}}{J}$$

In [115…

```
ltot = 1.1*N*lt
print("O comprimento total do condutor é", "%.3f" % ltot, "m")
Sfio = Il_rms/J
print("O valor mínimo da bitola é", "%.3f" % (Sfio*10**2), "mm^2")
```

```
O comprimento total do condutor é 3190.880 m
O valor mínimo da bitola é 0.922 mm^2
```

Tabela: Condutores de cobre:



(Medir o fio sem esmalte)

AWG	Diâmetro m/m	Secção m/m²	Peso Cobre Nú P/ Km.	AWG	Diâmetro m/m	Secção m/m²	Peso Cobre Nú P/ Km.
0	8,252	53,480		22	0,643	0,3247	2,90
1	7,348	42,410		23	0,574	0,2588	2,30
2	6,544	33,630		24	0,511	0,2051	1,82
3	5,827	26,670		25	0,455	0,1626	1,44
4	5,189	21,147	188,00	26	0,404	0,1282	1,15
5	4,620	16,764	149,10	27	0,361	0,1024	0,90
6	4,115	13,299	118,30	28	0,320	0,0804	0,71
7	3,665	10,550	93,78	29	0,287	0,0647	0,57
8	3,264	8,367	74,37	30	0,254	0,0507	0,45
9	2,906	6,633	58,98	31	0,226	0,0401	0,35
10	2,588	5,260	46,77	32	0,203	0,0324	0,28
11	2,304	4,169	37,09	33	0,180	0,0254	0,22
12	2,052	3,307	29,42	34	0,160	0,0201	0,18
13	1,829	2,627	23,33	35	0,142	0,0158	
14	1,628	2,082	18,50	36	0,127	0,0127	
15	1,450	1,651	14,67	37	0,114	0,0102	
16	1,290	1,307	11,63	38	0,102	0,0082	
17	1,151	1,040	9,30	39	0,089	0,0062	
18	1,024	0,8235	7,30	40	0,079	0,0049	
19	0,912	0,6533	5,80	41	0,071	0,0040	
20	0,813	0,5191	4,60	42	0,064	0,0032	
21	0,724	0,4117	3,65	43	0,056	0,0025	

In [116…

```
Sfio = 1.040
print("O valor da bitola escolhido pela tabela acima foi", Sfio, "mm^2 - AWG=17")
```

```
O valor da bitola escolhido pela tabela acima foi 1.04 mm^2 - AWG=17
```

$$A_{wmin} = \frac{n_{cond}S_{fio}10^{-2}N}{K_w}$$

In [117…

```
ncond = 1
A_w_min = ncond*Sfio*10**(-2)*N/Kw
print("A valor mínimo da área da janela do núcleo será", A_w_min,"cm^2")
```

```
A valor mínimo da área da janela do núcleo será 3.397333333333334 cm^2
```

Como Aw_min é menor que Aw a execução do projeto é possível!!!

Projeto do Dissipador de Calor:

Para o caso sem dissipador, o calor circula da junção para o ambiente através do encapsulamento. Assim, o circuito térmico se resume apenas à resistência Rjc (junção-case) em série com Rca (case-ambiente), sendo estes parâmetros encontrados no datasheet do componenete.

Para saber se o componente irá precisar de do dissipador de calor é estimado a temperatura de junção para a aplicação em específico.

$$T_j = R_{ja}P_T + T_a$$

MOSFET 1 (BSC100N10NSF): (pior caso = conversor Boost)

In [118…

```
P_T = 156.73*10**(-3)
R_ja = 62
Ta = 50
T_j_1 = R_ja*P_T + Ta
print("A temperatura de junção estimada é", "%.3f" % T_j_1, "°C")
```

```
A temperatura de junção estimada é 59.717 °C
```

Como 59.717°C é menor que 150°C (máxima temperatura de operação do transistor), o componente não precisa de um dissipador!!!

MOSFET 2 (BSC100N10NSF): (pior caso = conversor Boost)

In [119…

```
P_T2 = 182.39*10**(-3)
R_ja = 62
Ta = 50
T_j_1 = R_ja*P_T2 + Ta
print("A temperatura de junção estimada é", "%.3f" % T_j_1, "°C")
```

```
A temperatura de junção estimada é 61.308 °C
```

Como 61.308°C é menor que 150°C (máxima temperatura de operação do transistor), o componente não precisa de um dissipador!!!

DIODO 1 (MBR20100CT): (pior caso = conversor Buck-Boost)

In [120…

```
P_T3 = 575.9*10**(-3)
R_ja2 = 60
Ta2 = 50
T_j_2 = R_ja2*P_T3 + Ta2
print("A temperatura de junção estimada é", T_j_2, "°C")
```

```
A temperatura de junção estimada é 84.554 °C
```

Como 84.554°C é bem menor que 175°C (máxima temperatura de operação do diodo), o componente não precisa de um dissipador!!!

DIODO 2 (MBR20100CT): (pior caso = conversor Boost)

In [121…

```
P_T4 = 1.6855
R_ja2 = 60
Ta2 = 50
T_j_2 = R_ja2*P_T4 + Ta2
print("A temperatura de junção estimada é", T_j_2, "°C")
```

```
A temperatura de junção estimada é 151.13 °C
```

Como 151.13°C é maior que 80% de 175°C (máxima temperatura de operação do diodo), o componente precisa de um dissipador!!!

Dimensionamento do dissipador de calor para o diodo 2

O uso do dissipador de calor insere uma baixa resistência térmica em paralelo com a resistência R_{ca} do componente. Tal associação permite reduzir a resistência equivalente entre encapsulamento e ambiente e, assim, reduzir as temperaturas de cápsula e da junção.

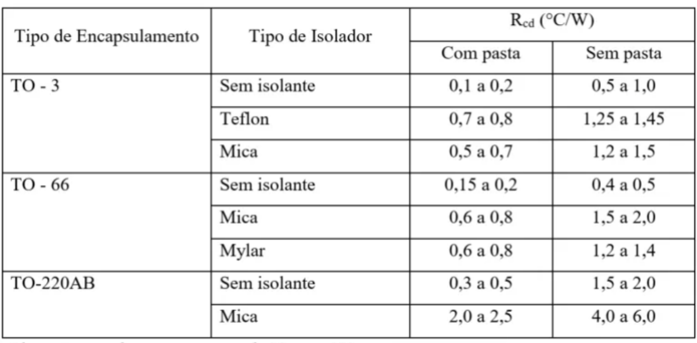
Equação que permite calcular a resistência térmica máxima do dissipador de calor:

$$R_{da} \leq \frac{T_{j,max} - T_a}{P_T} - R_{jc} - R_{cd}$$

Será usado 80% do valor fornecido no $T_{j,max}$ - margem de segurança.

Se o valor encontrado de R_{da} não for comercial, deve ser escolhido um valor menor e mais próximo do calculado.

R_{cd}, ressitência de interface, vai depender do material utilizado para separar o encapsulamento do dissipador de calor. Segue tabela a baixo.



Tipo de Encapsulamento	Tipo de Isolador	R_{cd} (°C/W)	
		Com pasta	Sem pasta
TO - 3	Sem isolante	0,1 a 0,2	0,5 a 1,0
	Teflon	0,7 a 0,8	1,25 a 1,45
	Mica	0,5 a 0,7	1,2 a 1,5
TO - 66	Sem isolante	0,15 a 0,2	0,4 a 0,5
	Mica	0,6 a 0,8	1,5 a 2,0
	Mylar	0,6 a 0,8	1,2 a 1,4
TO-220AB	Sem isolante	0,3 a 0,5	1,5 a 2,0
	Mica	2,0 a 2,5	4,0 a 6,0

Obs.: mica e mylar com espessura de 50 μm a 100 μm.

Foi escolhida o tipo TO-220AB, sem isolante e com pasta térmica.

In [122…

```
Rjc = 2
Rcd = 0.5
Tjmax = 140
Rda = Tjmax-Ta/P_T4 - Rjc - Rcd
print("A resistência térmica máxima do dissipador é", Rda, "°C/W")
```

```
A resistência térmica máxima do dissipador é 107.83521210323346 °C/W
```

HS 1509

Perímetro: 78 mm

Resistência Térmica: 19,8 ºC / W / 4

Peso Linear: 0,21 kg/m

Capacidade Térmica: 921 J/kg K

In [123…

```
Rda = 19.8
Tj = (Rjc + Rda + Rcd)*P_T4 + Ta
print("A temperatura de junção estimada com a presença do dissipador é", Tj, "°C")
```

```
A temperatura de junção estimada com a presença do dissipador é 87.58664999999999 °C
```

Circuito Integrado para Acionamento do MOSFET

Acionamento do mosfet IPB65R420CFD - boost:

In [124…

```
tr = 90*10**(-9)
Qg = 30.5*10**(-9)
Ig = Qg/tr
print("A corrente de gate é", "%.2f" % Ig, "A")
```

```
A corrente de gate é 0.34 A
```

Devido a magnitude da corrente, será usado um circuito de comando UCC27517 (suporta até 4A em sua saída).

In [125…

```
Vg = 12
Rg_tot = Vg/Ig
print("A resistência de gate total é", "%.2f" % Rg_tot, "ohms")
```

```
A resistência de gate total é 35.41 ohms
```

Analisando o datasheet do CI UCC27517:

- R_OL = 1 ohm
- R_OH = 1,4 ohms (1,4 x R_OL)
- R_G(int) = 1,75 (valor encontrado no modelo spice do componente)

In [126…

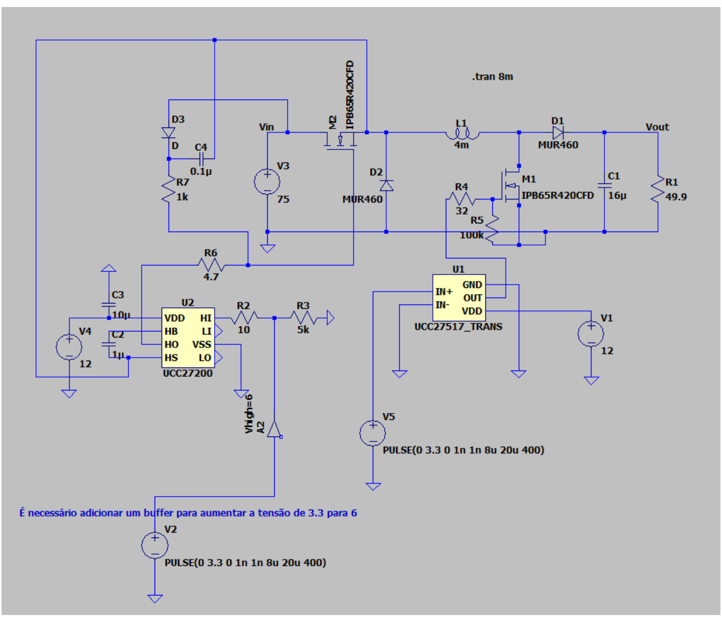
```
Roh = 1.4
Rg_int = 1.75
Rg_ext = Rg_tot - Roh - Rg_int
print("A resistência de gate é", "%.2f" % Rg_ext, "ohms")
```

```
A resistência de gate é 32.26 ohms
```

In [127…

```
Rs = 100000
print("A resistência Rs é", "%.2f" % Rs, "ohms - valor escolhido arbitrariamente")
```

```
A resistência Rs é 100000.00 ohms - valor escolhido arbitrariamente
```

Acionamento do mosfet IPB65R420CFD - buck:
Foi adicionado o CI UCC27200 - necessidade de colocar um circuito bootstrap para o controle do mosfet em high-side.