Termalização do Modelo de Ising

Nome: Arthur Pontes Nader

Matrícula: 2019022294

Bibliotecas

In [ ]:
```
from numba import jit
import numpy as np
import matplotlib.pyplot as plt
```

Funções

In [ ]:
```
def gerarRede(L):

    N = L*L
    rede = [np.random.choice([1,-1]) for i in range(N)]

    return np.array(rede)
```

In [ ]:
```
def gerarVizinhos(L):

  N = L*L
  vizinhos = np.zeros((N, 4), dtype = np.int32)

  for k in range(N):

      vizinhos[k][0] = k + 1
      if((k+1)%L) == 0:
          vizinhos[k][0] = k + 1 - L

      vizinhos[k][1] = k + L
      if(k > N - 1 - L):
          vizinhos[k][1] = k + L - N
      vizinhos[k][2] = k - 1

      if(k%L == 0):
          vizinhos[k][2] = k + L - 1

      vizinhos[k][3] = k - L
      if(k < L):
          vizinhos[k][3] = k + N - L

  return vizinhos
```

In [ ]:
```
def calcularEnergia(s, viz):

    energia = 0
    for i in range(len(s)):
        h = s[viz[i][0]] + s[viz[i][1]]
        energia = energia - s[i]*h

    return energia
```

In [ ]:
```
@jit(nopython=True)
```

```python
def expos(beta):

    ex = np.zeros(5,dtype=np.float32)
    ex[0]=np.exp(8.0*beta)
    ex[1]=np.exp(4.0*beta)
    ex[2]=1.0
    ex[3]=np.exp(-4.0*beta)
    ex[4]=np.exp(-8.0*beta)

    return ex
```

In [ ]:

```python
@jit(nopython=True)
def mcstep(beta,s,viz,ener,mag):

    N=len(s)
    ex=expos(beta)

    for i in range(N):
        h = s[viz[i,0]]+s[viz[i,1]]+s[viz[i,2]]+s[viz[i,3]] # soma dos vizinhos
        de = int(s[i]*h*0.5+2)
        if np.random.random() < ex[de]:
            ener=ener+2*s[i]*h
            mag -= 2*s[i]
            s[i]=-s[i]

    return ener,mag,s
```

In [ ]:

```python
def Metropolis(L, temperatura,  iteracoes = 1500):

    configuracao_atual = gerarRede(L)
    vizinhos = gerarVizinhos(L)

    beta = 1/temperatura
    energia = calcularEnergia(configuracao_atual, vizinhos)
    magnetizacao = np.sum(configuracao_atual)

    energias, magnetizacoes = [energia], [magnetizacao]

    for i in range(iteracoes):

      energia, magnetizacao, configuracao_atual = mcstep(beta, configuracao_atual, vizin
      energias.append(energia)
      magnetizacoes.append(magnetizacao)

    return energias, magnetizacoes
```

In [ ]:

```python
def gerarGrafico(dados, titulo, L, legenda):

  plt.figure(figsize=(16, 8))
  for dado in dados:
    plt.plot(dado)
  plt.legend(legenda, title = "Temperatura")
  plt.title(titulo, fontsize = 22)
  plt.show()
```

Resultados obtidos

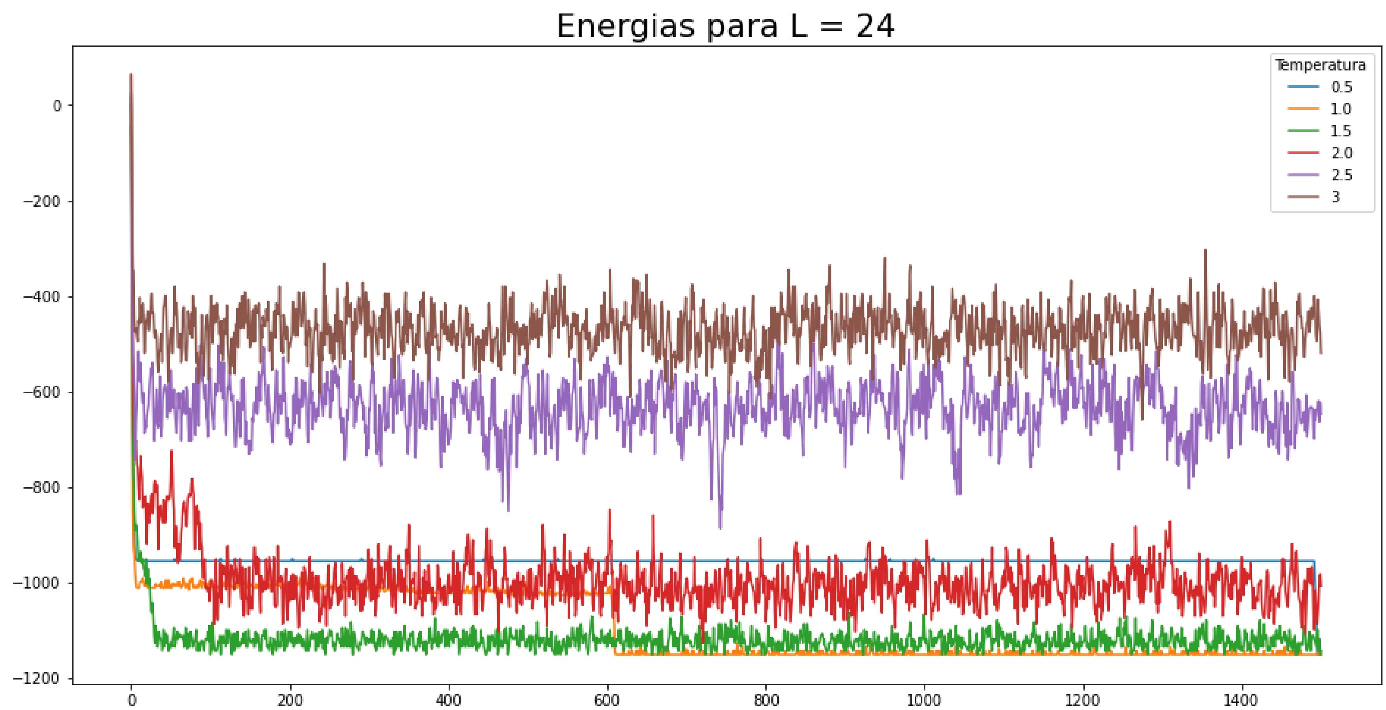
In [ ]:

```python
temperaturas = [0.5, 1.0, 1.5, 2.0, 2.5, 3]
```

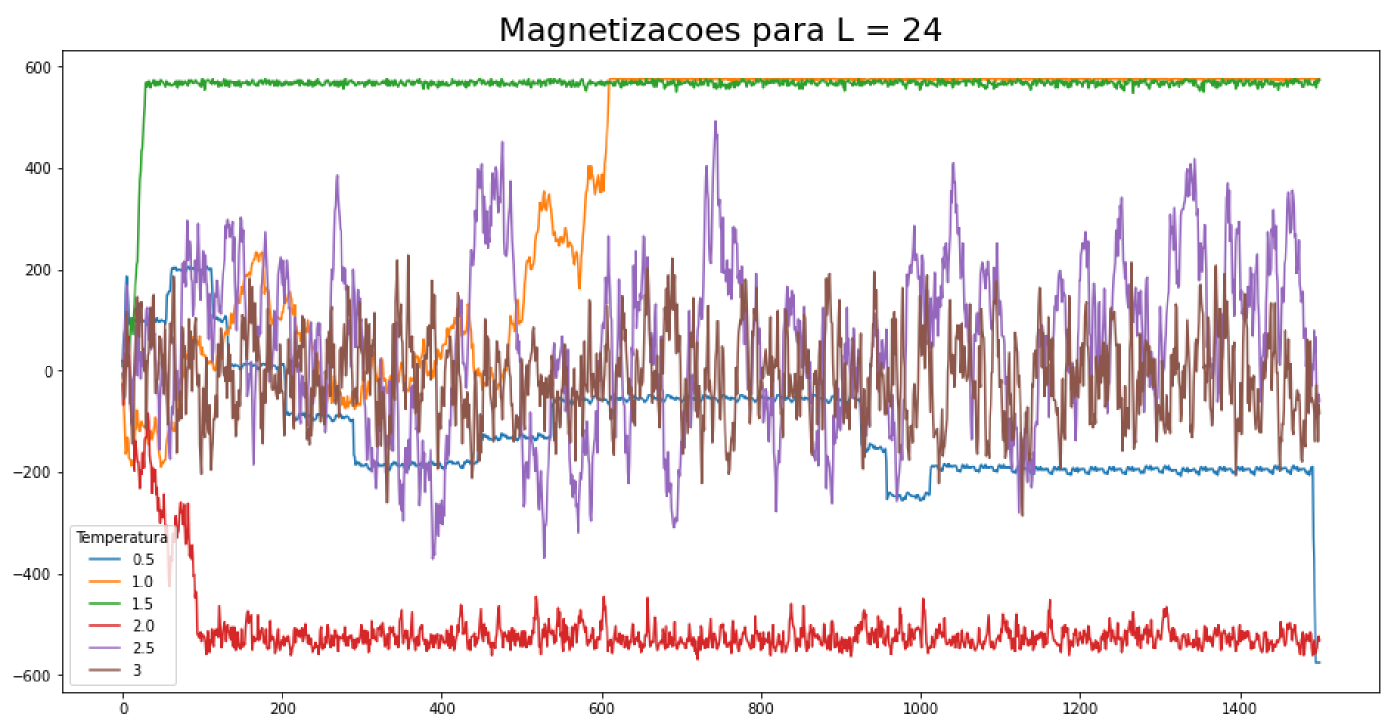
L = 24

In [ ]:
```python
L = 24
dados = []
for temp in temperaturas:
    dados.append(Metropolis(L, temp))
```

In [ ]:
```python
gerarGrafico([dado[0] for dado in dados], "Energias para L = " + str(L), L, temperaturas
```



In [ ]:
```python
gerarGrafico([dado[1] for dado in dados], "Magnetizacoes para L = " + str(L), L, tempera
```

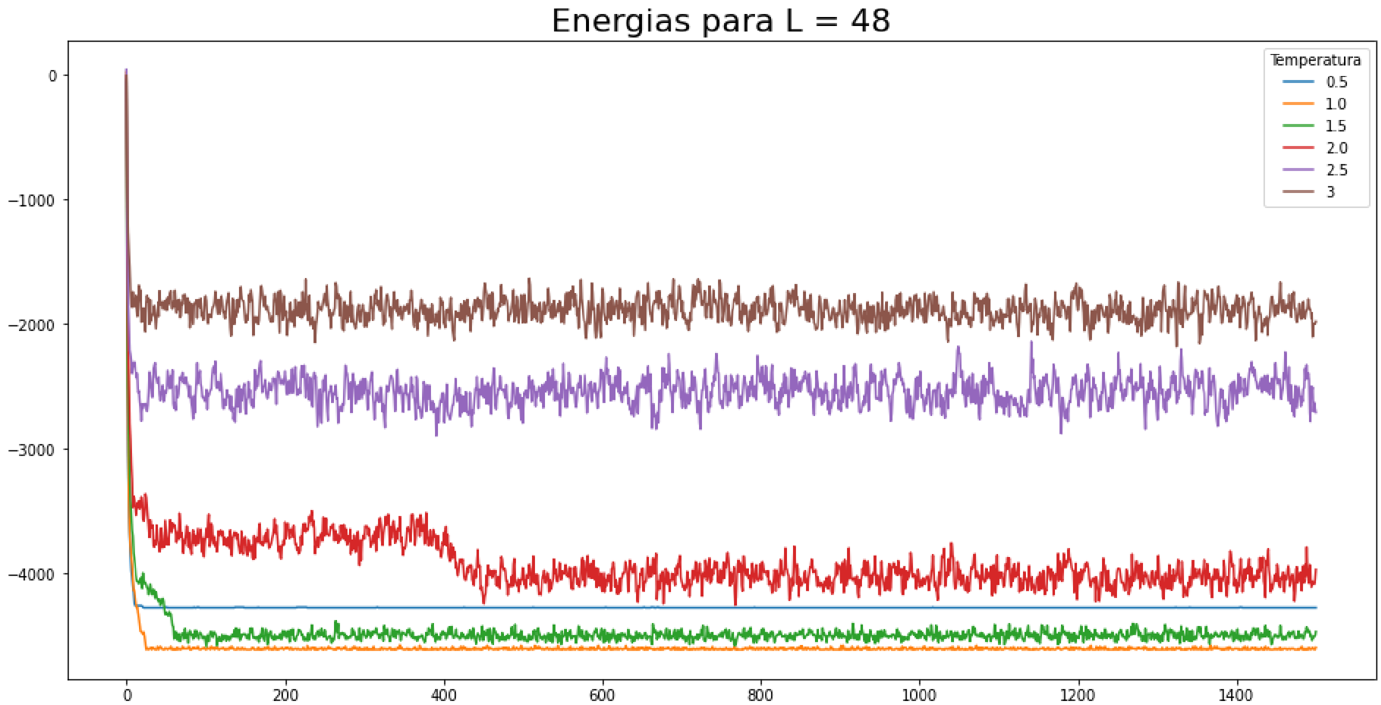


Percebe-se que para L = 24, os valores de magnetização e energia têm a tendência de variar muito ao passar das iterações, sendo difícil de perceber um valor estácionário para as temperaturas de 2.0, 2.5 e 3.0. \ Além disso, nota-se que a magnetização para a temperatura de 0.5 apresentou um comportamento peculiar, tendo vários trechos estacionários.

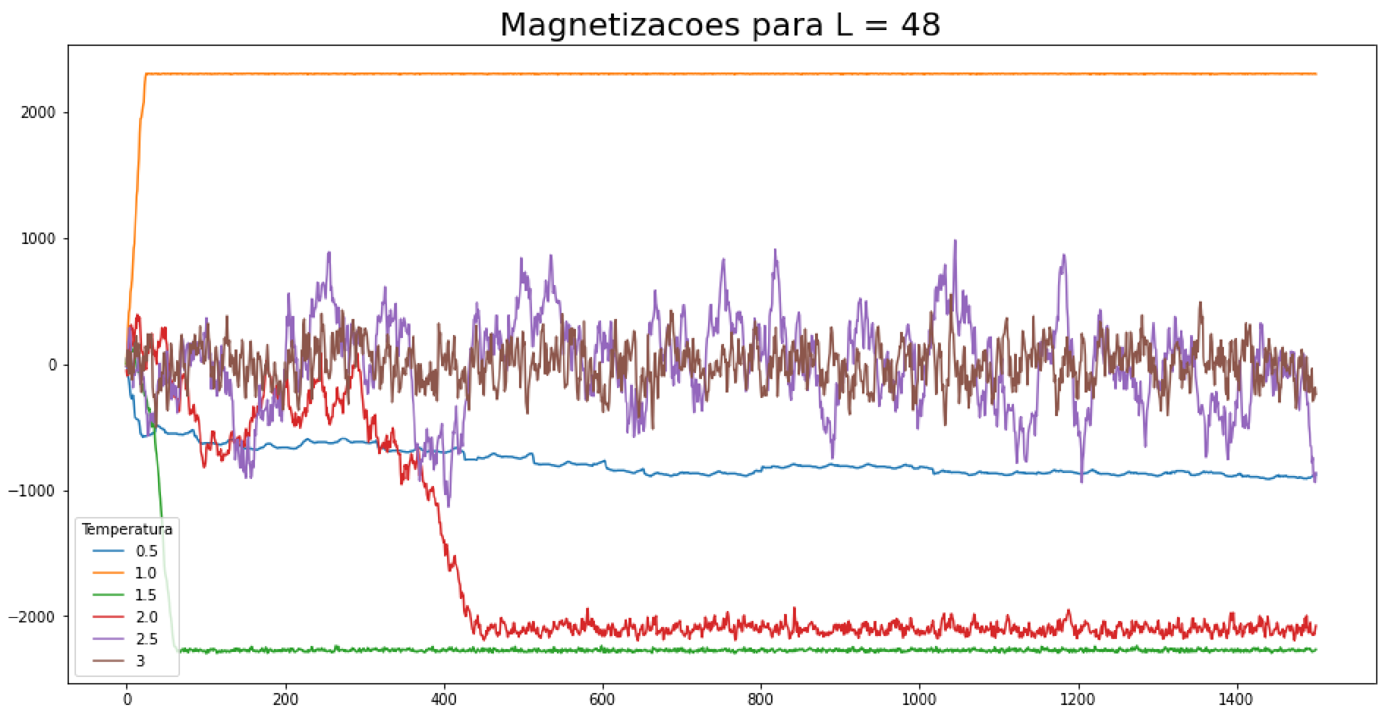
L = 48

In [ ]:
```
L = 48
dados = []
for temp in temperaturas:
    dados.append(Metropolis(L, temp))
```

In [ ]:
```
gerarGrafico([dado[0] for dado in dados], "Energias para L = " + str(L), L, temperaturas
```



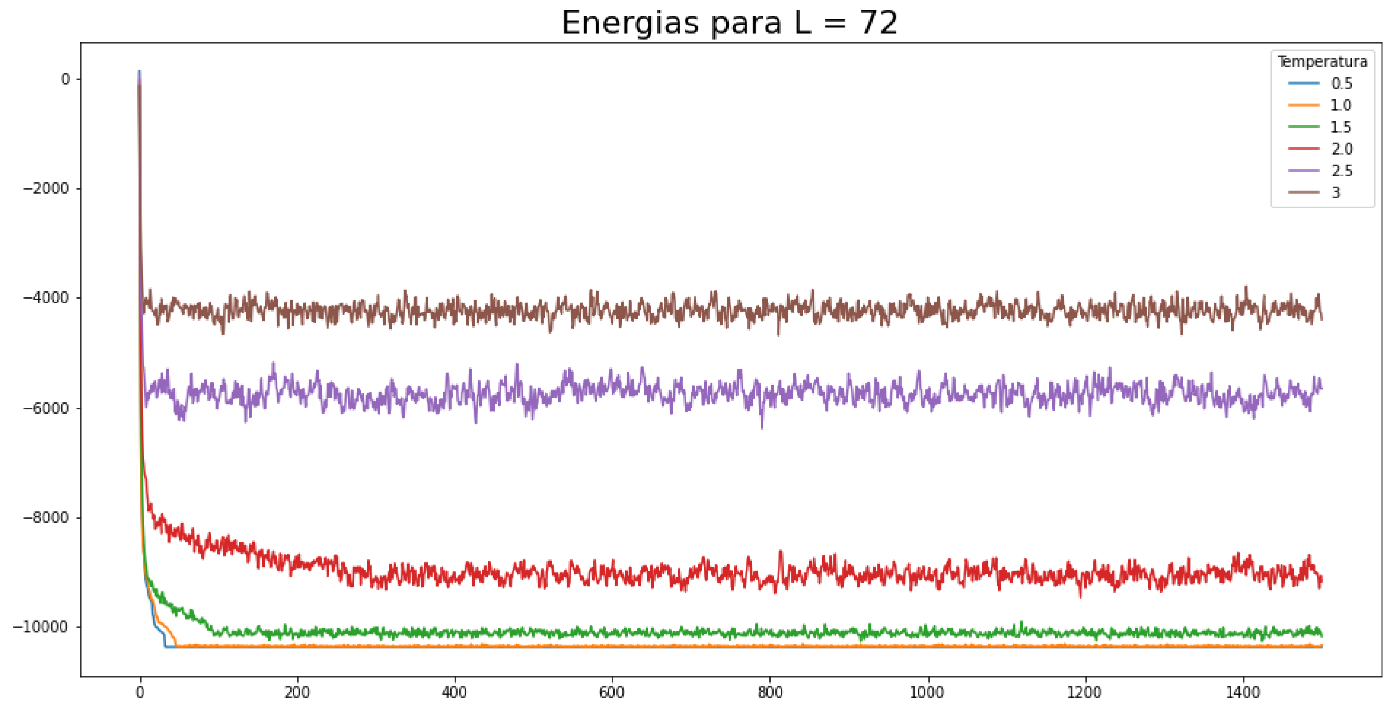
In [ ]:
```
gerarGrafico([dado[1] for dado in dados], "Magnetizacoes para L = " + str(L), L, tempera
```



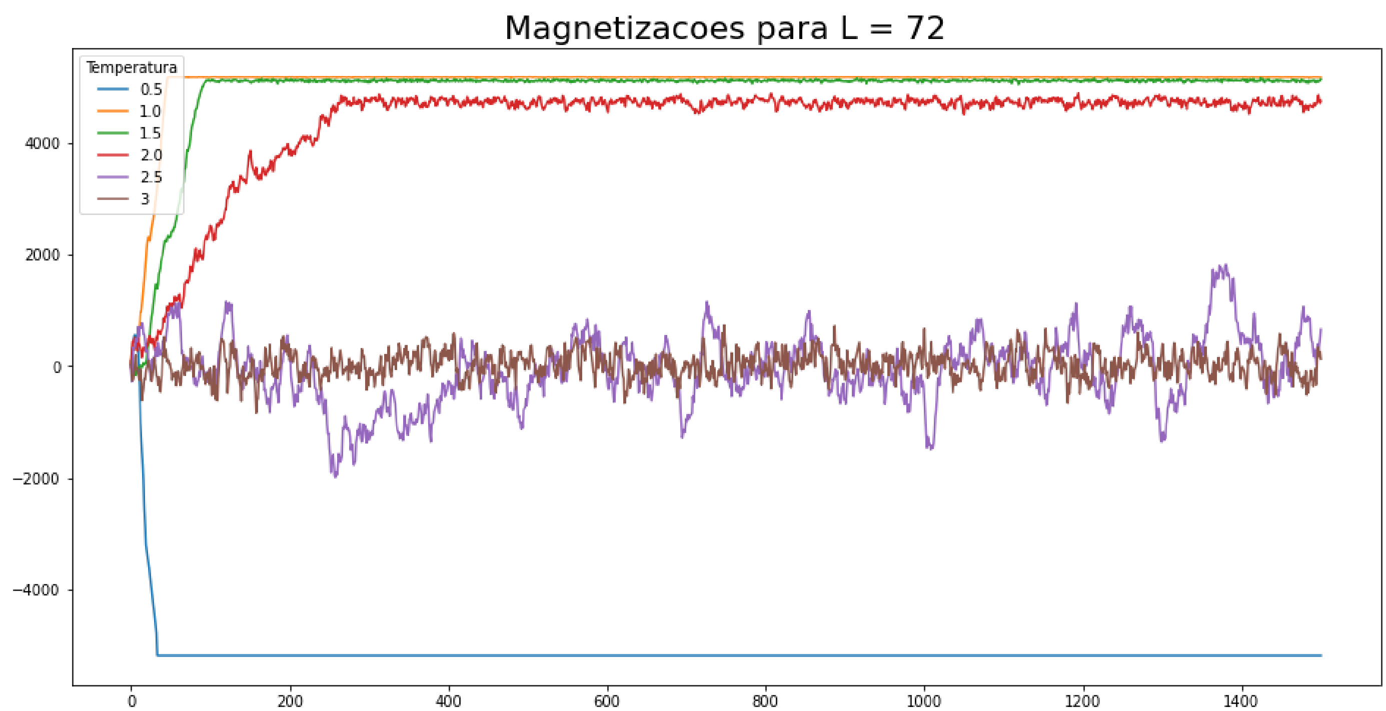
L = 72

In [ ]:
```
L = 72
dados = []
for temp in temperaturas:
    dados.append(Metropolis(L, temp))
```

In [ ]:
```
gerarGrafico([dado[0] for dado in dados], "Energias para L = " + str(L), L, temperaturas
```



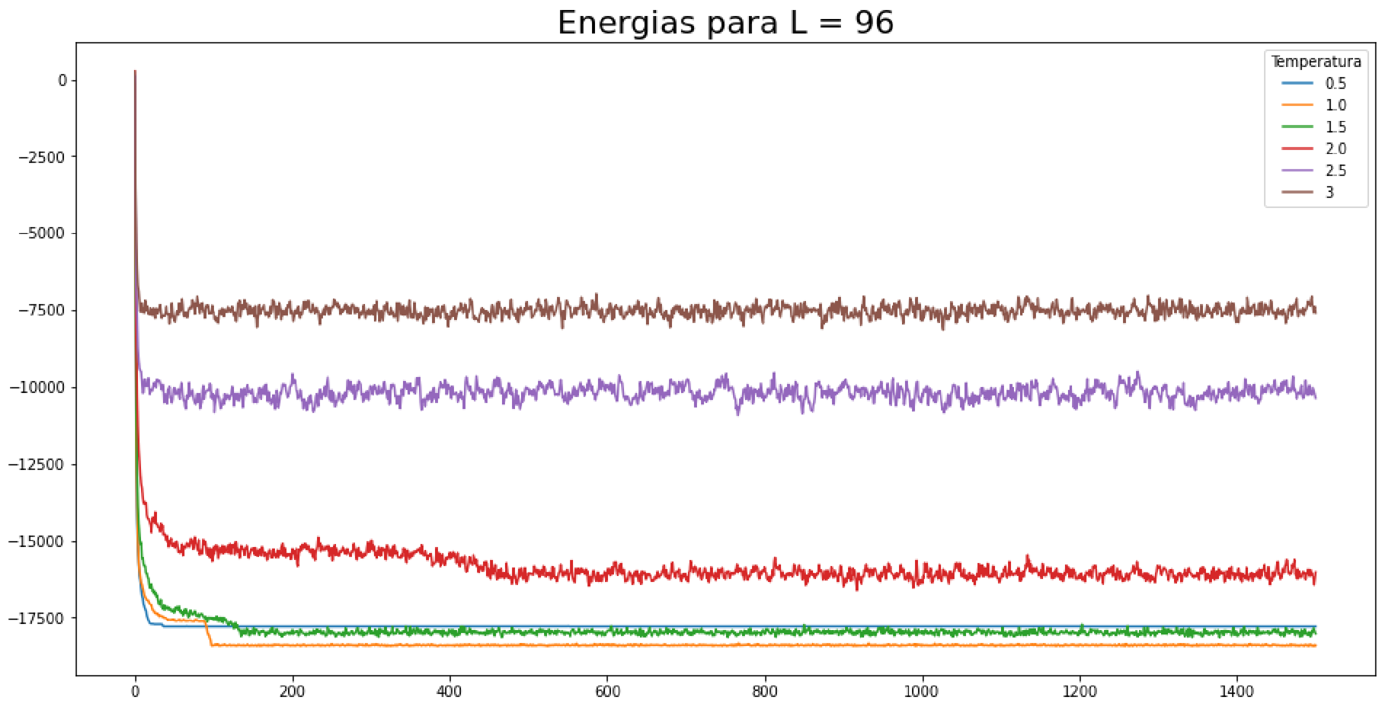
In [ ]:
```
gerarGrafico([dado[1] for dado in dados], "Magnetizacoes para L = " + str(L), L, tempera
```



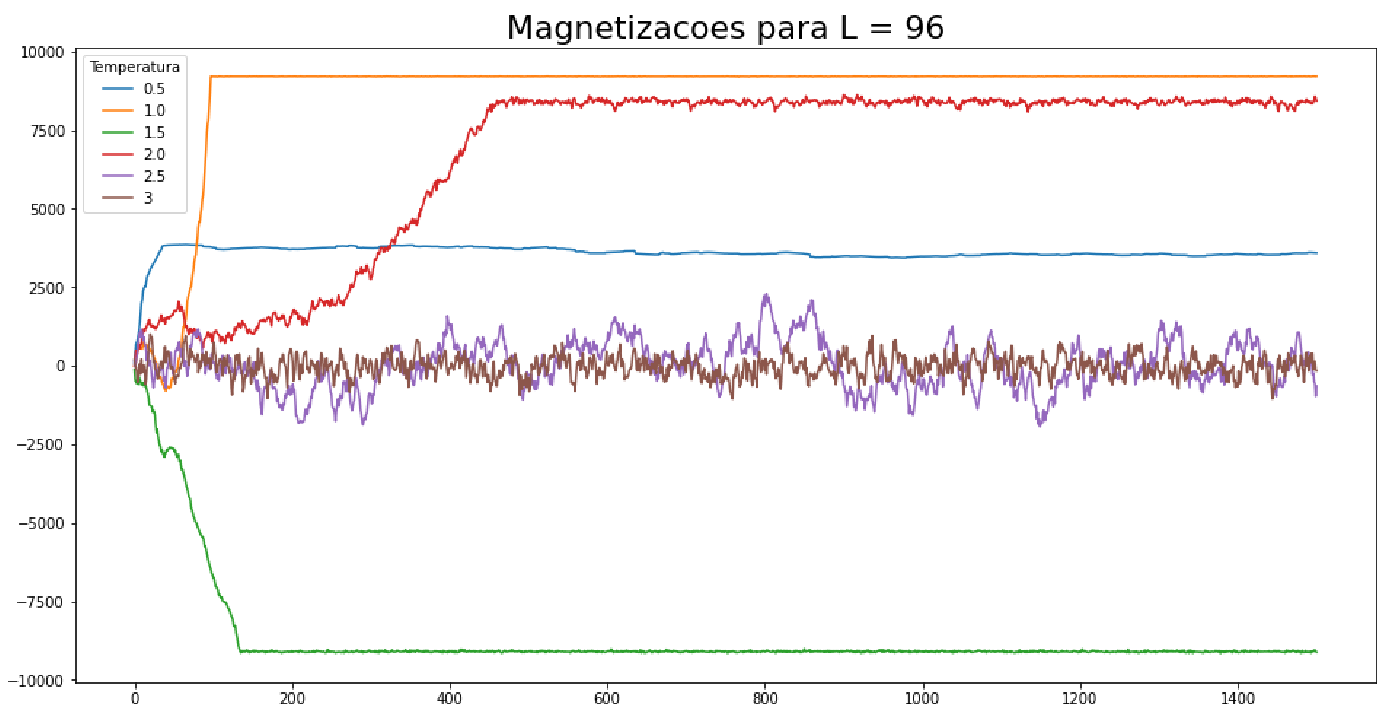
L = 96

In [ ]:
```
L = 96
dados = []
for temp in temperaturas:
    dados.append(Metropolis(L, temp))
```

In [ ]:
```
gerarGrafico([dado[0] for dado in dados], "Energias para L = " + str(L), L, temperaturas
```



In [ ]:
```
gerarGrafico([dado[1] for dado in dados], "Magnetizacoes para L = " + str(L), L, tempera
```



Análise dos Resultados e Conclusão

Percebe-se que para L = 24, os valores de magnetização e energia têm a tendência de variar muito ao passar das iterações, sendo difícil de perceber um valor estácionário para as temperaturas de 2.0, 2.5 e 3.0. Além disso, nota-se que a magnetização para a temperatura de 0.5 apresentou um comportamento peculiar, tendo vários trechos estacionários.

Para L = 48, ainda é perceptível que o estado estacionário possui uma margem muito grande para os valores de temperatura de 2.5 e 3.0. No gráfico de energias, já é possível observar uma certa tendência de quanto maior a temperatura, mais próximo de zero será a estabilização do sistema.

Já para L = 72 e L = 96, os gráficos de energias ficaram bem similares. Entretanto, as magnetizações e energias para as temperaturas de 2.5 e 3.0 continuaram a apresentar um comportamento relativamente instável ao decorrer das iterações.

As tabelas a seguir mostram qualitativamente a iteração em que ocorreu a termalização para cada um dos valores de L e para cada temperatura. Situações em que o sistema aparentemente não entra em estado estacionário foram marcadas com um traço (-).

Tabela 1 - Número de passos para energia estabilizar

Valor de L	Temperatura					
	0.5	1.0	1.5	2.0	2.5	3.0
24	5	610	25	-	-	-
48	10	20	60	-	-	-
72	20	30	100	-	-	-
96	25	100	150	440	-	-

Tabela 2 - Número de passos para magnetização estabilizar

Valor de L	Temperatura					
	0.5	1.0	1.5	2.0	2.5	3.0
24	-	600	10	-	-	-
48	-	20	60	420	-	-
72	30	35	120	230	-	-
96	20	100	140	440	-	-

Os resultados obtidos permitem concluir que quanto maior o tamanho da rede mais percetível se torna o momento em que ocorre a termalização e menor fica a variação dos valores no estado estacionário.

Em baixas temperaturas, nota-se que uma rede pequena tem um comportamento peculiar quando se trata dos valores de magnetização, enquanto em altas temperaturas, percebe-se que o sistema fica instável e não entra no estado estacionário.

Por fim, observa-se também que para os diversos tamanhos de rede e temperatura, a magnetização e a energia entram no estado estacionário na mesma ou em iterações bem próximas, tal qual esperado.