

Equações:

Tubarões:

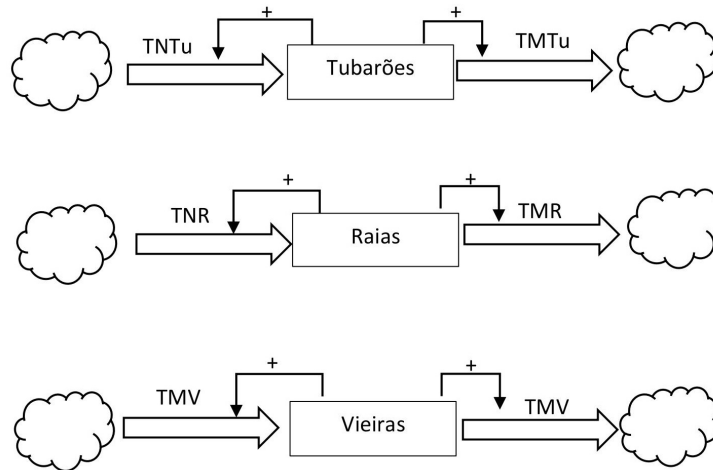
$$Tu(t + 1) = Tu(t) + TNTu \cdot Tu(t) - TMTu \cdot Tu(t)$$

Raias:

$$R(t + 1) = R(t) + TNR \cdot R(t) - TMR \cdot R(t)$$

Vieiras:

$$V(t + 1) = V(t) + TNV \cdot V(t) - TMV \cdot V(t)$$



In [9]:

```
import matplotlib.pyplot as plt
%matplotlib inline

#Sem predação

#Parâmetros Tubarão
TNTu = 0.21  #Taxa de Natalidade dos Tubarões em Tubarões/Tubarões.ano
TMTu = 0.15  #Taxa de Mortalidade dos Tubarões em Tubarões/Tubarões.ano

#Parâmetros Raias
TNR = 0.087  #Taxa de Natalidade das Raias em Raias/Raias.ano
TMR = 0.076  #Taxa de Mortalidade das Raias em Raias/Raias.ano

#Parâmetros Vieiras
TNV = 0.196  #Taxa de Mortalidade das Vieiras em Vieiras/Vieiras.ano
TMV = 0.146  #Taxa de Mortalidade das Vieiras em Vieiras/Vieiras.ano

#Listas Inicias
Tu = [0]*20
Tu[0] = 100  #População inicial dos Tubarões

R = [0]*20
R[0] = 1000  #População inicial das Raias

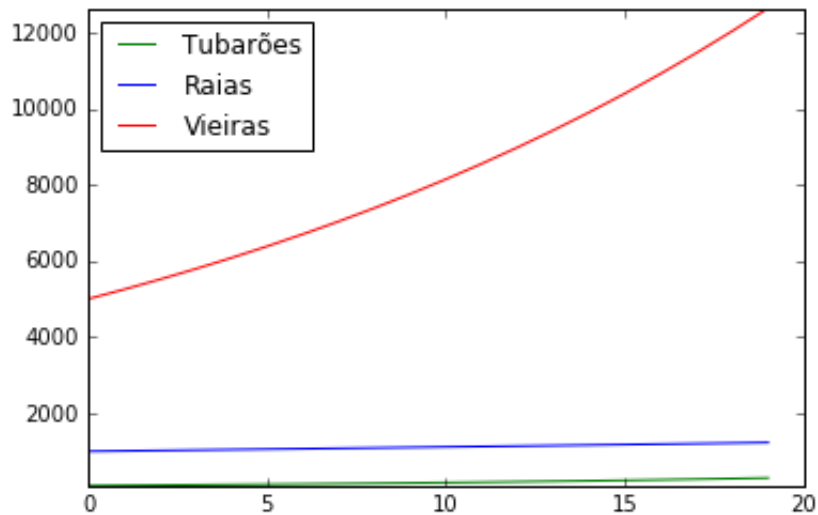
V = [0]*20
V[0] = 5000  #População inicial das Vieiras

#Equações a diferenças
for i in range(1,20):
    Tu[i] = Tu[i - 1] + ( (TNTu*Tu[i - 1]) - (TMTu*Tu[i - 1]) )
    R[i] = R[i - 1] + ( (TNR*R[i - 1]) - (TMR*R[i - 1]) )
    V[i] = V[i - 1] + ( (TNV*V[i - 1]) - (TMV*V[i - 1]) )
```

In [10]:

```
#Plotando os gráficos
Tempo = list(range(20))
plt.plot(Tempo, Tu, 'g', label = 'Tubarões')
plt.plot(Tempo, R, 'b', label = 'Raias')
plt.plot(Tempo, V, 'r', label = 'Vieiras')
plt.legend(loc = 'upper left')
plt.axis([0, 20, min(Tu), max(V)])
plt.show()
```

#
Legenda dos gráficos
#



Tubarões:

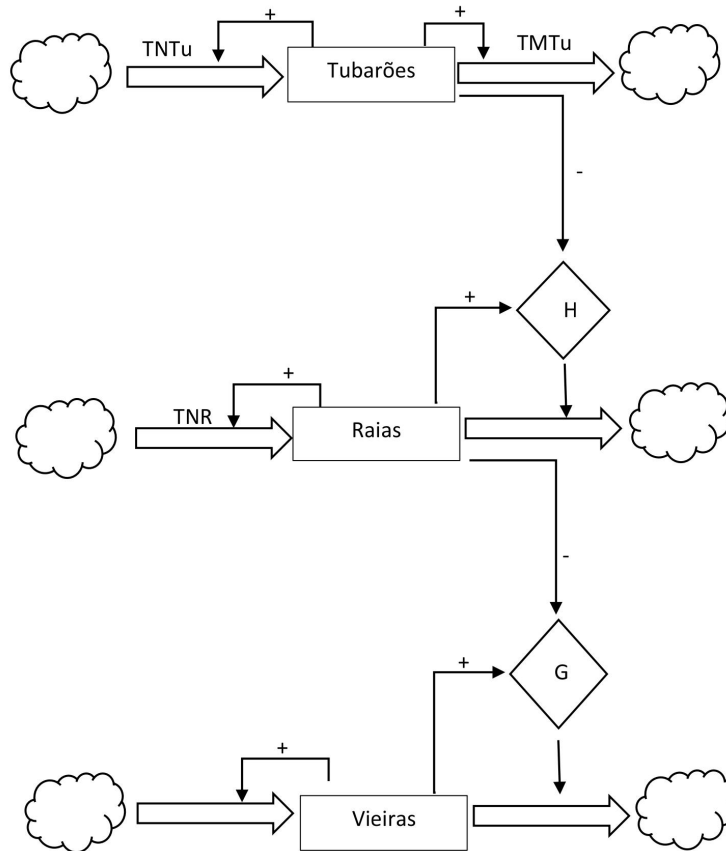
$$Tu(t + 1) = Tu(t) + (TNTu - TMTu) * Tu(t) * (R/Rc - 1)$$

Raias:

$$R(t + 1) = R(t) + (TNR - TMR) * R * (1 - T/Tc)$$

Vieiras:

$$V(t + 1) = V(t) + (TNV - TMV) * V * (1 - R/Rc)$$



In [3]:

```
#Parâmetros Tubarão
```

```
TNTu = 0.21
```

```
TMTu = 0.15
```

```
#Parâmetros Raias
```

```
TNR = 0.087
```

```
TMR = 0.076
```

```
Pr = 0.0000008
```

```
#Parâmetros Vieiras
```

```
TNV = 0.196
```

```
TMV = 0.146
```

```
Pv = 0.0000005
```

```
#Parâmetros
```

```
Rc = 700 #Taxa que determina o número mínimo de raias para que os tubaraões tenham o
```

```
Vc = 2000 # Taxa que determina o número mínimo de Vieiras para que as Raias tenham o
```

```
Tu = [0]*300
```

```
Tu[0] = 50
```

```
R = [0]*300
```

```
R[0] = 1000
```

```
V = [0]*300
```

```
V[0] = 5000
```

```
#Equações a diferenças
```

```
for i in range(1,300):
```

```
    Tu[i] = Tu[i - 1] + ( (TNTu - TMTu)*Tu[i - 1]*(( R[i - 1]/Rc)- 1) )
```

```
    R[i] = R[i - 1] + ( ((-1*(TNR - TMR))*R[i - 1]) + (((TNR - TMR)*R[i - 1]*V[i - 1]
```

```
    V[i] = V[i - 1] + ( (TNV - TMV)*V[i - 1]*(1 - (R[i - 1]/Rc)) )
```

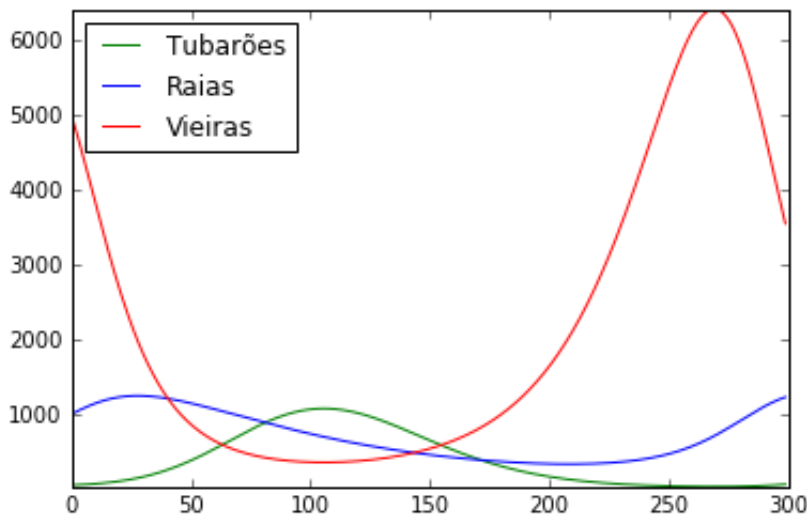
In [4]:

#Plotando os gráficos

```

Tempo2 = list(range(300))
plt.plot(Tempo2, Tu, 'g', label = 'Tubarões')
plt.plot(Tempo2, R, 'b', label = 'Raias')
plt.plot(Tempo2, V, 'r', label = 'Vieiras')
plt.legend(loc = 'upper left')
plt.axis([0, 300, min(Tu), max(V)])
plt.show()

```

**Tubarões:**

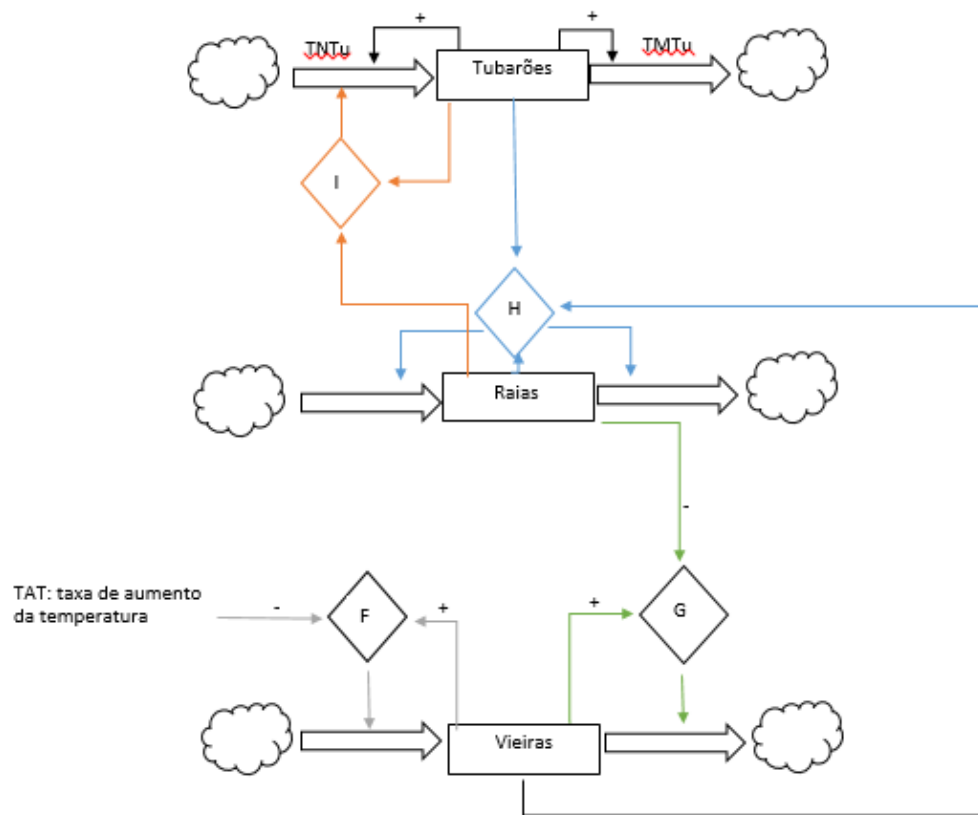
$$Tu_{t+1} = Tu_t + ((TNTu - TMTu) \cdot Tu_t \cdot \left(\frac{R}{R_c} - 1\right))$$

Raias

$$R_{(t+1)} = R_{(t)} + (-\gamma \cdot R + \gamma \cdot R \cdot \frac{V}{V_c} - Pr \cdot R \cdot T)$$

Vieiras

$$V_{t+1} = V_t + TNV \cdot V \cdot \left(1 - \frac{TAT}{TAT_c}\right) - V \cdot (TMV + Pv \cdot R)$$



In [5]:

```
#Quando a TAT da tese é considerada
```

```
#Parâmetros Tubarão
```

```
TNTu = 0.21
```

```
TMTu = 0.15
```

```
#Parâmetros Raias
```

```
TNR = 0.087
```

```
TMR = 0.076
```

```
Pr = 0.0000008
```

```
#Parâmetros Vieiras
```

```
TNV = 0.196
```

```
TMV = 0.146
```

```
Pv = 0.0000005
```

```
#Parâmetros
```

```
Rc = 700
```

```
Tc = 100
```

```
Vc = 2000
```

```
# A TATc foi definida por esse valor com base no fato de que, na chamada COP 21, o Li  
#definido em 2°C, portanto TATc = 2,5%
```

```
TAT = 0.0375
```

```
# A TAT foi definida por esse valor, pois segundo estudos, até o fim deste século a t  
#cerca de 3°C, portanto TAT = 3,75%
```

```
TATc = 0.025
```

```
#Listas Inicias
```

```
Tu = [0]*80
```

```
Tu[0] = 50
```

```
R = [0]*80
```

```
R[0] = 1000
```

```
V = [0]*80
```

```
V[0] = 5000
```

```
#Equações a diferenças
```

```
for i in range(1,80):
```

```
    Tu[i] = Tu[i - 1] + ( (TNTu - TMTu)*Tu[i - 1]*((R[i - 1]/Rc) - 1))
```

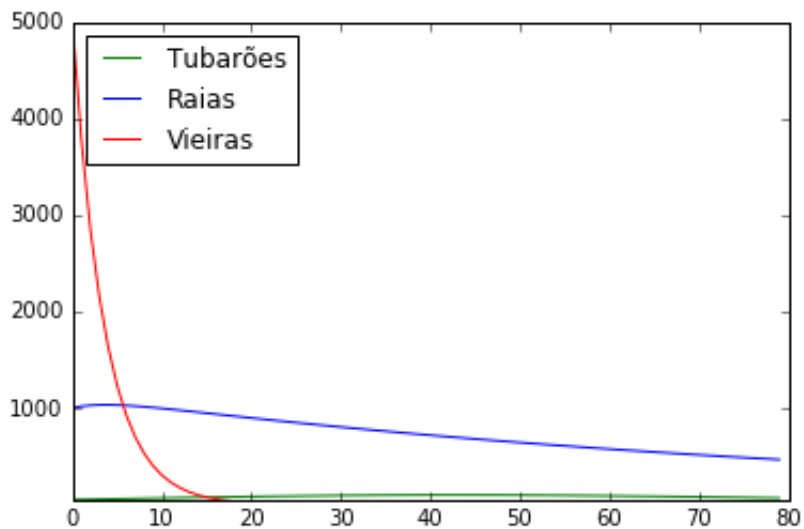
```
    R[i] = R[i - 1] + ( ((-1*(TNR - TMR))*R[i - 1]) + (((TNR - TMR)*R[i - 1]*V[i - 1]
```

```
    V[i] = V[i - 1] + ( (TNV*V[i - 1]*(1 - TAT/TATc)) - (V[i-1]*(TMV + Pv*R[i - 1])) )
```


In [6]:

#Plotando os gráficos

```
Tempo3 = list(range(80))  
plt.plot(Tempo3, Tu, 'g', label = 'Tubarões')  
plt.plot(Tempo3, R, 'b', label = 'Raias')  
plt.plot(Tempo3, V, 'r', label = 'Vieiras')  
plt.legend(loc = 'upper left')  
plt.axis([0, 80, min(Tu), max(V)])  
plt.show()
```



In [7]:

```
#Quando a TAT da tese é considerada
```

```
#Parâmetros Tubarão
```

```
TNTu = 0.21
```

```
TMTu = 0.15
```

```
#Parâmetros Raias
```

```
TNR = 0.087
```

```
TMR = 0.076
```

```
Pr = 0.0000008
```

```
#Parâmetros Vieiras
```

```
TNV = 0.196
```

```
TMV = 0.146
```

```
Pv = 0.0000005
```

```
#Parâmetros
```

```
Rc = 700
```

```
Tc = 100
```

```
Vc = 2000
```

```
TAT = [0]*41
```

```
TAT[0] = 0
```

```
Tu = [0]*41
```

```
Tu[0] = 50
```

```
R = [0]*41
```

```
R[0] = 1000
```

```
V = [0]*41
```

```
V[0] = 5000
```

```
TATc = 0.025
```

In [8]:

```

lista_de_variacao_vieiras_pela_variacao_do_TAT = []

#Equações a diferenças

for j in range (1,41):
    TAT[j] = TAT[j - 1] + 0.005

    for i in range(1,41):
        Tu[i] = Tu[i - 1] + ( (TNTu - TMTu)*Tu[i - 1]*((R[i - 1]/Rc) - 1))
        R[i] = R[i - 1] + ( ((-1*(TNR - TMR))*R[i - 1]) + (((TNR - TMR)*R[i - 1]*
        V[i] = V[i - 1] + ( (TNV*V[i - 1]*(1 -TAT[j]/TATc)) - (V[i-1]*(TMV + Pv*R
        variacao = V[40]/V[0]
        lista_de_variacao_vieiras_pela_variacao_do_TAT.append(variacao)

lista_de_variacao_vieiras_pela_variacao_do_TAT.append(0)    #Adicionando um elemento
                                                             #da população possui 80 numer
print(lista_de_variacao_vieiras_pela_variacao_do_TAT[5])

plt.plot(TAT, lista_de_variacao_vieiras_pela_variacao_do_TAT)
plt.axis([ 0, 0.025, min(lista_de_variacao_vieiras_pela_variacao_do_TAT), max(lista_d
plt.xlabel('TAT')
plt.ylabel('Vieiras')
plt.title(r'Variação da população de Vieiras em função da variação da TAT')
plt.show()

```

0.00027053622558248936



In []:

In []: