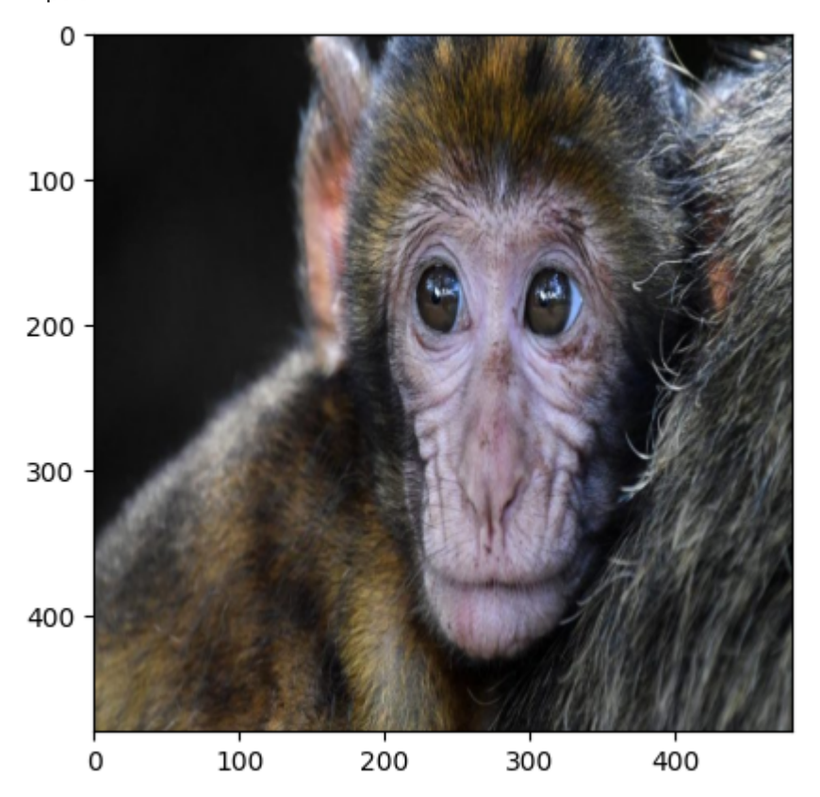
In [46]:
```
pip install opencv-contrib-python
```

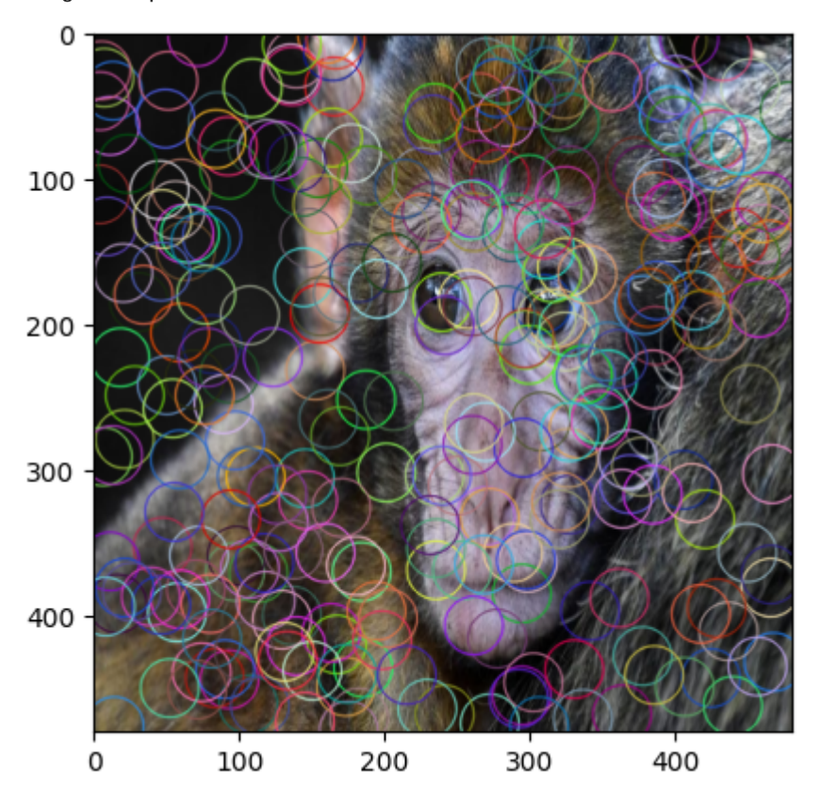
```
Collecting opencv-contrib-python
  Downloading opencv_contrib_python-4.7.0.72-cp37-abi3-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (67.9 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 67.9/67.9 MB 7.4 MB/s eta 0:00:00m eta 0:00:01[36m0:00:01
Requirement already satisfied: numpy>=1.17.0 in /home/luisa/Documents/faculdade/optativas/inf 492/inf492/lib/python3.10/site-packages (from opencv-contrib-python) (1.24.2)
Installing collected packages: opencv-contrib-python
Successfully installed opencv-contrib-python-4.7.0.72
Note: you may need to restart the kernel to use updated packages.
```

In [15]:
```python
import cv2 as cv
import matplotlib.pyplot as plt
import random
import numpy as np

#Amostragem aleatória de 300 pontos
image = cv.imread("image.jpg")
image = cv.resize(image, (480, 480))

height, width, channels = image.shape

print("input:")
plt.imshow(cv.cvtColor(image, cv.COLOR_BGR2RGB))
plt.show()

keypoints =[]

for i in range(300):
    keypoint = cv.KeyPoint()
    h = random.randint(0, height - 1)
    w = random.randint(0, width - 1)
    keypoint.pt = (h,w)
    keypoint.size = 40
    keypoints.append(keypoint)

image_rgb = cv.cvtColor(image, cv.COLOR_BGR2RGB)

output = cv.drawKeypoints(image_rgb, keypoints, np.array([]), flags=cv.DRAW_MATCHES_FLAGS_DRAW_RICH_KEYPOINTS)

#Descrevendo keypoints usando SIFT
print("===Describing keypoints using SIFT===")
image_gray = cv.cvtColor(image, cv.COLOR_BGR2GRAY)

descriptors = sift.compute(image_gray, keypoints)
print("keypoint dimension: ", len(descriptors[1][0]))
print("descriptor of first keypoint", descriptors[1][0])

print("image output:")
plt.imshow(output)
plt.show()

print("number of keypoints is", len(keypoints))
```

input:



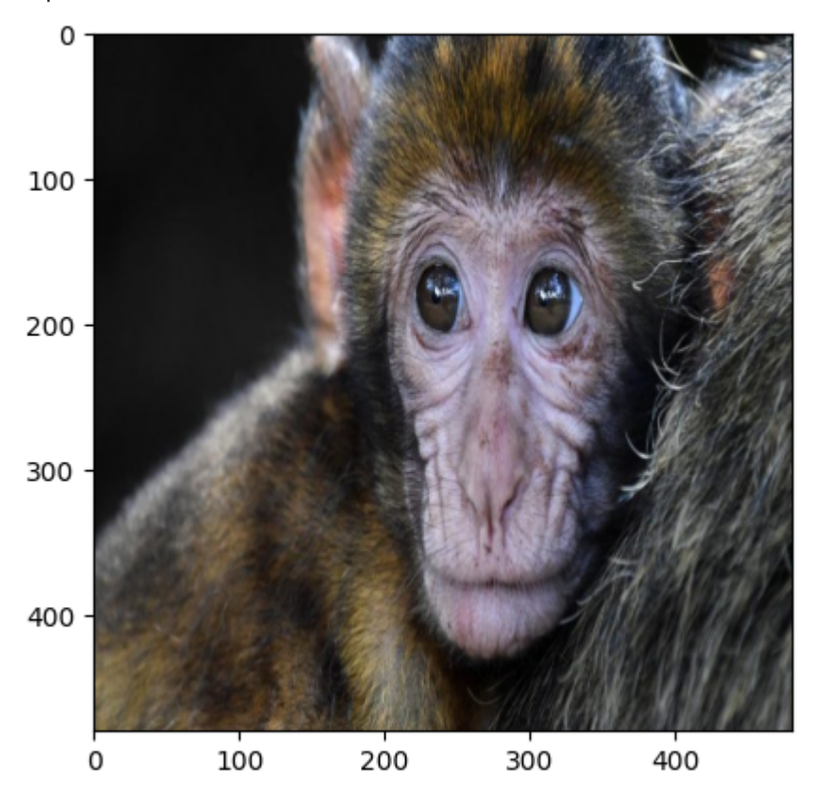
```
===Describing keypoints using SIFT===
keypoint dimension:  128
descriptor of first keypoint [ 42.  34.  25.  29.  27.  22.  20.  39.  72.  67.  72.  50.  44.  58.
  77.  92.  24.  24.  53.  64.  22.  24.  77.  67.   0.   0.   1.   1.
   0.   0.   1.   1.  32.  43.  37.  30.  34.  52.  51.  37.  62.  67.
 104.  90.  43.  41.  75. 104.  30.  21.  75. 100.  30.  15.  49.  90.
   0.   0.   1.   1.   0.   0.   0.   1.  46.  42.  44.  53.  63.  63.
  31.  27.  71.  48.  71.  92.  50.  31.  43.  95.  41.  19.  46.  85.
  27.  10.  26.  75.   0.   0.   0.   1.   0.   0.   0.   0.  28.  24.
  23.  46.  64.  44.  23.  20.  54.  31.  31.  62.  47.  28.  32.  81.
  27.   8.  15.  52.  25.   6.  16.  46.   0.   0.   0.   0.   0.   0.
   0.   0.]
image output:
```



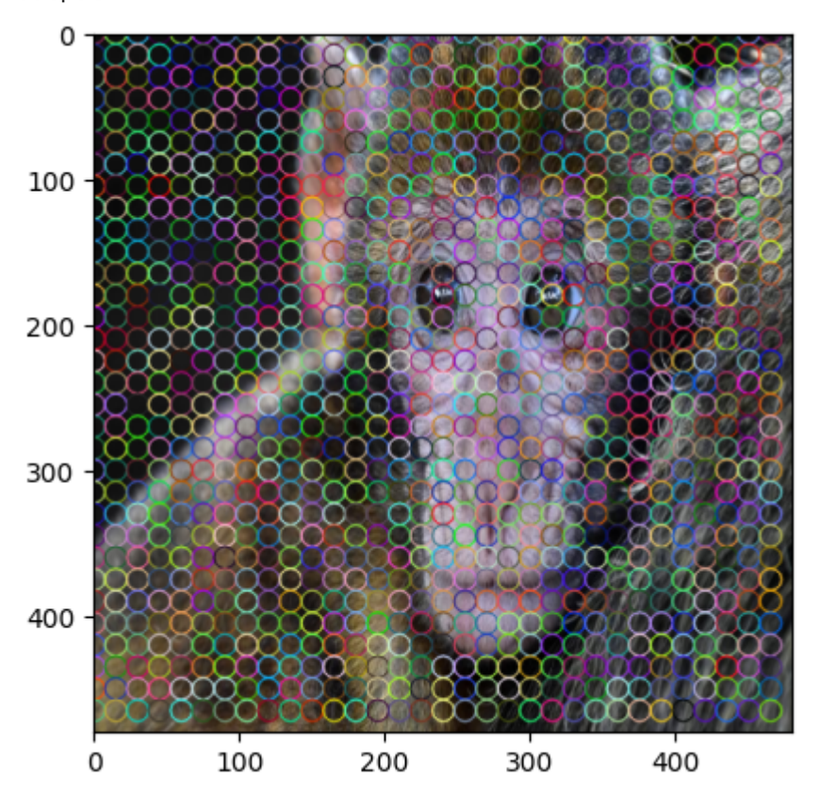
```
number of keypoints is 300
```

In [16]:
```python
import cv2 as cv
import matplotlib.pyplot as plt
import random
import numpy as np

#Amostragem em Grid de tamanho 15 x 15
image = cv.imread("image.jpg")
image = cv.resize(image, (480, 480))

height, width, channels = image.shape

print("input:")
plt.imshow(cv.cvtColor(image, cv.COLOR_BGR2RGB))
plt.show()

keypoints = []
for i in range(width):
    if(i%15 == 0):
        for j in range(height): #(0,0), (0,15), ... , (15,0), (15,15), ...
            if(j % 15 == 0):
                keypoint = cv.KeyPoint()
                keypoint.pt = (i, j)
                keypoint.size = 15
                keypoints.append(keypoint)

image_rgb = cv.cvtColor(image, cv.COLOR_BGR2RGB)
output = cv.drawKeypoints(image_rgb, keypoints, np.array([]), flags=cv.DRAW_MATCHES_FLAGS_DRAW_RICH_KEYPOINTS)

#Descrevendo keypoints usando SIFT
print("===Describing keypoints using SIFT===")
image_gray = cv.cvtColor(image, cv.COLOR_BGR2GRAY)

descriptors = sift.compute(image_gray, keypoints)
print("keypoint dimension: ", len(descriptors[1][0]))
print("descriptor of first keypoint", descriptors[1][0])

print("output:")
plt.imshow(output)
plt.show()

print("number of keypoints is", len(keypoints))
```

input:



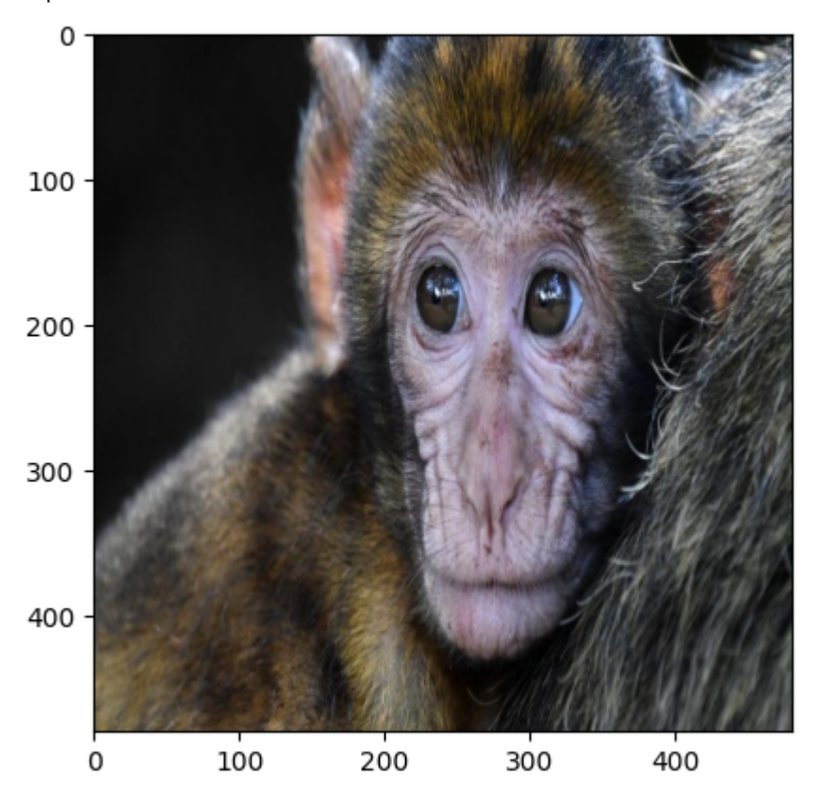
```
===Describing keypoints using SIFT===
keypoint dimension:  128
descriptor of first keypoint [  0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.
   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.
   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.  24.   0.
   0.   0.   0.   0.   0.   7.  79.   2.   2.   1.   2.   1.   1.  23.
  19.   1.   1.   2.  33.   3.   0.   1.   0.   0.   0.   0.   0.   0.
   0.   0. 132.  13.   5.   0.   0.   0.   4.  59. 147.  78.  50.  23.
  62.  20.  25. 147. 147.  14.   7.  64. 147.  36.  29.  42.   0.   0.
   0.   0.   0.   0.   0.   0.  31.   9.   4.   0.   0.   1.  14.  19.
 147.  55.  50.  18.  22.  16.  86. 144. 140.  35.  46.  82. 147.  57.
  70.  73.]
output:
```



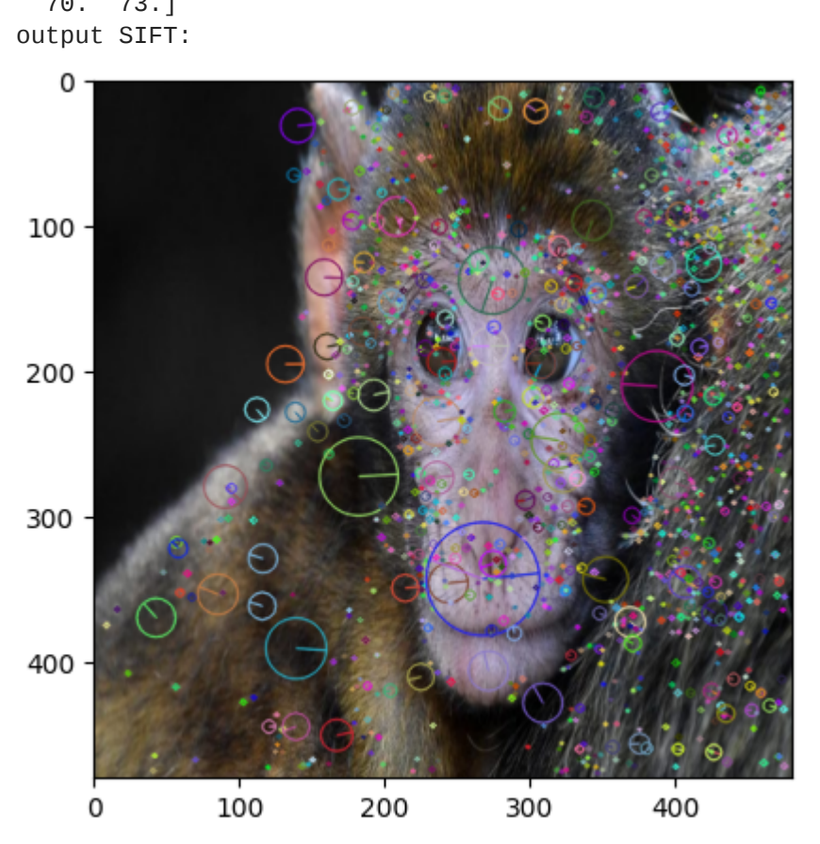
```
number of keypoints is 1024
```

In [18]:
```python
import cv2 as cv
import matplotlib.pyplot as plt
import random
import numpy as np

#Amostragem por detecção de keypoints usando um método pronto
sift = cv.xfeatures2d.SIFT_create()
orb = cv.ORB_create()

image = cv.imread("image.jpg")
image = cv.resize(image, (480, 480))

height, width, channels = image.shape

print("input:")
plt.imshow(cv.cvtColor(image, cv.COLOR_BGR2RGB))
plt.show()


image_gray = cv.cvtColor(image, cv.COLOR_BGR2GRAY)

#Usando SIFT
keypoints_sift = sift.detect(image_gray, None)

if len(keypoints_sift) > 0:
    image_rgb = cv.cvtColor(image, cv.COLOR_BGR2RGB)
    output = cv.drawKeypoints(image_rgb, keypoints_sift, np.array([]), flags=cv.DRAW_MATCHES_FLAGS_DRAW_RICH_KEYPOINTS)

    #Descrevendo keypoints usando SIFT
    print("===Describing keypoints using SIFT===")
    image_gray = cv.cvtColor(image, cv.COLOR_BGR2GRAY)

    descriptors = sift.compute(image_gray, keypoints)
    print("keypoint dimension: ", len(descriptors[1][0]))
    print("descriptor of first keypoint", descriptors[1][0])

    print("output SIFT:")
    plt.imshow(output)
    plt.show()
    print("number of keypoints found with SIFT is", len(keypoints_sift))

#Usando ORB
keypoints_orb = orb.detect(image_gray, None)

if len(keypoints_orb) > 0:
    image_rgb = cv.cvtColor(image, cv.COLOR_BGR2RGB)
    output = cv.drawKeypoints(image_rgb, keypoints_orb, np.array([]), flags=cv.DRAW_MATCHES_FLAGS_DRAW_RICH_KEYPOINTS)

    #Descrevendo keypoints usando SIFT
    print("\n ===Describing keypoints using SIFT===")
    image_gray = cv.cvtColor(image, cv.COLOR_BGR2GRAY)

    descriptors = sift.compute(image_gray, keypoints)
    print("keypoint dimension: ", len(descriptors[1][0]))
    print("descriptor of first keypoint", descriptors[1][0])

    print("output ORB:")
    plt.imshow(output)
    plt.show()
    print("number of keypoints found with ORB is", len(keypoints_orb))
```

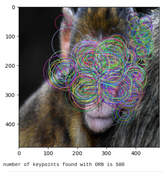
input:



```
===Describing keypoints using SIFT===
keypoint dimension:  128
descriptor of first keypoint [  0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.
   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.
   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.  24.   0.
   0.   0.   0.   0.   0.   7.  79.   2.   2.   1.   2.   1.   1.  23.
  19.   1.   1.   2.  33.   3.   0.   1.   0.   0.   0.   0.   0.   0.
   0.   0. 132.  13.   5.   0.   0.   0.   4.  59. 147.  78.  50.  23.
  62.  20.  25. 147. 147.  14.   7.  64. 147.  36.  29.  42.   0.   0.
   0.   0.   0.   0.   0.   0.  31.   9.   4.   0.   0.   1.  14.  19.
 147.  55.  50.  18.  22.  16.  86. 144. 140.  35.  46.  82. 147.  57.
  70.  73.]
output SIFT:
```



```
number of keypoints found with SIFT is 1695

 ===Describing keypoints using SIFT===
keypoint dimension:  128
descriptor of first keypoint [  0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.
   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.
   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.  24.   0.
   0.   0.   0.   0.   0.   7.  79.   2.   2.   1.   2.   1.   1.  23.
  19.   1.   1.   2.  33.   3.   0.   1.   0.   0.   0.   0.   0.   0.
   0.   0. 132.  13.   5.   0.   0.   0.   4.  59. 147.  78.  50.  23.
  62.  20.  25. 147. 147.  14.   7.  64. 147.  36.  29.  42.   0.   0.
   0.   0.   0.   0.   0.   0.  31.   9.   4.   0.   0.   1.  14.  19.
 147.  55.  50.  18.  22.  16.  86. 144. 140.  35.  46.  82. 147.  57.
  70.  73.]
output ORB:
```



number of keypoints found with ORB is 500

In [ ]: