Transformada Z

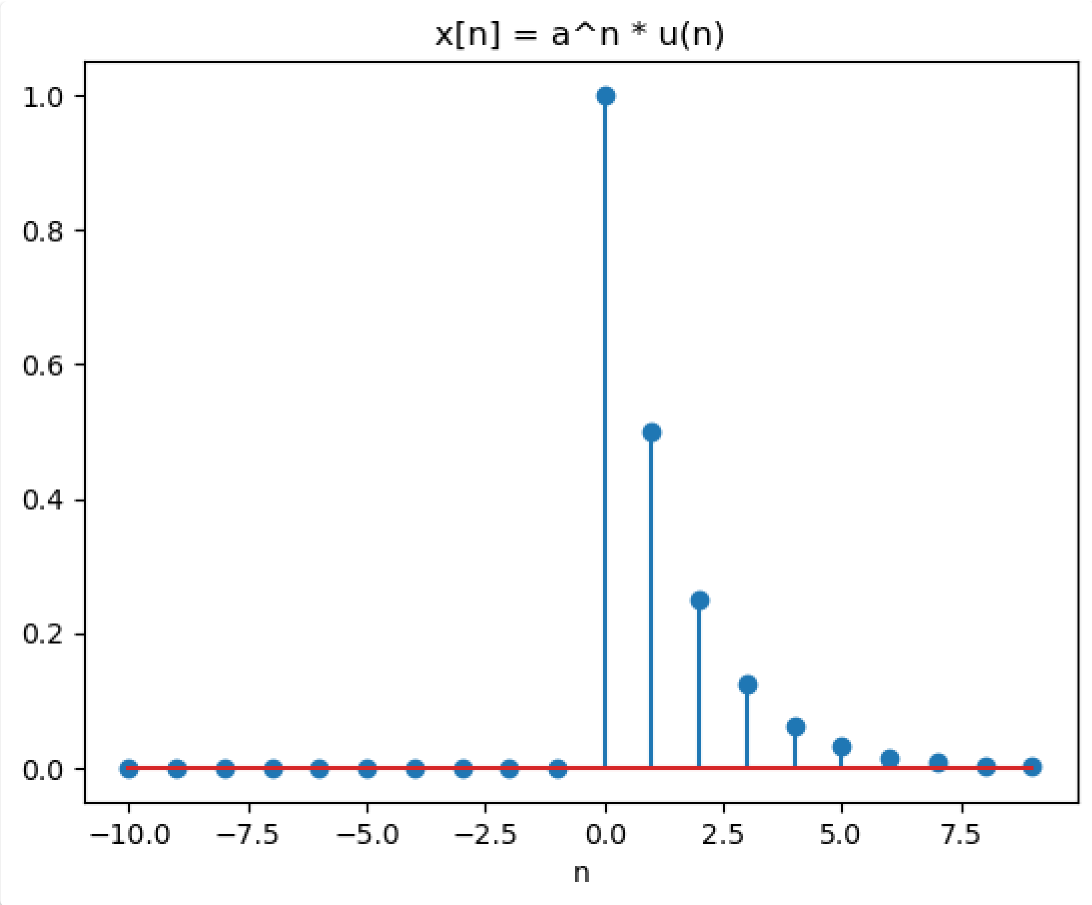
O objetivo principal deste projeto é utilizar a Transformada Z para vizualizar a Transformada de Fourier, pólos e zeros da função $a^n * u(n)$, cuja amplitude deve ser decrescente. Em Python, podemos representar essa função e transformada por:

```
  import numpy as np
import matplotlib.pyplot as plt

def transf_z(a, n):
        return np.array([a**k if k >= 0 else 0 for k in n]) # a^n*u(n)

def transf_fourier(a, w):
        return 1 / (1 - a * np.exp(-1j * w))

a = 0.5 # constante
n = np.arange(-10, 10)
w = np.linspace(-np.pi, np.pi, 1000) # frequencia para a Transf. Fourier

z = transf_z(a, n)
f = transf_fourier(a, w)
```

Este código utiliza a biblioteca Numpy e Matplotlib para calcular e visualizar a transformada de Fourier e os pólos e zeros de uma função. A função `transf_z` é responsável por calcular a transformada Z de uma função dada uma constante `a` e um índice `n`. A transformada de Fourier é calculada pela função `transf_fourier`, que usa a constante `a` e uma frequência `w`. Neste caso, temos que:

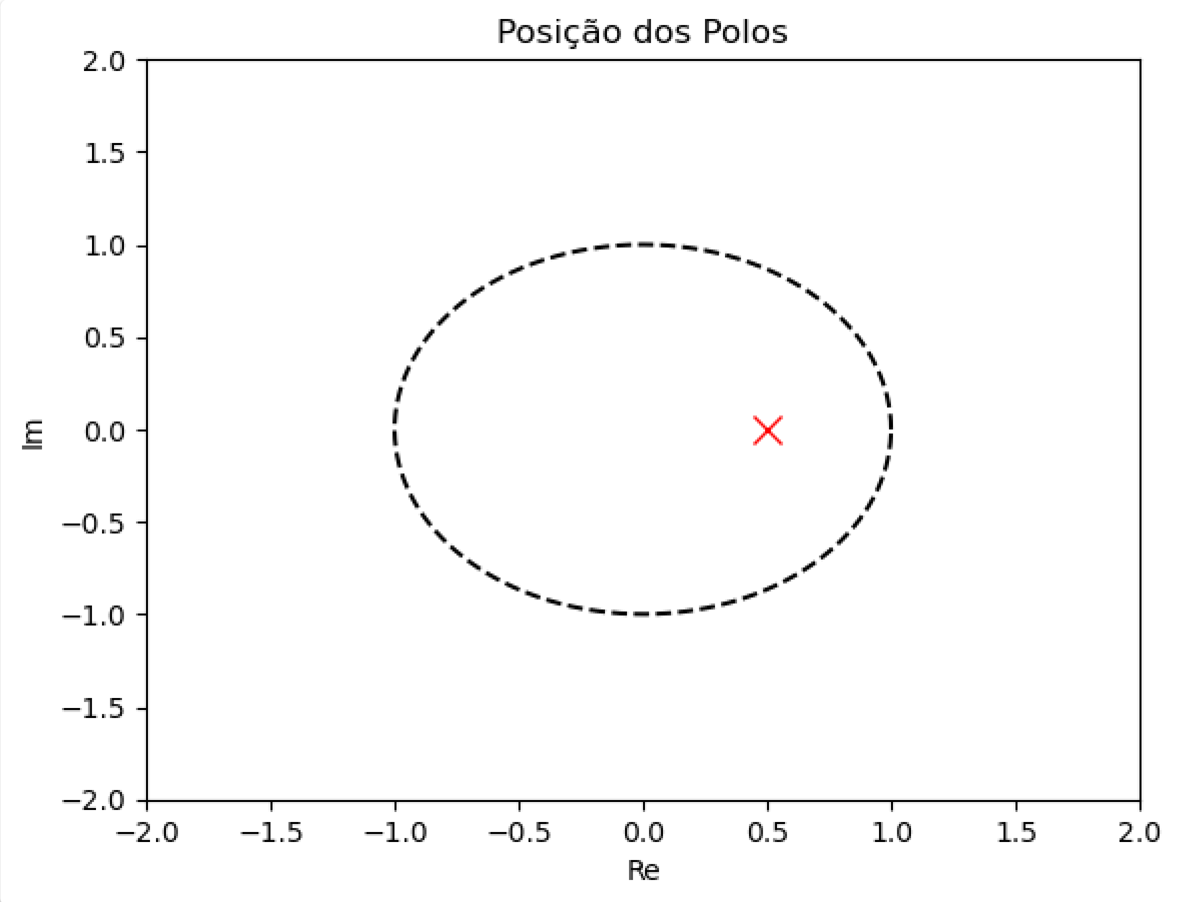
$$a^n * u(n) \to \frac{1}{1 - az^{-1}},\ com|z| > |a| \tag{1}$$

```
  # Plotar a funcao
plt.stem(n, z)
plt.title("x[n] = a^n * u(n)")
plt.xlabel("n")
plt.ylabel("a^n * u(n)")
plt.show()
```



Em seguida, o gráfico da função é plotado, mostrando a variação da função com relação ao índice `n`. Utilizando a Equação (1) podemos atribuir o valor para o pólo `a` e com isso definir comportamento de x(n). Para $0 < a < 1$, a amplitude do sinal $x(n) = a^n * u(n)$ decresce a medida que n cresce. Podemos plotar os pólos da seguinte forma:

```
  # Plotar os pólos
Circ_Unit = np.exp(1j * np.linspace(0, 2*np.pi, 1000)) plt.plot(Circ_Unit.real,
Circ_Unit.imag, 'k--')
plt.plot(a, 0, 'x', color='red', markersize=10)
plt.title("Posição dos Polos")

plt.xlabel("Re")
plt.ylabel("Im")
plt.xlim(-2, 2)
plt.ylim(-2, 2)
plt.show()
```



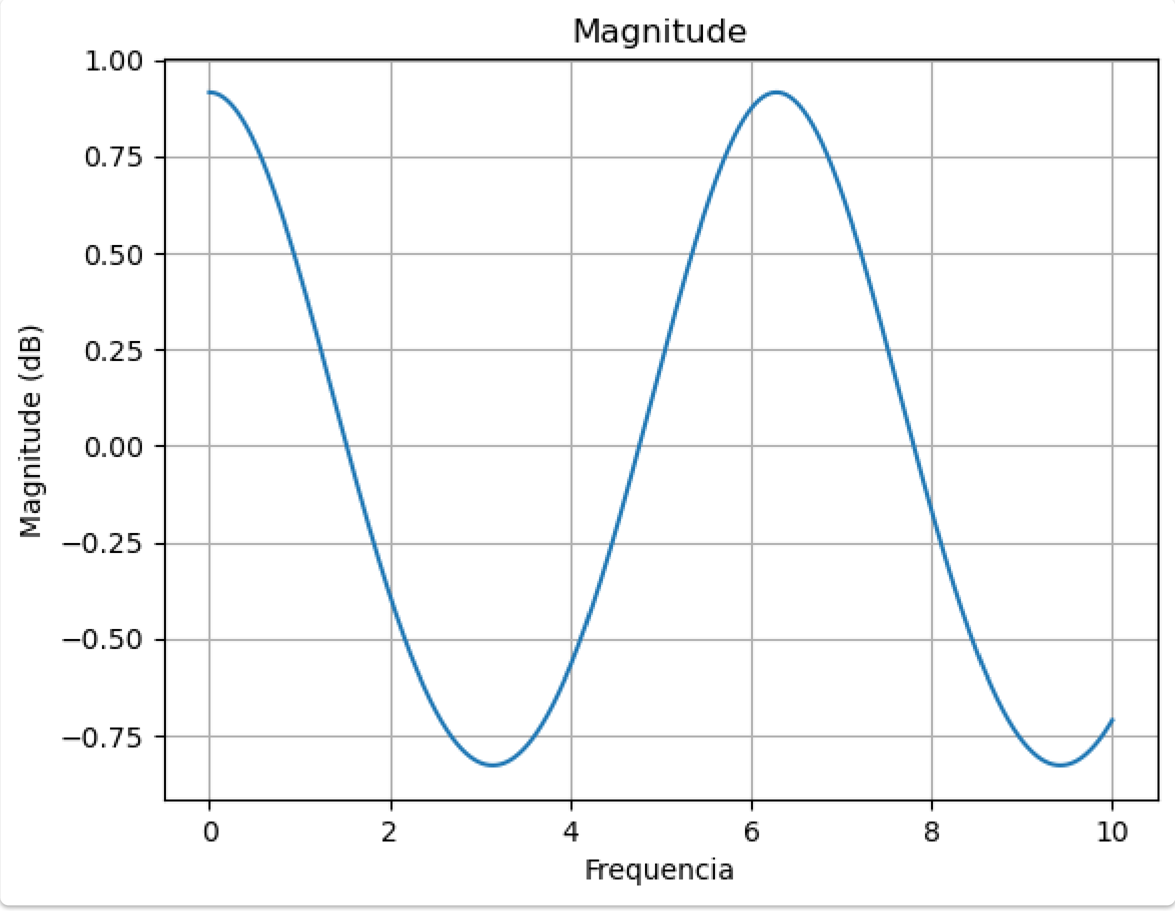
Por fim, para o valor de `a` escolhido a função $X(z)$ fica determinada, e com podemos usá-la para obter a Transformada de Fourier, fazendo $X(z)|_{z=e^{jw}}$. Por exemplo, para $a = 0,5$ temos:

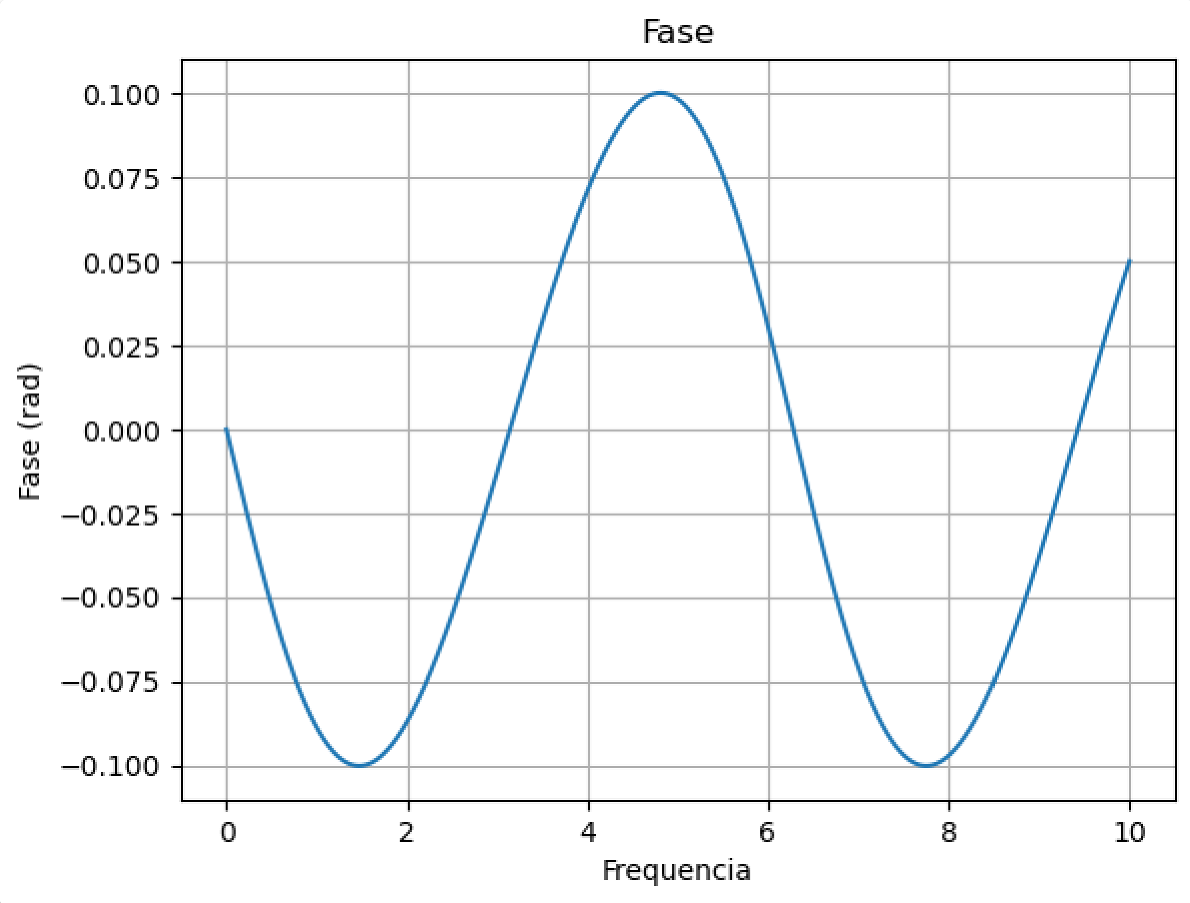
$$X(e^{jw}) = X(z)|_{z=e^{jw}} = \frac{1}{1 - 0.1e^{-jw}} \tag{2}$$

e variando o valor w podemos desenhar $X(e^{jw})$ e $\angle H(e^{jw})$

```
  # Definindo o tamanho da frequencia
w = np.linspace(0, 10, 1000)

# Definindo a função de transf.
H = transf_fourier(0.1, w)

# Plotando a magnitude
plt.figure()
plt.plot(w, 20*np.log10(abs(H)))
plt.xlabel('Frequencia')
plt.ylabel('Magnitude (dB)')
plt.title('Magnitude')
plt.grid()
```

```
# Plotando a fase
plt.figure()
plt.plot(w, np.angle(H))
plt.xlabel('Frequencia')
plt.ylabel('Fase (rad)')
plt.title('Fase')
plt.grid()

plt.show()
```





Assim, a Transformada de Fourier é calculada e plotada, mostrando a magnitude da transformada para diferentes frequências. Isso permite visualizar a distribuição de frequências na seqüência e compreender como a amplitude da seqüência se relaciona com diferentes frequências.