Rodrigo Fabrício Meneses - 376176

Código Fonte

In [19]:
```
%matplotlib inline
import random
import matplotlib.pyplot as plt
plt.style.use('seaborn-whitegrid')
```

In [2]:
```
# Código para gerar números aleatórios entre 0 e 1 usando mcl
def generator_mcl(a, b, m, seed):
    random_number = seed
    while True:
        random_number = (a * random_number + b) % m
        yield random_number / m
```

In [15]:
```
def pi_aproximator(seed=123456, n_pontos=100000, prm={'a':7**5, 'b':0, 'm':2**31}, log=False, draw_pi=False):

    if log:
        print('Parâmetros Utilizados')
        print(f'Número de pontos: {n_pontos}')
        print(f'mcl: {prm}')
        print(f'seed: {seed}')
    # Meu gerador
    rd_gen = generator_mcl(prm['a'], prm['b'], prm['m'], seed)

    # Contador de pontos que caíram dentro do círculo, inicialmente é 0
    contador = 0

    # Lançar n pontos aleatoriamente entre 0 e 1
    for _ in range(n_pontos):
        x = next(rd_gen)
        y = next(rd_gen)
        # Se o ponto gerado está dentro do círculo de raio 1
        if x**2 + y**2 < 1:
            # Se estiver, então incrementa o contador
            contador += 1
            # Se o ponto pertencer ao semi circulo, será desenhado um x vermelho
            if draw_pi: plt.plot(x, y, 'x', color='red')
            # Caso contrário, será desenhado um o azul
        elif draw_pi: plt.plot(x, y, 'o', color='blue')

    # 4 vezes pois os números gerados são apenas entre 0 e 1, então só abrangem 1 quadrante
    valor_pi = 4 * contador / n_pontos

    print('O valor de PI é aproximadamente: ', valor_pi)

    return valor_pi
```

Experimentos

Experimento com diferentes número de pontos

É fácil perceber o aumento da precisão a medida que o número de pontos aumenta

Todos experimento, com os mesmos parâmetros utilizados descritos a seguir

Os parâmetros utilizados passaram no teste de frequência e de execução, como descritos na atividade anterior

In [16]:
```
# Teste com 100 pontos e descrição dos parâmetros
pi_aproximator(n_pontos=100, log=True)

# Todos os outro seguem os mesmos parâmetros
for i in range(3, 8):
    print('\nNúmero de pontos: ', 10**i)
    pi_aproximator(n_pontos=10**i)
```

```
Parâmetros Utilizados
Número de pontos: 100
mcl: {'a': 16807, 'b': 0, 'm': 2147483648}
seed: 123456
O valor de PI é aproximadamente:  3.2

Número de pontos:  1000
O valor de PI é aproximadamente:  3.14

Número de pontos:  10000
O valor de PI é aproximadamente:  3.15

Número de pontos:  100000
O valor de PI é aproximadamente:  3.14508

Número de pontos:  1000000
O valor de PI é aproximadamente:  3.14374

Número de pontos:  10000000
O valor de PI é aproximadamente:  3.1414876
```

Experimentos com diferentes sementes

Experimentos feitos com 1000000 de pontos e variadas sementes

In [39]:
```
for i in range(10):
    seed = random.randint(1, 10000)
    print('\nSemente utilizada: ', seed)
    pi_aproximator(n_pontos=1000000, seed=seed)
```

```
Semente utilizada:  6534
O valor de PI é aproximadamente:  3.141976

Semente utilizada:  2902
O valor de PI é aproximadamente:  3.14154

Semente utilizada:  385
O valor de PI é aproximadamente:  3.14286

Semente utilizada:  7675
O valor de PI é aproximadamente:  3.139864

Semente utilizada:  1106
O valor de PI é aproximadamente:  3.139012

Semente utilizada:  7936
O valor de PI é aproximadamente:  3.141464

Semente utilizada:  8756
O valor de PI é aproximadamente:  3.143064

Semente utilizada:  7159
O valor de PI é aproximadamente:  3.13936

Semente utilizada:  4948
O valor de PI é aproximadamente:  3.14384

Semente utilizada:  4253
O valor de PI é aproximadamente:  3.14194
```

Visualização

Visualização para uma amostra de 1000 pontos, pois assim não ficará muito poluído

In [41]:
```
pi_aproximator(n_pontos=1000, draw_pi=True)
```

```
O valor de PI é aproximadamente:  3.14
```

Out[41]: 3.14