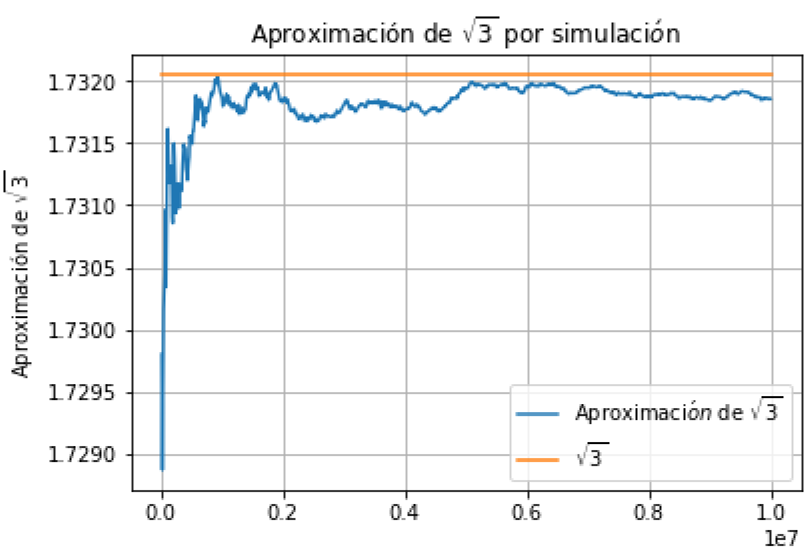
Estimar por simulacion $\sqrt{3}$ Cuantos ensayor tiene que realizar para asegurar que la probabilidad de tener un error mayor o igual que 0.01 sea menor que 0.1?

Sabemos que para generar un numero aleatorio entre 1 y 2 tenemos 1+random (0,1), lo que nos da un conjunto de numeros aleatorios uniformemente distribuidos entre 1 y 2, ademas se conoce que: $x^2 = n$ con $x = \sqrt{n}$ entonces x debe de ser aproximado a n que es la raiz del numero buscado, por la distribucion uniforma sabemos que los valores proximos a n deben de ser n-1 +random, esto es: x<3-1 x<2, por lo cual los numeros aleatorios deben de estar entre 1 y 2, en programa tenemos

In [1]:
```
from random import random
import matplotlib.pyplot as plt
from math import sqrt
```

In [2]:
```
N=10000000
n=0
lx=[]
ly=[]
lsqrt=[]
for i in range(N):
    x=1+random()
    if x**2<3:
        n=n+1
        if i%10000==0 and i>0:
            lx.append(i)
            ly.append(n/i+1)
            lsqrt.append(sqrt(3))
plt.plot(lx,ly)
plt.plot(lx,lsqrt)
plt.xlabel("")
plt.ylabel(r"Aproximación de $\sqrt{3}$")
plt.grid()
plt.legend([r"Aproximaci$\'on$ de $\sqrt{3}$",r"$\sqrt{3}$"])
plt.title(r"Aproximación de $\sqrt{3}$ por simulaci$\'o$n")
plt.show()
print(n/N+1)
```




```
1.7318468999999999
```

ahora bien, tenemos que $x_i = 1$ si cae dentro de $[1, \sqrt{3}]$ y 0 en caso contrario. como son experimentos independientes tenemos.

$\hat{Z}_m = \frac{x_1+x_2+x_3+...+x_n}{N} + 1$

$E(\hat{Z}_m) = \frac{1}{N}Np + 1$

$E(\hat{Z}_m) = \sqrt{3} - 1 + 1 = \sqrt{3}$

$V(\hat{Z}_m) = \frac{1}{N^2}(N(p(1-p))$

$V(\hat{Z}_m) = \frac{p(1-p)}{N^2}$

P(|$\hat{Z}_m$-$\sqrt{3}$| $\geq$ 0.1)$\leq$ $\frac{v(x)}{N(0.1)^2} < 0.01$
$\frac{p(1-p)}{N(0.1)^2} < 0.01$

$\frac{1}{(0.01)(0.1)^2} < N$

$N = 10,000$

2-. Estime por simulacion el volumen de una esfera de radio 1
Sabemos que el volumen de una esfera es $V = \frac{4}{3}\left(\pi r^3\right)$
Por lo cual, tenemos a traves del calculo de pi, lo siguiente:

In [3]:
```
n=0
v=0
for i in range(N):
    w=random()
    y=random()
    if w**2+y**2<1:
        n+=1
print("El volumen de la esfera, calculado por simulación es:\tV={:}".format(4/3*(4*n/N)*
```

```
El volumen de la esfera, calculado por simulación es:   V=4.188478933333333
```

3-. Se define a la variable aleatoria N como:

$$\text{N=Minimo}\{n : U_1 + U_2 + U_3 + ... + U_n > 1\}$$

Donde los $U_1, U_2, U_3, ..., U_n$ son uniformemente distribuidos en $[0, 1]$
Estime por simulación el valor esperado de N.

Dé un intervalo de confianza del 90% para E(N)

Solucion

In [7]:
```
s=0
contador=0
ss=0
va=0
for i in range (N):
    ss+=contador
    s=0
    contador=0
```

```
    while s<1:
        s=s+random()
        contador=contador+1
        va=va+(s-ss/N)**2
varianza=sqrt(1/(N-1)*va)
print("N={:}".format(ss/N))
print("El intervalo de confianza del 90% es:\n[{:}-{:.3e},{:}+{:.3e}]\t===>[{:},{:}]".fo
    ss/N,1.645*varianza/N,ss/N,1.645*varianza/N,ss/N-1.645*varianza/N,ss/N+1.645*varianz
```

```
N=2.7183391
El intervalo de confianza del 90% es:
[2.7183391-2.805e-07,2.7183391+2.805e-07]       ===>[2.7183388194934968,2.71833938050650
35]
```

4-.Estime por simulacion el valor de las siguientes integrales:

1. $\int_0^1 \sqrt{1+x^3}dx$
2. $\int_3^5 e^{-x^2}dx$
3. $\int_{0.5}^1 \frac{sin(x)}{x}dx$
4. $\int_0^\infty e^{-x^2}dx$
5. $\int_{-\infty}^\infty e^{-x^2}dx$
6. $\int_2^\infty e^{-x^2}dx$

In [8]:
```
from random import random
from math import sqrt, exp,sin
integral1=0
integral2=0
integral3=0
integral4=0
integral5=0
integral6=0
N=1000000
for i in range(N):
    x=random()
    integral1=integral1+sqrt(1+x**3)
    integral2=integral2+(5-3)*exp(-(3+(5-3)*x)**2)
    integral3=integral3+(1-0.5)*sin((0.5+(1-0.5)*x))/(0.5+(1-0.5)*x)
    integral4=integral4+exp(-(1/x-1)**2)/x**2
    integral5=integral5+2*exp(-(1/x-1)**2)*1/x**2
    integral6=integral6+(exp(-(1/x-1)**2)/x**2-2*exp(-(2*x)**2))
print("El valor de 1 es: {:}\nEl valor de 2 es: {:}\nEl valor de 3 es: {:}".format(integ
print("El valor de 4 es: {:}\nEl valor de 5 es: {:}\nEl valor de 6 es: {:}\n".format(int
                                                                                    (in
```

```
El valor de 1 es: 1.1117362483178834
El valor de 2 es: 1.9545040628626623e-05
El valor de 3 es: 0.45294354243922613
El valor de 4 es: 0.886444636570451
El valor de 5 es: 1.772889273140902
El valor de 6 es: 0.005252396078114682
```

5-. X sigue una distribucion geometrica con parametro $p = 0.2$. Estime por simulacion el valor esperado de $\sqrt{1+x}$. De un intervalo de confianza del 95% para la estimacion de E($\sqrt{1+x}$)

Primero debemos de generar la variable aleatoria, para despues estimar su valor esperado

In [9]:
```
from math import log
import numpy as np
def geometrica(p):
```

```
    u=random()
    return int(log(1-u)/log(1-p))+1
```

In [10]:
```
s=0
s2=0
var=0
N=100000
n=[]
var=0
varr=[]
for i in range(N):
    x=1+geometrica(0.2)
    s=s+sqrt(1+x)
    varr.append(x)
for i in varr:
    var=var+(i-s/N)**2
#print("[{:}-1.65*{:},{:}+1.65*{:}]".format(s/N,varianza/(N*sqrt(N)),s/N,varianza/(N*sqr
#print(s/N-1.65*varianza/(N*sqrt(N)))
print("El intervalo de confianza es: [{:}-1.65*{:},{:}+1.65*{:}]".format(s/N,sqrt(var*1/
print("El intervalo de confianza es: [{:},{:}]".format(s/N-1.65*sqrt(var*1/(N-1))/(N),s/
```

```
El intervalo de confianza es: [2.5388932361787804-1.65*5.656904683435026e-05,2.538893236
1787804+1.65*5.656904683435026e-05]
El intervalo de confianza es: [2.538799897251504,2.538986575106057]
```

6-.X sigue la siguiente funcion de distribucion de densidad de probabilidad:

|x|1||2|3|4|

|P(X=x)|0.1|0.1|0.3|0.5|

Estime por simulacion el valor esperado de $\frac{1}{1+x^2}$

Solucion

Generamos la variable aleatoria con numeros aleatorios y despue calculamos sus valor esperado

In [11]:
```
def fdp():
    u=random()
    if u<=0.1:
        return 1
    elif u>0.1 and u<0.2:
        return 2
    elif u>0.2 and u<0.5:
        return 3
    else:
        return 4
```

In [12]:
```
N=10**6
s=0
for i in range(N):
    x=fdp()
    s=s+1/(1+x**2)
print(s/N)
```

```
0.12928220588325418
```

7-.La variable aleatoria X sigue la siguiente funcion de densidad de probabilidad:

$f_x(i) = \frac{2}{3^i}, i = 1, 2, 3, 4, \ldots$

Estime por simulacion el valor esperado de X^2

Debemos de crear su funcion de disribucion

Tenemos que es una distribucion geometrica con parametro $p = \frac{2}{3}$

$P_k = P\left(X = k\right) = p(1-p)^{k-1}$

$P_1 + P_2 + \ldots + P_k = p(1-p)^0 + p(1-p)^1 + \ldots + p(1-p)^{k-1}$
$P_1 + P_2 + \ldots + P_k = p\left(1 + (1-p)^1 + \ldots + (1-p)^{k-1}\right)$
$P_1 + P_2 + \ldots + P_k = p\left(\frac{1-(1-p)^k}{1-(1-p)}\right)$
$P_1 + P_2 + \ldots + P_k = 1-(1-p)^k$
Entonces p se debe de contrar entre $1-(1-p)^k$ y entre $1-(1-p)^{k-1}$
$1-(1-p)^k < u < 1-(1-p)^{k-1}$
$-(1-p)^k < u-1 < -(1-p)^{k-1}$
$(1-p)^{k-1} < 1-u < (1-p)^k$
$(k-1)ln(1-p) < ln(1-u) < kln(1-p)$
$k < \frac{ln(1-u)}{ln(1-p)} < (k-1)$
Como se busca el valor de k, entonces tenemos:
$k = Int\left(\frac{ln(1-u)}{ln(1-p)}\right) + 1$ pero como (1-u) sigue siendo igualmente distribuido y aleatorio tenemos:

$$k = Int\left(\frac{ln(1-u)}{ln(1-p)}\right) + 1$$

In [13]:
```
def geometrica2(p):
    u=random()
    return int(log(u)/log(1-p))+1
```

In [14]:
```
s=0
for i in range(N):
    x=geometrica2(2/3)
    s=s+x**2
print(s/N)
```

```
2.998665
```

8-. La variable aleatoria X tiene la siguiente funcion de densidad de probabilidad.
$f_x(X) = \frac{1}{xLn(2)}$ si $x \in [1,2]$ 0 en caso contrario.

1. Estime por simulacion el valor esperado de $\sqrt{1+X^2}$
2. Estime por simulacion la probabilidad de que $1.3

Solución

tenemos la funcion de distribucion como sigue:
$F_x(X) = \int_1^x \frac{1}{tLn(2)}dt$
$F_x(X) = \frac{Ln(t)}{Ln(2)}|_1^x$
$F_x(X) = \frac{Ln(x)}{Ln(2)}$
Por lo cual, tenemos su inversa de la forma:
$y = \frac{Ln(x)}{Ln(2)}$
Cuando x=1, tenemos que y=0, cuando x=2, tenemos que y=1
$Ln(2)y = Ln(x)$
$e^{Ln(2)y} = e^{Ln(x)}$
$2^y = x$
$F_x^{-1}(X) = 2^x \quad x \in [0,1]$

In [15]:
```
s=0
```

```
n=0
for i in range(N):
    u=random()
    x=2**u
    s=s+sqrt(1+x**2)
    if s/N>1.3 and s/N<1.8:
        n+=1
print("El valor esperado es: {:}\nLa probabilidad es: {:}".format(s/N,n/N))
```

```
El valor esperado es: 1.7630188246832388
La probabilidad es: 0.262695
```

9-. Sean $U_1, U_2, \ldots, U_{10}$ variables aleatorias independientes y uniformemente distribuidas en $[0,1]$. Estime pos simulacion el valor esperado de la variable aleatoria M=Maximo{$U_1, U_2, \ldots, U_{10}$}

In [16]:

```
M=0
for i in range(N):
    u1=random()
    u2=random()
    u3=random()
    u4=random()
    u5=random()
    u6=random()
    u7=random()
    u8=random()
    u9=random()
    u10=random()
    M=M+max(u1,u2,u3,u4,u5,u6,u7,u8,u9,u10)
print(M/N)
```

```
0.9090280372314167
```

10-. Sean $X_1, X_2, \ldots, X_5$ variables aleatorias con la siguiente funcion de densidad de probabilidad.
$f_x(X) = 2x$ si $x \in [0,1]$, 0 en caso contrario.
Estime por simulacion el valor esperado de $X_1^1 + X_2^2 + \ldots + X_5^2$

Tenemos la funcion de densidad, por lo cual, usando el metodo de la inversas tenemos:
$F_x(X) = \int_0^x 2tdt = t^2|_0^x$
$F_x(X) = x^2$
$F_x^{-1}(X) = \sqrt{x}$ con $x \in [0,1]$ Finalmente tenemos:
$g(F_x^{-1}(X)) = (\sqrt{x})^2$
$g(F_x^{-1}(X)) = x$

In [17]:

```
s=0
for i in range(N):
    sumpar=sum(random() for i in range(1,6,1))
    s=s+sumpar
print(s/N)
```

```
2.499486305526461
```