

Prueba Wilson Calle

In [1]:

```
import pandas as pd
from collections import Counter
import matplotlib.pyplot as plt
```

Cuadrados Medios

In [2]:

```
x0 = 8952
digitos=4
simulaciones=100
iteraCC=[]
Ni = 1

for i in range(0,simulaciones):
    LXn=[]
    LXc=[]
    Li=[]
    LL=[]
    LUi=[]
    LRn=[]
    XNC=[]
    cont=0
    iteracion=1
    Xci=int(x0) * int(x0)
    Xcsi=str(Xci)
    while(len(Xcsi) < 8):
        Xcsi="0"+Xcsi

    Longitudi=len(Xcsi)
    Uii=Xcsi[int((Longitudi/2)-2):int((Longitudi/2)+2)]
    #print("Semilla X0= ",x0, "Semilla nueva= ", Uii)
    x0=int(Uii)+100
    Xn=x0
    while(iteracion==1):
        #print(iteracion)
        Xc=int(Xn) * int(Xn)
        Xcs=str(Xc)
        while(len(Xcs) < 8):
            Xcs="0"+Xcs

        Longitud=len(Xcs)
        Ui=Xcs[int((Longitud/2)-2):int((Longitud/2)+2)]
        Nn="1"
        while(len(Nn)<(digitos+1)):
            Nn=Nn+"0"
        Rn=int(Ui)/int(Nn)
        #print("Iteración: ",iteracion)
        Li.append(iteracion)
        #print("Xn: ",Xn)
        LXn.append(Xn)
        #print("Xn * Xn: ",Xcs)
        LXc.append(Xcs)
        #print("Longitud: ",Longitud)
        #print("Ui: ",Ui)
        LUi.append(Ui)
        #print(Nn)
        #print("Randomico: ",Rn)
        LRn.append(Rn)
        Xn=Ui
        #print(Xn)
        #print(LXn)
        if Xn in LXn:
            #print("Esta")
            #print(iteracion)
            iteraCC.append(cont)
```

```

        iteracc.append(cont)
        iteracion=0
        cont=cont+1

```

In [3]:

```
print(iteraCC)
```

```

[58, 29, 59, 88, 13, 91, 28, 28, 68, 55, 62, 8, 54, 28, 77, 41, 8, 48, 35, 37, 48, 51, 27, 3, 45,
12, 82, 9, 47, 13, 23, 18, 13, 86, 38, 7, 51, 5, 8, 3, 49, 54, 84, 34, 14, 55, 5, 27, 13, 19, 27,
9, 57, 55, 10, 3, 32, 56, 69, 4, 55, 36, 67, 6, 28, 70, 86, 31, 71, 18, 36, 53, 8, 9, 26, 15, 23,
47, 30, 58, 30, 77, 59, 5, 11, 6, 4, 5, 1, 1, 5, 1, 1, 5, 1, 1, 5, 1, 1, 5]

```

In [4]:

```

X=list(Counter(iteraCC).keys())
Y=list(Counter(iteraCC).values())

print(X)
print(Y)
fig, gra = plt.subplots(figsize = (18, 6))

gra.bar(X,Y)
plt.title('Historicos de Cuadrados Medios')
plt.xlabel('Iteración')
plt.ylabel('Numero de Repeticiones')

```

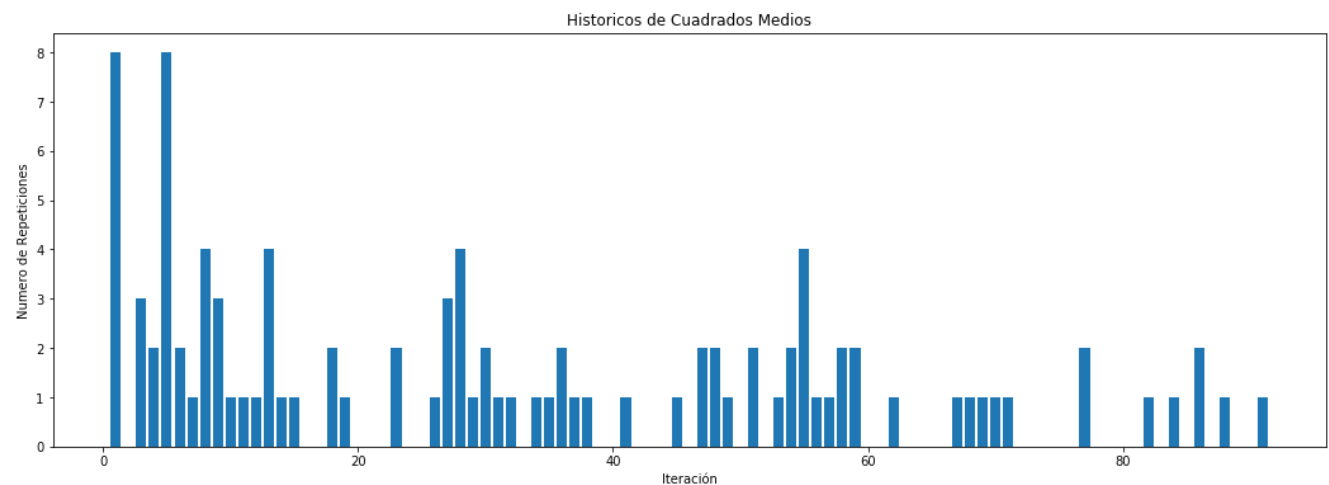
```

[58, 29, 59, 88, 13, 91, 28, 68, 55, 62, 8, 54, 77, 41, 48, 35, 37, 51, 27, 3, 45, 12, 82, 9, 47,
23, 18, 86, 38, 7, 5, 49, 84, 34, 14, 19, 57, 10, 32, 56, 69, 4, 36, 67, 6, 70, 31, 71, 53, 26, 1
5, 30, 11, 1]
[2, 1, 2, 1, 4, 1, 4, 1, 4, 1, 4, 2, 2, 1, 2, 1, 1, 2, 3, 3, 1, 1, 1, 3, 2, 2, 2, 2, 1, 1, 8, 1, 1,
1, 1, 1, 1, 1, 1, 1, 2, 2, 1, 2, 1, 1, 1, 1, 1, 1, 2, 1, 8]

```

Out[4]:

```
Text(0, 0.5, 'Numero de Repeticiones')
```



Congruencia Lineal

In [5]:

```

a=3
b=5
m=19
x0=1520

conteo=[]
i=0
Xn=0.0
Un=0.0

Iter=1

```

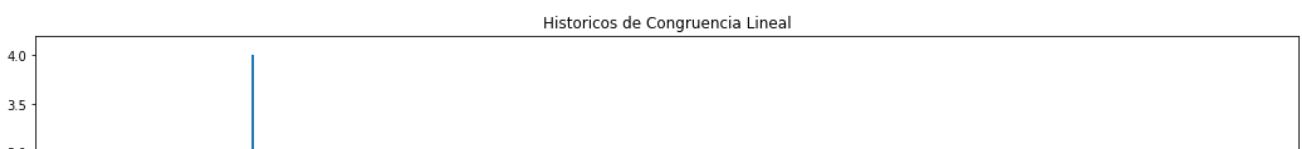
```
#print (conteo)
```

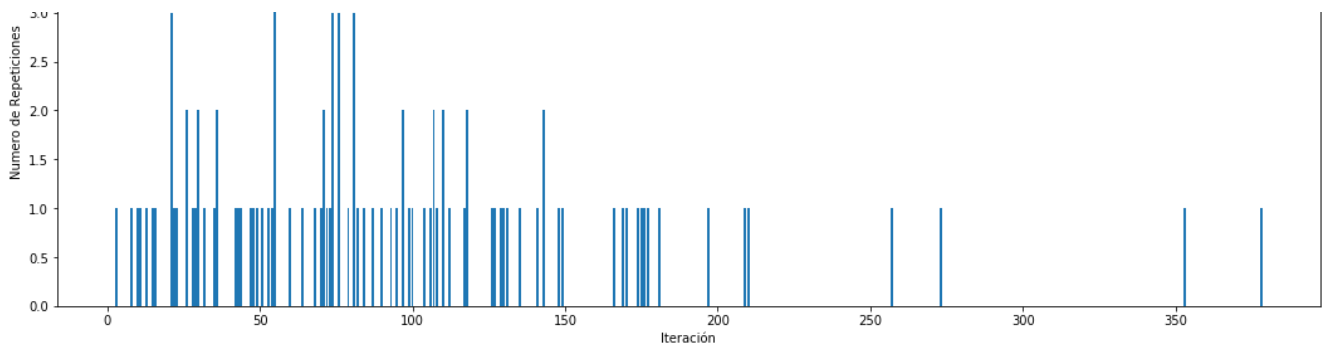
```
print (conteo)
```

In [7]:

[10, 11, 3, 13, 21, 23, 8, 30, 51, 16, 26, 43, 15, 81, 22, 35, 36, 97, 55, 32, 29, 70, 28, 129, 5
4, 47, 53, 90, 143, 44, 49, 42, 177, 110, 60, 73, 74, 71, 48, 209, 130, 127, 64, 76, 72, 100,
257, 68, 273, 170, 95, 141, 82, 107, 84, 117, 174, 104, 93, 126, 99, 79, 210, 148, 353, 118, 131,
87, 181, 378, 108, 149, 166, 135, 112, 197, 106, 175, 176, 169]
[1, 1, 1, 1, 3, 1, 1, 2, 1, 1, 2, 1, 1, 3, 1, 1, 2, 2, 4, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1,
2, 1, 1, 3, 2, 1, 1, 1, 1, 1, 3, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]

```
Text(0, 0.5, 'Numero de Repeticiones')
```





conclusiones

Dentro de el motodo de Cuadrados Medios, al tomar una semilla partiendo de un valor dado, si generamos el método desde aquí vamos a notar un decrecimiento en cuanto a los numeros repetidos dada su posición y solo repeticion en un valor por muy alto, de lo contrario si hacemos lo mismo en el método de Congruencia Lineal notaremos un crecimiento pero los numeros repetidos serán iguales.

opiniones

Los números aleatorios existirán un punto en el que estos se repetirán pero, mientras más grande sea este existirá una distancia de repetición mas prolongado con respecto a un número más pequeño. Es por ello que a opinion personal el método de congruencia lineal es el más efectivo para tener numeros aleatorios ya que existe una repetición muy baja tendiendo a 1.

recomendaciones

Para que exista más aleatoriedad en cuanto a los números recomiendo sumar si es en el método de cuadrados medios, un valor cualquiera para que sea una semilla totalmente distinta y en el método de congruencia lineal a las variables de a, b y m se le suman numeros primos para tener asi mas notación en cuanto a la aleatoriedad nos referimos.