

Deber Seminario
Fausto Fabian Crespo Fernandez
Atractor de Mapa de Henon

El mapa de Henon clásico se define como:

$$f(x, y) = (1 - ax^2 + by, bx)$$

(https://en.wikipedia.org/wiki/H%C3%A9non_map).

Sin embargo la versión del mapa de Henon estudiada en clase y que aparece en la página 51 del libro “An Introduction to dynamical systems” Alligood, Yorke, Sauer (Springer 2000) fue:

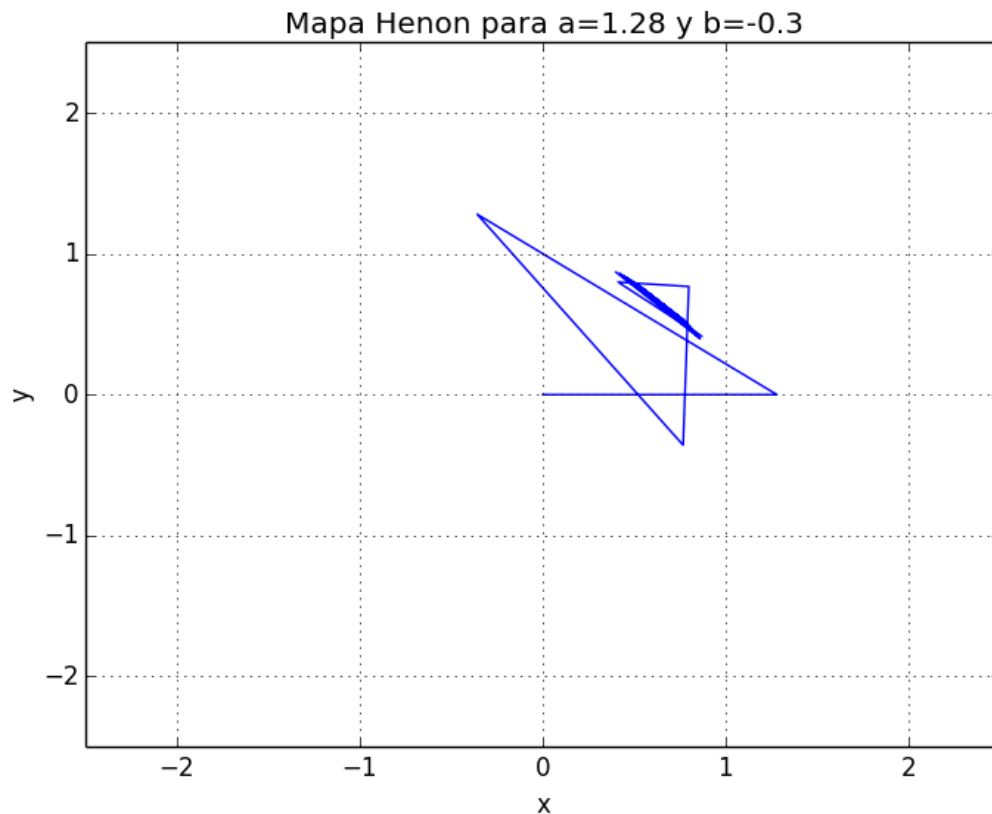
$$f(x, y) = (a - x^2 + by, x)$$

Para

$a=1.28$

$b=-0.3$

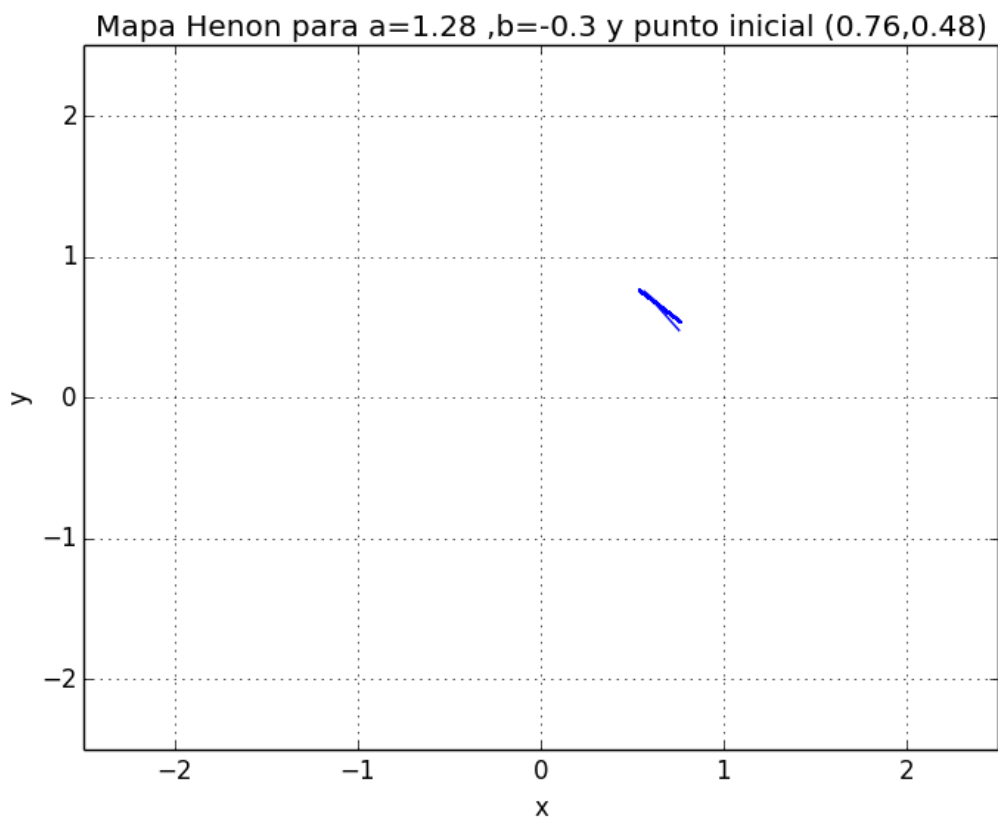
Escogimos como punto inicial al (0,0) y al iterar 1000 veces obtenemos el grafico siguiente:

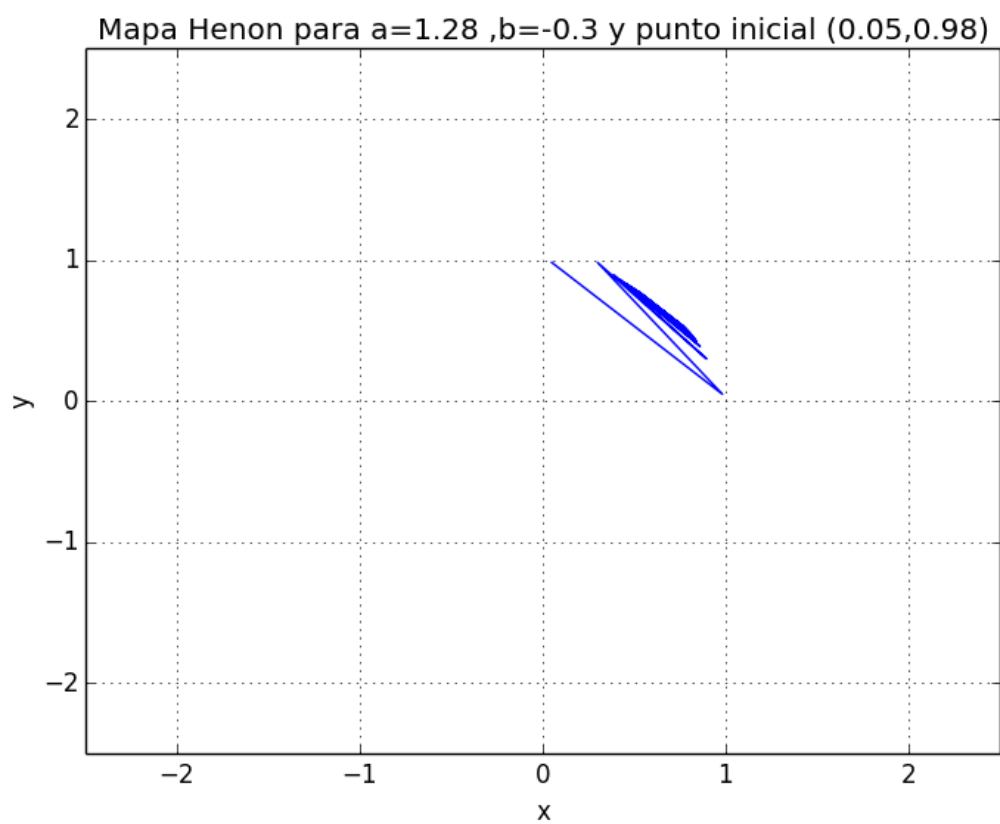


Donde se observa la presencia de 2 puntos de periodo 2 ubicados en $(0.538, 0.76)$

Y $(0.76, 0.538)$ respectivamente.

Si para los mismos valores de a y b , escogemos un punto aleatorio entre 0 y 1 para la x y la y , obtenemos los mismos puntos periódicos de periodo 2:



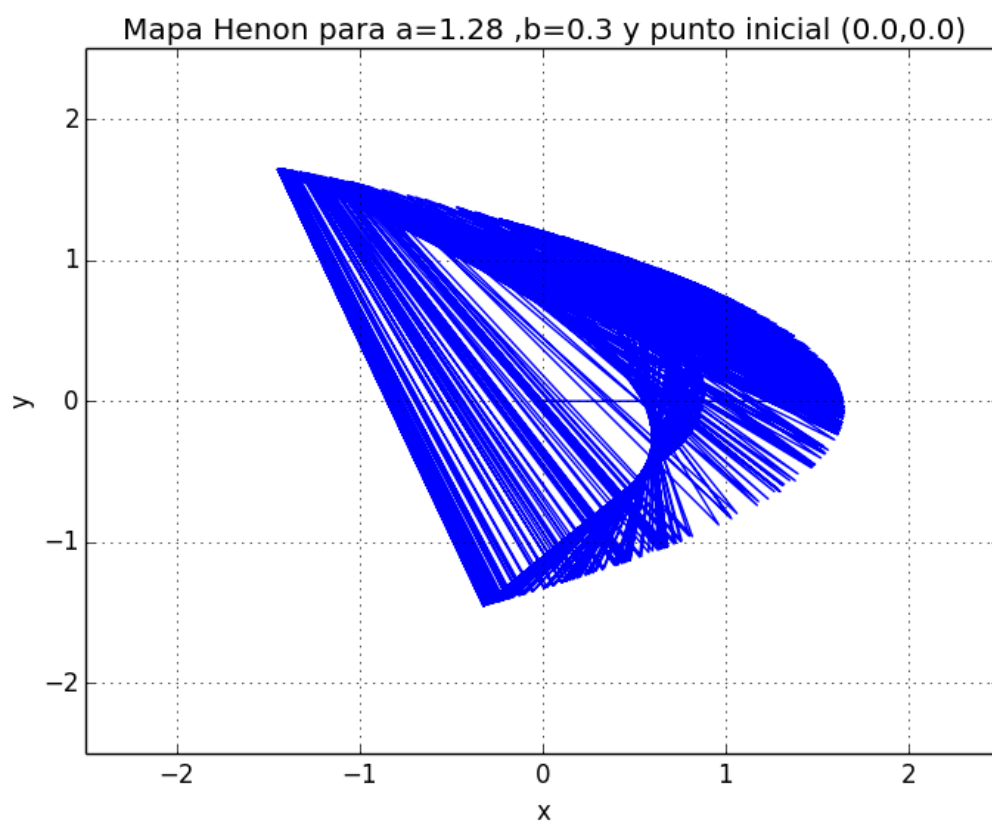


Ahora si hacemos:

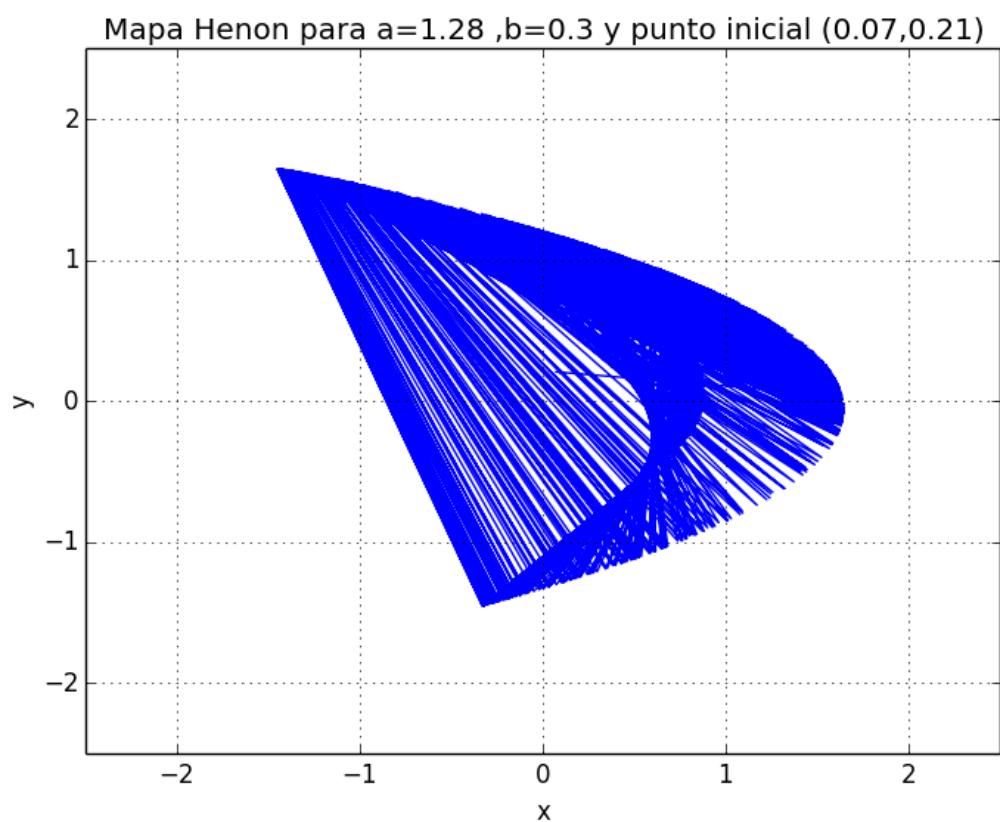
$a=1.28$

$b=0.3$

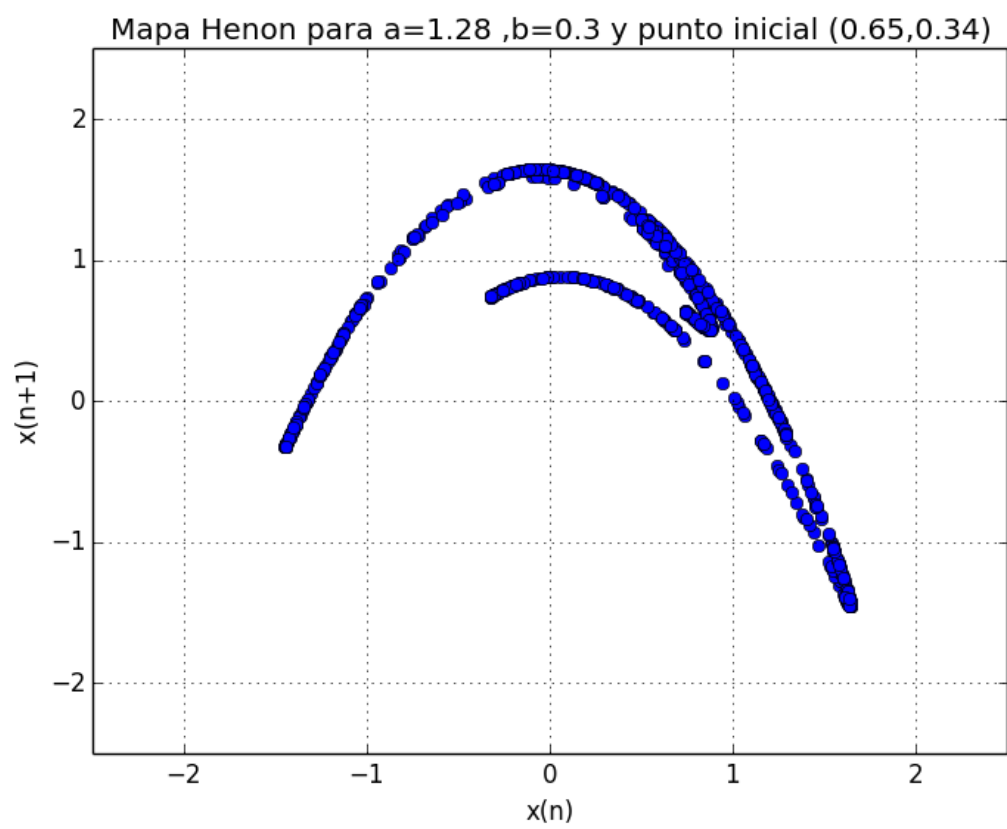
y comenzamos las iteraciones con el punto $(0,0)$, obtenemos:



Y para un punto (x,y) aleatorio entre 0 y 1:



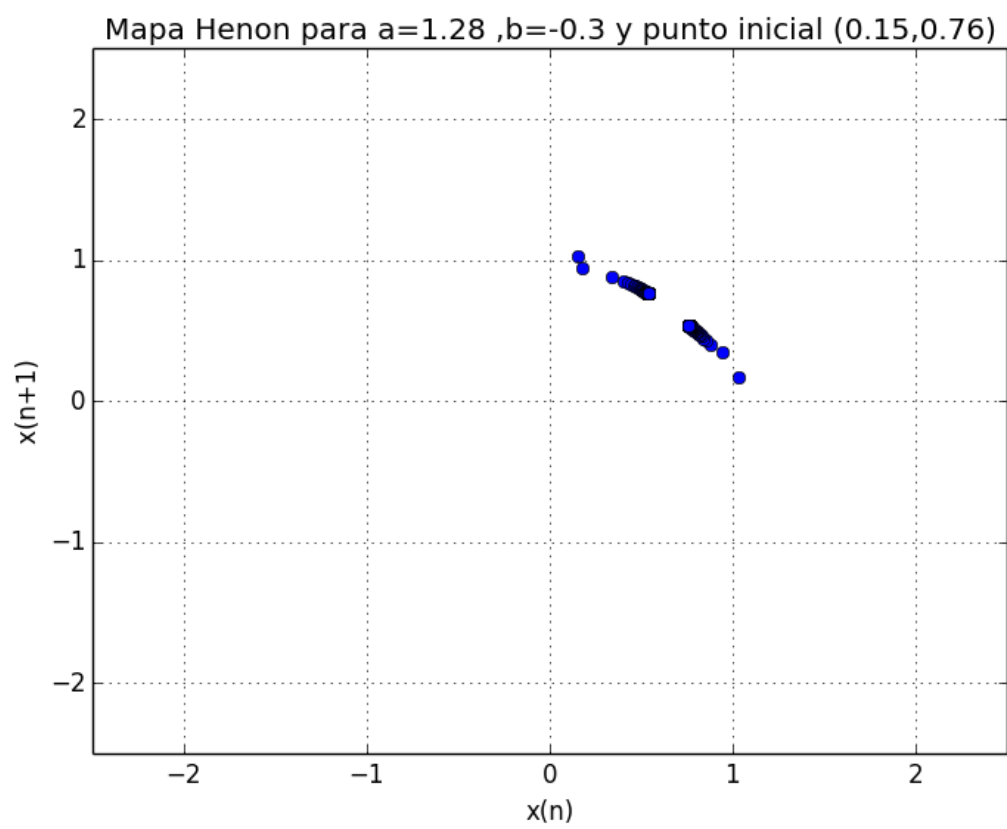
Ahora si queremos visualizar el atractor del mapa de Henon como se hace en al artículo <http://chaosbook.org/projects/Wen14.pdf> podemos graficar x_{n+1} en función de x_n :



Y para

$a=1.28$

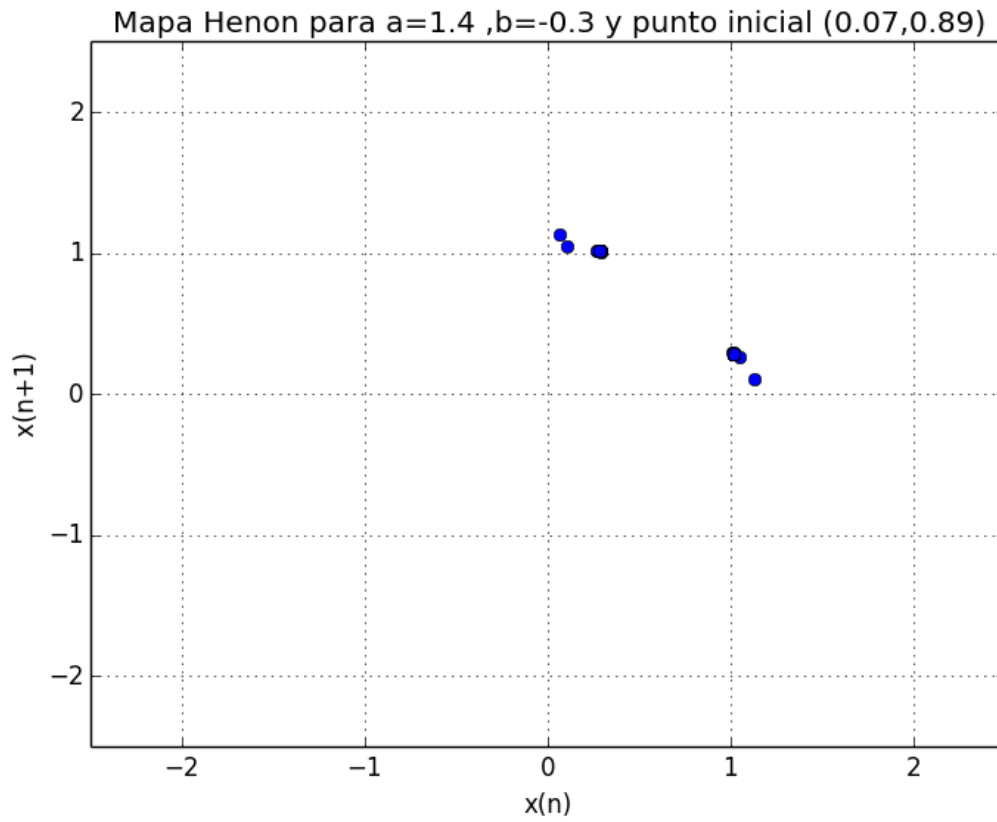
$b=-0.3$



Y para

$a=1.4$

$b=-0.3$



Código en Python (2 archivos):

Archivo 1:

```
import math
```

```
import random
```

```
import matplotlib.pyplot as plt
```

```
a=1.28
```

```
b=0.3
```

```
def RoundTokDecimals(n,k):
```

```
temp = int(n * math.pow(10,k) + 5*(1/math.pow(10,k-1))) / float(math.pow(10,k))
```

```
return temp
```

```
def HenonMap(x,y):
```

```
return [a-x*x+b*y, x]
```

```
x0=random.random()
```

```
y0=random.random()
```

```
print(x0)
```

```
print(y0)
```

```
yarray=[]
```



```

xarray=[]
xarray.append(x0)
yarray.append(y0)
tempx=x0
tempy=y0
for i in range(1000):
    [tempx, tempy]=HenonMap(tempx, tempy)
    xarray.append(tempx)
    yarray.append(tempy)

plt.plot(xarray,yarray, 'b-')
plt.axis([-2.5, 2.5, -2.5, 2.5])
plt.xlabel('x')
plt.ylabel('y ')
plt.title('Mapa Henon para a='+str(a)+" ,b="+str(b)+ " y punto inicial (" +
str(RoundTokDecimals(x0,2))+", "+ str(RoundTokDecimals(y0,2))+")")

plt.grid(True)
plt.show()

```

Archivo 2:

```
import math
```

```
import random
```

```
import matplotlib.pyplot as plt
```

```
a=1.28
```

```
b=-0.3
```

```
def RoundTokDecimals(n,k):
```

```
temp = int(n * math.pow(10,k) + 5*(1/math.pow(10,k-1))) / float(math.pow(10,k))
```

```
return temp
```

```
def HenonMap(x,y):
```

```
return [a-x*x+b*y, x]
```

```
x0=random.random()
```

```
y0=random.random()
```

```
print(x0)
```

```
print(y0)
```

```
yarray=[]
```

```
xarray=[]
```

```
#xarray.append(x0)
```

```
#yarray.append(y0)
```

```
tempx=x0
```

```
tempy=y0
```

```

xarrayAnt=[]
for i in range(1000):
    xarrayAnt.append(tempx)
    [tempx, tempy]=HenonMap(tempx, tempy)
    xarray.append(tempx)
    yarray.append(tempy)
for i in range(len(xarrayAnt)):
    plt.plot(xarrayAnt[i],xarray[i], 'bo')
plt.axis([-2.5, 2.5, -2.5, 2.5])
plt.xlabel('x(n)')
plt.ylabel('x(n+1) ')
plt.title('Mapa Henon para a='+str(a)+" ,b="+str(b)+ " y punto inicial (" +
str(RoundTokDecimals(x0,2))+", "+ str(RoundTokDecimals(y0,2))+")")

plt.grid(True)
plt.show()

```