



UNIVERSIDAD DE SONORA

DIVISIÓN DE CIENCIAS EXACTAS Y NATURALES

FÍSICA COMPUTACIONAL

14 DE MAYO DEL 2017

---

# Act 9: Teoría del Caos y el Mapeo Logístico

---

Alumna: Chávez Gutiérrez Yanneth Tzitzin  
Profesor: Carlos Lizárraga Celaya.

## 1. Breve resumen

En la práctica anterior se habló un poco sobre los conceptos de la teoría del caos, con ayuda de la teoría ya vista, en esta práctica nos apoyaremos del mapeo logístico. Basándonos principalmente en el trabajo de Geoff Boeing, sobre Teoría del Caos y el Mapeo Logístico se interpretarán datos a partir de condiciones iniciales dadas. Se analizará un conjunto de población y se analizará su comportamiento dadas algunas condiciones.

## 2. Introducción

La teoría del Caos es una rama de las Matemáticas que estudia el comportamiento de los sistemas dinámicos, que son aquellos sistemas que son muy sensibles de las condiciones iniciales. El Caos es una teoría multidisciplinaria que muestra que en la aleatoriedad aparente en algunos sistemas caóticos complejos, existen patrones de repetición, auto-similitud, fractales, y auto-organización. La propiedad de la gran sensibilidad a las condiciones iniciales del Caos o Caos determinista, fue expresada por Edward Lorenz, como "Caos: Cuando el presente determina el futuro, pero una aproximación del presente no determina aproximadamente el futuro".

El mapeo logístico muestra propiedades de una bifurcación de doblamiento de periodo y esto a su vez conduce al caos. Existen muchos fenómenos en la naturaleza que muestran un comportamiento caótico, un ejemplo de ello es el Estado del Tiempo y el Clima.

Para ello se requerirá instalar el paquete de Python, llamado pynamical, que puede ser instalado localmente en el sistema, mediante el comando: `pip install pynamical`

Haciendo uso del artículo de Geoff Boeing te cambiarán los colores, etiquetas y títulos de las animaciones obtenidas. Para ello se requerirá instalar el paquete de Python, llamado pynamical, que podrá ser instalado localmente en el sistema, mediante el comando:

```
pip install pynamical
```

### 3. Procedimiento

Para poder realizar la práctica se requirió instalar el paquete de Python ***Pynamical*** el cual se instaló desde la terminal de la siguiente forma:

```
pip install --user pynamical
```

Para poder instalar correctamente pynamical se requería una versión de matplotlib específica, en este caso podía ser la 2.0.0 o 1.5.3 en este caso se utilizó la versión 1.5.3

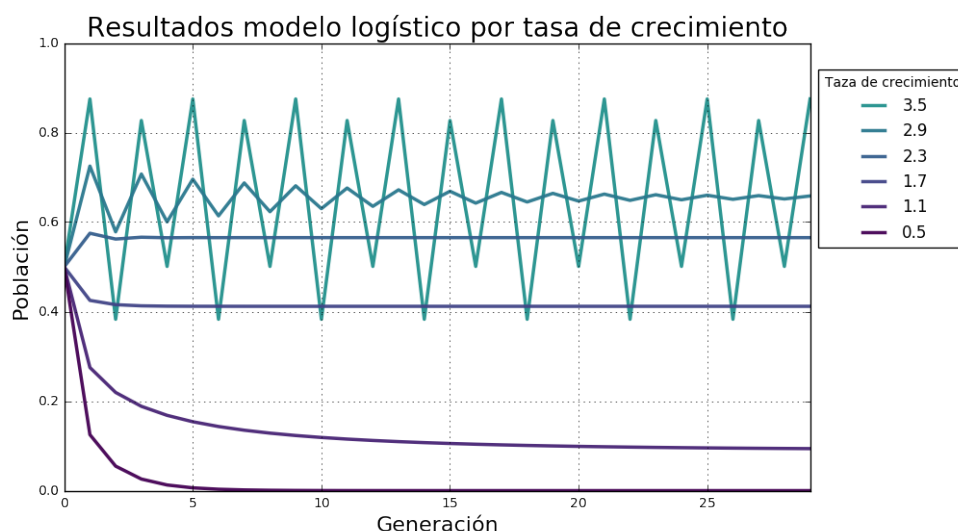
**¿Qué es pynamical?** Pynamical es un paquete de python utilizado para modelar, simular, visualizar y animar sistemas dinámicos discretos no lineales y el caos.

Los códigos utilizados se obtuvieron de **Github gboeing/pynamical** se aquí se pudieron obtener las figuras y se cambiaron títulos y colores de ellas.

### 4. Resultados

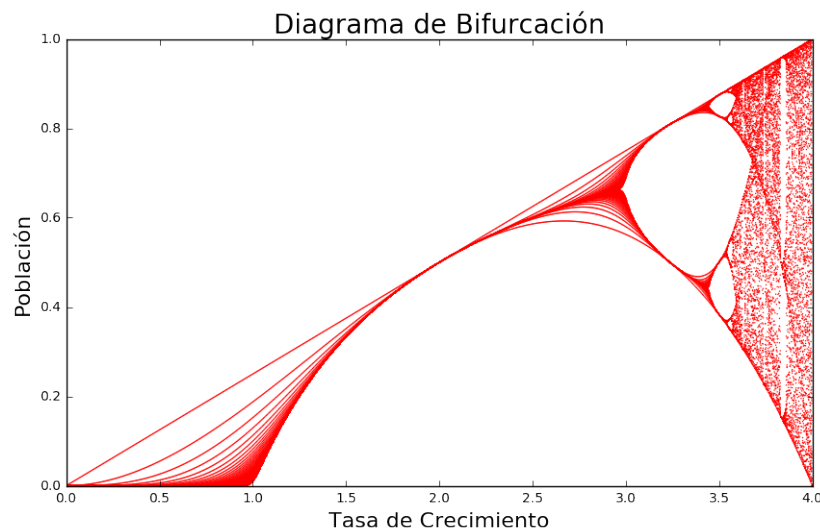
#### Comportamiento del sistema y atractores

Al darle los parámetros específicos para la población que se estará utilizando se lleva a cabo un modelo logístico con parámetros de crecimiento en este caso de: 0.5, 1.0, 1.5, 2.0, 2.5, 3.0, y 3.5. Con estos datos, primeramente se analizó el comportamiento general de nuestra población haciendo variar el parámetro de las generaciones a 30:

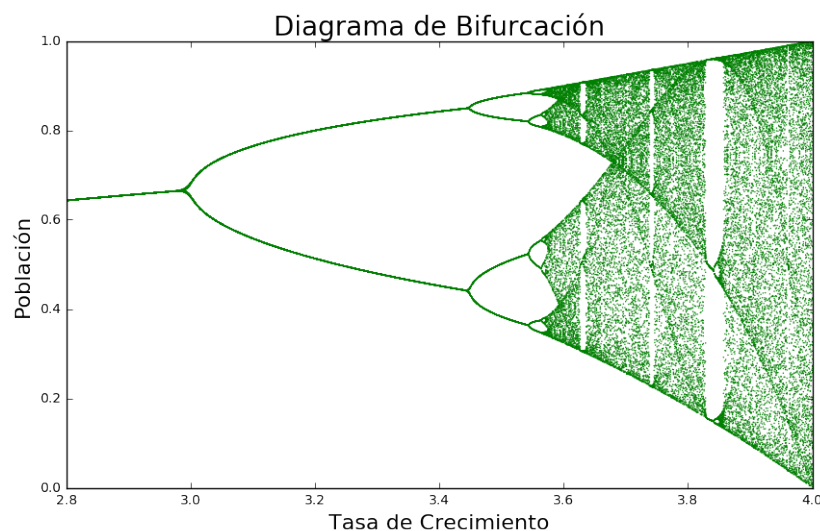


## Bifurcaciones y el camino al caos

Se corre ahora un modelo logístico para 200 generaciones en 1,000 tasas de crecimiento entre 0.0 y 4.0. En la figura pasada se visualizaron 7 tasas de crecimiento, aquí tendremos 1000 para lo que se necesitará visualizarlo de otra manera utilizando un *diagrama de bifurcación*:

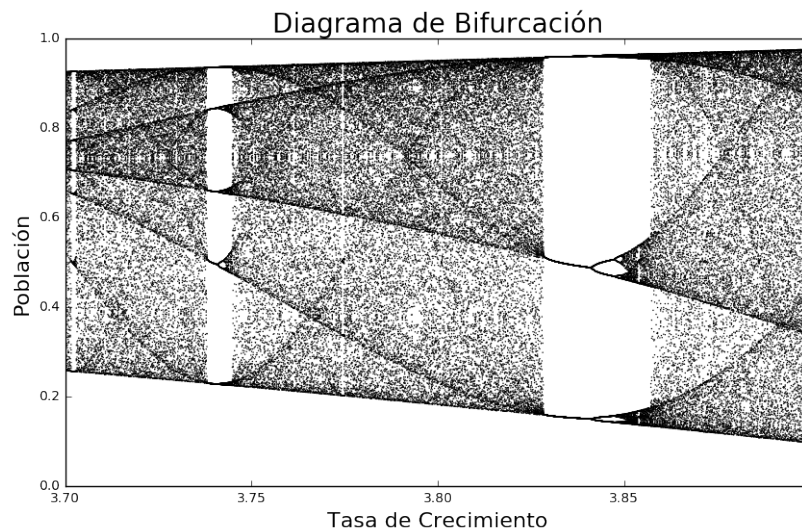


Y ¿por qué lo llamamos diagrama de bifurcaciones? A continuación hagamos zoom a la tasa de crecimiento entre 2.8 y 4.0 para ver que es lo que esta pasando:



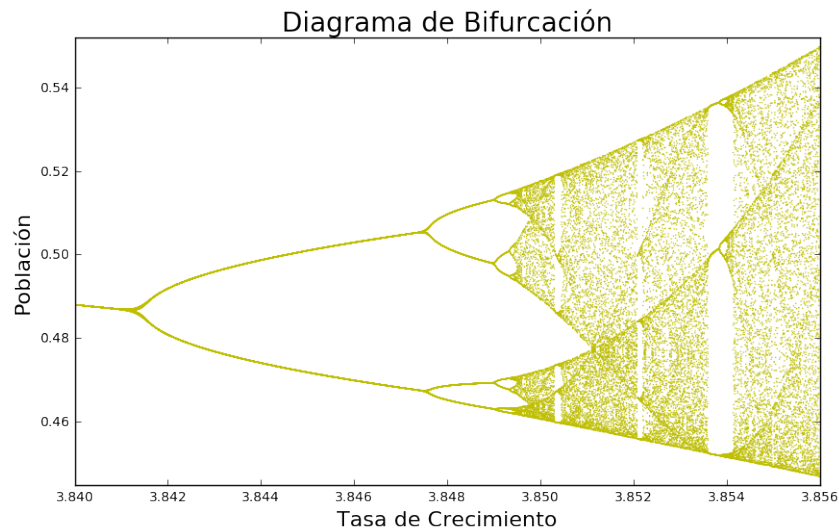
## El comienzo del caos

Haciendo zoom de nuevo, para tasas de crecimiento entre 3.7 y 3.9 obtenemos:



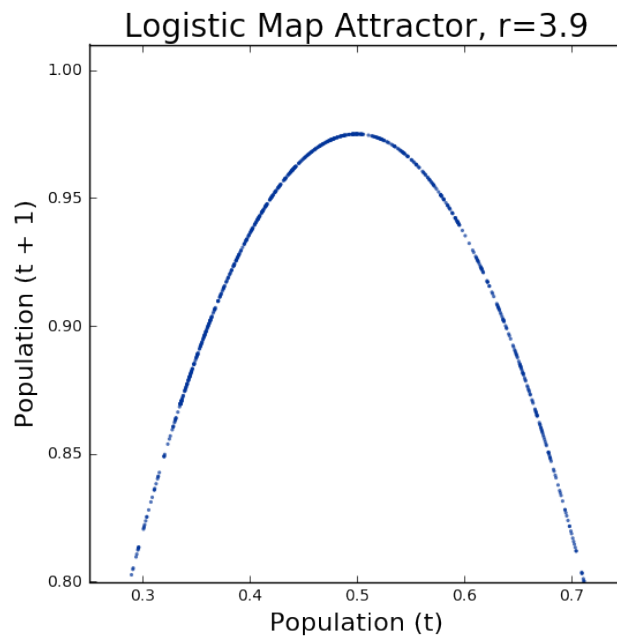
Al hacer más acercamientos podemos comenzar a notar el caos haciendose presente. Del ruido emergen patrones en los que el sistema se comporta de una manera muy diferente. Entre la tasa de crecimiento de los parámetros 3.82 y 3.84 el sistema se comporta en orden pero retorna al caos en parámetros más alla de 3.86.

## Fractales y atractores extraños



En esta figura podemos notar especies de fractales. Los fractales tienen la misma estructura a diferentes escalas.

Otra manera de visualizarlo es con un diagrama de fase que grafica el valor de la población como  $(t+1)$  en y contra la población en  $t$  en el eje x.

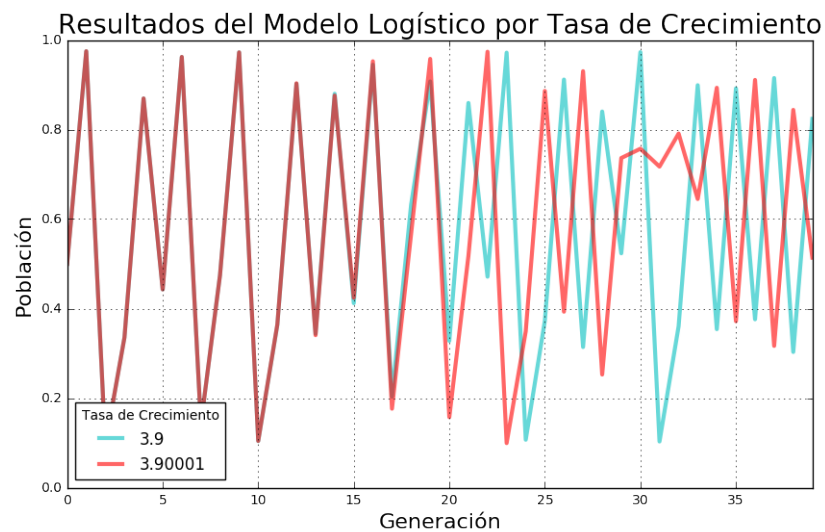


## Act 9: Teoría del Caos y el Mapeo Logístico

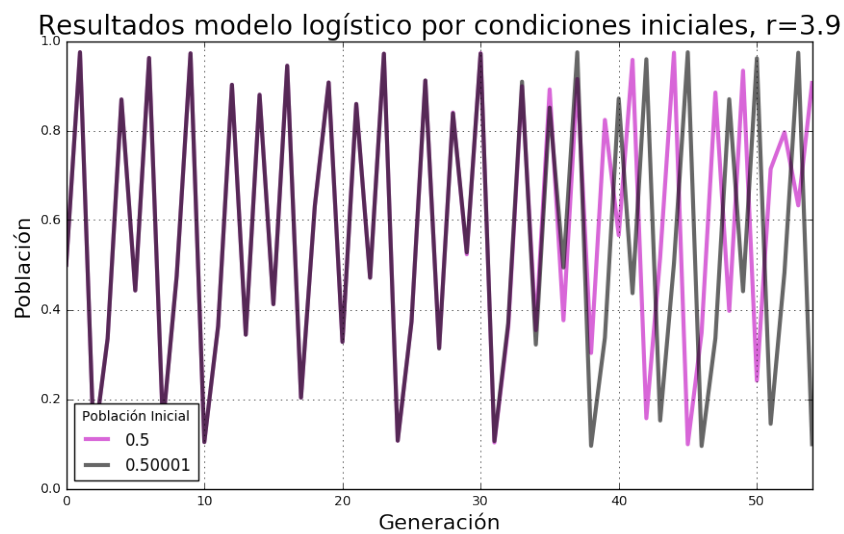
---

Ahora se visualizará el sistema con dependencia en las condiciones iniciales, dos sistemas que divergen al caos y otro que no:

Rango de 3.9 y 3.90001. Caótico.

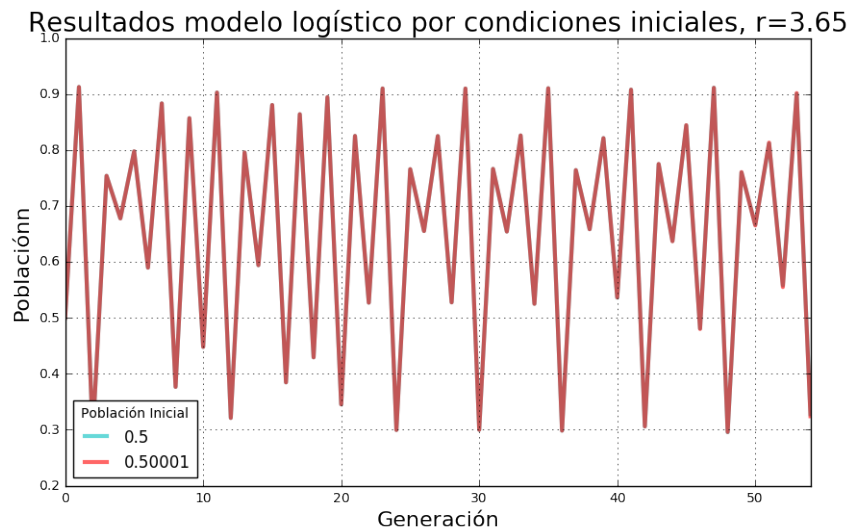


Rango de 3.9 para dos valores de poblaciones similares. Caótico.



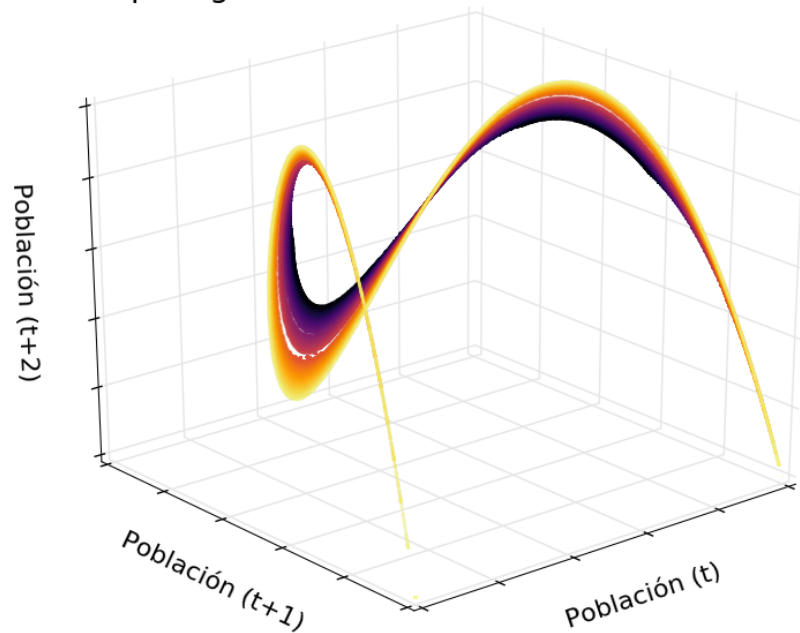
Tasa de crecimiento 3.65 para dos inicios similares de población. El rango es no caótico.

---



Los diagramas de fase permiten visualizar nuestros atractores extraños y identificar el caos del ruido aleatorio:

Mapa logístico de atractor de  $r=3.6$  a  $r=4.0$





## 5. Conclusiones

Se analizó como algunos sistemas divergen al caos y por medio de herramientas computacionales como Python y su biblioteca de Pynamical se pudo apreciar gráficamente lo que estaba ocurriendo para cierto modelo utilizado por medio de diferentes tipos de diagramas en los cuales en cada uno se pudo apreciar de una perspectiva diferente la representación de estos ya sea con diagramas de bifurcación, diagramas de fase etc.

Esta herramienta es muy útil ya que permite visualizar de manera muy interactiva estos sistemas.

## 6. Referencias

- (1) [http://computacional1.pbworks.com/w/page/117841293/Actividad %209 %20 %282017-1 %29](http://computacional1.pbworks.com/w/page/117841293/Actividad%209%20%282017-1%29) (2) <https://pypi.python.org/pypi/pynamical>
- (3) <https://github.com/gboeing/pynamical/tree/master/examples>
- (4) <http://geoffboeing.com/2015/03/chaos-theory-logistic-map/>