CB24

FUNCIONES RECURSIVAS, CAOS Y DINÁMICA DE POBLACIONES

Fernando López Gregorio & Pedro A. Willging

Facultad de Ciencias Exactas y Naturales - UNLPam
Uruguay 151 – 6300 – Santa Rosa – La Pampa
flopezgregorio@gmail.com, pedro@exactas.unlpam.edu.ar

Categoría del Trabajo, Nivel Educativo y Metodología de Investigación:
Propuestas didácticas, Educación Secundaria y Superior, Nuevas tecnologías y su impacto en la enseñanza y el aprendizaje de la Matemática

Palabras clave: caos, recursividad, programación, estrategia didáctica, modelos matemáticos

RESUMEN

Una estrategia para comprometer a los estudiantes en el análisis de problemas matemáticos es presentarles situaciones que relacionen el contenido matemático con aplicaciones a otras ciencias. Por otra parte, hay problemas que conectan una diversidad de áreas temáticas y cuyo abordaje involucra conceptos a los que se les da un tratamiento escaso o marginal en los cursos de matemática.

En este trabajo, presentamos una manera de abordar el estudio de una función recursiva, que puede aplicarse al estudio de modelos poblacionales, al mismo tiempo que conduce de manera natural al estudio de sistemas caóticos. Por medio del desarrollo de un algoritmo, que puede generarse con distintos lenguajes de programación, es posible introducir conceptos matemáticos por medio del estudio de imágenes complejas y asombrosas.

INTRODUCCIÓN

El encontrarnos con un viejo artículo: “Salto al espacio de Lyapunov” de A.K. Dewdney (1991) nos motivó a analizar una particular función recursiva, que aún siendo muy sencilla es muy susceptible a pequeños cambios en los parámetros iniciales. Empezamos por hacer un análisis computacional de la variante de la función logística desarrollada por Markus (1990) y calculando el exponente de Lyapunov para distintas secuencias de pares de valores. Comenzamos a investigar los usos que esta función tenía en distintas ciencias y cómo poder llevarla al aula para que los estudiantes se involucraran en el estudio de la recursividad mediante el uso de algoritmos computacionales. Vamos a comentar este recorrido, introduciendo previamente algunos conceptos necesarios para desarrollar el tema.

MARCO TEÓRICO

Las **Funciones Recursivas** son funciones que toman como argumentos sus mismos resultados (Mota, 2015). Con una función $f{:}A \rightarrow A$ tal que $y=f(x)$, queremos hacer $f(f(\ldots f(f(f(x)))\ldots))$ y ver qué ocurre al aplicar gran cantidad de veces la función f al resultado obtenido en el paso anterior. Entonces llamaremos a_0 a un valor elegido arbitrariamente y a partir de aquí, calculamos $a_1=f(a_0)$, $a_2=f(a_1)=f(f(a_0))$ y en general, $a_n=f^n(a_0)$ o lo que es lo mismo, el resultado de aplicar n veces la función f al valor a_0.

Un típico ejemplo de función recursiva se puede expresar como $a_0=constante$, y $a_n=f(a_{n-1})$. En esta definición queda claro que para conocer un valor de la sucesión tenemos que conocer el anterior; y para ese tenemos que conocer el resultado anterior a él, y así sucesivamente hasta llegar a a_0.

Aplicaciones de las Funciones Recursivas

¿Pero qué tiene que ver esto con la biología, la sociología, la tecnología y otras ciencias? Pensemos en un secreto que sólo conoce una persona. Ese individuo se lo dice en confianza a un amigo pero le pide que no lo divulgue. Pero resulta que el amigo y él mismo se lo cuentan en confianza a un par de terceros con la misma indicación de confidencialidad. Ya son cuatro que tienen conocimiento del secreto y de a poco se va divulgando. Supongamos que el universo de este problema se reduce a un grupo de 1000 personas. Cuando va pasando el tiempo, cada vez más personas lo saben y entonces el crecimiento de quienes conocen la confidencia (que parecía exponencial) tendrá un "freno" ya que cada vez será más difícil encontrar alguien que no lo sepa, sobre todo al acercarnos al millar de personas. Y al hacer crecer el tiempo indefinidamente, estaremos cada vez más cerca de que el total de personas conozca lo que comenzó sabiendo uno solo. Ese crecimiento se denomina logístico y se da mucho en la naturaleza en casos como la diseminación de una enfermedad, la reproducción de individuos de una especie en un determinado ecosistema, el porcentaje de personas de una comunidad que disponen de una cierta nueva tecnología, la cantidad de material que ha reaccionado a un experimento químico, y muchos otros casos (Ulloa & Rodríguez, 2010).
Existen muchas situaciones de la vida cotidiana y en particular en la naturaleza, donde se manifiestan las matemáticas (Stewart, 2011), vamos a ilustrar con algunos ejemplos una clase particular de funciones matemáticas que son las recursivas.

Los Conejos de Fibonacci: Se asocia con un matemático italiano del siglo XIII a uno de los problemas más conocidos para ilustrar la recursividad: es aquel que enuncia que "se tiene una pareja de conejos y se desea saber cuántos se podrían reproducir en un año a partir de la pareja inicial, teniendo en cuenta que de forma natural tienen una pareja en un mes, y que a partir del segundo se empiezan a reproducir". Como sabemos este problema lleva a la fórmula recursiva:

$$\begin{aligned} &F_1 = F_2 = 1 \\ &F_n = F_{n-1} + F_{n-2} \end{aligned} \qquad (1)$$

que genera la famosa sucesión de Fibonacci. Este es un ejemplo de la biología fácil de trabajar con nuestros estudiantes para generar la idea de recursividad. Ellos tendrán que ir calculando cuántas parejas de conejos hay en cada mes y luego buscar una fórmula que permita calcular el total de parejas que habrá en un mes determinado, para lo que deberán basarse en los pasos anteriores. Es decir, si les pedimos la cantidad de parejas de conejos que habrá al final del primer año, tendrán que calcular los meses anteriores para luego llegar a eso.

Árboles Fractales: Pensemos en un segmento vertical. Del extremo superior saquemos dos segmentos de la mitad de la longitud del segmento original, formando cada uno un ángulo de 120° con el segmento anterior. Repitamos el proceso descripto con cada uno de los dos nuevos extremos, y así sucesivamente. Se nos irá presentando un fractal (Guzman, Martin, Morán & Reyes, 1993) hecho recursivamente y con la forma de un árbol muy vistoso. Podemos cambiar el ángulo que forman las nuevas ramas con la anterior y la proporción que representa cada nueva rama con respecto a la previa y estas modificaciones harán que cambie la fisonomía del árbol, pero siempre tendrá la componente recursiva que permitirá ir haciendo aparecer dos ramas donde había una (ver animaciones en http://sabia.tic.udc.es/gc/Contenidos%20adicionales/trabajos/Imagenyvideo/fractales/arboles_fractales.htm).

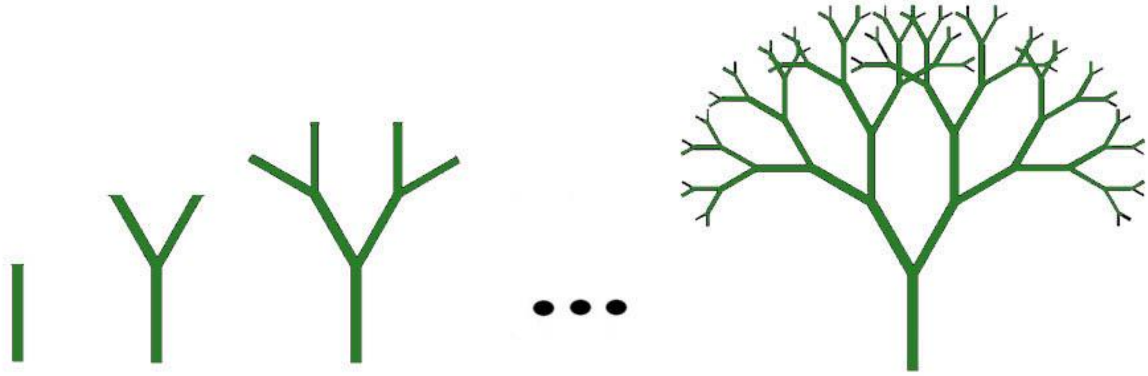

Figura 1.Creación de un árbol fractal de modo recursivo.

El Modulor de Le Corbusier: ¿No debería ser raro el hecho de que cuando vamos a sentarnos en una silla en cualquier lugar en el que nunca hemos estado sabemos la altura de la misma de forma casi “natural”? ¿O que los picaportes de las puertas se hallan todos a igual altura? Estas, claramente, no son casualidades. Las medidas a las que se encuentra el dintel de una puerta, la altura de una mesa, los artefactos de la cocina y el baño tienen dimensiones similares que provienen de un estudio hecho sobre la persona, en principio sobre una persona promedio. El arquitecto Le Corbusier tomó la altura de un hombre con la mano en alto y partió de que llegaría a 2,26 m de alto. A partir de allí, dividió en dos esa medida y obtuvo 1,13 m que es la altura del ombligo de esa persona. A cada uno de estos valores lo fue multiplicando y dividiendo recursivamente por el número de oro (aproximadamente 1,618) y obteniendo dos series de valores. Usando el 2,26 se define la serie azul y a partir del 1,13 surge la serie roja (Ostwald, 2001). Estas medidas que van surgiendo al usar recursividad nos dan las alturas de las mesas y de las sillas, de los dinteles y de los picaportes, de las perillas de la luz y de las mesadas. Y por eso, usando recursividad en el estudio de las proporciones del cuerpo humano, hallamos las medidas que nos hacen más fácil la vida, más cómoda a la hora de interactuar con los objetos cotidianos y más manejable en lugares desconocidos.

Figura 2: La serie azul en la casa del Modulor. Ilustración de Tobías Garzarón.

Algoritmos y Programación

En estos últimos años, se viene manifestando la necesidad de incorporar el estudio de la programación en la currícula de las escuelas y colegios, a fin de que los niños y jóvenes adquieran las habilidades necesarias para codificar en lenguajes de computación. Por un lado está la necesidad del mercado laboral que requiere de mano de obra calificada en el área del software y la ingeniería informática y por el otro, se indica que el pensamiento computacional es un requerimiento de la sociedad digitalizada (Wing, 2006). Las tecnologías informáticas son parte de la experiencia humana cotidiana y modifican las interacciones en el trabajo, en el aprendizaje, en la industria y el mercado, en el entretenimiento y prácticamente en todas las acciones de la sociedad. Al utilizar el pensamiento computacional se desarrollan las habilidades de pensamiento crítico y resolución de problemas y se equipa al estudiante con destrezas cognitivas de orden superior, que no sólo las puede aplicar al ámbito de la programación, sino que las traslada a todas las asignaturas en las que participe.

Tanto en Argentina como en el resto del mundo existen en la actualidad varios movimientos que están abogando por la implementación de programas de aprendizaje de la programación y la resolución de problemas mediante el uso de algoritmos como por ejemplo la Fundación Sadovsky (http://www.fundacionsadosky.org.ar/). Iniciativas como Program.ar, Mumuki.org o Code.org (ver http://program.ar/ , http://www.mumuki.org/, y https://code.org/) dan cuenta de la importancia que se le está dando a la necesidad de lograr que los jóvenes se motiven a aprender a codificar.

El Mundo del Caos

La utilización de las computadoras y el desarrollo de algoritmos que se programan en lenguajes de computación que pueden resolver infinidad de cálculos de manera precisa y rápida, facilita el estudio de problemas matemáticos complejos, como es la aparición de funciones con un comportamiento caótico. Se denomina como teoría del caos a la rama de las matemáticas, la física y otras ciencias que trata sistemas complejos y dinámicos muy sensibles a las variaciones en las condiciones iniciales. Una pequeña variación en las condiciones iniciales pueden significar una gran diferencia en el comportamiento futuro, imposibilitando la predicción a largo plazo. La historia reciente del caos se sitúa en la década de 1950 con la aparición de las computadoras y el desarrollo de las primeras gráficas sobre el comportamiento de sistemas no lineales. Edward Lorenz se encontró con ecuaciones que tenían comportamiento caótico cuando intentaba predecir condiciones climáticas en la atmósfera. Los modelos de la Naturaleza (como los atmosféricos), son extremadamente sensibles a pequeños cambios. A esta característica de los estados caóticos se la conoce como "efecto mariposa" (Devaney, 1992).

Uno de los tests disponibles para la detección del caos en un sistema es el del cálculo del exponente de Lyapunov. Un estudio detallado de los fundamentos matemáticos de este tipo de cálculo está más allá de las pretensiones de nuestro trabajo, pues involucra conceptos de topología y análisis matemático muy avanzados para el nivel secundario. Pero conformémonos con establecer que cuando este exponente es positivo, el sistema es caótico (Lyapunov, 1907).

El Enfoque Pedagógico

Cada aproximación a la enseñanza-aprendizaje de las ciencias se sustenta en teorias pedagógicas que van modificándose y adaptándose. Dentro de los sustentos teóricos que consideramos en nuestra propuesta se encuentran el aprendizaje por descubrimiento, el construccionismo y el aprendizaje basado en la resolución de problemas. El aprendizaje por descubrimiento (Bruner, 1966) es aquel en el que los estudiantes construyen conocimiento por sí mismos, generalmente guiados por el docente, y en contraposición al modelo de enseñanza trasmisivo de información. Esto requiere de un rol activo del estudiante (Martínez y Zea, 2004).

Según Eleizalde et al. (2010) la enseñanza por descubrimiento guiada provee a los estudiantes con "oportunidades para manipular activamente objetos y transformarlos por la acción directa, así como actividades para buscar, explorar y analizar. Estas oportunidades, no solo incrementan el conocimiento de los estudiantes acerca del tema, sino que estimulan su curiosidad y los ayudan a desarrollar estrategias para aprender a aprender, descubrir el conocimiento, en otras situaciones" (p. 272). En el área de la enseñanza de la computación, son muy conocidos los trabajos seminales de Seymour Papert (1980) con el lenguaje Logo. Sus trabajos son continuación de la obra de Piaget y generan el construccionismo, una teoria del aprendizaje en la cual el sujeto aprende en la medida que construye su propio conocimiento por medio de la acción, involucrándose activamente e interactuando con su contexto (Ackermann, 2001). El aprendizaje basado en problemas (ABP) es una metodología en la cual los estudiantes adquieren conocimientos mientras aprenden a aprender de forma progresivamente independiente, con la guía del docente. Los estudiantes aprenden a aplicar los nuevos conocimientos en la resolución de problemas similares, a identificar aspectos del problema que necesitan mayor exploración y a investigarlos por iniciativa propia (Norman & Schmidt, 1992). La metodología ABP permite superar los límites de las asignaturas tradicionales, ya que se concreta con una serie de problemas elaborados por profesores de materias afines. El ABP es un método de trabajo activo que se centra en el estudiante y no en el docente o en los contenidos (Restrepo, 2005).

¡VAMOS AL AULA!

Con todo lo anterior en mente es que planteamos una secuencia didáctica para introducir el tema de la programación recursiva en nuestros cursos, ya sean de secundaria como de nivel superior.

Podríamos comenzar con el siguiente problema:

En una isla de la Patagonia se introduce un grupo de 30 ciervos a compartir y competir con la flora y la fauna autóctona. Se estima que el número máximo de ejemplares de esta especie foránea que puede albergar el territorio es de 500, por cuestiones de espacio y de alimento. El lugar tiene pocos predadores, algo de competencia por la comida y fuentes de agua dulce que ayudan al desarrollo de los animales introducidos, por lo que fácilmente comienza a aumentar el número de ciervos. Pero, ¿qué pasará a lo largo del tiempo? ¿Llegarán a llenar por completo la isla o por el contrario desaparecerán de aquel lugar ajeno?

Empecemos a estudiar el problema...

Un modelo matemático que permite trabajar el problema introductorio es el de una función logística de fórmula

$$a_{n+1} = r\, a_n\, (1 - a_n), \qquad (2)$$

donde a_0 toma un valor entre 0 y 1 que indica el porcentaje del máximo teórico que representan actualmente los individuos presentes. Cada término subsiguiente caerá en el mismo intervalo.

El comportamiento logístico de esta fórmula deberá surgir como parte del análisis de los estudiantes, luego de que efectúen varios cálculos, en los que pueden anticipar un crecimiento exponencial, o según las reglas de Fibonacci, o más probablemente lineal.

El parámetro *r* está definido por las condiciones que facilitan o dificultan el progreso a los nuevos habitantes del lugar (la fecundidad). En este caso, tomaremos un *r* de 2 para el ejemplo anterior. Las razones que justifican esta elección exceden lo pretendido en este problema aunque el análisis que hagamos del parámetro será lo que nos lleve al caos o a la regularidad.
Hagamos algunas cuentas con el ejemplo de los ciervos. La proporción de animales que había inicialmente sobre el total de los que puede albergar la isla es de 30/500. Por lo tanto, $a_0=.06$.
Un periodo después, pongamos por caso un año,

$$a_1= 2\, a_0\, (1 - a_0) = 2 * .06 * (1 -.06) = .1128, \qquad (3)$$

o sea que en un año alcanzaron un 11.28% del total que pueden lograr. En otras palabras, en un año aumentaron hasta 56 ciervos. Los siguientes años se pueden ver en la Tabla 1 que se presenta a continuación:

Año	0	1	2	3	4	5	6	7	8
$a_{Año}$	.06	.11	.20	.32	.44	.49	.50	.50	.50
$Ciervos_{Año}$	30	56	100	160	218	246	250	250	250

Tabla 1: Evolución de la cantidad de ciervos en el tiempo (*r*=2)

Podemos ver que a partir del sexto año el número se mantiene estable, lo que quiere decir que el grupo de ciervos se estabilizará en los 250 teniendo aún espacio y alimento para crecer a una cantidad mayor pero no pudiendo hacerlo por motivos ajenos a los recién mencionados.
Ahora bien. ¿Qué pasaría si *r* fuera 3.05 en lugar de 2? Parecería que el comportamiento no debería ser demasiado diferente del anterior aunque tal vez tendiendo a un valor distinto de 250 ciervos, incluso pudiendo ser que se extingan. Pero veámoslo en la Tabla 2:

Año	0	1	2	3	4	5	6	...	26	27	28	29	30
$a_{Año}$	.06	.17	.43	.75	.57	.75	.58	...	.59	.74	.59	.74	.59
$Ciervos_{Año}$	30	86	217	375	286	373	289	...	295	369	295	369	295

Tabla 2: Evolución de la cantidad de ciervos en el tiempo (*r*=3.05)

En este caso podemos observar que a partir del año 26 la secuencia se estabiliza pero en dos subsucesiones que convergen a valores distintos: 369 y 295. Esto querría decir que el número de animales en la isla va a tener un efecto rebote cada dos años. ¿Cómo entender esto? Seguramente la sobrepoblación que se presenta un año con 369 ciervos hará que durante ese año los predadores tengan mayor facilidad para atrapar sus presas haciendo decaer el número hacia los 295, año en el cual la facilidad de los ciervos para conseguir alimento y asilo les facilite reproducirse y volver al número de 369, y así sucesivamente. Esta nueva situación se diferencia bastante del caso anterior en el cual la población se estabilizaba, pero aún más se diferencia del caso que sigue a continuación. Tomemos *r*=3.6 y armemos la Tabla 3 similar a las anteriores:

Año	0	1	2	3	4	5	6	7	8
$a_{Año}$	.06	.20	.58	.88	.39	.86	.44	.89	.36
$Ciervos_{Año}$	30	102	291	438	196	429	219	443	182

Tabla 3: Evolución de la cantidad de ciervos en el tiempo (*r*=3.6)

Y aunque sigamos calculando no se estabiliza en ningún punto ni encontramos periodos en los cuales se repita la secuencia y alcanzamos a vislumbrar el caos en una forma sencilla de armar pero complicada de entender. Esta población de ciervos vivirán de cambio en cambio, año tras año, sin lograr un equilibrio siquiera periódico. Y podríamos seguir cambiando valores de *r* entre 0 y 4 para ver qué va ocurriendo, pero nos encontraremos que con pequeños cambios en su valor el comportamiento pasa de ser estable a caótico. El análisis de estos casos es relativamente sencillo y puede verse resuelto en el mencionado artículo de A.K. Dewdney (1991).

Forzamiento periódico

Entonces pasamos a la variante que Markus diseñó con forzamiento periódico: hizo que el valor de *r* cambiara periódicamente entre dos constantes fijas *a* y *b*. Es decir, el valor inicial de *r* sería *a*, en la siguiente iteración *r* valdría *b*. Luego volvería a tomar el valor *a*, después el *b*, y así sucesivamente.

Aquí el comportamiento se vuelve más complicado de estudiar ya que dependerá de dos valores y de la secuencia que use, ya que lo expuesto en el párrafo anterior es lo que llamaremos secuencia *ab*, pero podríamos usar *aab*, que indicaría que *r* toma un par de veces el valor *a*, luego una vez el *b*, para volver a comenzar con el patrón. Entonces, a la sensibilidad que el problema ya tenía para *r* constante hay que sumarle estas nuevas variables que lo hacen más interesante pero a la vez mucho más complejo para predecir qué pasará para cada elección de los parámetros.

El análisis numérico que puede realizarse de estas variantes de la función logística está en base al mencionado exponente de Lyapunov. Para una secuencia dada de valores *a* y *b* nos interesará ver el grado de caoticidad que presenta la recursividad, por lo que tendremos una función con dominio en (0,4)x(0,4) y cuya imagen nos indicará el exponente calculado para esos valores de *a* y de *b*.

¿Y cómo calcularemos el exponente Lyapunov para esta función?

$$x_n = r\,x_{n-1}\,(1 - x_{n-1}) \rightarrow x_n' = r - 2\,r\,x_{n-1} \rightarrow \textit{exponente Lyapunov} = \lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} log_2 \left| r - 2\,r\,x_i \right| \qquad (4)$$

Para aproximar el exponente, tendremos que calcular el argumento del límite anterior para valores de *n* lo suficientemente grandes; y esto deberemos hacerlo para cada par de valores que tomen los parámetros *a* y *b*, por lo que es inviable resolver las cuentas a mano y tendremos que valernos de un código que nuestros mismos estudiantes pueden generar, una vez sugerida la idea de límite en infinito como “*n* muy grande”.

Nosotros a continuación proponemos uno realizado en Octave, que puede servir de modelo.

El Código

```
#Paisajes de Lyapunov
# Willging & Lopez Gregorio - 2016
# Facultad de Ciencias Exactas y Naturales
# Universidad Nacional de La Pampa
#Calculo del exponente de Lyapunov

x=3.0;
y=3.0;

function [lyap]=ExpLyapunov(x,y)
```

```

r1=x;
r2=y;
x=0.016;
s='aabb';
l=length(s);
total=0.0;

for i=1:400
  if(s(mod (i,l)+1)=='a')
  r=r1;
  else r=r2;
  endif

  x= r*x*(1-x);
  total=total+log(abs(r-2*r*x))/log(2);
endfor

lyap=total/400;

endfunction

#Graficos de Lyapunov

M=zeros(120,100);
G=zeros(120,100);
x0=2.5;
y0=2.5;

 for i=1:120
    x=x0+(0.0125*(i-1))/1.0;
    for j=1:100
      y=y0+(0.0125*(j-1))/1.0;
      M(i,j)=ExpLyapunov(x,y);
    endfor
   endfor

mini=min(min(M))

#Definir la paleta de colores

  for i=1:120
    for j=1:100
      if (M(i,j) > 0)
          G(i,j)=0;
      endif

      if (M(i,j) <= 0)
          G(i,j)=abs(M(i,j)/mini);
      endif
```

```
61    endfor
62   endfor
63
64   figure;
65   imshow(G)
```

La salida de este programa, que tardó 8 minutos en ejecutarse, es la siguiente imagen:

Figura 3: Caos y regularidad en los espacios Lyapunov

En la línea 10 del código comienza la definición de la función ExpLyapunov que toma como entrada un par de valores para *a* y *b* (los llama *x* e *y*) y calcula, con 400 iteraciones en el programa que mostramos, el exponente de Lyapunov para esos valores. En la línea 40 se carga la matriz con los exponentes Lyapunov, en la 48 se calcula el mínimo exponente para "normalizar" la matriz, y en la línea 50 se carga la matriz G que se representa al final del código y cuyo gráfico es el que vemos en la Figura 3. Creemos que el algoritmo es sencillo y que también es fácil encontrar cómo adaptarlo para usarlo para otras secuencias (modificables en la línea 15), para otros valores iniciales de *a* y *b* (en las líneas 37 y 38), distintos incrementos y otras cantidades de iteraciones.

Hemos probado diferentes variantes de este código básico, entre ellas hicimos algunas con gráficos en colores o en 3D, con posibilidades de elegir los valores iniciales y finales de *a* y *b*, así cómo la cantidad de valores tomados entre estos límites, con distintas cantidades de iteraciones, pero cada nueva función que le agregábamos incrementaba el tiempo de ejecución, haciendo en ciertas ocasiones, que tuviéramos que dejar el programa trabajando algunas horas. Igualmente, con lo expuesto cumplimos con el objetivo del trabajo que era ver cómo introducir en forma sencilla el caos, la recursividad y el algoritmo correspondiente a un problema muy sensible a cambios de los parámetros.

Para estudiar y experimentar el problema con otro software, en el sitio web http://www.efg2.com/Lab/FractalsAndChaos/Lyapunov.htm se puede descargar un programa que realiza el gráfico con cálculos optimizados que abrevian el tiempo de ejecución. Es configurable y la página trae enlaces a sitios con imágenes resultantes asombrosas.

Para seguir analizando...

Presentamos a continuación un par de variaciones de este problema que se pueden llevar adelante con nuestros estudiantes, utilizando estas mismas ideas y aún el mismo código, pero haciéndole modificaciones que permitan tanto aprender nuevos conceptos como reforzar los aprendidos y que también giran en torno a la idea de que con variaciones pequeñas en los parámetros el caos y la regularidad se mezclan.

- Un grupo de conejos crece según la misma función logística que empleamos en este trabajo, pero si bien cambian la proporción de conejos mes a mes, en verano el valor de *r* es 3.5, en invierno baja *r* a 2.5, mientras que en otoño y primavera *r* toma el valor 3.
    - ¿Para qué proporciones iniciales de conejos la población se estabilizará?
    - ¿Para qué proporciones se presentará el caos?
    - ¿Cambia la respuesta si iniciamos el estudio en distintas estaciones?
    - Experimentar con otros valores de *r* para cada estación del año.

- Una colonia de insectos ve afectada su proporción de individuos en relación con un total teórico de diferente forma de día que de noche. De día el parámetro a emplear es $r=a$, con a entre 0 y 4, mientras que de noche es $4-a$.
    - ¿Qué ocurre con la colonia a lo largo del tiempo?
    - ¿Es igual el comportamiento para cualquier proporción inicial?
    - ¿Qué ocurriría si se usaran valores $r=a$ entre 1 y 2, mientras que el valor de noche siguiera siendo $4-a$?

CONCLUSIONES

En este trabajo, hemos presentado una forma de introducir la idea de recursividad, la generación de algoritmos recursivos, la codificación de éstos y la idea de caoticidad mediante un problema interdisciplinario, que provoque y movilice a los estudiantes a buscar respuestas que difícilmente lograrían alcanzar sin el uso de las herramientas tecnológicas propuestas, y que al mismo tiempo les permiten experimentar como con cambios sutiles se pueden provocar grandes diferencias en los resultados.

Quedan abiertas muchas otras posibilidades y caminos a explorar, que busquen la interacción y complementariedad entre las ciencias, en nuestro caso particular las aplicaciones de la matemática. Creemos que esta propuesta puede ser modificada y adaptada a distintas situaciones áulicas, cambiando el nivel de complejidad y dificultad, de acuerdo al grupo de estudiantes y el nivel educativo. Las posibilidades en cuanto a la programación son muchas, ya que pueden hacerse modificaciones y agregados en el código de manera de obtener imágenes de distinta calidad gráfica y artística, así como utilizar otras variantes de software. Esperamos que lo planteado en este trabajo sea de utilidad y motive a otros docentes a probarlo en sus clases.

REFERENCIAS

- Ackermann, E. (2001). Piaget’s constructivism, Papert’s constructionism: What’s the difference. *Future of learning group publication*, *5*(3), p. 438. Recuperado el 22 de abril de 2016, de http://learning.media.mit.edu/content/publications/EA.Piaget%20_%20Papert.pdf
- Bruner, J. (1966). *Toward a Theory of Instruction*. Cambridge, MA: Harvard University Press.

- Dewney, A. K. (1991). Salto al espacio de Lyapunov. *Investigación y Ciencia*, 182, 94-97.
- Devaney, R. L. (1992). *A First Course in Chaotic Dynamical Systems: Theory and Experiment*. Reading, MA: Addison Wesley.
- Eleizalde, M., Parra, N., Palomino, C., Reyna, A., & Trujillo, I. (2010). Aprendizaje por descubrimiento y su eficacia en la enseñanza de la Biotecnología. Revista de Investigación, 34(71), 271-290. Recuperado el 28 de abril de 2016, de http://www.scielo.org.ve/scielo.php?script=sci_arttext&pid=S1010-29142010000300014&lng=es&tlng=es.
- Guzmán, M., Martín, M., Morán, M., & Reyes, M. (1993). *Estructuras fractales y sus aplicaciones*. Barcelona: Editorial Labor.
- Lyapunov, A.M. (1907): Problème général de stabilité de mouvement. *Annales de la Faculté des Sciences de l'Université de Toulouse*, 9, 203-475.
- Markus, M. (1990). Chaos in maps with continuous and discontinuous maxima. *Computers in Physics*, 481-493.
- Martínez, E. R., & Zea, E. (2004). Estrategias de enseñanza basadas en un enfoque constructivista. *Revista Ciencias de la Educación, 2* (24), 69-90.
- Mota, S. (2015). Sobre el concepto de recursion y sus usos. *Praxis Filosófica*, 40, pp. 153-181. Recuperado el 2 de abril de 2016, de http://www.scielo.org.co/pdf/pafi/n40/n40a07.pdf
- Norman, G.R., & Schmidt, H.G. (1992). The Psychological Basis of Problem-Based Learning: A Review of the Evidence. *Academic Medicin*e, 67 (9), 557-565.
- Ostwald, M. J. (2001). Le Corbusier (Charles Edouard Jeanneret) The Modulor and Modulor 2, *Nexus Network Journal*, **3,** 145–147.
- Papert, S. (1980). *Mindstorms. Children, Computers and Powerful Ideas*. New York: Basic books.
- Restrepo, B. (2005). Aprendizaje basado en problemas (ABP): una innovación didáctica para la enseñanza universitaria. *Educación y Educadores*, *8*, 9-19.
- Stewart, I. (2011). *Las matemáticas de la vida*. Barcelona:Critica.
- Ulloa Ibarra, J.T., & Rodríguez Carrillo, J.A. (2010) El modelo logístico: Una alternativa para el estudio del crecimiento poblacional de organismos. *REDVET. Revista electrónica de Veterinaria*, 11 (3). Recuperado el 3 de marzo de 2016, de http://www.veterinaria.org/revistas/redvet/n030310/031004.pdf
- Wing, J. M. (2006). Computational Thinking, *Communications of the ACM*, 49(3), 33-35.