



## BICHOS DISTRIBUIDOS

Ma. Paola Neri-Ortiz

paoneri@hotmail.com

<http://www.geocities.com/paolanerortiz>

Facultad de Física e Inteligencia Artificial - Universidad Veracruzana  
Sebastian Camacho No. 5 Col. Centro, Xalapa, Ver.



### RESUMEN

En este trabajo, se presenta una extensión del desarrollo de Mundo Artificial. Una aplicación de los Sistemas Evolutivos, los autómatas celulares y las matrices evolutivas al campo de la Vida Artificial, el cual se enfoca a generar ecosistemas simulados por computadora. Algunas posibles áreas de aplicación de este tipo de sistema, se encuentran en el modelado de ecosistemas reales y en el desarrollo de videojuegos evolutivos, los cuales generen nuevos personajes, no planeados, durante el desarrollo del juego.

Su funcionamiento se basa en el uso de matrices evolutivas, generadas a partir de valores aleatorios, las cuales representan insectos virtuales. La interacción constante entre insectos produce cambios permanentes en las matrices que describen los insectos, y, en algunos casos, estos cambios generan nuevas entradas en las matrices, con lo que se pueden representar bichos nuevos, no predefinidos, creados con características nuevas basadas en las características de los padres. Estos cambios en las matrices es lo que le da el carácter de evolutivo al sistema

La extensión de este proyecto pretende generar ecosistemas complejos donde la emergencia pueda surgir mostrando comportamientos en mandada no prediseñados. Para esto se utilizan varios mapas comunicados a través de una red con el fin de “migrar” los insectos de un mapa a otro cuando se cumplan ciertas condiciones dotando al sistema de distribución.

Palabras clave: Sistemas Evolutivos, Vida Artificial, Autómatas Celulares, Matrices Evolutivas, Juego de la Vida, insectos virtuales, Sistema Distribuido.

## INTRODUCCIÓN

Mundo Artificial<sup>1</sup> esta basado en el juego de la vida diseñado por John Conway que intenta modelar el desarrollo de una población con el uso de puntos llamados “células”, los cuales interaccionan usando tres reglas preestablecidas.

- 1.-Cada elemento permanece vivo, si tiene dos o tres *vecinos* (necesita compañía para vivir).
- 2.-Un elemento muere, si tiene más de tres *vecinos* (por sobrepoblación).
- 3.-Un elemento nace, en un espacio vacío, si tiene exactamente tres *vecinos*

Mundo Artificial Basado en Sistemas Evolutivos es un espacio simulado por computadora llamado *mapa* donde existen toda clase de seres parecidos a insectos llamados “bichos”, los cuales poseen características que los distinguen como forma, color y tiempo de vida.

Los bichos pueden nacer, alimentarse, reproducirse y morir dentro de un ambiente simulado llamado *mapa*.

Un bicho, posee diferentes características que lo distinguen, como forma, color, tiempo de vida, madurez reproductiva y fuerza de ataque. Todas estas características, definen una especie.

Dos bichos de la misma especie pueden reproducirse y crear otro bicho idéntico a ellos. Y también, dos bichos de diferente especie pueden reproducirse para generar una nueva especie. Debido a lo anterior, habrá especies que desaparezcan y otras nuevas que surjan con el paso del tiempo haciendo que las especies de bichos se mantengan en constante evolución.

Las reglas que rigen a los bichos para reproducirse y alimentarse, dependen directamente de las características de la especie a la que pertenecen. *Como los bichos, son evolutivos, las reglas que los rigen, se establecen cada vez que una nueva especie es creada, así, estas reglas y las características de cada bicho, están en constante evolución.*

La presente versión de Mundo Evolutivo pretende extender el alcance del espacio de vida de los bichos o *mapa* usando el modelo cliente – servidor generalizado o distribuido.

Una vez que la aplicación es ejecutada podrá crearse otra en el mismo computador o en otro remoto conectado a través de una red. Los bichos que existen en uno de los mapas pueden migrar a otro siempre y cuando haya espacio en el mapa a donde se desea trasladar al bicho.

Para esto será necesario que el applet que contiene al mapa sea cliente de un programa servidor que hace de receptor de peticiones de migración a otros mapas y a la vez de solicitador de migraciones de los bichos de su propio mapa.

---

<sup>1</sup> Aunque aquí se hará una breve mención del funcionamiento de mundo artificial, se recomienda leer la documentación del proyecto base Mundo Artificial en la pagina web <http://www.geocities.com/paolaneriortiz/>.

## FUNCIONAMIENTO

El sistema está codificado en lenguaje Java de Sun Microsystems. Tiene dos módulos principales: un programa applet cliente que genera un *Mapa* y un servidor que atiende las peticiones de los mapas de migrar sus bichos a otros mapas.

### EL APPLET CLIENTE

El programa principal es el applet cliente que ejecuta el sistema. Está organizado en clases como sigue.

- *Mapa*. En él se guarda la definición de la matriz de bichos vivos en pantalla, el tiempo de vida y la ingesta actual de cada bicho.
- *Especie*. Definición de las características que distinguirán a una especie de bicho.
- *Regla*. Definición de las reglas de vida (alimentación y reproducción) que seguirá una especie.
- *Util*. Conjunto de herramientas para obtener números aleatorios.

La clase *Mapa* tiene la siguiente forma:

```
class Mapa{
    int Estado; //indica si hay o no un bicho en esta posicion
    int IdEspecie; //identificador de la especie del bicho existente en esta posicion
    int TiempoVida; //tiempo de vida actual del bicho en esta posicion
    int Ingesta; //ingesta actual del bicho en esta posición
}
```

En la función *paint* (equivalente a *main* en c) se declaró un arreglo de clase *Mapa*. Cada elemento del arreglo almacena los datos de un bicho; el estado indica si hay o no un bicho vivo en la *i*-ésima posición del arreglo y sus datos de especie, tiempo de vida e ingesta.

La clase *Especie* se determina como sigue:

```
class Especie{
    //Estructura logica del bicho
    int TiempoVidaMax; //Tiempo maximo de vida del bicho
    int MadurezRep; //Tiempo que debe esperar el bicho
    //para poder reproducirse
    int IngestaMax; //Cantidad maxima de alimento que puede ingerir el bicho
    int Fuerza; //Fuerza de ataque del bicho

    //Estructura fisica del bicho
    int Npuntos; //Numero de puntos que forman el cuerpo
    int []px; //puntos en el eje x que forman el cuerpo
    int []py; //puntos en el eje y que forman el cuerpo
    int Npatas; //numero de patas del insecto
}
```

Color color; //color del bicho

Esta clase determina las características de la especie tanto la forma como la estructura lógica; tiempo de vida máxima en ciclos que puede vivir el bicho, el tiempo en ciclos que debe dejar pasar el bicho para alcanzar la madurez reproductiva, la cantidad máxima de alimento que puede almacenar y la fuerza de ataque y defensa contra otros bichos.

La clase Regla es como sigue:

```
class Regla{
int [ ][ ]Alimentacion;
int [ ][ ]Reproduccion;
int MaxNumEspecies;
int Nespecies;
int Nreglas;
}
```

Esta clase almacena dos matrices de datos correspondientes a las reglas de alimentación y reproducción donde la intersección de una fila y columna indican si la especie  $i$  y  $j$  son presa y depredador, o bien si son compatibles para combinarse y crear nuevas especies.

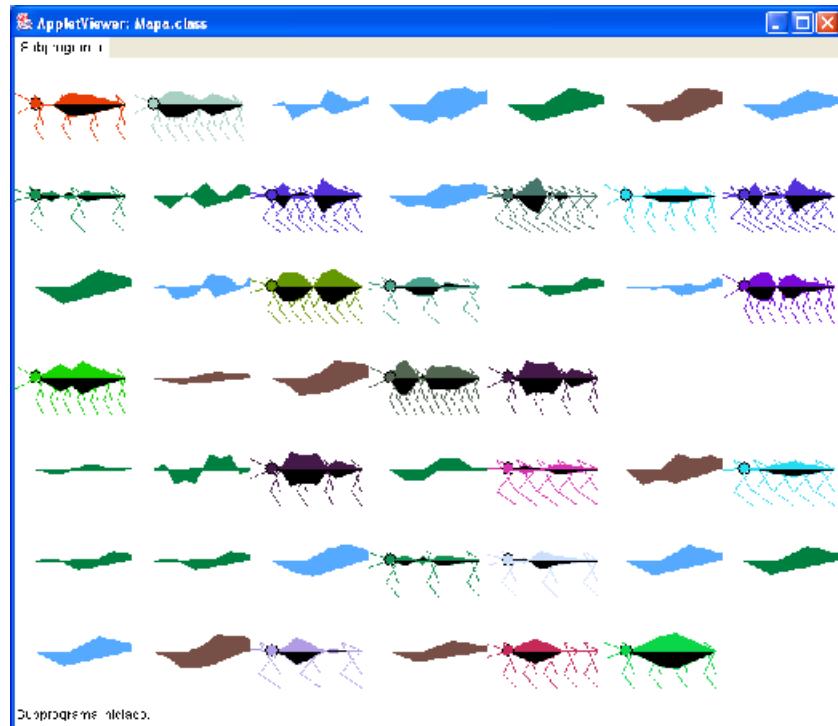
Para la comunicación con el programa servidor utiliza dos funciones: enviaBicho y recibeBicho. Ambas funciones reciben una posición del mapa. La función enviaBicho tiene el siguiente algoritmo envía los datos de un bicho al servidor y usa el siguiente algoritmo.

- Solicita enviar un bicho al servidor
- Si la solicitud es aceptada se envía una cadena que contiene los datos que describen a un bicho y su estado actual. Al terminar de enviar, el bicho es eliminado del mapa.
- De lo contrario termina.

La función recibeBicho solicita un bicho al servidor para insertarlo en una posición que previamente se ha marcado como vacía. Tiene la siguiente estructura.

- Solicita un bicho del buffer del programa servidor
- Si la solicitud es aceptada se capturan los datos del bicho proveniente de algún mapa remoto. Al terminar la recepción, el bicho es insertado en el mapa.
- De lo contrario, termina.

El applet cliente se ve como sigue:



Applet Mundo Artificial

## EL PROGRAMA SERVIDOR

El programa servidor se encarga de trasladar los bichos de un mapa a otro. Funciona con el siguiente algoritmo:

- Al entrar una solicitud
- Si la solicitud es recibir un bicho y el buffer esta vacío
  - recibe los datos del bicho y los guarda en una variable llamada bichoString que hace de buffer.
- Si la solicitud es enviar un bicho y hay uno en el buffer
  - envía la cadena que representa a las características de un bicho almacenado en la variable bichoString.

```
Acceso directo a servBicho.bat
C:\jdk1.6\bin>java servBicho
Escucho...
..Recibe una conexión
buffer:
CLIENTE:ENVI0_BICHO
SERVIDOR: Recibiendo bicho de mapa remoto
SERVIDOR: Bicho recibido 4215 593 512 88 5 3 74 139 59 4115 511
Escucho...
..Recibe una conexión
buffer:4215 593 512 88 5 3 74 139 59 4115 511
CLIENTE: ENVI0_BICHO -peticion denegada
Escucho...
..Recibe una conexión
buffer:4215 593 512 88 5 3 74 139 59 4115 511
CLIENTE:RECIBO BICHO
SERVIDOR: Enviando bicho al mapa local
SERVIDOR: Bicho enviado
Escucho...
..Recibe una conexión
buffer:4215 593 512 88 5 3 74 139 59 4115 511
CLIENTE:ENVI0_BICHO
SERVIDOR: Recibiendo bicho de mapa remoto
SERVIDOR: Bicho recibido 3797 692 542 92 3 4 247 252 35 3397 538
Escucho...
..Recibe una conexión
buffer:3797 692 542 92 3 4 247 252 35 3397 538
CLIENTE: ENVI0_BICHO -peticion denegada
Escucho...
..Recibe una conexión
buffer:3797 692 542 92 3 4 247 252 35 3397 538
CLIENTE: ENVI0_BICHO -peticion denegada
Escucho...
-
```

Programa Servidor

## CONCLUSIONES

En el presente trabajo se propuso la aplicación de los *Sistemas Evolutivos* al área de la *Vida Artificial* como una herramienta en la simulación de espacios de vida, incorporando herramientas como los *autómatas celulares* y las *matrices evolutivas*. Dando como resultado un espacio simulado donde seres virtuales llamados *bichos* nacen, crecen, se reproducen y mueren bajo el control de reglas que se crean y modifican constantemente.

Durante la ejecución del sistema fue posible distinguir los patrones de comportamiento de algunas especies. Por ejemplo, se encontraron algunas especies cuyo comportamiento era similar a la de una manada de leones, es decir, el número de bichos creados era relativamente pequeño mientras que la fuerza de ataque era grande, haciendo que la manada se separara del resto de los bichos, dejando un espacio claramente delimitado a su alrededor.

Algunos mapas concluyen con la muerte de todos los bichos, mientras que otros, logran establecer un equilibrio entre la reproducción y la alimentación de algunos grupos de especies. Este proceso es el que resulta mas parecido al funcionamiento que tiene un ecosistema en equilibrio, en donde, una especie produce suficientes crías como para no extinguirse cuando otra especie mas fuerte se alimenta de ella.

De forma aleatoria los bichos de un mapa migran a otro funcionando como agentes externos que alteran al sistema. Cuando un bicho se traslada de mapa puede ocurrir que siendo dominante en su ecosistema se comporte igual en su nuevo ambiente, pero también puede ocurrir que muestre un comportamiento emergente y el bicho se comporte diferente a lo esperado en su nuevo habitat.

La herramienta generada presenta una opción de laboratorio de observación de un ecosistema cuyas interacciones entre los elementos que viven en él se realiza directamente en pantalla dando mayor posibilidad de emergencia.

## REFERENCIAS

- 1.-GALINDO Soria, Fernando. "Matrices Evolutivas". *Revista Científica num. 8. Editorial de la ESIME del IPN. Cd. de México. Marzo – Abril de 1998*
- 2.-MEDINA May, José Armando. "Sistemas Evolutivos Generadores de Escenarios Fractales". *Memorias del X Congreso nacional ANIEI sobre Informática y Computación. Monterrey, N. L. Octubre 1997.*
- 3.-GALINDO Soria, Fernando. "Sistemas Evolutivos". *Boletín de Política Informática del INEGI- SPP. Cd. de México. Septiembre de 1986.*
- 4.-OLIVARES Ceja, Jesús. "Agente Evolutivo Manejador del Conocimiento". *Memorias del X Congreso Nacional ANIEI sobre Informática y Computación, Monterrey, N. L. Octubre 1997.*
- 5.-EMMCHE, Claus. "Vida Simulada en el ordenador. La nueva ciencia de la Inteligencia Artificial. Colección Límites de la Ciencia. Editorial Gedisa (<http://www.gedisa.com>). Octubre de 1988
- 6.-ADAMI, Christoph. "Introduction to Artificial Life". *Editorial Telos (<http://www.telospub.com>)*
- 7.-JIMÉNEZ Morales, Francisco. "Autómatas Celulares y Vida Artificial". *Francisco Jiménez Morales. <http://complex.us.es/~jimenez/CA/ac/ac.html>*
- 8.-TORRES Hernández, Luis E., LONGORIA G., Alfonso, ROJAS Salinas, Antonio. "Aplicación de los Sistemas Evolutivos en el Análisis de Espectros de Rayos Gamma". *Memorias de la Primera Conferencia de Ingeniería Eléctrica CIE/95. CINVESTAV del IPN. Cd. de México. Septiembre 1995.*
- 9.-GALINDO Soria, Fernando. "Desarrollo Histórico de los Sistemas Evolutivos". *UPIICSA del IPN. Cd. de México. Noviembre 1989.*
- 10.-NERI Ortiz, Ma. Paola. "Mundo Artificial Basado en Sistemas Evolutivos". *<http://paoneri.topcities.com/> ESCOM del IPN, Cd. de México. Septiembre 2000*
- 11.-La Vida Llegó a Maloka. *Página web con información descriptiva sobre el área de la Vida Artificial. <http://maloka.org/virtual/vida/evolucion.htm>*
- 12.-GARCÍA V., Angélica, MUÑOZ Meléndez, Angélica, NEGRETE, José. "Propuesta de un Mundo Sintético Fundado en Metodologías Basadas en el Comportamiento". *Memorias del VI Congreso Nac. sobre informática y computación organizada por la Asociación Nac. de Instituciones de Educación en Informática (ANIEI). Mérida, Yucatán. Octubre 1993*
- 13.-GARCÍA García, Karla, SALCIDO Bustamante, Sergio, VENTURA Silva, Alfonso. "Sistema Evolutivo de Reconocimiento de Formas en dos dimensiones". *Teoría y Práctica de los Sistemas Evolutivos. Editor Jesús Olivares Ceja ([www.jesusolivares.com](http://www.jesusolivares.com)). Cd. de México, 1997*