



---

---

**INSTITUTO TECNOLÓGICO DE IZTAPALAPA.**

**INGENIERÍA EN SISTEMAS COMPUTACIONALES**

**ACTIVIDAD:**

PRACTICAS Y APUNTES.

**MATERIA:**

LEGUAJES Y AUTOMATAS I.

**PROFESOR:**

ING. ABIEL TOMAS PARRA HERNANDEZ.

**ALUMNO:**

REYES GOMEZ JAVIER ROBERTO 181080144.

**GRUPO:**

6BM.

**FECHA DE ENTREGA:**

CDMX A 27 DE JUNIO DEL 2021.



## SESION I.

### TEORIA DE LENGUAJES FORMALES

Sabemos que los alfabetos son un sistema finito y no vacío que cuyos elementos, que estos se denominan símbolos. Estos se designan normal mente con letras:  $\Sigma$

Algunos ejemplos que vimos en clase fueron:

$\{0,1\}$

$\{a,b,c,\dots,x,y,z\}$

$\{0,1,2,3,4,5,6,7,8,9\}$

$\{a,b\}$

Recordemos que el autómata finito aceptará una cadena en este caso será W, esta cadena comenzará desde el estado de inicio hasta un estado de aceptación.

En lo mencionado en clase la teoría de autómatas tiene cuatro niveles que se mencionaron:

- Sistema combinacional
- Autómata finito
- Autómata con pila
- Máquina de Turing

En base a la estructura de un traductor, observamos que tiene una sucesión de pasos en base a eso lo primero es generar un código intermedio, después optimizar el código intermedio y posterior mente generar código objeto (debemos gestionar los errores, tabla de símbolos y enfocarnos al objetivo del texto).

Las Facetas de un compilador se basan en programas de computadora que traducen de un lenguaje a otro.

El compilador toma como su entrada un programa escrito en lenguaje fuente y produce un programa equivalente escrito en lenguaje objeto.

Un compilador se compone internamente de varias etapas, o fases, que realizan operaciones lógicas.

## SESION II

### DEFINICION FORMAL DE UNA ER

Lo que se comentó en clase de la definición formal de expresión ER, es que es un equivalente algebraico para un autómata debido a que son patrones utilizados para determinar combinaciones de caracteres dentro de una cadena. La cual podemos expresar de dos maneras muy simples:

1. Representación literal ( var re = /ab + c/ ; ) patron cerrado entre diagonales.
2. Función constructora var re = new RegExp('ab + c') recopilación en tiempo de ejecución de la expresión.

### DISEÑO DE ER.

El diseño de ER se basa en un diseño que se miro en clase , el ejemplo de ellos es;

Dado el alfabeto  $\Sigma = \{ a,b,c \}$  ,  $(a \cup b)^*a^*(bc)^*$  según lo comentado en clase esto es una expresión regular que esto va a representar al lenguaje .

$(\{a\} \cup \{b\})^* \cdot \{a\}^* \cdot \{bc\}^*$

### APLICACIONES EN PROBLEMAS REALES.

Las expresiones regulares facilitan la construcción de un compilador. A menudo se utiliza una expresión regular larga y compleja para validar la sintaxis de un programa, si el código del programa no concuerda con la expresión regular, sabremos que hay un error de sintaxis dentro del código.



### SESION III.

#### AUTOMATAS FINITOS.

Lo que se comentó en clase es que los autómatas finitos es un modelo computacional que realiza cálculos en una forma automática sobre una entrada para así producir una salida. A si mismo se conforma por medio de un alfabeto, se podría decir por un conjunto de estados finitos, es como una función de transición, un estado inicial y también un conjunto de estados finales.

Estos autómatas deterministas se definen como una variable (a) estaría formado, también se define como una quintupla que se forma mediante 5 elementos:

Q será el número de estados que se encuentran en el autómata.

$\Sigma$  se refiere al alfabeto

$\delta$  esta es la función de transición

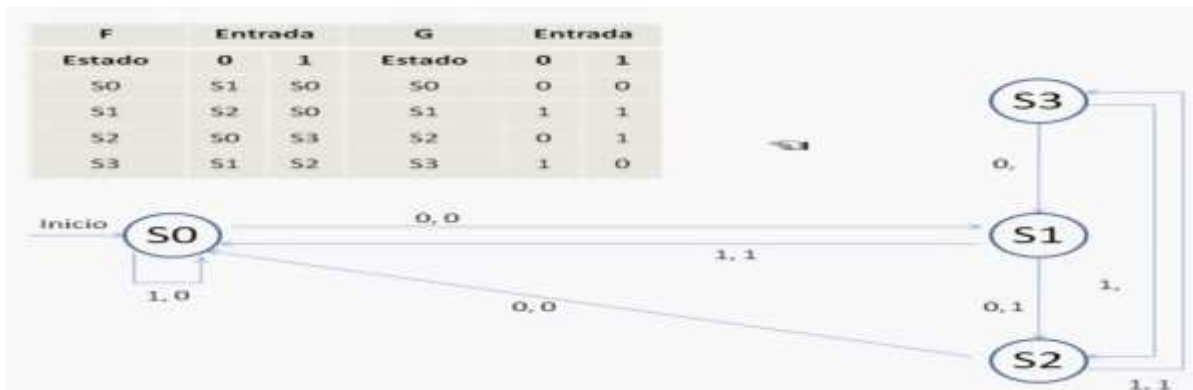
$q_0$  estado inicial

F sería conjunto de estados finales.

Deterministas: Cada combinación (estado, símbolo de entrada) produce un solo estado.

No Deterministas: Cada combinación (estado, símbolo de entrada) produce varios estados y además son posibles las transiciones con  $\lambda$ .

En clase también se mostró como se representa un ER a AFND, la tabla nos indica los resultados de los estados que obtenemos mediante el diagrama.



### SESION IV

#### FUNCION ANALISIS LEXICO

A la conclusión que llegamos en clase es que un análisis léxico es una herramienta la cual nos ayuda a leer los caracteres de entrada de un código y así poder identificarlos, para poder realizar la función de leer la secuencia de caracteres del programa, carácter a carácter y así producir la salida compuesta de tokens.

#### COMPONENTES LEXICOS.

Un analizador léxico, inicialmente lee lexemas y le asigna un significado propio.

Componente Léxico: Secuencia lógica y coherente de caracteres relativos a una categoría (Identificador, palabras reservadas, espacios en blanco, paréntesis).

Patrón: Es la secuencia que genera la secuencia de caracteres que puede representar una expresión regular.

Lexema: Cadena de caracteres que concuerden con un patrón que describa el valor de la cadena.

Atributos: Proporciona información adicional sobre los tokens en sus atributos asociados.

#### CREACION DE TABLA DE TOKENS.



La tabla de Tokens es una serie de renglones cada uno de los cuales contiene una lista de valores de atributos que sean asociados con una variable en particular.

Símbolos de mayor importancia para poder proveer la información:

- Da un identificador.
- Que información es asociada con un nombre.
- Como se asocia es información con el nombre.
- Como acceder a la información

Funciones que realiza la tabla de tokens:

Validad que la semántica sea correcta.

Ayudar en la generación apropiada del código.

**ERRORES LEXICOS.**

Los errores léxicos los detectamos cuando el proceso de reconocimiento de componentes léxicos y la cadena de caracteres de la entrada no encajan con ningún patrón.

Algunos errores son:

- Nombres incorrectos de los identificadores: Se debe a que utilizamos caracteres inválidos (paréntesis, empezamos por un numero).
- Numero incorrectos: Los colocamos con caracteres inválidos (puntos en lugar de comas o no esta escrito correctamente).
- Palabras reservadas escritas incorrectamente: Problemas de ortografía.
- Caracteres que no pertenecen al lenguaje: @, ¿?

**GENERADORES DE ANALIZADORES LEXICOS.**

El generador léxico nos permite leer caracteres del archivo de entrada, donde encontraremos la cadena a analizar y poder reconocer las subcadenas que corresponden a los símbolos del lenguaje y retornar los tokens.

Existen tipos de generadores léxicos:

- Generador Lex: Este generador nos ayuda a generar análisis sintáctico, genera código fuente en c, a partir de una serie de especificaciones escritas en el lenguaje.
- Generador Flex: Este generador nos brinda la realización para patrones lexicos en texto.
- Generador JTLex: Este generador nos brinda expresar sintaxis y semántica en los esquemas de traducción.

## **SESION V.**

### **ANÁLISIS SINTÁCTICO**

Mediante el transcurso de la clase vivos gramática la cual tipo 3 son regulares de acuerdo al formato de producción puede ser:

Lineales a la derecha en ejemplo de ello que visualizamos en la sesión sincrónica

A → aB

A → a

Y las lineales a la izquierda

A → Ba

A → a

El tipo 2: se refiere al libre contexto como, por ejemplo:

.A → B es siempre indispensable del contexto.

La gramática libre de contexto es formal en la que cada una de las reglas de producción es de la forma  $V \rightarrow W$  donde se puede decir que V será el símbolo no terminal y W será siempre una cadena de terminales y/o no terminales. Con esto nos referimos que al hecho de que el no terminal V puede siempre ser sustituto por W sin haber tenido en cuenta el contexto en el que ocurra.



Recordemos que los árboles de derivación es un lenguaje que se genera por una gramática libre de contexto se representa gráficamente mediante un árbol con raíz ordenado, a eso se le conoció como árbol de derivación. Damos a entender que la raíz del árbol presenta un símbolo inicial. Los nodos internos del árbol representan los símbolos no terminales, que parecen en la derivación. Las hojas de los árboles esas representan los símbolos terminales.

En las formas normales de Chomsky: remplazaremos toda la producción de una forma como la que se muestra a continuación:

$A \rightarrow C_1 C_2 \dots C_n$  por una cadena de producciones

$A \rightarrow C_1 V_1 A \rightarrow C_1 V_2 V_1 A \rightarrow C_1 V_2 V_3 V_1 A \rightarrow \dots \rightarrow C_1 V_2 V_3 \dots V_{n-2} V_{n-1} C_n$

Donde los  $V_1 \dots V_{n-2}$  son nuevas variables, intermedias.

Lo visto del diagrama de sintaxis es otra forma al igual que los árboles de derivación de poder especificar gramáticas de cualquier tipo en especial de tipo 2. Este esquema permite visualizar las derivaciones al instante de que ocurren. Es una forma visual de poder representar la gramática de un lenguaje.

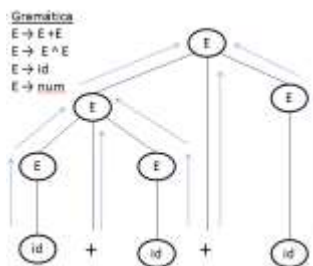
También vimos la ambigüedad que nos indica la eliminación de ella, para las construcciones de los lenguajes de programación comunes, en la mayoría de las gramáticas, damos entender que no se respeta la precedencia de operadores, también esa secuencia puede agruparse desde la izquierda y desde la derecha. Lo convencional es agrupar desde la derecha.

En el Análisis Sintáctico Ascendente se parte de las hojas y se intenta construir el árbol hacia arriba hasta llegar al símbolo inicial de la gramática.

En un análisis top-down un parser hacer corresponder cadenas de entrada con sus correspondientes derivaciones izquierdas.

En un análisis bottom-up un parser hace corresponder cadenas de entrada con las inversas de las correspondientes derivaciones derechas.

EJEMPLO:



La generación de matriz predictiva nos menciona que los First y Follows son significados para verificar y construir pares predictivos, son un conjunto de tokens que se encuentran en el tope de la pila.

Pseudocódigo y algoritmo para calcular los First de una gramática:

1.-Si  $X$  es un terminal  $FIRST(X) = \{X\}$  2.-Si  $X \rightarrow e$  es una producción  $e$  (pertenece)  $FIRST(X)$

3.-Si  $X \rightarrow Y_1 Y_2 \dots Y_n$  es una producción.

a.  $FIRST(Y_1)$  (es subconjunto de)  $FIRST(X)$

b. Si  $e$  (pertenece)  $FIRST(Y_k)$  (para todo)  $k < i$   $FIRST(Y_i)$  (es subconjunto de)  $FIRST(X)$

c. Si  $e$  (pertenece)

$FIRST(Y_k)$  (para todo)  $1 \leq n$   $e$  (pertenece a)  $FIRST(X)$  Nota (a,b y c pertenecen al paso 3 y solo se toma uno dependiendo el caso).





Los errores sistemáticos se suelen producir cuando la sintaxis del código es correcta, pero la semántica o significado no es el que se pretendía.

La construcción obedece las reglas del lenguaje, y por ello el compilador o intérprete no detectan los errores semánticos.

Los compiladores e intérpretes sólo se ocupan de la estructura del código que se escribe, y no de su significado. Un error semántico puede hacer que el programa termine de forma anormal, con o sin un mensaje de error.

## SESION VI

### DEFINICION FORMAL MT.

En clase se determinó que en 1936 se definió la máquina de Turing por Alan Turing, la cual es un modelo matemático (computacional) el cual nos permite interpretar problemas matemáticos a través de un algoritmo, debido a que manipula símbolos mediante una tira de cinta con reglas (Tabla), debido a que es un dispositivo hipotético el cual representa una maquina de computación.

### CONSTRUCCION MODULAR DE UNA MT.

La construcción modular tiene como finalidad desarrollar maquinas complejas a partir de bloques elementales (a partir de máquinas pequeñas), para la creación de máquinas de Turing se llevan a cabo diagramas de transición para poder combinarlos de manera parecida y así realizar la formación de la unión y concatenación de los autómatas finitos

### LENGUAJES ACEPTADOS POR LA MT.

La máquina de Turing puede considerarse como un autómata capaz de reconocer lenguajes formales, es capaz de reconocer los lenguajes recursivamente enumerarles ( lenguajes más generales, o tipo cero (0) ), de acuerdo a la jerarquía de Chomsky .

Su potencia es, superior a otros tipos de autómatas, como el autómata finito, o el autómata con pila, o igual a otros modelos con la misma potencia computacional.

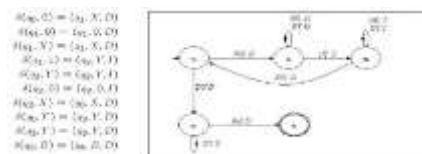
El siguiente ejemplo se visualizó en una de las sesiones:

Una cadena  $\omega \in A^*$ , es aceptada por una MT, si comienza en el estado  $e_0$ , con la cabeza de lectura/escritura en el símbolo más a la izquierda, luego de leer toda la cadena  $\omega$ , llega a un estado  $e_f \in F$ .

El lenguaje aceptado por MT, es el conjunto de todas las cadenas que son aceptadas por MT:

$L(MT) = \{ \omega / e_0 \omega \vdash^* \alpha_1 e_f \alpha_2 \text{ y } e_f \in F \text{ y } \alpha_1, \alpha_2 \in A^* \text{ y } \omega \in A^* \}$

Tenemos por ejemplo una MT que reconoce el lenguaje  $\{0^n 1^n : n \geq 1\}$ . Las transiciones de la máquina se representan como sigue:



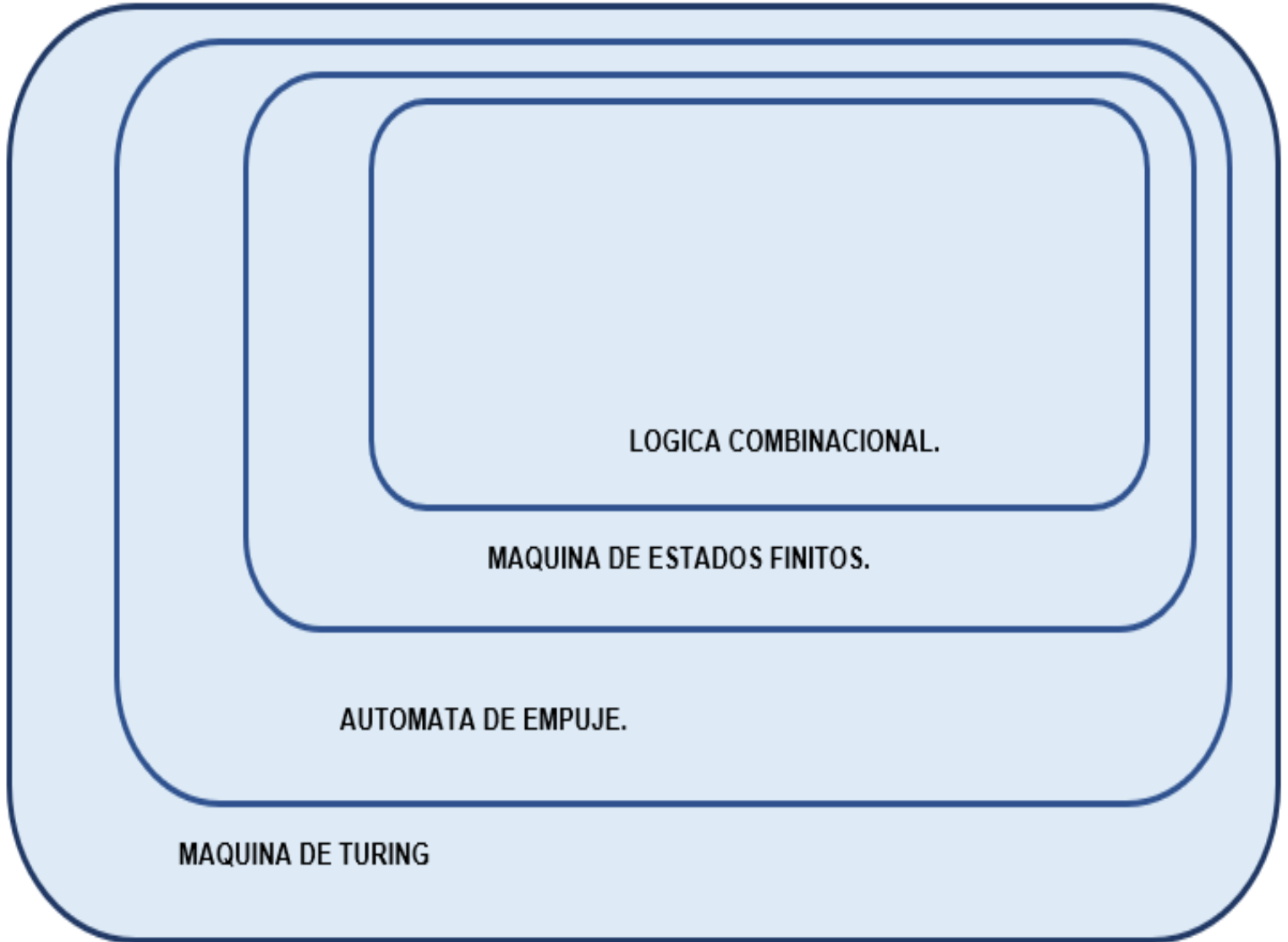
Se evalúa la cadena  $w = 1100$ , arrojando el siguiente resultado:

$q_0 0011$	$\vdash$	$X q_1 11 \vdash X 0 q_1 11 \vdash X q_2 0 Y 1$
	$\vdash$	$q_2 X 0 Y 1 \vdash X q_0 0 Y 1 \vdash X X q_1 Y 1$
	$\vdash$	$X X Y q_1 1 \vdash X X q_2 Y Y \vdash X q_2 X Y Y$
	$\vdash$	$X X q_0 Y Y \vdash X X Y q_3 Y$
	$\vdash$	$X X Y Y q_3 B \vdash X X Y Y B q_4 B$

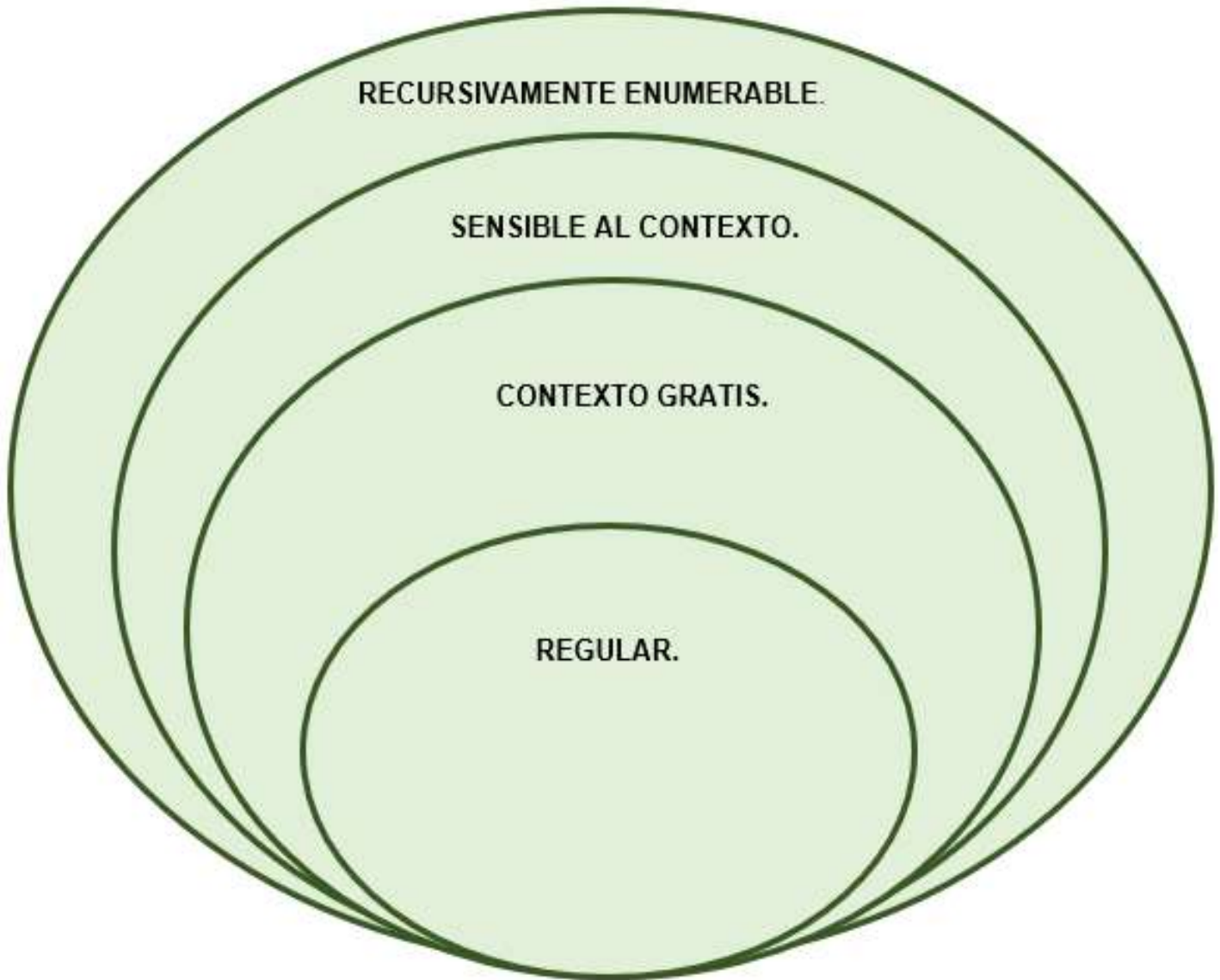
Otras cadenas también aceptadas por esta MT son 11110000, 10, 11111110000000.

ANEXOS:

## TEORIA DE LOS AUTOMATAS



FILE: CHOMSKY-HIERARCHY.SVG





GRAMÁTICA	IDIOMAS	AUTOMATA	REGLAS DE PRODUCCIÓN (RESTRICCIONES)	EJEMPLO
TIPO-0	RECURSIVAMENTE ENUMERABLE	MAQUINA DE TURING	$\gamma \rightarrow \alpha$ SIN RESTRICCIONES	$L = \{w   w \text{ DESCRIBE UNA MAQUINA DE TURING TERMINANDO}\}$
TIPO 1	SENSIBLE AL TEXTO.	MAQUINA DE TURING NO DETERMINISTA DELIMITADA LINEAL.	$\alpha A \beta \rightarrow \alpha \gamma \beta$	$L = \{a^n b^n c^n   n > 0\}$
TIPO 2	LIBRE DE TEXTO	AUTOMATA PUSHDOWN NO DETERMINISTA	$A \rightarrow \alpha$	$L = \{a^n b^n   n > 0\}$ —
TIPO 3	REGULAR	AUTOMATA DE ESTADO FINITO.	$A \rightarrow \alpha \vee A \rightarrow \alpha B$	$L = \{a^n   n > 0\}$



# INSTITUTO TECNOLÓGICO DE IZTAPALAPA 1

## LENGUAJES Y AUTOMATAS 1

### **RESUMEN. DESENREDANDO LA HISTORIA DE ADA LOVELACE.**

#### INTEGRANTES:

ANGELICA BEATRIZ CAMACHO CAPULÍN 181080421

OSVALDO RUIZ BARCO 181080148

JAVIER ROBERTO REYES GÓMEZ 181080144

AVILEZ ARELLANO RAFAEL IVAN 181080381

ISC-6BM



## RESUMEN. DESENREDANDO LA HISTORIA DE ADA LOVELACE.

Ada Lovelace nació hace 200 años hoy. Para algunos, es una gran heroína en la historia de la informática, nació en Londres el 10 de diciembre de 1815 de padres de la alta sociedad. Ada aprendió historia, literatura, idiomas, geografía, música, química, costura, taquigrafía y matemáticas (enseñadas en parte a través de métodos experimentales) hasta el nivel de geometría elemental y álgebra. Cuando Ada tenía 11 años, fue con su madre y un séquito en una gira de un año por Europa. Cuando regresó, estaba haciendo cosas con entusiasmo, cómo estudiar lo que ella llamaba "flyología", e imaginar cómo imitar el vuelo de las aves con máquinas de vapor. Su padre era un orfebre y banquero emprendedor y exitoso fue a Cambridge para estudiar matemáticas, pero pronto se propuso modernizar la forma en que se realizaban allí, fundó la Sociedad Analítica para impulsar reformas cómo reemplazar la notación basada en puntos de Newton para el cálculo con la notación basada en funciones de Leibniz. A principios de 1822, Babbage, de 30 años, estaba ocupado estudiando diferentes tipos de maquinaria y produciendo planos y prototipos de lo que podría ser el Motor Diferencial. El encuentro de Ada con el motor de diferencias parece ser lo que encendió su interés por las matemáticas. Ella escribió: "Creo que poseo una combinación sumamente singular de cualidades que encajan exactamente para hacerme preeminentemente un descubridor de las realidades ocultas de la naturaleza". Ada había tenido problemas de salud intermitentes durante años, pero en 1841 aparentemente empeoraron y comenzó a tomar opiáceos de forma sistemática. Tenía muchas ganas de sobresalir en algo y comenzó a tener la idea de que tal vez debería ser música y literatura en lugar de matemáticas. Pero su esposo William parece haberla convencido de que no lo hiciera y, a fines de 1842, volvió a estudiar matemáticas.

Había estado haciendo todo tipo de cosas, con distintos grados de éxito, después algunos intentos fueron nombrado profesor LUCASIANO DE MATEMÁTICAS en Cambridge, escribiendo un libro bastante influyente, "sobre la economía de la maquinaria y las manufacturas", trataba de dividir las tareas en fábricas (un tema surgido en relación al cálculo humano en operaciones matemáticas). En 1837 intervino con un tema popular de la teología natural, agregando su noveno tratado de bridgewater. La pregunta central era si hay evidencia de una deidad a partir del diseño aparente visto en la naturaleza, su libro Babbage es difícil de leer,; "Las nociones que adquirimos sobre la invención y el diseño surgen al comparar nuestras observaciones sobre las obras de otros seres con las intenciones de las que somos conscientes en nuestras propias empresas", alrededor de 150 años después, habla de la relación entre procesos mecánicos, las leyes naturales y el libre albedrío, realizando afirmaciones como "los cálculos de gran complejidad se pueden realizar por medios mecánicos", pero afirma con ejemplos que son bastante débiles que un motor mecánico puede producir secuencias de números que muestran cambios inesperados podría ocurrir con milagros. Después de tiempo Babbage probó suerte en la política, postulándose para el parlamento dos veces en una plataforma orientada a la fabricación, pero no pudo ser elegido debido a denuncias de uso indebido de fondos gubernamentales en Difference Engine, Babbage celebraba con fiestas de lujo en su casa grande y cada vez más desorganizada de Londres, atrayendo a luminarias como Charles Dickens, Charles Darwin, Florence Nightingale, Michael Faraday y el Duque de Wellington. a pesar de los grados y honores que enumeró su nombre tenía 6 líneas, estaba cada vez más amargado por su falta de reconocimiento. Babbage había contratado a un ingeniero de su época para que construyera el motor, pero en una década de trabajo, a pesar del desarrollo de máquinas, herramienta de precisión, el motor real no estaba terminado, poco después de conocer a Ada, Babbage había intentado frenar el proyecto, pero obtuvo como resultado el renunció del ingeniero a cargo e insistió en que podía conservar todos los planos de la máquina diferencial, incluso los propios de Babbage, dio un cambio por completo y decidió hacer una máquina que sólo calculara las diferencias, imaginó un "motor analítico". En 1826 había inventado la notación mecánica, que tenía la intención de proporcionar la representación simbólica del funcionamiento de la maquinaria, pero tuvo una decepción ya que la gente no apreciaba su invento. Después de un tiempo en Inglaterra, las ideas de Babbage adquirieron cierta popularidad en otros lugares, y en 1840 fue invitado a dar una conferencia sobre la Máquina Analítica en Turín, y recibió honores del gobierno italiano, nunca había



publicado una descripción sería del motor de diferencias ni del motor analítico, pero se decidió por fin hablar sobre la máquina analítica en Turín. En octubre de 1842, Menabrea publicó un artículo en francés basado en sus notas. Cuando Ada vio el periódico, decidió traducirlo al inglés y enviarlo a una publicación británica, después de años Babbage afirmó que le sugirió a Ada que escribiera su propio relato de la máquina analítica, ella había respondido que no se le había ocurrido esa idea, en febrero de 1843, Ada había resuelto hacer la traducción, pero agregó extensas notas propias a finales de 1843 de julio Ada había terminado de escribir sus notas estaba orgullosa de ella pero para Babbage los elogiaba quería agregar prefacio anónimo explicando que el gobierno británico no lo había apoyado con el proyecto Ada se negó a ayudarlo Babbage insistía bastante tanto que llegó a pensar que podrían retirar la nota de la publicación. Ada estaba entusiasmada con lo que había logrado lo compartió con familia y amigos su triunfo explicándoles que nadie puede estimar el problema y el trabajo interminable de tener que revisar la impresión de fórmulas matemáticas. En cuestión de días ya había chismes de la sociedad sobre la publicación de Ada, no le importaban las críticas y los chismes ya que ella estaba dedicada a convertirlo en algo real y no en un secreto, aunque no le importaba que fuera algo exagerado, sino que fuera sobrevalorado. Babbage le expresa su admiración por ella describiéndola como una hechicera de los números. Al concluir sus notas decidió dedicarse por completo a su hogar y a su familia, sin embargo, tuvo problemas de salud cáncer de cuello uterino los dolores no los toleraba ya, pasaron unos meses y a Ada se le complicaba más sus dolores, durante su ausencia llegó a sentirse cerca de la muerte finalmente terminaron sus sufrimientos a su edad de 36 años el día 27/Nov/1852. El funeral de Ada fue pequeño su madre, ni Babbage asistieron, a los 41 años William finalmente se volvió a casar su hijo mayor con quien tuvo muchas dificultades Ada se unió a la Marina varios años antes de que ella muriera, falleció a sus 26 años trabajando en un astillero en Inglaterra la hija de Ada se casó con un poeta un tanto salvaje pasó años en el Medio Oriente y se convirtió en el principal creador de caballos árabes, el hijo menor de edad es el heredero titular de la familia y pasó la mayor parte en la propiedad, su madre de Ada murió en 1960 todos hablaban de ella pareciendo que estaban en contra de ella. Babbage vivió 18 años más después de la muerte de Ana muriendo en 1871 intento trabajar en el motor analítico de nuevo en 1856 pero no hizo grandes progresos escribiendo artículos sobre las estadísticas de la casa de luz en 1864 publicó su autobiografía pasajes de la vida de un filósofo un documento extraño bastante amargo el capítulo sobre el mentor analítico se abre como una cita de un poema de Byron nunca se volvió a casar después de la muerte de su esposa y sus últimos años parecen haber sido solitarios. aparentemente le gustaba decir que con mucho gusto renunciaría al resto de su vida si pudiera pasar 3 días 500 años en el futuro, su cerebro se conservó y todavía se muestra en exhibición, aunque nunca terminó Babbage su difference Engine, una empresa sueca lo hizo y pasaron sus proyectos a su hijo.

Las computadoras mecánicas continuaron desarrollándose, dando paso gradualmente para la programación que comenzó a entenderse donde en la década de 1940, se redescubrió el trabajo de Babbage y Ada's Notes y fue un tal Bertram Bowden, quien "redescubrió" a Ada. Al investigar su libro de 1953 Más rápido que el pensamiento. El conocimiento acerca de Ada Lovelace y Charles Babbage comenzó a aumentar, y comenzaron a aparecer biografías, aunque a veces con errores espeluznantes "problema de los tres cuerpos" en una carta de Babbage se refiere al problema de los tres cuerpos en la mecánica celeste. A medida que aumentaba el interés por Babbage y Ada, también lo hacía la curiosidad por saber si el motor de diferencia realmente habría funcionado por esa razón se montó un proyecto y en 2002, después de un esfuerzo heroico, se construyó un Motor Diferencial completo, con solo una corrección en los planos. Sorprendentemente, la máquina funcionó. El contenido real de sus notas sobre el motor analítico explica que el motor de diferencia puede calcular valores de cualquier polinomio de sexto grado, pero el motor analítico es diferente porque puede realizar cualquier secuencia de operaciones. Ada analizó cómo funcionaría una secuencia de tipos específicos de cálculos en el motor analítico, con "Tarjetas de operación" que definen las ubicaciones de los valores. Ada habla de "ciclos", ahora conocidos como bucles y bucles anidados, dándoles una notación matemática. La idea de usar bucles para reducir la cantidad de tarjetas necesarias y el valor de reorganizar las operaciones para optimizar su ejecución en el motor analítico. Ada parece haber entendido con cierta claridad la visión tradicional de la programación: que diseñamos programas para hacer cosas que sabemos hacer. Lo que se ha convertido en la parte más famosa de lo que escribió Ada es el cálculo de los números de Bernoulli, en la Nota G.



Donde Ada tenía que especificar cómo calcular realmente esto en el motor analítico. El algoritmo que utilizó para el cálculo fue (en términos modernos):

En el motor analítico, la idea era tener una secuencia de operaciones (especificadas por "Tarjetas de operación") realizadas por el "Molino", con operandos provenientes de la "Tienda" (con direcciones especificadas por "Tarjetas variables"). Para calcular los números de Bernoulli de la forma en que Ada quería, se necesitan dos ciclos de operaciones anidados.

Los números de Bernoulli hace unos años, básicamente estábamos usando el mismo algoritmo que Ada, aunque ahora hay algoritmos un poco más rápidos que calculan efectivamente los numeradores de números de Bernoulli. El logro de Ada fue extraer de estos detalles una exposición clara del funcionamiento abstracto de la máquina, algo que Babbage nunca hizo. Es una forma bastante agradable de representar cómo funciona un sistema, similar en algunos aspectos a un diagrama de tiempos moderno, pero no exactamente igual. Y a lo largo de los años que Babbage trabajó en el motor analítico, sus notas. La "Notación Mecánica" es el volante que había impreso para la Gran Exposición de 1851, aparentemente una propuesta para la estandarización en dibujos de componentes mecánicos.

Charles Babbage era un hombre enérgico que tenía muchas ideas, algunas buenas. Ada Lovelace era una mujer inteligente que se hizo amiga de Babbage. Como un favor a Babbage, escribió una exposición de la máquina analítica y, al hacerlo, desarrolló una comprensión más abstracta de la misma que Babbage, y vislumbró la increíblemente poderosa idea de la computación universal. Y es esta idea de computación universal la que, por ejemplo, hace posible el software y la que lanzó toda la revolución informática en el siglo XX.

Gottfried Leibniz había ya tenía un concepto filosófico de algo así como volver cómputo universal en la década de 1600. Y el motor analítico de Babbage es el primer ejemplo explícito que conocemos de una máquina que habría sido capaz de realizar cálculos universales. Sin embargo, Babbage no lo pensó en estos términos. Pero en el esfuerzo por diseñar esto, terminó con una computadora universal.

Otra forma importante en que las computadoras modernas reducen el hardware es manejando números en binario en lugar de decimal.

Leibniz sabía de binario. Y si George Boole hubiera seguido su reunión con Babbage en la Gran Exposición, tal vez eso hubiera llevado a algo. La concepción principal de Babbage del motor analítico era como una máquina para producir automáticamente tablas matemáticas, ya sea imprimiéndolas por composición tipográfica o dándolas como tramas dibujándolas en una plancha. Imaginó que los humanos serían los principales usuarios de estas tablas, aunque pensó en la idea de tener bibliotecas de tarjetas precalculadas que proporcionarían versiones legibles por máquina.

Pero en la época de Babbage, con la idea de un motor analítico masivo, esta forma de hacer las cosas habría sido impensable. Entonces, ¿qué pasa con Ada? Por todo lo que puedo decir, ella hablaba con claridad y pensaba con claridad. Ella venía de las clases altas, pero no usaba ropa especialmente a la moda y se comportaba mucho menos como una condesa estereotipada que como una intelectual. De adulta, era emocionalmente bastante madura, probablemente más que Babbage, y parece haber tenido una buena comprensión práctica de las personas y el mundo.

Al igual que Babbage, era rica de forma independiente y no tenía necesidad de trabajar para ganarse la vida. Pero era ambiciosa y quería hacer algo por sí misma. Lo que dice suele ir vestido con agradables detalles victorianos, pero en el fondo, las ideas son claras y, a menudo, bastante contundentes.







---

---

**INSTITUTO TECNOLÓGICO DE IZTAPALAPA.**

**INGENIERÍA EN SISTEMAS COMPUTACIONALES**

**INFORME TÉCNICO DE RESIDENCIA  
PROFESIONAL.**

**TEMA:**

AUTOMATA CELULAR.

**LUGAR DE REALIZACIÓN:**

INSTITUTO TECNOLÓGICO DE IZTAPALAPA.

**PROFESOR:**

ING. ABIEL TOMAS PARRA HERNANDEZ.

**ALUMNOS:**

**PARTICIPACIÓN:**

RUIZ BARCO OSVALDO. 181080148	25%
REYES GÓMEZ JAVIER ROBERTO. 181080144	25%
CAMACHO CAPULIN ANGELICA BEATRIZ 181080421	25%
AVILEZ ARELLANO RAFAEL IVAN 181080381	25%





## INDICE

RESUMEN.....	3
Introducción.....	4
Objetivos .....	5
Justificación.....	6
Marco teórico.....	7
Metodología de trabajo.....	11
Desarrollo E Implementación .....	13
Resultados .....	16
Conclusiones.....	23
Referencias .....	24
Anexos .....	25





## RESUMEN.

El presente documento trata los conceptos fundamentales de autómatas celulares, como técnicas para modelar sistemas naturales que muestran un comportamiento complejo en función del tiempo, por medio de reglas lo mas simple y sencillas posibles.

En primer lugar, se define a los autómatas celulares "los cuales son sistemas dinámicos discretos que se definen de una manera sencilla y poseen una gran complejidad" y los conceptos asociados a estos.

Después de esto presentaremos el desarrollo en autómatas celular por John Conway, conocido como el juego de la vida" este es interesante porque es equivalente a una máquina universal de Turing, es decir, todo lo que se puede computar algorítmicamente se puede computar en el juego de la vida".

En segundo lugar, se presenta el modelo propuesto por el físico Per Bak y Kim Sneppen, conocido como el modelo de Bak-Sneppen de evolución biológica "es un modelo sencillo de coevolución entre especies interactuantes, la dinámica de modelo elimina repetidamente la especie menos adaptada y lo muta a él y a sus vecinos para recrear la interacción entre especies". Con este modelo se busca reproducir algunos resultados que el análisis de este modelo ha proporcionado, con el objetivo de obtener experiencia en el tratamiento de un fenómeno y la identificación de reglas que lo gobiernan.

### Palabras Clave

Autómatas celulares, modelado de sistema, simulación





## Introducción

Los autómatas celulares son sistemas dinámicos discretos que se definen de una manera sencilla y poseen una gran complejidad, aparte de muchas otras bondades. Como veremos, un autómata celular es esencialmente una regla de evolución temporal sobre un conjunto discreto, es decir, dado un punto de este conjunto, presentamos una regla que le asocia otro punto del conjunto. El estudio de sistemas cuya construcción es sencilla pero cuyo comportamiento resulta extremadamente complicado ha llevado a una nueva disciplina de estudio llamada teoría de sistemas complejos, en la cual los métodos computacionales juegan un papel central. Partiendo de algunos ejemplos sencillos, queremos presentar una breve introducción a esta disciplina.

Se representaran los conceptos fundamentales que permiten entender lo que son los autómatas celulares acompañados de algunos ejemplos de estos que nos ayudaran a la comprensión de estos, estará avanzado del trabajo de Wolfram y su trabajo nombrado el juego de la vida y podremos observar por que para muchos aficionados, el juego de la vida solo era un desafío de programación y una manera divertida de usar ciclos de la CPU.<sup>1</sup> Para otros, sin embargo, el juego adquirió más connotaciones filosóficas.

Se hablará del modelo Bak-Sneppen de evolución el cual es un modelo sencillo de coevolución entre especies interactuantes en el cual el algoritmo consiste en escoger la especie menos adaptada, y entonces reemplazarlo con sus dos vecinos más cercanos (entero anterior y próximo) por una especie nueva, con un factor de adaptación aleatorio nuevo. El cual a través de simulaciones podremos obtener resultados con el modelo de computación.





## Objetivos

### Objetivo General

Analizar de manera discreto modelos basados en autómatas celulares que permitan mejor su entendimiento

### Objetivos específicos

- 1.-Describir los procesos de generación de modelos, se presenta la división en modelos continuos y modelos discretos.
- 2.- Hacer una descripción de los autómatas celulares, revisando los conceptos fundamentales de los autómatas celulares.
- 3.- Dar ejemplos clásicos de autómatas celulares de Stephen wólfam teórico que ha estudiado a fondo a los autómatas celulares en una dimensión y del "juego de la vida" desarrollado por conway.
- 4.- Reproducir y analizar el modelo de evolución biológico de bak-Sneppen





## Justificación

### ¿Por qué usar autómatas celulares?

Los autómatas celulares han sido estudiados por un considerable número de investigadores en todo el mundo, existe al menos cuatro características que motivan al estudio de este

Los autómatas celulares pueden verse como poderosos motores de cómputo.

Como simuladores discretos de sistemas dinámicos.

Como los vehículos conceptuales para el estudio de la formación de patrones y la complejidad.

Como modelos originales de física fundamental.

Vistos como simuladores discretos de sistemas dinámicos, los autómatas celulares permiten la investigación sistemática de fenómenos complejos que contienen cualquier número de propiedades físicas deseables.

Los modelos generalmente a partir de autómatas celulares, los cuales han sido adecuadamente generalizados según el fenómeno que estudien, son usados para el estudio de crecimiento de formación de cristales de dendrita, patrones especiales generados por la reacción de los sistemas de difusión, auto – organización de redes neuronales y turbulencia de sistemas hidrodinámicos, estos últimos son capaces de reproducir el comportamiento continuo del sistema a gran escala.

Por otro lado, los autómatas celulares como modelos originales de física fundamental, permite el estudio de la física con una aproximación microscópica.

La importancia de esta clase de modelos radica, no en el hecho de construir una red de calibre similar a la teoría, si no en reproducir con éxito el fenómeno a pesar de nunca haber escuchado hablar de las ecuaciones que lo gobiernen, la esperanza es de establecer un conjunto abstracto de leyes de la microfísica que reproducen el comportamiento conocido en la macroescala; en este sentido autores como Fredkin sostiene que ha llegado al extremo de afirmar que el universo es, en su núcleo, un autómata celular.

Son estas ideas las que dan origen a pensar que sistemas como la evolución biológica puede ser estudiada desde el punto de vista de sistemas continuos y discretos, concretamente los autómatas celulares





## Marco teórico.

### MODELADO DE SISTEMAS

La historia del desarrollo de las ciencias naturales marca que el esfuerzo para entender el entorno comienza con la observación de fenómenos naturales, frecuentemente seguido de la clasificación de estos fenómenos, sobre todo siguiendo aspectos morfológicos. Pasando este análisis, se tiene el conocimiento sobre las componentes de un fenómeno, pero no se tiene conocimiento de las reglas que lo regulan.

Los sistemas interactúan con su entorno de una manera única, se conocen sus límites, puede formar parte de otros sistemas y pueden estar constituidos de sistemas, ejemplo de esto puede verse en un humano que es un sistema por sí solo, la economía de un país, etc. Para estudiar un sistema particular se hace necesaria una descripción de este, que no necesariamente será única dado que dependerá del punto de vista del cual se pretenda estudiar.

### Modelos Continuos

Los modelos generalmente empleados para la descripción de sistemas o fenómenos son los modelos matemáticos, los cuales conjuntan una serie de reglas lógicas que conforman el modelo.

En la física matemática, tanto clásica como cuántica, prevalece la noción del "continuo". Las ecuaciones diferenciales forman la base matemática para los modelos más usuales de sistemas naturales; las ecuaciones diferenciales ordinarias son convenientes para sistemas con un pequeño número de grados continuos de la libertad, que evoluciona en una manera continua [6].

Los modelos continuos hacen una descripción de evolución en el tiempo y cada uno de

estos modelos son expresados como una función continua. Estas ecuaciones que representan el modelo continuo de un fenómeno son eficientes para obtener información de un elemento del sistema en un instante de tiempo determinado, sin embargo, si la información que se quiere obtener no obedece a una sola componente sino a un conjunto de componentes o a todas las del sistema, los problemas para hallar la solución se hacen más complicados.

### Modelos Discretos

Los modelos de escala celular o modelos discretos consideran que el elemento (no el fenómeno en sí) juega un papel fundamental, dado que la dinámica de todo el fenómeno se observa en la simulación de efectos colectivos.





En general, los modelos matemáticos de sistemas naturales están basados usualmente en ecuaciones diferenciales que describen una ligera variación de un parámetro como una función de algunos otros. Los autómatas celulares, que son sistemas discretos, son una alternativa a fenómenos constituidos por un gran número de componentes idénticas. Los modelos basados en autómatas celulares son más apropiados en sistemas físicos con un régimen altamente no lineal, y en sistemas químicos y biológicos donde tienen lugar umbrales discretos.

## **SIMULACIÓN**

Para construir un modelo es necesario experimentar con el sistema, en cierto sentido decimos que el modelo almacena el entorno experimental que le ha dado origen.

La finalidad de un modelo es proporcionar información sobre el sistema. La simulación es el proceso de diseñar un modelo de un sistema real y llevar a término experiencias con él, con la finalidad de comprender el comportamiento del sistema o evaluar nuevas estrategias -dentro de los límites impuestos por un cierto criterio o un conjunto de ellos- para el funcionamiento del sistema.

## **Autómatas Celulares**

Un autómata celular (A.C.) es un modelo matemático y computacional para un sistema dinámico que evoluciona en pasos discretos. Es adecuado para modelar sistemas naturales que puedan ser descritos como una colección masiva de objetos simples que interactúen localmente unos con otros.

Los autómatas celulares (AC) surgen en la década de 1940 con John Von Neumann, que intentaba modelar una máquina que fuera capaz de autorreplicarse, llegando así a un modelo matemático de dicha máquina con reglas complicadas sobre una red rectangular. Inicialmente fueron interpretados como conjunto de células que crecían, se reproducían y morían a medida que pasaba el tiempo. Su nombre se debe a esta similitud con el crecimiento de las células.

Un autómata celular es un modelo matemático para un sistema dinámico compuesto por un conjunto de celdas o células que adquieren distintos estados o valores. Estos estados son alterados de un instante a otro en unidades de tiempo discreto, es decir, que se puede cuantificar con valores enteros a intervalos regulares. De esta manera este conjunto de células logra una evolución según una determinada expresión matemática, que es sensible a los estados de las células vecinas, y que se conoce como regla de transición local.

El aspecto que más caracteriza a los AC es su capacidad de lograr una serie de propiedades que surgen de la propia dinámica local a través del paso del tiempo y no desde un inicio, aplicándose a todo el sistema en general. Por lo tanto, no es fácil



analizar las propiedades globales de un AC desde su comienzo, complejo por naturaleza, si no es por medio de una simulación, partiendo de un estado o configuración inicial de células y cambiando en cada instante los estados de todas ellas de forma síncrona.



### Stephen Wolfram

Realizó un extenso análisis experimental de los patrones de crecimiento de los AC en una dimensión.

Para Wolfram, los autómatas celulares pueden ser usados como modelos matemáticos para sistemas físicos, biológicos y computacionales. Dado que son simples en la construcción, y potencialmente amigables con el análisis matemático preciso, son capaces de mostrar un comportamiento complejo. Define los AC como idealizaciones matemáticas simples de sistemas naturales, los cuales están constituidos de un arreglo de sitios discretos idénticos y cada sitio puede tomar un conjunto finito de estados.

### El juego de la vida

Conway desarrolló otro espacio celular siendo éste un avance dado que es uno de los espacios celulares más simple que exhibe cómputo y un constructor universal.

Aunque Conway construyó las reglas de su espacio celular de manera que debe permitir la interacción compleja, las estructuras que son posibles de generar en este espacio se han descubierto después de que las reglas del espacio se establecieron. En contraste, la mayoría de los patrones utilizados por von Neumann para la construcción de su autómata se han desarrollado al mismo tiempo que las normas que rigen sus espacios, las normas son complejas y parece claro que eran elaboradas específicamente para permitir ciertas estructuras.

La idea básica es comenzar con una configuración simple de organismos, distribuidos en una retícula, uno por célula, luego observar cómo estos se cambian cuando se aplican lo que Gardner llamó "las leyes genéticas de Conway" para nacimientos, muertes, y supervivencia.





## MODELO BAK-SNEPPEN DE EVOLUCIÓN.

El modelo de evolución de Bak-Sneppen, considera el proceso evolutivo grosso modo, no se presenta de manera explícita un paisaje adaptativo, como existen en otros modelos. Sin embargo, el modelo toma la idea de los paisajes adaptativos, imitando los efectos que este produce en las especies, en términos de una sola medida de aptitud, barreras óptimas. La aptitud de cada especie se ve afectada por otras especies con las que esté relacionada en el ecosistema, esto significa, que los movimientos adaptativos de una especie asociada co-evolutivamente a otra, afectan la aptitud y los grados adaptativos de sus socios co-evolutivos.

La estabilidad de cada especie se caracteriza por una barrera de cierta altura que separa su aptitud local máxima de otras máximas mejores. La altura de la barrera es la medida del número de fragmentos de código genético (genotipo) que debe ser cambiado. La mutación de fragmentos ocurre a menudo, pero modificaciones complicadas, como el desarrollo de alas para permitir que una criatura vuele, son poco probables que ocurran ya que involucran grandes movimientos evolutivos coordinados. La escala de tiempo para cada mutación es exponencial. Cuando la aptitud es alta, es difícil encontrar una mejor máxima cercana de tal forma que esos estados son relativamente estables. Cuando la aptitud es baja es más probable encontrar estados mejores cercanos, de modo tal que las barreras son bajas.

Por lo tanto, para cada especie, sólo se considera la barrera más pequeña.





## Metodología de trabajo

### Origen de los Automatas Celulares

Los autómatas celulares no son sino una instancia especializada de la temática general de teoría de autómatas, la diferencia radica en el hecho de que los autómatas son impulsados por señales de entrada y producen señales de salida. Los AC disfrutan de todas las simetrías, mayormente de traslación, inherente a su disposición cristalográfica, pero los usos de determinados estados vecinos para señales de entrada no se consideran en general para producir una salida.

### Definición Formal de un Autómata Celular

Una "lattice" o "retícula" es un arreglo uniforme, generalmente infinito, formado por objetos idénticos llamados "células". Este arreglo puede ser n-dimensional, pero para efectos de simulación de sistemas naturales se implementa de 1, 2 ó 3 dimensiones, de tamaño finito.

### El juego de la vida

La idea básica es comenzar con una configuración simple de organismos, distribuidos en una retícula, uno por célula, luego observar como estos se cambian cuando se aplican lo que Gardner llamó "las leyes genéticas de Conway" para nacimientos, muertes, y supervivencia. Conceptualmente el AC del juego de la vida, fue diseñado para capturar, de una manera simple, la reproducción y muerte de una población. GL está definido en una lattice 2-dimensional donde cada célula que conforma el lattice puede tomar uno de dos estados: viva o muerta (1 ó 0, respectivamente).

Los AC reversibles, los cuales presentan una evolución en el tiempo que puede ser compleja, pero se describe de manera precisa, estos AC tienen la característica de que se puede encontrar una función de evolución que permite obtener la evolución del AC original, pero hacia atrás en el tiempo.

Conway construyó las reglas de su espacio celular de manera que debe permitir la interacción compleja, las estructuras que son posibles de generar en este espacio se han descubierto después de que las reglas del espacio se establecieron. En contraste, la mayoría de los patrones utilizados por von Neumann para la construcción de su autómata se han desarrollado al mismo tiempo que las normas que rigen sus espacios, las normas son complejas y parece claro que eran elaboradas específicamente para permitir ciertas estructuras. La idea básica es comenzar con una configuración simple de organismos, distribuidos en una retícula, uno por célula, luego observar cómo estos se cambian cuando se aplican lo que Gardner llamó "las leyes genéticas de Conway" para nacimientos, muertes, y supervivencia.







El AC del juego de la vida (GL por sus siglas en inglés), fue diseñado para capturar, de una manera simple, la reproducción y muerte de una población. GL está definido en una lattice 2-dimensional donde cada célula que conforma el lattice puede tomar uno de dos estados: viva o muerta (1 ó 0, respectivamente). La función de transición para una célula toma en cuenta una vecindad de Moore descritas a continuación:

- **Sobrevive:** una célula en estado 1, tiene 2 o 3 vecinos en estado 1, entonces la célula sobrevive, es decir, continua en estado 1.
- **Nacimiento:** si una célula se encuentra en estado 0 y tiene exactamente 3 vecinos en estado 1, en el siguiente paso de tiempo la célula cambia a estado 1.
- **Muerte:** una célula en estado 1 muere por inanición si en su vecindad existe 1 o menos células en estado 1, ó muere por hacinamiento si existen 4 o más células en estado 1 en su vecindad. Deacuerdo a la definición 2.2, el juego de la vida es un AC 2-dimensional.



## Desarrollo E Implementación

Los ACs consisten en un sistema discreto de rejillas que mantiene algunos elementos (células idénticas) iniciales, donde su evolución se basa en la información local. Estos contienen: 1) un espacio  $d$ -dimensional que es dividido por rejillas idénticas, b) un conjunto finito de posibles estados, y c) una regla de transición que define el estado de la célula al tiempo  $t + 1$ , el cual depende de su estado anterior y de su vecindad. En este apartado se expone la formulación formal de esta herramienta y los aspectos generales de El juego de la vida.

### Formulación formal

El AC es un sistema dinámico discreto (en espacio y tiempo). Este se define como sigue de acuerdo con:

Sea  $s: Z \times N \rightarrow \{0, 1\}$  La función satisface la ecuación

$$(\forall i \in Z) (\forall t \in Z) s(i, t+1) = f(s(i-r, t), s(i-r+1, t), \dots, s(i+r, t))$$

$$(\forall i \in Z) (\forall t \in Z) s(i, t+1) = f(s(i-r, t), s(i-r+1, t), \dots, s(i+r, t))$$

Tal que

$$(\forall i \in Z) s(i, 0) = s_0(i), (\forall i \in Z) s(i, 0) = s_0(i),$$

Donde  $N$  es el conjunto de enteros no negativos,  $Z$  es el conjunto de todos los enteros, y  $s_0: Z \rightarrow \{0, 1\}$  una función dada que especifica la condición inicial, para un sistema de una-dimensión. Para  $d$ -dimensiones el AC podría ser definido de forma similar. El mapeo  $f: \{0, 1\}^{2r+1} \rightarrow \{0, 1\}$  determina el dinamismo. Esto refiriéndose a la regla local del AC. El entero positivo  $r$  es el rango -o el radio- de la regla. La función  $S: t \mapsto s(i, t)$  es el estado del AC al tiempo  $t$ .  $S: \{0, 1\}^Z$  es el estado del espacio. Un elemento del estado del espacio se le llama configuración. A partir del estado al tiempo  $t + 1$  es enteramente determinado por el estado al tiempo  $t$  y la regla  $f$ ,  $f$  induce el mapeo  $f: S \rightarrow S$ , llamada regla global -u operador de evolución- tal que,

$$s_{t+1} = f(S_t) \quad s_{t+1} = f(S_t)$$

Dada la regla  $f$ , que está limitada al conjunto  $A_f$  está definida por

$$A_f = \lim_{t \rightarrow \infty} f^t(S) = \bigcup_{t \geq 0} f^t(S) \quad A_f = \lim_{t \rightarrow \infty} f^t(S) = \bigcup_{t \geq 0} f^t(S)$$

Donde, para cualquier  $t \in N$ ,  $f_{t+1} = f \circ f^t$  con  $f^t = f^t|_{A_f}$  es claramente invariante, esto es,  $f(A_f) = A_f$ . Desde que cualquier  $f$ -subconjuntos invariantes que pertenecen a  $A_f$ , el límite del conjunto de  $f$  es el máximo  $f$ - del subconjunto invariante de  $S$ .

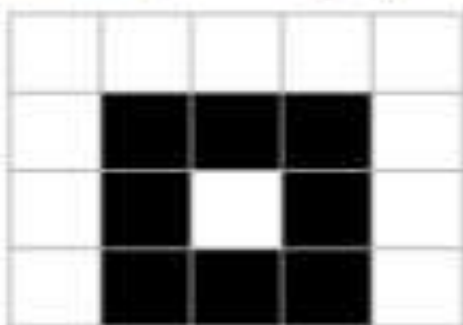


## El juego de la vida

En esta investigación se emplea el más famoso diseño de ACs: El juego de la vida, de Conway. Para este juego se tiene:

1) un espacio de 2 dimensiones, 2) los estados son de ceros (0, muerte de la célula) y unos (1, nacimiento o vida de la célula), y 3) la regla de transición que genera un patrón que indica cuando una célula cambia de estado o permanece en el mismo.

La primera consideración pertinente en El juego de la vida es la influencia que existe entre las células, es decir, el criterio que se mantiene para especificar un tipo de vecindad. Para este caso, se considera el tipo de vecindad Moore, en este son ocho vecinos (de color negro) quienes ejercen influencia sobre la célula central.

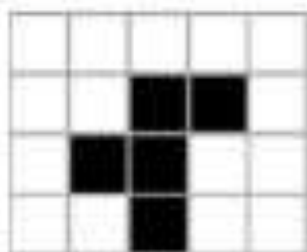


Una segunda consideración es la estructura de las células iniciales que, en este caso, es una poliforma de cinco células. En *El juego de la vida* se emplea una estructura en forma de F (que se mantiene en movimiento, *glider*) que junto con el tipo de vecindad Moore se aplican las siguientes reglas:

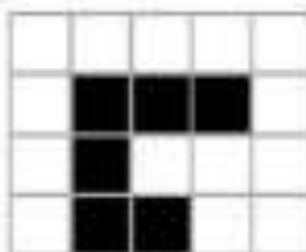
1. Si hay tres células vecinas vivas para una célula cero, entonces nace una célula (de 0 a 1).
2. Si hay entre una y tres células vecinas vivas para una célula uno, entonces permanece viva la célula (se mantiene en 1).
3. Si hay más de tres células vecinas vivas para una célula uno, entonces muere la célula (de 1 a 0).
4. Si hay una célula vecina viva para una célula cero, entonces se mantienen muerta (se mantienen en 0).

Se muestran tres columnas con rejillas que contienen tanto células vivas (en negro) como células muertas (en blanco). en la parte superior de la primera columna se presentan las células iniciales (forma de F) o la generación cero, mientras que en la parte inferior se visualizan cuales células cambiarán de estado (por ejemplo, las que indican un tres y un tono gris oscuro son aquellas células que nacerán, y las que indican un seis y un tono de gris claro son aquellas que morirán) en la próxima

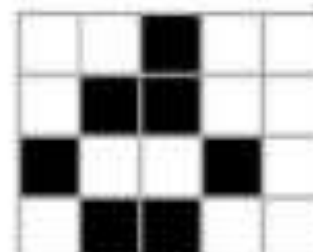
generación. En la segunda columna se representan las células de la primera generación (aplicando la vecindad Moore y las reglas), en este punto ya no es posible seguir omitiendo una tercera consideración: el tipo de frontera, pues en este juego se emplean fronteras periódicas que significa una conexión entre los bordes (o fronteras) laterales, y el superior con el inferior (por ejemplo, la célula que indican un cuatro morirá en la siguiente generación debido a la influencia de la célula situada en el extremo del otro borde, es decir, por influencia de la célula situada en el extremo izquierdo); en la tercera columna se muestra el resultado de los criterios anteriores que llevan a la segunda generación.



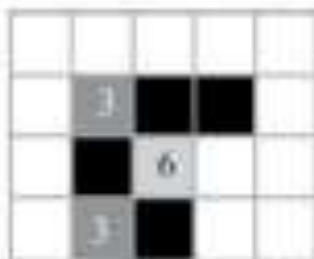
Celulas iniciales



Primera generación



Segunda generación



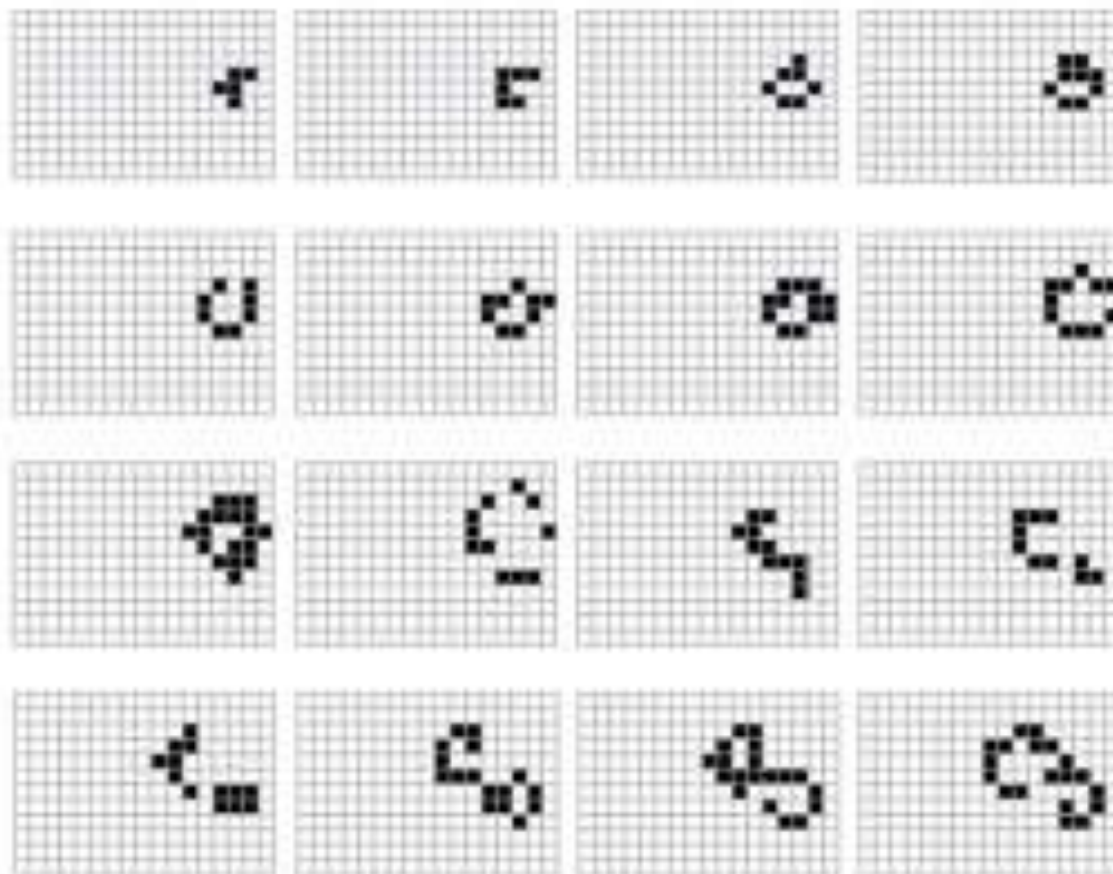
Para la primera generación



Para la segunda generación

## Resultados

Se muestra *El juego de la vida* y las primeras 30 generaciones. En el extremo izquierdo superior se ubica la rejilla que contiene las células iniciales, la primera generación se encuentra a la derecha y así, sucesivamente. Mientras que la cuarta generación se localiza por debajo de la generación cero y se continúa con el criterio anterior. En esencia, se muestra el comportamiento dinámico de las estructuras en espacio (rejilla) y tiempo (30 generaciones) discreto. Para el caso de la última generación, el número de células vivas se ha quintuplicado y ha recorrido casi todo el espacio.







Av. Telecomunicaciones S/N, Col. Chinampac de Juárez, C.P.  
09208, Alcaldía de Iztapalapa, Ciudad de México Tel. 5773-8210, e-mail:  
[division@iztapalapa.tecnm.mx](mailto:division@iztapalapa.tecnm.mx)  
[www.tecnm.mx](http://www.tecnm.mx) | [www.iztapalapa.](http://www.iztapalapa.)



```
private static final int COLS_NUMBER = 45;
private int stationary;
private boolean leftButtonPressed;
private boolean change;
private boolean click;
private boolean isStandalone;
private Vivarium vivarium;
private JLabel[][] area;
private JPanel jPanel1;
private JPanel jPanel2;
private JPanel jPanel3;
private JPanel jPanel4;
private GridLayout GridLayout1;
private JButton nextGenButton;
private JButton aboutButton;
private JButton eraseButton;
private JCheckBox ballCheckbox;
private JLabel stationaryLabel;
```



```
private void actualizar() {
    try {
        String sql = "UPDATE usuarios SET nombre = ?, correo = ?, password = ? WHERE id_usuario = ?";
        PreparedStatement pstmt = con.prepareStatement(sql);
        pstmt.setString(1, txtNombre.getText());
        pstmt.setString(2, txtCorreo.getText());
        pstmt.setString(3, txtPassword.getText());
        pstmt.setInt(4, Integer.parseInt(txtID.getText()));
        int rows = pstmt.executeUpdate();
        if (rows > 0) {
            JOptionPane.showMessageDialog(this, "Usuario actualizado correctamente");
        } else {
            JOptionPane.showMessageDialog(this, "Error al actualizar usuario");
        }
    } catch (SQLException e) {
        JOptionPane.showMessageDialog(this, "Error al actualizar usuario: " + e.getMessage());
    }
}

private void btnEliminarActionPerformed() {
    try {
        String sql = "DELETE FROM usuarios WHERE id_usuario = ?";
        PreparedStatement pstmt = con.prepareStatement(sql);
        pstmt.setInt(1, Integer.parseInt(txtID.getText()));
        int rows = pstmt.executeUpdate();
        if (rows > 0) {
            JOptionPane.showMessageDialog(this, "Usuario eliminado correctamente");
        } else {
            JOptionPane.showMessageDialog(this, "Error al eliminar usuario");
        }
    } catch (SQLException e) {
        JOptionPane.showMessageDialog(this, "Error al eliminar usuario: " + e.getMessage());
    }
}
```

```
package tecnm_izt;

public class VistaInicio extends javax.swing.JFrame {

    private Usuario[] usuarios;
    private Usuario usuario;
    private int id_usuario;
    private int id_usuario_nuevo;

    public VistaInicio() {
        initComponents();
    }

    private void initComponents() {
        // ...
    }

    private void btnRegistrarActionPerformed() {
        // ...
    }

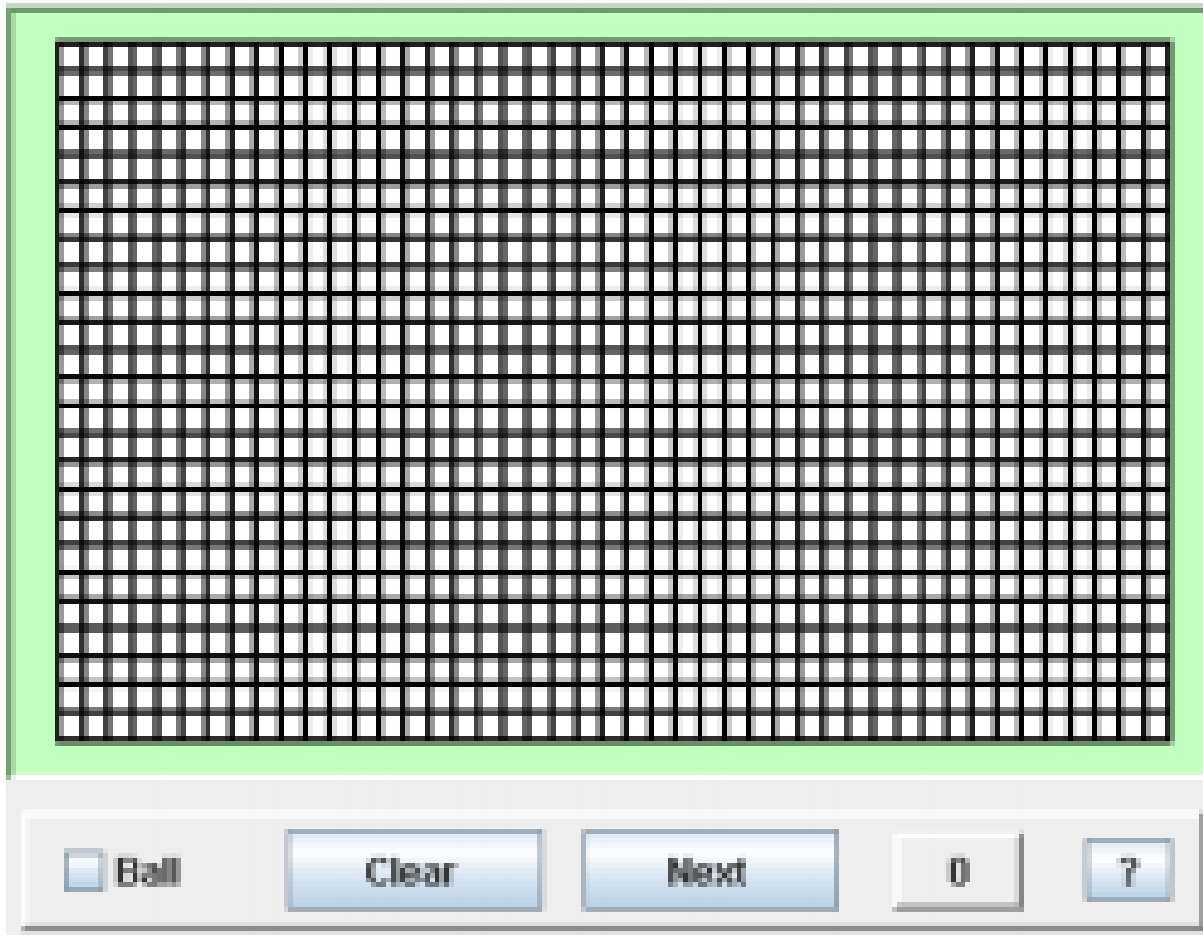
    private void btnActualizarActionPerformed() {
        // ...
    }

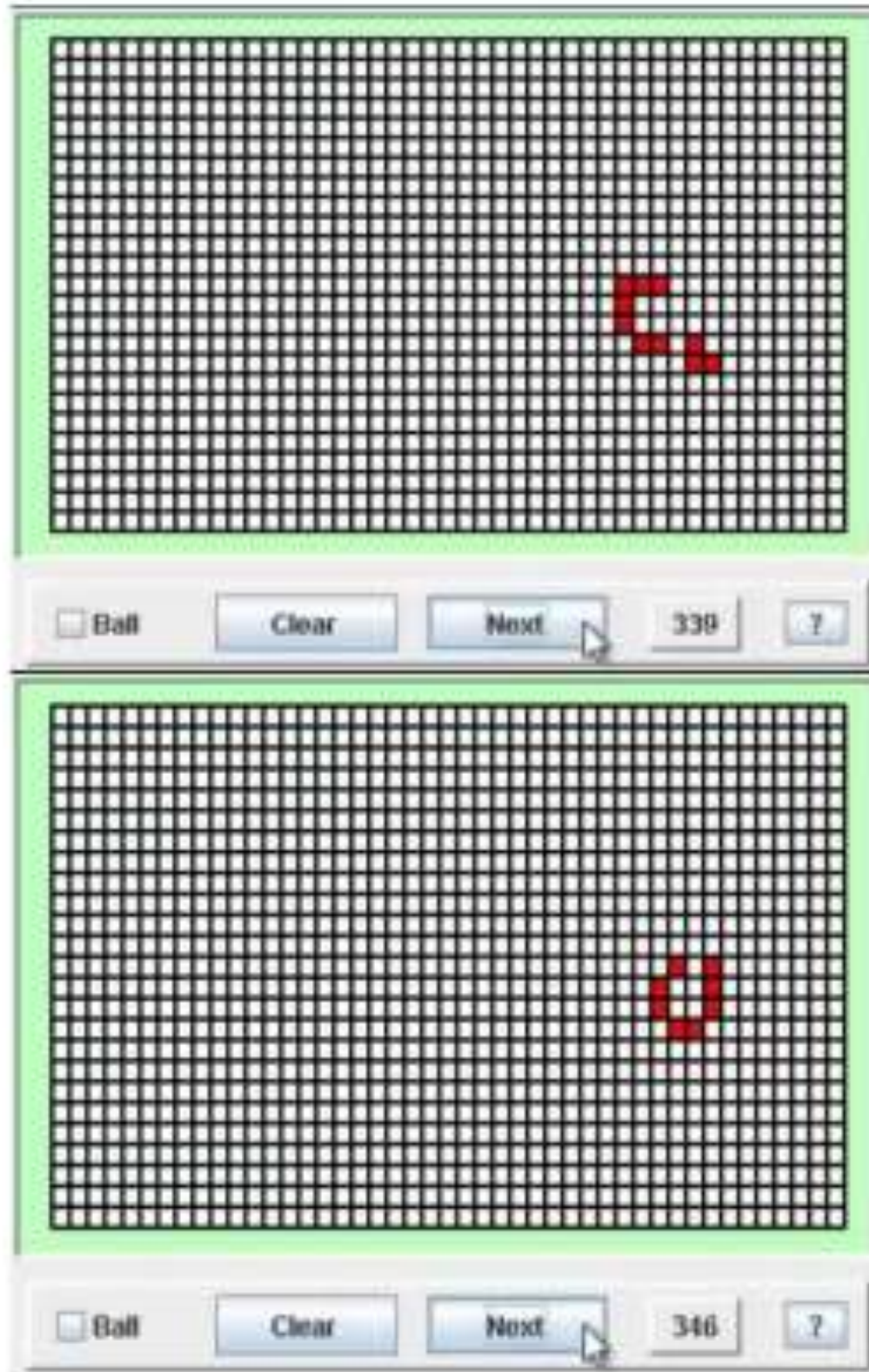
    private void btnEliminarActionPerformed() {
        // ...
    }

    private void btnSalirActionPerformed() {
        // ...
    }
}
```



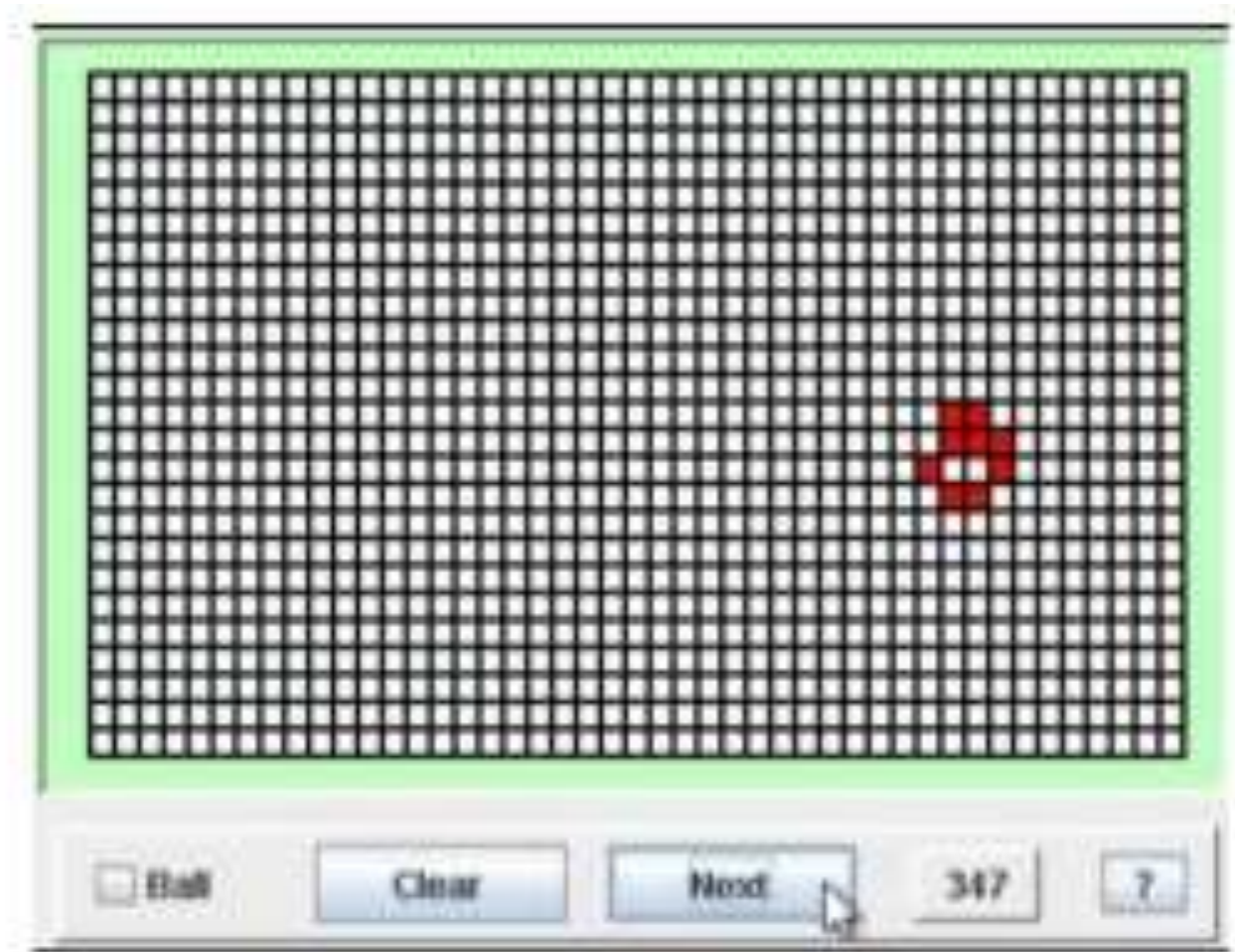






Av. Telecomunicaciones S/N, Col. Chinampac de Juárez, C.P.  
09208, Alcaldía de Iztapalapa, Ciudad de México Tel. 5773-8210, e-mail:  
[division@iztapalapa.tecnm.mx](mailto:division@iztapalapa.tecnm.mx)  
[www.tecnm.mx](http://www.tecnm.mx) | [www.iztapalapa.mx](http://www.iztapalapa.mx)





## Conclusiones

Es notable la gran capacidad que tienen para simular fenómenos naturales, ayudando a una interpretación más completa de los resultados de una investigación o incluso favoreciendo la prevención de situaciones no deseadas. También contribuye en las actividades humanas, al lograr que muchas operaciones matemáticas y de cómputo sean más rápidas y permitan un manejo de datos superior a otros modelos abstractos.







## Referencias

autor, s. (10 de noviembre de 2019). wikipedia. Obtenido de Modelo de Bak-Sneppen:

[https://es.wikipedia.org/wiki/Modelo\\_de\\_Bak%E2%80%93Sneppen](https://es.wikipedia.org/wiki/Modelo_de_Bak%E2%80%93Sneppen)

autor, s. (19 de Diciembre de 2020). wikipedia. Obtenido de Autómata celular:

[https://es.wikipedia.org/wiki/Aut%C3%B3mata\\_celular](https://es.wikipedia.org/wiki/Aut%C3%B3mata_celular)

autor, s. (7 de junio de 2021). wikipedia. Obtenido de Cellular automaton:

[https://en.wikipedia.org/wiki/Cellular\\_automaton](https://en.wikipedia.org/wiki/Cellular_automaton)

autor, s. (5 de junio de 2021). wikipedia. Obtenido de Juego de la vida:

[https://es.wikipedia.org/wiki/Juego\\_de\\_la\\_vida](https://es.wikipedia.org/wiki/Juego_de_la_vida)

Caparrini, F. S. (30 de Octubre de 2016). Fernando Sancho Caparrini. Obtenido de Autómatas Celulares: <http://www.cs.us.es/~fsancho/?e=86>

TRUJILLO, I. I. (Diciembre de 2009). pdf. Obtenido de SIMULACIÓN DE SISTEMAS NATURALES USANDO AUTÓMATAS CELULARES: <https://www.repositoriodigital.ipn.mx/bitstream/123456789/5696/1/Tesis12181.pdf>







## Anexos

**AUTÓMATA CELULAR:** Sistema dinámico que evoluciona a través de lapsos de tiempo discretos.

**AVALANCHA:** Término empleado en la teoría de la criticalidad auto - organizada que se refiere a la actividad generada por una perturbación a un sistema que se encuentra en estado de aparente equilibrio.

**CASO DE USO:** Representación de un elemento funcional en el diseño de software, usado en los diagramas de casos de uso.

**CÉLULA:** Para este caso, es una máquina de estados numerables, que conforman una retícula.

**CLASE:** Una clase es una abstracción de la realidad la cual define atributos y métodos a los objetos que de ella deriven. **DENDRITA:** Concreción mineral que en forma de ramas de árbol suele presentarse en las fisuras y juntas de las rocas.

**ESTRÉS:** Tensión provocada por situaciones agobiantes que originan reacciones psicosomáticas o trastornos psicológicos a veces graves.

**FASE:** Indica la situación instantánea en dos procesos periódicos, de una magnitud que varía cíclicamente.

**LATTICE:** También llamada retícula, define el espacio de acción para el AC.

**MEDIA:** También llamada media aritmética, es una medida estadística como medida de centralización.

**NICHO ECOLÓGICO:** Término que describe la posición relacional de una especie o población en un ecosistema o el espacio concreto que ocupa en el ecosistema.

**SISTEMA CONTINUO:** Se dice que un sistema es continuo, si dados dos puntos en el tiempo cercanos uno de otro, existen una infinidad de configuraciones de la evolución del sistema.

**STASIS:** palabra utilizada en la teoría del equilibrio puntuado, hace referencia en español a estasis, que significa detención, dilatación.

**VARIANZA:** Medida de dispersión de los valores de una variable aleatoria alrededor de la media.

**VECINDAD:** Conjunto de células, las cuales generan una influencia sobre la célula a evaluar.



**EXPOSICIONES (LIGAS).**

**(RESUMEN)** <https://youtu.be/eGW8VACOPmk>

**(MODELOS)** <https://youtu.be/lvTfkxsJnm4>



Av. Telecomunicaciones S/N, Col. Chinampac de Juárez, C.P.  
09208, Alcaldía de Iztapalapa, Ciudad de México Tel. 5773-8210, e-mail:  
[division@iztapalapa.tecnm.mx](mailto:division@iztapalapa.tecnm.mx)  
[www.tecnm.mx](http://www.tecnm.mx) | [www.iztapalapa.mx](http://www.iztapalapa.mx)

