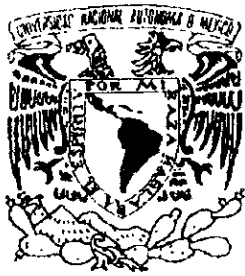


33
2e.



UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO

FACULTAD DE ESTUDIOS SUPERIORES
"ZARAGOZA"

MODELACIÓN DE LA MORFOLOGÍA
DE ORGANISMOS VEGETALES MEDIANTE
EL USO DE SISTEMAS LINDENMAYER.

T E S I S

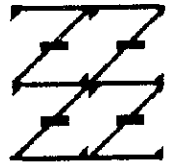
QUE PARA OBTENER EL TITULO DE

B I Ó L O G O

PRESENTA

YOLANDA RODRÍGUEZ PAGAZA

U N A M
FES
ZARAGOZA



LO HUMANO EJ
DE NUESTRA REFLEXIÓN

DIRECTOR: M. EN C. ARMANDO CERVANTES SANDOVAL

ASESOR: M. EN C. PATRICIA RIVERA GARCÍA

MÉXICO, D F

JUNIO 1998

2633 11

TESIS CON
FALLA DE ORIGEN



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

AGRADECIMIENTOS

A mis padres:

YOLANDA PAGAZA GUERRERO

Por creer en mí y por luchar por mí aún muchos años antes de que me tuoiera en su vientre

RODRIGO RODRIGUEZ ROMERO

Por guiarme con su ejemplo y hacerme sentir orgullosa de él y de mí.

A mis asesores:

M. EN C. ARMANDO CERUANTES S.

M. EN C. PATRICIA RIVERA G.

Gracias por su apoyo, dirección, amistad y paciencia

A mis sinodales:

DR. FRANCISCO MONROY A.

M. EN C. AMARDO BARBA A.

M. EN C. ALEJANDRINA ÁVILA O.

Gracias por sus comentarios que enriquecieron este trabajo

A mis maestros:

A todos y cada uno de los que he conocido a lo largo de mi vida

Gracias por haberme dado el valioso tesoro de su amistad, su confianza, sus consejos y enseñanzas.

Gracias porque de todos aprendí más de lo que se aprende en el aula.

A Fundación UNAM por la beca otorgada para la realización de esta tesis

EN PAZ

*Muy cerca de mi ocaso, yo te bendigo vida,
porque nunca me diste ni esperanza fallida
ni trabajos injustos ni pena inmerecida;*

*porque veo al final de mi rudo camino
que yo fui el arquitecto de mi propio destino;
que si extraje la hiel o la miel de las cosas,
fue porque en ellas puse hiel o mieles sabrosas;
cuando planté rosales coseché siempre rosas.*

*...Cierio, a mi lozanía va a seguir el invierno
¡Mas tú no dijiste que mayo fuese eterno!*

*Hallé sin duda largas las noches de mis penas;
mas no me prometiste solo noches buenas,
y en cambio tuve algunas santamente serenas...*

*Amé, fui amado, el sol acarició mi faz.
¡Vida, nada me debes! ¡Vida, estamos en paz!*

AMADO NERVO

RESUMEN

Los sistemas Lindenmayer son una herramienta lógico matemática, que permite generar modelos del crecimiento de organismos vegetales, la cual se ha venido utilizando y desarrollando desde los años 60's en diversos países, como Estados Unidos, Alemania, Canadá y Japón. En México no se han encontrado trabajos que utilicen este tipo de herramienta, a pesar de que se ha demostrado que es útil para estudios en áreas como botánica, agronomía y paleontología, en otros países del mundo.

Para responder al por qué no se utiliza esta herramienta, se plantearon dos posibles causas: a) que existe poca información al respecto, o b) que no se cuenta con la infraestructura necesaria para usar el software especializado o para crearlo.

Para resolver estos planteamientos, se buscó bibliografía respecto al crecimiento vegetal y los factores que influyen en este, para verificar si existían patrones de desarrollo constantes y predecibles, y si pueden ser modelados con los sistemas Lindenmayer, para lo cual, se revisaron los fundamentos de la herramienta y de los modelos en general. También se crearon cuatro programas desarrollados con el lenguaje de programación Visual Basic 4.0, los cuales fueron "Alga" en dos versiones, "Planta", y "Pruebagra". El primer programa, en sus dos versiones modela el crecimiento de una alga roja (*Callithamnion roseum*) reportada en la bibliografía. El segundo programa "Planta" muestra el crecimiento de una planta hipotética reportada por Chien y Jürgensen, y el último programa, "Pruebagra", que muestra, con ayuda de los sistemas Lindenmayer, la descripción de la forma de un árbol de ciprés (*Cupressus lindleyi* Klotzsch) y una planta de hinojo (*Foeniculum vulgare* Will).

Con la información bibliográfica referente a los factores que influyen en el crecimiento de los vegetales, la existencia de caracteres constantes y predecibles como la filotaxia, la ramificación, la floración y la fructificación; y la fundamentación de los sistemas Lindenmayer, así como la creación de los cuatro programas, se llegó a la conclusión de que esta herramienta tiene un alto potencial de uso, y su nula utilización se debe a la poca información de su aplicación en el terreno biológico, así como a la dificultad de conseguir el *hardware* y *software* en nuestro país, lo que se puede solucionar creando software propios con la infraestructura disponible. Por otra parte, se sentaron las bases teóricas para la comprensión de los fundamentos matemáticos y computacionales de la herramienta y su uso en la investigación biológica.

INDICE DE CONTENIDO

	PAG.
RESUMEN	i
INTRODUCCIÓN	1
CAPITULO I : PATRONES DE DESARROLLO EN VEGETALES	4
1.1 Importancia de los patrones de desarrollo en vegetales.	4
1.2 Partes de las plantas donde se han encontrado patrones de crecimiento.	7
1.2.1 Disposición foliar o filotaxia.	7
1.2.2 Ramificación.	11
1.2.3 Flores y frutos	16
1.3 Factores que afectan los patrones de crecimiento	19
CAPITULO II: REPRESENTACIÓN DE LOS PATRONES DE DESARROLLO MEDIANTE UN MODELO: LOS SISTEMAS LINDENMAYER.	20
2.1 Modelos.	20
2.1.1 Definición de modelo y conceptos asociados.	20
2.1.2 Porqué y para qué se utilizan los modelos.	21
2.1.3 Validez de los modelos	22
2.1.4 Tipos de modelos.	22
2.2 Los lenguajes Lindenmayer como herramienta para modelar los patrones de crecimiento de las plantas.	23
2.2.1 Lenguajes formales	24
2.2.1.1 Gramáticas de tipo 0.	27
2.2.1.2 Gramáticas de tipo 1	27
2.2.1.3 Gramáticas de tipo 2.	28
2.2.1.4 Gramáticas de tipo 3.	28
2.2.2 Diferencias entre lenguajes formales y sistemas Lindenmayer.	29
2.2.3 Tipos de sistemas Lindenmayer.	30
2.2.3.1 Sistemas 0L.	31
2.2.3.2 Sistemas 0L determinísticos (sistemas-D0L).	32
2.2.3.3 Sistemas-T0L	33
2.2.3.4 Sistemas-0L estocásticos.	34
2.2.3.5 Sistemas-IL	35
2.2.3.6 Sistemas 0L y IL paramétricos	36

CAPITULO III : MODELACIÓN DE LOS PATRONES DE DESARROLLO DE LAS PLANTAS UTILIZANDO LA COMPUTADORA.	41
3.1. Historia del enlace entre los sistemas Lindenmayer y la computación.	42
3.2. Tipos de hardware y software utilizados y potencialmente utilizables en los sistemas Lindenmayer.	45
3.2.1 Hardware	45
3.2.2 Software	46
3.2.2.1 Lenguajes de programación.	46
3.2.2.1.1 LOGO.	46
3.2.2.1.2 BASIC.	47
3.2.2.1.3 FORTRAN	47
3.2.2.1.4 Lenguaje de programación C.	48
3.2.2.2 Software especializado.	48
3.2.2.2.1 Pfg. (Plant and fractal generator)	48
3.2.2.2.2 Spiral (Modeling program for phyllotactic patterns.	48
3.2.2.2.3 Ise (Interactive surface editor)	49
3.2.2.2.4 Mapl (Modeling program for cellular structures).	49
3.2.2.2.5 Panel (Control panel manager)	49
3.2.2.2.6 Rayshade (Ray tracer)	49
3.2.2.2.7 CELIA (Celular Linear Interactive Array simulator)	49
3.2.2.2.8 AMAP	49
3.3. Ejemplos	50
CAPITULO IV: MÉTODO	52
4.1 Revisión de bibliografía.	52
4.2 Elaboración de programas que modelen el crecimiento de los vegetales utilizando los sistemas Lindenmayer y la computadora.	54
4.2.1 Determinación del tipo de programas a elaborar.	55
4.2.2 Elaboración de los programas.	55
4.3 Descripción de la elaboración de programas.	56
4.3.1 Primera versión del programa <i>Alga: Alga1</i> .	56
4.3.2 Segunda versión del programa <i>Alga: Alga2</i> .	60
4.3.3 Programa <i>Planta</i> .	65
4.3.4 Programa <i>Pruebagra</i>	66
4.4 Creación de la versión ejecutable de los programas elaborados.	67
4.5 Observación y reconocimiento de patrones de desarrollo en plantas.	68
CAPITULO V: RESULTADOS	70
5.1 Material bibliográfico.	70
5.2 Elaboración de programas que modelen a los organismos vegetales.	73
5.2.1 Primera versión del programa <i>Alga: Alga1</i> .	73
5.2.2 Segunda versión de <i>Alga: Alga2</i>	74
5.2.3 Programa <i>Planta</i> .	76

5.2.4 Programa Pruebagra.	79
5.3 Observación y reconocimiento de patrones de desarrollo en plantas.	80
5.3.1 Árbol de ciprés (<i>Cupressus lindleyi</i> Klotzsch).	80
5.3.2 Planta de hinojo (<i>Foeniculum vulgare</i> Will).	85
CAPITULO VI: DISCUSIÓN	90
CAPITULO VII : CONCLUSIONES	97
BIBLIOGRAFÍA	99
DIRECCIONES EN INTERNET	103
ANEXO I: CÓDIGO DEL PROGRAMA “ALGA 1”	I
ANEXO II: CÓDIGO DEL PROGRAMA “ALGA 2”	IV
Codigo para el botón “Mostrar cadena de caracteres”.	IV
Código para el botón “Graficar”	IV
ANEXO III: CÓDIGO DEL PROGRAMA “PLANTA”	VI
ANEXO IV. CÓDIGO DEL PROGRAMA PRUEBAGRA	X
ANEXO V: GRAMÁTICA QUE DESCRIBE AL ÁRBOL DE CIPRÉS (<i>Cupressus lindleyi</i> Klotzsch).	XI
ANEXO VI: GRAMÁTICA DE LA PLANTA DE HINOJO (<i>Foeniculum vulgare</i> Will).	XIX

INDICE DE FIGURAS Y TABLAS

	PAG.
Figura 1. Ejemplos de disposición foliar en ramas.	8
Figura 2. Ejemplo de una filotaxia 5/13	9
Figura 3. Ejemplo de una filotaxia 3/8.	10
Figura 4. Ejemplos de dicotomías y pseudodicotomías.	12
Figura 5. Ejemplo de crecimiento monopodial y simpodial.	12
Figura 6. Ejemplos de crecimiento en troncos.	13
Cuadro 1. Clasificación de los tipos de ramificación.	14
Cuadro 2. Tipos de inflorescencias.	16
Figura 7. Ejemplo de paracladía	18
Figura 8. Flor de girasol mostrando los tipos de parásitos.	18
Figura 9. Clasificación de modelos y simulaciones.	23
Figura 10. Tres estados transitorios de una célula y sus reglas.	32
Figura 11. Ejemplo de desarrollo en <i>Anabaena catenula</i>	33
Figura 12. Tablas de decisión para la floración de una planta dependiendo de las horas-luz.	34
Figura 13. Ejemplo de la acción de hormonas en el crecimiento de <i>Anabaena catenula</i> .	38
Figura 14. Interpretación de una cadena de símbolos.	43
Figura 15. Ramificación en un árbol representado por la cadena $F[+F][-F[-F]F][+F][-F]$	43
Figura 16. Representación de los tres vectores propuestos por Albelson y diSessa para representar a los sistemas Lindenmayer en tres dimensiones.	44
Figura 17. Estructura tridimensional de un arbusto.	44
Figura 18. Aspecto del paquete llamado Ise.	51
Figura 19. Gramática del crecimiento del alga roja <i>Callithamnion roseum</i> .	56
Figura 20. Crecimiento del alga <i>Callithamnion roseum</i> .	57
Figura 21. Crecimiento del alga <i>Callithamnion roseum</i> en dos dimensiones	57
Figura 22. Interface de usuario para el programa Alga1	57
Figura 23. Crecimiento del alga <i>Callithamnion roseum</i> en dos dimensiones con células numeradas.	58
Cuadro 3. Arreglo para el crecimiento del alga <i>Callithamnion roseum</i> .	59
Figura 24. Cadena de caracteres para cada etapa del crecimiento del alga <i>Callithamnion roseum</i> .	60
Figura 25. Ejemplo de graficación utilizando las bases propuestas por Prusinkiewicz.	61
Figura 26. Interface de usuario para el programa Alga2 en la creación de cadenas.	62
Figura 27. Interface de usuario para la graficación del alga.	62
Figura 28. Diseño de la parte "Mostrar cadena de caracteres" en el programa Alga2.	64
Figura 29. Diseño del programa Planta.	65
Figura 30. Diseño del programa Pruebagra	67
Figura 31. Interface de usuario del programa Alga1	73
Figura 32. Modelación del desarrollo del alga <i>Callithamnion roseum</i> creada con el programa Alga1	74
Figura 33. Interface del usuario del programa Alga2.	75
Figura 34. Modelación del desarrollo del alga <i>Callithamnion roseum</i> creada con el programa Alga2	75
Figura 35. Interface del usuario del programa Planta.	76
Figura 36. Simulación de una planta artificial creada por Chien.	77
Figura 37. Simulación del crecimiento de una planta artificial con el programa Planta.	78
Figura 38. Interface de usuario del programa Pruebagra.	79
Figura 39. Foto del árbol de ciprés.	80

Figura 40. Ramificación secundaria del árbol de ciprés.	81
Figura 41. Ramificación terciaria del árbol de ciprés.	81
Figura 42. última ramificación del árbol de ciprés.	81
Figura 43. Comparación entre la última ramificación del árbol y su gramática.	82
Figura 44. Comparación entre la ramificación secundaria del árbol de ciprés y la gramática creada	84
Figura 45. Comparación entre el árbol gramatizado y el real.	85
Figura 46. Planta de hinojo.	85
Figura 47. Ramificación de la planta de hinojo.	86
Figura 48. Ramificación secundaria en la planta de hinojo	86
Figura 49. Ramificación terciaria en el tallo de hinojo.	86
Figura 50. Comparación entre la planta de hinojo real y gramatizada.	87
Figura 51. Inflorescencia del hinojo.	88
Figura 52. Detalle de la inflorescencia del hinojo.	88
Figura 53. Esquema del esqueleto de la inflorescencia del hinojo.	88
Figura 54. Inflorescencia de hinojo generada con el programa Pruebagra.	89
Figura I. Interface de usuario usada en el programa Alga1	III
Figura II. Interface del usuario utilizada en el programa Alga2	V
Figura III. Interface del usuario utilizada en el programa Planta	IX
Figura IV. Interface de usuario utilizado en el programa Pruebagra	X

INTRODUCCIÓN

El hombre ha estudiado a las plantas desde que inició su cultivo, realizando observaciones cuantitativas y cualitativas de ellas (Weier, 1992:4) La ciencia encargada de estudiar vegetales recibe el nombre de Botánica (Fuller, 1974:1-2) y de ésta han surgido ramas que se han expandido para conocer de manera más profunda el comportamiento y las características de las plantas.

Una de estas ciencias, la **morfología**, que estudia la forma de las plantas, la cual sirvió como base para desarrollar una teoría que describe el desarrollo de estas mediante un lenguaje formal. los *Sistemas Lindenmayer*, nombre que recibe en honor a su autor, el biólogo húngaro Aristid Lindenmayer (Mahoney, 1993 40; Prusinkiewicz y Hanan, 1989:3-85; Valdés, 1991:46).

Estos sistemas, y haciendo uso de la computadora, han dado como resultado imágenes sorprendentemente semejantes a las plantas reales (Reeting, 1993:15; Mahoney, 1993:40, Prusinkiewicz y Hanan, 1989:3-85). Además, han mostrado su utilidad y tienen un amplio potencial de investigación, por que modelan el crecimiento de raíces, plantas herbáceas, árboles y enredaderas, considerando la acción de ciertas sustancias en el suelo, factores climáticos y ecológicos que determinan y predicen el crecimiento de las plantas. Sin embargo, después de una intensa búsqueda bibliográfica, no se encontraron trabajos desarrollados en México, lo que nos hizo suponer que las posibles razones por las que no se utilizan en México son: que se tiene poca información respecto a los sistemas Lindenmayer, o que en México no se cuenta con la infraestructura necesaria para usar el software especializado o para crearlo

El presente trabajo tuvo como objetivo general analizar los sistemas Lindenmayer como herramienta para modelar la morfología de organismos vegetales. De este objetivo general se desprendieron los siguientes objetivos particulares:

1. Explicar los fundamentos de los sistemas Lindenmayer en la modelación de organismos vegetales;
2. Describir algunos de los patrones morfológicos más comunes en los organismos vegetales, los factores que los afectan así como su relación con los sistemas Lindenmayer;
3. Comparar los modelos generados usando a los sistemas Lindenmayer contra las plantas reales para determinar la similitud, utilidad y validez de esta herramienta en la investigación biológica; y

4. Mostrar algunos modelos de formas vegetales en computadora utilizando los sistemas Lindenmayer para determinar la accesibilidad a estos con la infraestructura disponible.

Para cubrir estos objetivos, se realizó una búsqueda bibliográfica para establecer y desarrollar el marco teórico y referencial del trabajo, así como para conocer y analizar los fundamentos teóricos de los sistemas Lindenmayer. Por un lado, se encontró información referente a los factores que afectan el desarrollo de los vegetales, y como estos son tomados en cuenta al generar modelos con los sistemas Lindenmayer. También se encontraron los elementos necesarios para determinar que los sistemas-L son válidos y lo que se puede esperar de ellos.

Dentro de la información de cómputo relevante, se encontraron los principios que Prusinkiewicz y Hanan (1989:6) usaron para elaborar los primeros programas que mostraban el desarrollo de vegetales usando a los sistemas Lindenmayer, como la utilización de una fórmula trigonométrica y la notación utilizando los símbolos +, -, [, y], para mostrar las ramificaciones a la derecha y a la izquierda del eje principal. Con estos elementos, se elaboraron 4 programas que modelan el crecimiento de los vegetales utilizando los sistemas Lindenmayer y la computadora, para determinar que tan fácil es crear un programa utilizando la infraestructura disponible, y por último se observaron y reconocieron patrones de desarrollo en plantas, para determinar que tipo de vegetales se pueden modelar.

A partir del trabajo realizado, el capitulado del trabajo es el siguiente:

En el capítulo I llamado *patrones de desarrollo en vegetales*, se describen algunos de estos que han mostrado ser constantes y predecibles, tales como la filotaxia, la ramificación y los tipos de flores y frutos que existen y se encuentran dentro de una clasificación. También se revisan los factores externos e internos que afectan el desarrollo de una planta y su forma.

El capítulo II, *representación de los patrones de desarrollo mediante un modelo: los sistemas Lindenmayer* introduce al lector al conocimiento de que es un modelo, para que se usa, que se puede esperar de él y como saber si un modelo es válido. después, se revisan los diferentes tipos de modelos que existen y se ubica a los sistemas Lindenmayer dentro de estos tipos de modelos. Posteriormente, se revisan los fundamentos de los sistemas-L: la teoría de los lenguajes formales, desarrollada por Noam Chomsky a finales de los años cincuenta. Se establecen algunas diferencias entre los lenguajes formales y los sistemas Lindenmayer para

entrar de lleno a la definición de estos y sus diferentes tipos, así como algunos ejemplos de en que circunstancias se utilizan.

En el capítulo III, *modelación de los patrones de desarrollo de las plantas usando la computadora*, describe como los sistemas Lindenmayer pasaron de ser una herramienta teórica a utilizar la computadora para optimar su aplicación. Dentro de este capítulo se abordan los fundamentos gráficos que se utilizaron en los primeros programas que mostraban el desarrollo de los vegetales usando a los sistemas Lindenmayer via computadora, en dos y tres dimensiones. También se analiza el hardware y software que se utiliza actualmente y que se puede utilizar. Por último, se dan algunos ejemplos de como se están utilizando los sistemas Lindenmayer en la investigación biológica a nivel internacional.

El capítulo IV muestra el método que se siguió para realizar este trabajo, así como para elaborar los programas que modelan el crecimiento de un organismo vegetal obtenido de la bibliografía via computadora.

En los capitulos V, VI y VII se dan los resultados, la discusión y las conclusiones de este trabajo

CAPITULO I : PATRONES DE DESARROLLO EN VEGETALES

1.1 Importancia de los patrones de desarrollo en vegetales.

Desde que el hombre comenzó a cultivar plantas se interesó en sus características y en acumular información respecto a ellas. En las primeras clasificaciones que hizo de las plantas tomó en cuenta sus usos, su cultivo, sus propiedades medicinales, alimenticias o alucinógenas, como lo indican descubrimientos arqueológicos y estudios en las sociedades aborígenes de diversas partes del mundo, como en Egipto y África (Fuller, 1974:1-2). Clasificar las plantas responde a la necesidad que el hombre tiene de ordenarlas haciendo más fácil su entendimiento y eliminando el aprendizaje repetido¹ (Backer y Allen, 1970:454).

La primer persona reportada en clasificar a las plantas fue el griego Teofrasto (370-285 a. de C.), quien tuvo en cuenta su forma, su periodo de vida y su habitación². A Cesalpino (1583 d. C.) se le reconoce como el primero que generó un sistema botánico de clasificación, aunque se basó en la forma de crecimiento de las plantas tal como Teofrasto lo había propuesto, dividiéndolas primordialmente en árboles, arbustos, matas y hierbas. Otras personas que trataron de modificar este sistema de clasificación fueron Rivio (1690 d. C.) y Tournefort (1694 d. C.), pero no lograron tener el éxito de el sueco Carlus Von Linneo, quien en 1753 publicó su trabajo llamado *Species Plantarum*, donde agrupa a las plantas según sus semejanzas estructurales. Linneo se dio cuenta, al igual que sus antecesores, que la aparentemente infinita variabilidad entre los seres vivientes no era casual y que los organismos estaban distribuidos en grupos y tipos distintos. A cada tipo le dio el nombre de *especie*³, y esta se convirtió en la unidad elemental de su sistema de clasificación⁴ que se sigue utilizando hasta nuestros días. El hecho de que entre especies emparentadas exista una semejanza, se atribuye a la evolución, la selección natural y la reproducción exitosa de las especies, según lo afirman Jones (1988:87), Koleff (1992:26) y Weier, Stonick y Barbour (1992:4)

¹ El término de aprendizaje repetido se refiere a que con una clasificación existente del reino vegetal, se evita el que una misma planta se estudie dos o más veces.

² El término habitación hace referencia al lugar donde normalmente viven las plantas, pudiendo ser este un pantano, sobre las rocas o cerca de un río.

³ Una especie es una categoría para definir a los seres vivos y que abarca a un conjunto de individuos que tienen caracteres comunes y son fecundos entre sí.

⁴ Como siguiente unidad, después de especie, Linneo utilizó el término *género* para designar a un grupo de especies similares en muchos aspectos y por consiguiente presumió que estaban más relacionadas entre sí que entre otras especies. Posteriormente, reunió estos géneros en grupos mayores llamados *órdenes*, y estos últimos en grupos aún más grandes llamados *clases*.

Para diferenciar unas especies de otras, Linneo utilizó características estructurales de las plantas, las cuales se denominan *caracteres*. Jones (1988:66) los define como los rasgos o atributos de una planta que pueden ser medidos, cuantificados, descritos o evaluados de alguna manera. Linneo utilizó dos grandes grupos de caracteres: los reproductivos (que involucran a los tejidos asociados a la reproducción); y los vegetativos, que forman todas las demás estructuras del cuerpo vegetal. A este tipo de caracteres se les conoce como *caracteres morfológicos*⁵. Este tipo de caracteres son los más accesibles y los más empleados en el trabajo de clasificar y determinar organismos. A la rama de la ciencia que hace este trabajo se le conoce como *Taxonomía*⁶.

Una planta posee diferentes caracteres morfológicos, como la forma de ramificación, la disposición de las hojas en el tallo, y determinada estructura de reproducción. Estos caracteres son los que le dan una forma definida al organismo, y van variando dependiendo de la etapa de desarrollo en que se encuentre. Por ejemplo: una planta de maíz (*Zea mays*) es diferente en su forma y tamaño cuando está germinando que cuando fructifica, porque tienen diferentes caracteres morfológicos (la segunda tiene determinada flor, determinado fruto y determinada altura a diferencia de la primera). A la forma en general que adquiere una planta en cierta etapa de su desarrollo se le llamará *forma arquitectural*, y a las formas arquitecturales por las que pasa a lo largo de su vida se les llamará *patrón de desarrollo*.

Muchos de los patrones de desarrollo de las plantas son claramente precisos, geométricos y predecibles, tal como lo afirma Bell (1991:312). Si estos patrones de crecimiento pueden ser predecibles, se puede desarrollar una herramienta o modelo⁷, que los simule y permita observar el desarrollo de una planta a futuro. Pero, para modelar el crecimiento de una planta, un modelo no solo se puede basar en la unión de los

⁵ Existen otros tipos de caracteres empleados en la taxonomía para clasificar las plantas, como los caracteres *citológicos*, *fisiológicos* y *químicos*. Los caracteres *citológicos* son, por ejemplo, el número de cromosomas en una especie. Pero especies diferentes como el hombre, el erizo y el murciélago pueden tener el mismo número de cromosomas, por lo que este tipo de caracteres no es confiable. Los caracteres *fisiológicos* hacen referencia a la posible fecundidad de un híbrido, ya que la mayor parte de estos, si provienen de especies distintas son estériles, pero existen sus excepciones, como en el caso de la cruce de conejo y liebre, o perro y chacal, etc. Los caracteres *químicos* hacen referencia al protoplasma de la especie, cuya composición acarrea la de todos sus humores y todas sus secreciones. Por esta causa existen hemoglobinas, clorofilas, taninos, etc. Por ejemplo, el cardo corredor tiene el riesgo de ser clasificado como un cardo, por su apariencia, pero si se mastica su tallo tiene sabor a zanahoria; en efecto, el cardo corredor es una umbelífera. Este tipo de caracteres son potencialmente utilizables pero necesitan de análisis químicos complejos. (Larousse, 1989: 40-41)

⁶ George Ball define a la taxonomía como el “estudio y aplicación de los principios de la clasificación a los organismos y nominación de los taxa reconocidos[. . .]”. Esta rama debe diferenciarse de la sistemática, la cual realiza sus estudios a nivel poblacional ocupándose de las relaciones filogenéticas entre los organismos.

⁷ Un *modelo*, según Shannon (1975:4) es una representación, de cualquier tipo, de un objeto, sistema o idea en otra forma distinta a la que se presenta. Una *modelación* es la creación de un modelo.

caracteres morfológicos de una planta, sino en la interacción que hay entre estos y en como se relaciona todo el organismo con el medio que le rodea. La rama de la ciencia que estudia estos aspectos recibe el nombre de *Morfología*⁸.

Existen varios factores que son determinantes para que una planta tenga una cierta estructura. La herencia es una de ellas (factor genético). Otra es la capacidad de que su estructura resista las condiciones ambientales (factores anatómicos, morfológicos y ambientales), y por último, la forma en que responde a la competencia por luz, agua, nutrimentos o a los depredadores (factor ecológico)

Conociendo los factores que afectan el crecimiento de la planta, se puede crear un modelo que incluya como variables los factores genéticos, anatómicos y ambientales de los cuales depende la forma que tome en su desarrollo. Esta herramienta sería de gran utilidad en el estudio de las plantas, ya que, aparte de comprobar si realmente una planta sigue un determinado patrón de crecimiento en diferentes circunstancias (en diversos climas, por ejemplo) lo cual ayudaría al taxónomo, también ayudaría a otros profesionistas en su área de estudio como al biólogo y al botánico, prediciendo el desarrollo de una planta dependiendo de la competencia, la simbiosis y la acción de ciertas fitohormonas; al agrónomo, prediciendo el desarrollo de la planta dependiendo de los fertilizantes o tipos de suelo; y al paleontólogo, extrapolando el desarrollo a plantas ya extintas apoyándose en el registro fósil. Esta herramienta ahorraría tiempo, al poder predecir el desarrollo de la planta en horas; dinero, ya que muchos estudios requieren salidas continuas a campo; y por lo tanto, esfuerzo.

⁸ Lawrence da una definición de lo que es la morfología: "La forma y estructura de un organismo como su forma fisiológica, etc./el estudio de la forma y estructura" siendo esta definición muy general. Otra definición es la dada por Parker: "Rama de la biología que estudia la forma y estructura de un organismo y sus estadios en la vida del organismo". Sin embargo, la dada por Font es la más completa, ya que especifica lo que es la morfología desde el punto de vista botánico. El la define como "[...] la ciencia de la forma. La morfología botánica estudia la forma de las plantas. Se divide en *morfología general*, que también comprende la organografía, o descripción de la forma de los diversos órganos vegetales; *morfología experimental*, que se sirve de la experimentación para investigar el origen de las formas vegetales; y la *anatomía vegetal*, que atiende a la estructura microscópica de los órganos". Esta definición, a diferencia de las anteriores, establece las ramas en las que se divide la morfología botánica, y con ello se puede establecer una diferencia entre morfología y morfogénesis, que es otro elemento asociado. Robbins define a esta ciencia como "[...] el origen y desarrollo de la forma de las plantas desde la relativa sencillez de la célula inicial, el cigoto, hasta alcanzar la complejidad de la planta madura [...]", con lo cual no se establece una clara diferencia entre morfología y morfogénesis. Lo mismo sucede con la definición dada por Font de morfogénesis: "Conjunto de fenómenos relativos a la plasmación de la forma de un organismo, así como a la evolución de los caracteres morfológicos". Lawrence da otra definición: "El desarrollo de la forma y estructura; origen y desarrollo de órganos o partes de organismos". En esta definición ya se notan diferencias, como el nivel de organización al que se trabaja la morfogénesis, ya que la morfología estudia a todo el organismo y la morfogénesis los órganos de este. Con la definición dada por Parker se puede establecer la relación que guarda la morfogénesis con la morfología, ya que de acuerdo a la definición de morfología dada por Font, la morfogénesis sería parte de la morfología utilizando como herramientas a la anatomía vegetal y a la morfología experimental. (Robbins, 1976: 301; Font, 1977:735; Lawrence, 1989:326; y Parker, 1985:234)

Una herramienta de este tipo han desarrollado investigadores como Aristid Lindenmayer y Przemyslaw Prusinkiewicz desde los años sesenta, utilizando la simulación por computadora. Esta simulación se compone de un modelo matemático basado en las gramáticas Chomsky⁹ y software desarrollado en lenguajes de programación, como BASIC, C y C++, entre otros

El principal software, desarrollado por Przemyslaw Prusinkiewicz y James Hanan, recibe el nombre de CELIA¹⁰, y se ha utilizado para simular el crecimiento de diferentes especies de plantas, algas y bacterias considerando factores genéticos, ambientales y ecológicos. Desgraciadamente, no se han encontrado trabajos semejantes en México, lo cual hace suponer que:

- 1) Existe poca información al respecto, o
- 2) No se cuenta con la infraestructura necesaria para usar el software o para crearlo.

Este trabajo tiene como meta el desglosar el funcionamiento de la herramienta creada por Aristid Lindenmayer y Przemyslaw Prusinkiewicz, recalcando tanto sus bases biológicas como su potencial y limitaciones para aplicarlos en la investigación biológica en México; así como comprobar la accesibilidad a esta herramienta utilizando un lenguaje de programación y computadoras con procesamiento simple.

1.2 Partes de las plantas donde se han encontrado patrones de crecimiento.

Para modelar la forma arquitectural que tiene un organismo vegetal, es necesario conocer si posee un patrón de desarrollo en el organismo completo o en alguno de sus órganos, como puede ser en las hojas, flores o ramas. Además, es preciso conocer si estos patrones están influidos por diferentes factores y cuales son estos, para tenerlos como variables de las cuales va a depender el modelo.

Existen muchos caracteres morfológicos. Dentro de los más comunes, conspicuos y predecibles se encuentran la disposición foliar o filotaxia, la ramificación, la floración y la fructificación, por lo que se verán en este apartado sus formas y diferentes clasificaciones.

1.2.1 Disposición foliar o filotaxia.

La *filotaxia* o el patrón de disposición foliar de la planta es la secuencia en que se originan las hojas en el tallo así como su disposición a lo largo de los ejes de crecimiento. La filotaxia de una planta es usualmente

⁹ Las gramáticas Chomsky se verán con mayor detenimiento en el capítulo 2 de este trabajo.

¹⁰ Siglas en inglés de Celular Linear Interactive Array Simulator (Para más información véase a Johnson, 1996:118, Retting, 1993:15; y Robertson, 1989:73)

constante. De hecho, la filotaxia se utiliza como un carácter importante cuando se determina a una planta, y una filotaxia particular caracteriza a un género de plantas dado, o a un grupo de estas. Por ejemplo, en las monocotiledóneas, generalmente se presenta una hoja por nodo, mientras que en las dicotiledóneas se presenta más de una. La posición de las hojas en una planta afecta la interceptación de luz y además, fija la posición de los subsecuentes botones axilares, por lo que la filotaxia puede determinar los patrones de ramificación, sobre todo en plantas perennes. La filotaxia es uno de los patrones de crecimiento que se pueden predecir en una planta, de allí que pueda diseñarse un modelo que la represente a lo largo del crecimiento de una planta.

Las filotaxias se clasifican¹¹ generalmente en tres grupos de la siguiente manera:

- I. **Una hoja por nodo (alternas).** Existen varios tipos de arreglo de estas hojas en el tallo.
 - A. **Monósticas.** La hoja se desarrolla siempre del mismo lado del tallo, por lo que forma una fila visto desde arriba (Fig. 1 a).
 - B. **Dísticas.** Las hojas forman dos filas en el tallo vistas desde arriba y el ángulo entre hojas consecutivas es de 180° (Fig. 1 b).
 - C. **Trísticas.** Las hojas forman tres filas en el tallo vistas desde arriba y el ángulo entre hojas consecutivas es de 120° (Fig. 1 c).
 - D. **Espirales.** Se dice que son espirales si vistas desde arriba forman más de tres filas. A las filas que forman también se les conoce como *ortósticos*
- II **Dos hojas por nodo (opuestas).** En este patrón, las hojas forman cuatro ortósticos, y un par de hojas guarda un ángulo de 90° con el subsecuente par de hojas (Fig. 1 d).
- III. **Tres o más hojas por nodo.** También reciben el nombre de verticiladas (Fig. 1 e).

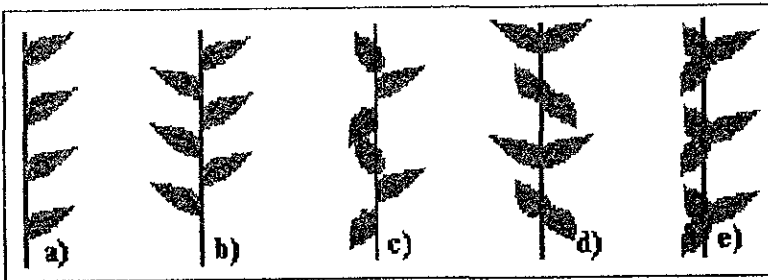


Figura 1. Ejemplos de disposición foliar en ramas.
a) monósticas b) Dísticas c) Trísticas d) Opuestas. e) Verticiladas.

¹¹ La mayor parte de los autores (Greulanch y Adams, 1990:425 y Jones, 1988:222) clasifican a las filotaxias en a) *Alternas*, con una hoja por nodo; b) *Opuestas*, con dos hojas por nodo; y c) *Verticiladas*, con tres o más hojas por nodo. En este trabajo se reporta la clasificación utilizada por Bell, quien las clasifica en función de las hojas por nodo. Aunque es básicamente la misma clasificación, este autor es más explícito en los tipos de hojas alternas que existen. (Bell, 1991:218-220)

Cuando la filotaxia es de tipo espiral (una hoja por nodo) se puede designar como un quebrado. Para determinar el tipo de filotaxia que sigue una planta, se cuentan las hojas que hay que recorrer alrededor de este para dar un número de vueltas exacto, es decir, hasta encontrar otra hoja paralela y en la misma posición que la hoja de la que se está partiendo. Si por ejemplo, se necesitan 2 hojas para dar una vuelta completa, la filotaxia se representa como $1/2$, que es el caso de las hojas alternas disticas (fig1 b); si se necesitan 13 hojas para dar 5 vueltas completas, como muestra la figura 2, se designa como $5/13$, donde:

Vueltas alrededor del tallo \longrightarrow $5/13$ \longleftarrow Hojas (o nodos) necesarios.

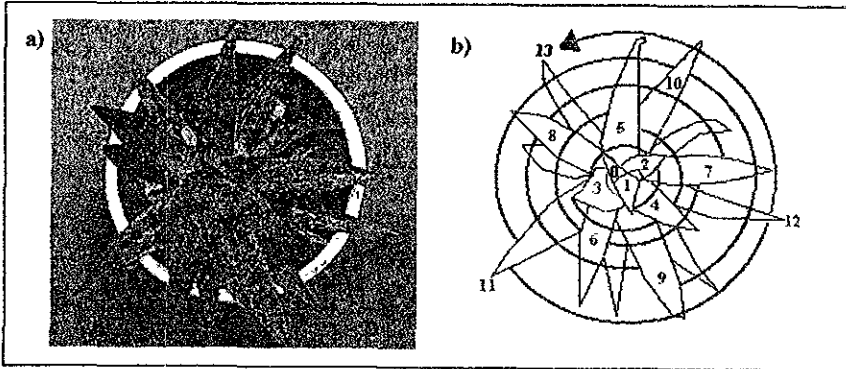


Figura 2. Ejemplo de una filotaxia $5/13$

a) Foto de una Agavácea vista desde arriba. Para dar 5 vueltas completas se necesitan recorrer 13 hojas como se muestra en b). Su filotaxia es de $5/13$, y el ángulo entre hojas es de 138.46° . Obsérvese que la hoja 0 y la 13 son paralelas y están en la misma posición.

El ángulo entre 2 hojas adyacentes se determina de la siguiente manera:

$$\begin{array}{c}
 \text{Filotaxia representada en un quebrado} \\
 \downarrow \qquad \qquad \qquad \downarrow \\
 1/2 \times 360^\circ = 180^\circ \qquad \text{ó} \qquad 5/13 \times 360^\circ = 135^\circ \\
 \uparrow \qquad \qquad \qquad \uparrow \\
 \text{Ángulo entre hojas adyacentes}
 \end{array}$$

Otro ejemplo de una filotaxia se puede observar en la figura 3.

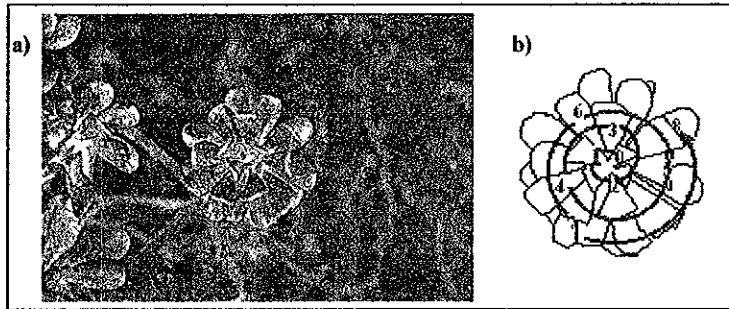


Figura 3. Ejemplo de una filotaxia 3/8.
 a) Foto de siempreviva (*Sedum paldaeatum* D.C) vista desde arriba. Para dar tres vueltas completas alrededor del tallo se necesitan recorrer 8 hojas como lo muestra la figura b) Su filotaxia es de 3/8 y el ángulo de separación entre ellas es de 135°. Obsérvese que la hoja 0 y la hoja 8 son paralelas y están en la misma posición.

Respecto a la filotaxia, Bell (1990:220-223), Greulich y Adams (1990:425) y Prusinkiewicz y Hanan (1989 26), explican que existe un número determinado de filotaxias de acuerdo a la serie:

$$\frac{1}{2}, \frac{1}{3}, \frac{2}{5}, \frac{3}{8}, \frac{5}{13}, \frac{8}{21}, \frac{13}{34} \dots$$

donde, tanto numerador como denominador siguen la serie de Fibonacci¹², en la cual el número siguiente es la suma de los dos anteriores (Por ejemplo $1+1=2$, $1+2=3$, $2+3=5$, $3+5=8$, etc.). Estas series pueden seguir indefinidamente, pero los ángulos formados hasta ahora se acercan sin sobrepasar los $137^{\circ} 30' 28''$. Una hipótesis sostiene que si el ángulo entre hojas fuera exactamente de $137^{\circ} 30' 28''$, ninguna hoja se superpondría, captando mejor la luz del sol.

La filotaxia no siempre es tan clara y predecible Esta puede cambiar en algunas plantas debido a factores ambientales, y se pueden presentar dos o más filotaxias en la misma planta, siendo confuso el arreglo de hojas en la porción de tallo que queda entre las dos series filotácticas.

Steeves y Sussex (1991:114), afirman que los estudios sobre filotaxia son puramente descriptivos y proveen poca información sobre los mecanismos que operan para designar el acomodo de las hojas en el tallo. Los estudios realizados hasta ahora revelan que este acomodo no es azaroso. El primer primordio que se genera en el ápice del tallo regula la distancia que habrá entre hojas posteriores, por lo que su tamaño y posición son

¹²La serie de Fibonacci (llamada así por el apodo de Filius Bonacci de Leonardo de Pisa, quién la descubrió en 1202) es una sucesión de números donde, cada número nuevo se origina de los dos anteriores. Además de la botánica, a esta serie de números también se le halla en otras áreas como matemáticas, biología y música. (Para más información de esta serie, su historia, y su relación con otras áreas, consulte a Newman y Boles, 1992:168-195)

importantes, según explica Lyndon¹³. Se cree que el patrón de filotaxia se puede deber a la interacción de unidades de crecimiento, con promotores o inhibidores de desarrollo como las auxinas, y que pueden ser susceptibles de modificación

Hasta aquí se ha visto que la filotaxia es un patrón de crecimiento que siguen las hojas al rededor del tallo en las plantas. La mayor parte de las filotaxias están determinadas por la genética de la planta, aunque también dependen del ambiente en algunos casos como asegura Bell (1991:312).

Al igual que las hojas, las ramas alrededor del tronco en las plantas siguen un patrón que puede ser predecible y constante, como se verá en el apartado siguiente

1.2.2 Ramificación.

La ramificación también tiene patrones característicos. Generalmente se le clasifica en tres grupos

1. **Dicotómica.** Cuando el tallo se bifurca en dos ramas más o menos iguales (Fig. 4 a y b), las cuales siguen creciendo y se pueden dividir a su vez en dos. Cuando una de las ramas es abortada o forma una estructura temporal como una inflorescencia se llama *pseudodicotomía* (Fig. 4c y d)
2. **Monopodial.** Existe un eje principal y ejes laterales reducidos o ausentes (Fig. 5a). Si las ramas laterales quedan subordinadas al crecimiento del eje principal, hablamos de monopodios, como en el abeto (*Abies*). Si los brotes laterales siguen ramificándose, se les puede denominar brotes laterales de primer, segundo y tercer grado
3. **Simpodial.** No existe un eje principal, en cambio, existen varias ramas laterales semejantes (Fig. 5b). A veces el eje principal se atrasa en su crecimiento o incluso este cesa.

¹³Lyndon explica que, al formarse los primordios al alrededor del ápice, dependiendo del tamaño de ellos es la cantidad que se forman, determinando el ángulo entre ellos y su filotaxia.

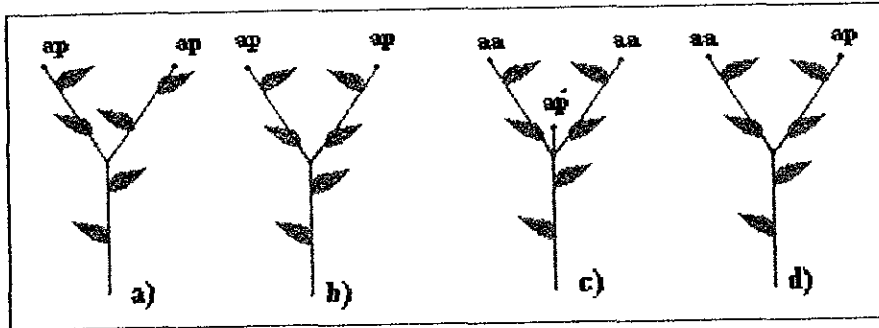


Figura 4. Ejemplos de dicotomías y pseudodicotomías. (ap= ápice principal aa= ápice axilar)
 a) Dicotomía sin reflejo en el espejo. b) Dicotomía con reflejo en el espejo. En estas dos dicotomías, ambas ramas provienen de ápices principales c) Pseudodicotomía Se trata de un crecimiento simpódico. Nótese que el ápice principal se encuentra en medio de las dos ramas bifurcadas, las cuales provienen de un ápice axilar. d) Pseudodicotomía debida a un desarrollo precoz de la rama. Nótese que la rama izquierda proviene de un ápice axilar mientras la rama derecha proviene de un ápice principal.

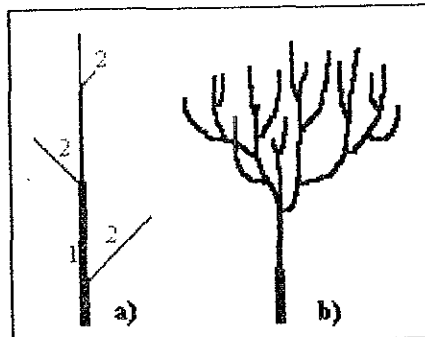


Figura 5. Ejemplo de crecimiento monopodial y simpodial.
 a) Crecimiento monopodial. b) Crecimiento simpodial

Hallé y Oldeman¹⁴, además de usar la clasificación de falsa y verdadera dicotomía, menciona 23 modelos diferentes para árboles, los cuales se reportan en el cuadro 1. Esta clasificación se puede utilizar también en plantas herbáceas, según lo afirma Bell (1991.288).

Estos modelos de ramificación consideran 5 características diferentes:

1. **Tronco.** Este puede ser monopodial o simpodial. Bell usa las palabras “monopodial” y “simpodial” con un concepto distinto al visto anteriormente. Para él, un tronco monopodial es aquel que se desarrolla de yemas apicales, mientras que un tronco simpodial es el que se desarrolla de yemas axilares (fig. 6 a y b respectivamente). El tipo de tronco puede ser determinado (con flor terminal) o indeterminado (sin flor).

¹⁴ Citados por Bell, 1991: 288

2. **Crecimiento del tronco.** Este puede ser rítmico (con periodos de crecimiento y periodos de dormancia) o continuo (siempre en crecimiento).
3. **Ramas con crecimiento plagiotrópico** (en dirección vertical) u **ortotrópico** (en dirección horizontal).
4. **Ramas simpodiales y unidades simpodiales.**
5. **Floración.** Puede ser lateral o terminal.

Aunque estos tipos de ramificación fueron propuestos para los árboles, Bell explica que también se pueden encontrar en arbustos y plantas herbáceas.

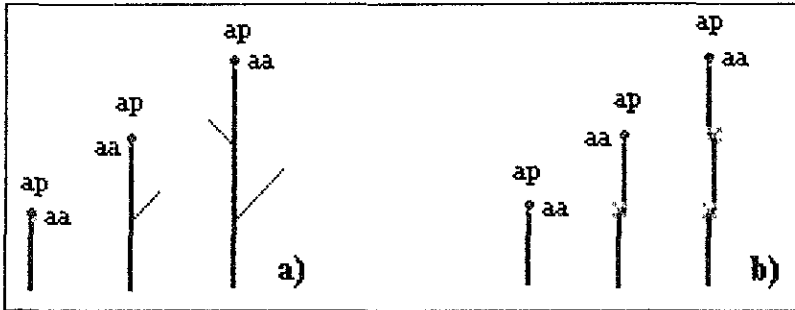
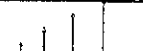


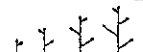

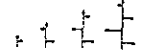

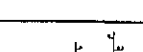
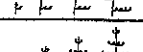
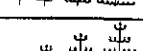
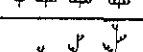
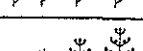
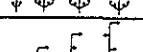
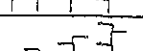


Figura 6. Ejemplos de crecimiento en troncos. (ap=ápice apical. aa=ápice axilar).

a) Tronco monopodial. Se desarrolla de ápices apicales b) Tronco simpodial. Del ápice apical se desarrolla una flor y del axilar el tronco

El cuadro 1 es una muestra de que existen patrones de crecimiento en árboles, que pueden ser clasificados y predichos en cuanto a la forma que puede adquirir un árbol en su crecimiento. Otras partes que están clasificadas y que se reconocen como un patrón de crecimiento son los tipos de flores, como se verá en el siguiente apartado

Cuadro 1. Clasificación de los tipos de ramificación.

MODELO	DESCRIPCIÓN	EJEMPLO	ESQUEMA
Hulttun	Tronco monopodial, determinado y con inflorescencia terminal. Ramas que acompañan a la inflorescencia.	<i>Corypha utan</i>	
Corner	Tronco monopodial, indeterminado y con inflorescencias laterales. Sin ramas, excepto las que acompañan a la inflorescencia.	<i>Bulbostylis vestita</i>	
Cook	Tronco monopodial e indeterminado, crecimiento continuo. Algunas ramas temporales.	<i>Phyllanthus grandifolius</i>	
Attims	Tronco monopodial con crecimiento continuo. Ramas monopodiales y ortotrópicas.	<i>Ficus pumila</i>	
Rauh	Tronco monopodial con crecimiento rítmico. Ramas monopodiales y ortotrópicas.	<i>Cecropia obtusa</i>	
Roux	Tronco monopodial con crecimiento continuo. Ramas monopodiales y plagiotrópicas	<i>Laetia procera</i>	
Massart	Tronco monopodial con crecimiento rítmico. Ramas plagiotrópicas	<i>Alisma plantago-aquatica</i>	
Petit	Tronco monopodial con crecimiento continuo. Ramas compuestas de unidades simpodiales determinadas.	<i>Piper sp.</i>	
Fagerlind	Tronco monopodial con crecimiento rítmico. Ramas compuestas por unidades simpodiales determinadas	<i>Paulownia tomentosa</i>	
Aubreville	Tronco monopodial con crecimiento rítmico. Ramas plagiotrópicas compuestas de unidades simpodiales indeterminadas.	<i>Terminalia catappa,</i>	
Stone	Tronco monopodial con crecimiento continuo. Ramas ortotrópicas simpodiales	<i>Rhipsalis bambusoides</i>	
Scarrone	Tronco monopodial con crecimiento rítmico. Ramas ortotóficas simpodiales	<i>Phellodendron chinense</i>	
Troll	Tronco monopodial, plagiotrópico, reorientado en posición vertical por un cambio de actividad. Ramas plagiotrópicas.	<i>Prunus sp.</i>	
Troll	Tronco simpodial, plagiotrópico, reorientado en posición vertical. Ramas plagiotrópicas.	<i>Platanus hispanica</i>	


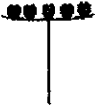


Cuadro 1. Clasificación de los tipos de ramificación. (Cont.)

MODELO	DESCRIPCIÓN	EJEMPLO	ESQUEMA
Mangenot	Tronco simpodial y ortotrófico. La porción distal de cada unidad simpodial del tronco se desarrolla lateralmente como una rama plagiotrópica.	<i>Strychnos sp.</i>	
Champagnat	Tronco simpodial ortotrófico. La parte distal de cada unidad simpodial se desarrolla de lado cuando crece cayendo por su propio peso.	<i>Salix babylonica</i>	
McClure	Secuencia de ramas simpodiales en las que la parte proximal de cada unidad simpodial determinada es plagiotrópica y la parte distal forma un tronco ortotrófico. El tronco soporta ramas determinadas.	<i>Bambusa arundinacea</i>	
Tomlinson	Secuencia de ramas simpodiales donde cada unidad simpodial nace en la parte proximal de la unidad previa. Las unidades simpodiales son determinadas o indeterminadas.	<i>Alpinia speciosa</i>	
Chamberlain	Tronco simpodial. Cada unidad simpodial soporta una unidad similar justo en su parte terminal. Sin ramas.	<i>Epiphyllum sp.</i>	
Leeuwenberg	Secuencia de ramas simpodiales. Cada unidad simpodial soporta solo una unidad similar en su parte final.	<i>Euphorbia punicea</i>	
Schoute	Verdadera dicotomía en el ápice a intervalos. Floración lateral	<i>Hyphaene thebaica</i>	
Koriba	Tronco simpodial. Cada unidad simpodial de tronco soporta mas de una rama lateralmente extendida en su parte final. Una de estas ramas es secundariamente reorientada en posición vertical para recibir la siguiente unidad del tronco.	<i>Alstonia macrophylla</i>	
Prévost	Tronco simpodial. Cada unidad simpodial soporta mas de una rama en su extremo distal. Una de estas ramas es retardada en su extensión y crece verticalmente para soportar la siguiente unidad del tronco. las otras ramas son ortotrópicas inicialmente pero se convierten en plagiotrópicas por aposición o substitución.	<i>Cyphomandra betacea</i>	
Nozeran	Tronco simpodial, cada unidad simpodial soporta una o más ramas en su parte distal. Una de estas ramas es retrasada en su extensión y crece verticalmente para recibir la siguiente unidad del tronco. Las otras ramas son plagiotrópicas conservando este carácter incluso si son cortadas.	<i>Geissospermum serviceum</i>	



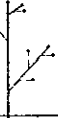

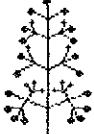
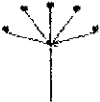
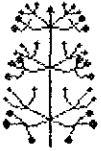
1.2.3 Flores y frutos

Los órganos reproductivos de las plantas también muestran patrones, sobre todo la inflorescencia. El término inflorescencia, se refiere al arreglo de las flores en la planta (Jones, 1988:251). Una *Inflorescencia determinada* es aquella en la que la secuencia de floración comienza con la flor terminal en la punta del tallo o en el centro del grupo de flores. Una *inflorescencia indeterminada* presentan una secuencia de floración que comienza en o cerca de la base hacia arriba o hacia el centro. Algunas inflorescencias son simples y fáciles de distinguir. Otras son agregados complicados difíciles de caracterizar. Los tipos más comunes de inflorescencias se describen en el cuadro 2:

Cuadro 2. Tipos de inflorescencias.

TIPO	DESCRIPCIÓN	EJEMPLO
Amento	Inflorescencia en forma de espiga, decidua, con brácteas escamosas y flores unisexuales y apétalas. <i>La inflorescencia puede ser erecta o laxa.</i>	
Capítulo o Cabezuela	Agrupación densa de flores sin pedicelo. Cuando las flores se originan en un receptáculo pueden ser guardadas dentro de esta en un hipantodio. Puede ser determinada o indeterminada	
Cima	Construcción simpodial consiste de una serie de flores que nacen en la axila de la bractea de una flor precedente Si el cimo soporta una flor es monocasio. Si soporta 2 flores es dicasio, y más de dos es pleiocástico. Determinada.	
Corimbo	Inflorescencia amplia, donde los pedicelos se van alargando sucesivamente dando la apariencia de que las flores se distribuyen mas o menos al mismo nivel horizontal. Las flores no se originan en el mismo punto dentro del eje principal como en la umbela. Indeterminada.	

Cuadro 2. Tipos de inflorescencias. (cont.)

TIPO	DESCRIPCIÓN	EJEMPLO
Espádice	Inflorescencia parecida a la espiga pero gruesa y carnosa, con flores muy pequeñas que se encuentran reunidas y por lo común incluidas en una espata. Indeterminada	
Espiga	Inflorescencia con un solo eje, donde las flores se arreglan a lo largo de este sin pedicelos. Indeterminada	
Panicula	Inflorescencia compuesta, donde el eje principal se ramifica una o más veces y puede sostener espigas, racimos o corimbos.	
Racimo	Inflorescencia con un solo eje, donde las flores se arreglan a lo largo de este sobre sus pedicelos. Indeterminada.	
Tirso	Un conjunto de secuencias simpodiales que se arreglan a lo largo de un tallo en series consecutivas. El eje principal es indeterminado y los ejes laterales son determinados.	
Umbela	Si las flores se distribuyen mas o menos al mismo nivel horizontal y las flores se originan en un mismo punto. Indeterminada	
Verticilo	Inflorescencia que presenta flores arregladas en vértices o espirales en nodos.	

Otro concepto utilizado en las inflorescencias es la *paracladia*, que se refiere a que ocurre una secuencia regular de un patrón a lo largo de la estructura completa de la flor. Por ejemplo, en una ramificación dicotómica se repite de forma general la división de un eje en dos. En una inflorescencia como la de la figura 7,

lo que se encuentra encerrado en líneas discontinuas es el patrón que se repite a lo largo de la inflorescencia. La unidad que se repite recibe el nombre de *paracladium*.

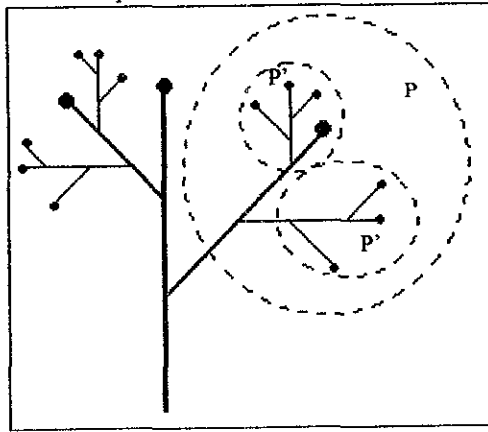


Figura 7. Ejemplo de paracladia.
Inflorescencia donde se representa la paracladia. P y P' son las unidades repetibles, el *paracladium*.

Otra propiedad de algunas flores y frutos es que presentan su arreglo en forma de una espiral o una hélice. Este crecimiento se da en los meristemas apicales de una planta cuando el primordio de un tamaño inicial se desarrolla pero no necesariamente ocupa la misma área de la superficie apical. La consecuencia de este empacamiento puede ser visto en el fruto de la piña (*Ananus comosus*) o en la flor de girasol (*Heliantus sp.*), donde las semillas tienen el mismo aspecto pero no el mismo tamaño. Las semillas se arreglan en hileras espirales en dos direcciones: una, en el sentido de las manecillas del reloj y la otra en contra del sentido de las manecillas del reloj. Estas hileras se denominan *parásticos* y forman espirales logarítmicas (fig. 8)

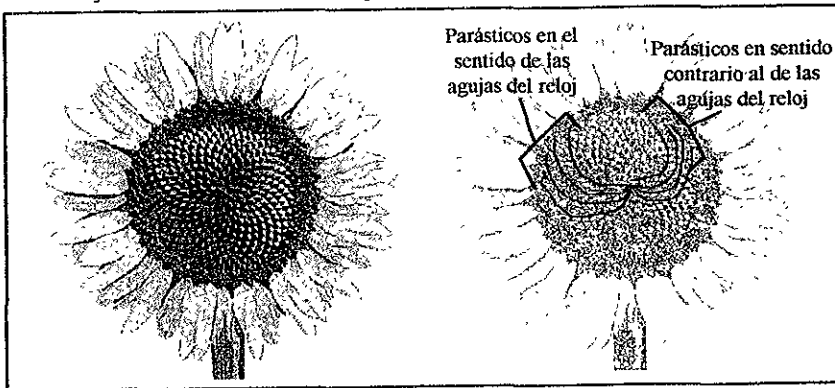


Figura 8. Flor de girasol mostrando los tipos de parásticos.

Bell (1991:216-217) menciona que las plantas no crecen al azar, sino de un modo flexiblemente organizado, controlado por factores internos (como la genética y la fisiología) y factores externos (como el

clima y la competencia). En algunas plantas, el patrón de crecimiento es visualmente preciso, geométrico y predecible. En otras plantas, el patrón no es detectable aún usando análisis estadísticos.

1.3 Factores que afectan los patrones de crecimiento

Fuller (1974:199-215) afirma que hay varios factores que son determinantes para que una planta tenga una cierta estructura. Existen factores externos e internos, que son inherentes a la planta.

Dentro de los factores internos está la herencia y la producción de expresores o supresores de características de la planta, como las fitohormonas. Dentro de los caracteres externos están los ambientales, como la capacidad que tiene una planta para que su estructura resista condiciones climáticas como la irradiación solar, la humedad y el viento. Los factores ecológicos también son considerados como factores externos, y un ejemplo es la forma en que responde la planta a la competencia por luz o a los depredadores.

Fuller (1974:201) considera que las condiciones ambientales pueden afectar la forma del crecimiento de una planta, pero hasta cierto punto. Este límite está dado por un factor muy importante en la planta: su genética. Otros autores como Chien y Jürgensen (1992:213-229) afirman que los factores genéticos en las plantas herbáceas juegan un papel predominante a diferencia de las plantas leñosas donde el desarrollo se halla más determinado por el ambiente y la competencia.

Nultsch (1975:398), por su parte menciona que en el crecimiento de una planta, no se debe discutir que es más importante: si la herencia o el medio ambiente. La herencia limita lo que un organismo podrá ser, mientras que el medio ambiente determina lo que el organismo será.

Los patrones de crecimiento revisados en este capítulo son los generalmente predecibles y constantes en las plantas debido a que son el resultado de la genética del organismo. Cuando los patrones de crecimiento se ven afectados por factores externos, estos se deben tomar en cuenta para generar un modelo del crecimiento de la planta, pero antes de crear un modelo, es necesario conocer qué es un modelo para saber qué se puede esperar de él.

CAPITULO II: REPRESENTACIÓN DE LOS PATRONES DE DESARROLLO MEDIANTE UN MODELO: LOS SISTEMAS LINDENMAYER.

2.1 Modelos.

2.1.1 Definición de modelo y conceptos asociados.

Cuando un fenómeno (sea físico, químico, biológico, social, etc.) sigue un patrón que se puede predecir, se puede generar un *modelo* que ayude a comprender mejor los procesos que se llevan al cabo dentro de dicho fenómeno. Un *modelo*¹⁵ en su definición más general es “[...] una representación de un objeto, sistema o idea en otra forma distinta a la que se presenta” (Shannon, 1975:4). Achinstein (1987:6-8) explica que un modelo describe un tipo de objeto o sistema atribuyéndole lo que podría llamarse una estructura interna, una composición o un mecanismo que explicará, al tomarlo como referencia, diversas propiedades de ese objeto o sistema modelado. Un modelo se considera como una aproximación de la realidad útil para ciertos propósitos, ya sean estos didácticos, explicativos, o predictivos, como se verá mas adelante.

Según la definición dada por Tejera (1994:820.), un modelo se puede analizar directamente, por medios matemáticos o con técnicas de simulación, aunque por medios matemáticos siempre resulta útil, ya que sirve para conocer más a fondo el problema que se estudia. También resalta que no se debe olvidar que un modelo es solo eso, un modelo, y entre más detallado sea, más difícil será el último análisis del problema real al cual se le busca *solución*.

Aquí surge otro concepto que vale la pena aclarar, el de *simulación*. Shanon (1975:2) explica que la “*simulación* es el proceso de designar un modelo de un sistema real y conducir experimentos con este modelo con el propósito de entender el funcionamiento del sistema o evaluar varias estrategias (dentro de los límites

¹⁵ Existen varias definiciones de lo que es un modelo, como la dada por Pfaffenberger: “Simulación de un sistema que existe en el mundo real, como el fuselaje de una aeronave o el flujo de efectivo de una empresa. La creación de un modelo pretende una mejor comprensión del prototipo -el sistema que se está modelando.[...]” Esta definición da un ejemplo de modelo pero no especifica los tipos de modelo que existen. Otra definición es la dada por Lapedes: “Sistema matemático o físico que cumple con ciertas condiciones especificadas, de las cuales se usa el funcionamiento para comprobar la analogía en algún aspecto con los sistemas físicos, biológicos o sociales.” pero esta definición tiene el mismo problema que la anterior. (Véase a Pfaffenberger, 1996:330; y a Lapedes, 1981:1335.)

impuestos por un criterio o un conjunto de criterios) para la operación del sistema.”¹⁶ Este término es diferente del concepto de *modelación*, que aunque Tejeda lo maneje como un sinónimo, de acuerdo a la definición de modelo presentada en el párrafo anterior, el concepto de *modelación* dentro de este trabajo será *la creación de un modelo para un sistema* (sea matemático, físico o cualquier otro), mientras que la *simulación* utilizará este modelo para *alcanzar por lo menos uno de tres objetivos*:

1. **Describir** el comportamiento del sistema;
2. **Construir teorías o hipótesis** sobre la causa del comportamiento observado; o
3. **Usar estas teorías para predecir** su funcionamiento futuro, como los efectos que puedan ser producidos por los cambios en el sistema o en su método de operación (Shannon, 1975:2).

Otros de los fines para los que se utilizan los modelos son de tipo didáctico, ya que es más fácil enseñar lo que sucede en un fenómeno a través de un modelo ya que este es una representación simplificada de la realidad.

2.1.2 Por qué y para qué se utilizan los modelos.

La modelación se utiliza desde que el hombre a buscado entender y manipular su medio ambiente, desde el momento en que ha tratado de representar ideas y objetos; esta ha ido desde las pinturas rupestres, pasando por el lenguaje hasta las expresiones matemáticas. Los modelos se han usado como ayuda para pensar, comunicarse, predecir y experimentar. Shanon¹⁷ explica que un modelo debe servir para una de dos metas:

1. **describir**, es decir, para explicar y/o entender mejor el comportamiento del fenómeno estudiado en la realidad o para
2. **prescribir** su comportamiento, predecirlo.

Pero, ¿por qué se simula si se puede experimentar directamente con la realidad? la simulación ofrece varias ventajas si se le compara con la experimentación directa, tales como:

- **Costos.** La simulación requiere de gastar en términos de horas-hombre (al armar un modelo a escala, escribir un programa en computadora), y la experimentación, además de gastar en horas-hombre, también requiere de diversos materiales que por lo regular son caros (como equipo y reactivos) En el caso de que la experimentación, además de costosa sea peligrosa para el hombre, la simulación es una mejor alternativa.

¹⁶ Este concepto tiene varias definiciones las cuales explican que la simulación es la representación de un sistema por otro, lo cual no diferencia a un modelo de una simulación. (Consulte a Lapedes, 1981:1870 y a Tejera, 1994:1177)

¹⁷ Shanon, 1975:7

- **Tiempo.** La experimentación puede necesitar de mucho tiempo para obtener resultados, sobre todo cuando se trabaja con organismos con ciclo de vida largo, mientras que la simulación, ayudada por la computadora reduce ese tiempo de años, meses o días a segundos en la computadora.
- **Replicación.** Uno de los requerimientos de cualquier experimento es que se pueda repetir por otros científicos. En la realidad, repetir un experimento de manera precisa es raro, debido a la gran cantidad de variables a controlar. En cambio, las simulaciones pueden ser repetibles y precisas.
- **Seguridad.** Cuando los experimentos requieren de medir el comportamiento de un organismo en condiciones extremas, y esto puede ser peligroso para el experimentador o ilegal, la simulación es ideal, ya que no ofrece peligro alguno para el experimentador (Pidd, 1988:7-8).

Para decidir si es mejor experimentar o simular al estudiar cualquier fenómeno, es necesario evaluar los anteriores cuatro puntos.

2.1.3 Validez de los modelos.

Un punto importante en la utilización de los modelos es determinar si realmente un modelo es adecuado para el fenómeno que está representando. Achinstein (1987:8) propone que para juzgar esto, se deben analizar dos aspectos del modelo:

1. **Utilidad.** Cuando se utiliza un modelo es para obtener propósitos definidos antes de utilizar el modelo mismo. La pregunta obligada para saber que el modelo utilizado es el mejor para los propósitos establecidos es: ¿Qué tan bien me sirve para las metas que me propongo?
2. **Representatividad.** Este es el otro aspecto clave en la validez de un modelo. ¿Qué tan exacta y completamente representa al fenómeno que trata de modelar?

Un buen modelo debe tener estas dos características ser útil para ciertos propósitos y ser representativo del fenómeno real que modela

2.1.4 Tipos de modelos.

Existen diversos tipos de modelos. La clasificación más general es la dada por Shannon (1975:2), quien clasifica a los modelos y las simulaciones en función de que tan exactos o abstractos son, e ilustra esto a través de un espectro continuo como el de la fig. 9:

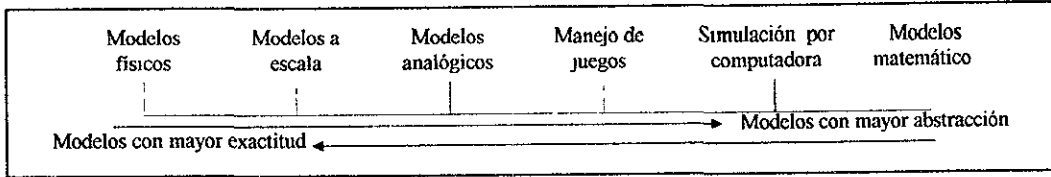


Figura 9. Clasificación de modelos y simulaciones.

De estos modelos, varios se han usado en la ciencia, siendo los modelos a escala los más simples. Estos representan al original tridimensionalmente y guardan la misma proporción en su estructura que el original¹⁸. Este tipo de modelo tiene sus desventajas, ya que al experimentar con el modelo requiere de alteración física de este, lo cual puede ser tedioso y caro. Por otro lado, los modelos a escala son estáticos, ya que no se pueden ver varios factores interactuando dentro del modelo. Otro tipo de modelos muy utilizados son los matemáticos¹⁹ y los lógicos. En ellos se representa al sistema en cuestión como una serie de ecuaciones que permiten su solución.

Para representar el crecimiento de las plantas, los sistemas Lindenmayer se basan en un modelo matemático. Con el posterior desarrollo de la computación²⁰, fue posible, a través de la computadora, observar las plantas en dos y tres dimensiones como si fuera un modelo a escala de la planta.

Para entender como representan los sistemas Lindenmayer a las plantas, es necesario conocer el modelo matemático en el que se basan. Para ello se describirá este modelo en el siguiente apartado.

2.2 Los lenguajes Lindenmayer como herramienta para modelar los patrones de crecimiento de las plantas.

Los sistemas Lindenmayer son una variación de la teoría de los lenguajes formales, desarrollada a finales de los años cincuenta por Noam Chomsky²¹. Los sistemas Lindenmayer están basados en estos lenguajes formales, por lo que, para comprender mejor qué son y cómo llegan a modelar el patrón de

¹⁸ Una definición similar la da Lapedes: "Representación tridimensional de un objeto o estructura que tiene todas sus partes en la misma proporción de su tamaño verdadero" (Lapedes, 1981:1335)

¹⁹ Tejera define a un modelo matemático como "Caracterización general de un proceso, objeto o concepto en términos matemáticos, lo cual permite la manipulación relativamente sencilla de variables, la cual debe realizarse para poder determinar cómo se comportaría el proceso, objeto o concepto en diferentes situaciones" (Tejera, 1994:821)

²⁰ De hecho, con el desarrollo de la computación y en particular de la simulación por computadora se requiere que los modelos se expresen en forma matemática, para que sea más fácil de elaborar un programa.

²¹ Noam Chomsky, lingüista norteamericano, desarrolló en los cincuenta la teoría de los lenguajes formales para explicar como se comportan los idiomas utilizados por el hombre (Fu, 1974:25; Prusinkiewicz, 1989:39)

crecimiento de las plantas es necesario comprender sus raíces, de allí que se le dedique este primer apartado a los lenguajes formales.

2.2.1 Lenguajes formales

Un *lenguaje formal* o un *sistema formal*, se define como “[...] un sistema lógico sin interpretación definida” (Donovan, 1986:214). Según Donovan, un lenguaje formal consta de un alfabeto, que es un conjunto de palabras llamadas axiomas, y un conjunto finito de relaciones llamadas reglas de inferencia. Se hace uso de este tipo de lenguajes para obtener modelos formalizados de nociones intuitivas o informales. Los modelos formales pueden ser abstraídos y estudiados matemáticamente, y si el modelo es adecuado para el problema que representa, los resultados pueden explicar mucho del fenómeno que se estudia.

Como todo lenguaje, los sistemas formales tienen una *sintaxis*²², donde se establecen cuáles son las condiciones para construir palabras válidas dentro de este, y una *semántica*²³, donde se asigna una interpretación a los signos lingüísticos. Para definir matemáticamente a un lenguaje es preciso definir un alfabeto. Un *alfabeto*, según Herman (1975:57), es “un conjunto finito de símbolos [...]” que se denotarán como el conjunto V . Una serie (también conocida como sentencia, fórmula o cadena) es la concatenación de símbolos del alfabeto V , y el conjunto de estas posibles concatenaciones finitas se denotará como V^* , que incluye la serie nula λ , donde no existen símbolos. La mayoría de los lenguajes no incluyen todas las posibles concatenaciones finitas de los símbolos del alfabeto, ya que no todas tienen un significado, por ello, un *lenguaje* L es un subconjunto de V , y se expresa como $L \subset V$.

Para entender y definir una gramática formal tomemos como ejemplo una oración o sentencia en español: “La planta fotosintetiza eficientemente”, donde “La planta” es una frase nominal, y “fotosintetiza eficientemente” es una frase verbal; a su vez, la frase nominal está compuesta por un artículo “La”, y un sustantivo “planta”; y la frase verbal está compuesta por un verbo “fotosintetiza” y un adverbio, “eficientemente”.

²² Una definición de este concepto es: “Parte de la gramática que estudia la concordancia o armonía que entre varias clases de palabras ha establecido el uso, su régimen o dependencia mutua, y el orden en que se colocan los varios elementos de que se compone la oración”. Esta definición también es válida para el trabajo. (Reader’s Digest, 1986:3530)

²³ Este concepto se define como “Estudio del significado de los signos lingüísticos” (Reader’s Digest, 1986:3471)

Estas frases se pueden formar siguiendo los siguientes pasos:

Paso 1: <sentencia>

Paso 2: <frase nominal><frase verbal>

Paso 3: <artículo><sustantivo><frase verbal>

Paso 4: La <sustantivo><frase verbal>

Paso 5: La planta <frase verbal>

Paso 6: La planta <verbo><adverbio>

Paso 7: La planta fotosintetiza <adverbio>

Paso 8: La planta fotosintetiza eficientemente

El conjunto de palabras entre corchetes (<>) forman parte de un lenguaje que se utiliza para hablar de otro lenguaje, es decir, un *metalenguaje*, y que son estados de transición para describir al lenguaje objeto. Los símbolos utilizados en el metalenguaje pertenecen a un conjunto V_N y se conocen como *símbolos no terminales*, debido a que aparecen solo en pasos intermedios de la generación formal. En contraparte, los *símbolos terminales* son símbolos del alfabeto V_T , como “La”, “planta”, “fotosintetiza” y “eficientemente” en el ejemplo. Además, otro elemento importante es el *símbolo inicial* denotado como S , a partir del cual se derivan todas las series del lenguaje. En el ejemplo, “<sentencia>” es el símbolo inicial y este pertenece siempre a V_N . Retomando el ejemplo anterior, el conjunto de pasos pueden ser descritos siguiendo *reglas de reescritura*²⁴ como sigue:

<sentencia> → <frase nominal><frase verbal>

<frase nominal> → <artículo><sustantivo>

<frase verbal> → <verbo><adverbio>

<artículo> → La

<sustantivo> → planta

<verbo> → fotosintetiza

<adverbio> → eficientemente

²⁴ Una regla de reescritura se da de la forma $a \rightarrow b$, y se entiende como “a puede ser reescrita como b”, o “a puede ser sustituida por b”.

Estas reglas de reescritura o sustitución constituyen un algoritmo para generar sentencias o cadenas de símbolos, y el proceso de generación consiste en aplicar estas reglas hasta que no puedan aplicarse nuevas producciones²⁵, o la serie quede constituida por símbolos terminales. Con el ejemplo visto, una gramática formal G se define como una cuádrupla $G = (V_N, V_T, P, S)$ en donde:

1. V_N y V_T son el vocabulario no terminal y terminal de G , respectivamente. En el ejemplo visto,

$$V_N = \{ \langle \text{sentencia} \rangle, \langle \text{frase verbal} \rangle, \langle \text{frase nominal} \rangle, \langle \text{artículo} \rangle, \langle \text{verbo} \rangle, \langle \text{sustantivo} \rangle, \langle \text{adverbio} \rangle \}$$
 y

$$V_T = \{ \text{La, fotosintetiza, planta, eficientemente} \}.$$
 La unión de V_N y V_T constituyen el vocabulario total V de G , y la intersección de V_N y V_T da por resultado el conjunto vacío, representado por el símbolo λ .
2. P es un conjunto finito de reglas de reescritura o producciones denotadas como $\alpha \rightarrow \beta$ (se lee: α genera a β), donde α y β son cadenas de V y donde α pertenece a V_N .
3. S pertenece a V_N y es el símbolo inicial.

Por otra parte, es necesario definir algunos conceptos más, como el proceso para generar una serie. Una serie η genera inmediatamente a una serie γ (se escribe $\eta \Rightarrow \gamma$) en una gramática G si y sólo si $\eta = \sigma \alpha \tau$, $\gamma = \sigma \beta \tau$, y $\alpha \rightarrow \beta \in P$ de G en la que σ y τ ²⁶ representan series arbitrarias que pueden estar vacías. Pongamos un ejemplo.

$$V_N = \{ A, \Sigma \}, \quad V_T = \{ a, b \}, \quad S = \{ \Sigma \}$$

$$P = \{ \Sigma \rightarrow A \quad \dots (1)$$

$$A \rightarrow aAb \quad \dots (2)$$

$$A \rightarrow ab \quad \dots (3)$$

Supongamos que $\eta = aAb$ y $\gamma = aabb$. Entonces $aAb \Rightarrow aabb$ es una generación inmediata con $\sigma = a$, $\tau = b$, $\alpha = A$, $\beta = ab$ y la producción $A \rightarrow ab$. Ahora, una serie η genera a una serie γ (se escribe $\eta \Rightarrow^* \gamma$) en una gramática G si y solo si hay una secuencia de series $\omega_0, \omega_1, \dots, \omega_n$, para $n \geq 0$, de modo que $\eta = \omega_0$, $\gamma = \omega_n$ y para

²⁵ Una producción y una regla de reescritura son sinónimos, de manera que $a \rightarrow b$ se lee como "a produce a b".

²⁶ A estas letras (σ y τ) también se les conoce como contexto. La expresión $\sigma \alpha \tau$, se entiende como α en el contexto $\sigma \tau$.

todo $0 \leq i \leq n$, $\omega_i \Rightarrow \omega_{i+1}$ (es decir, $(\eta = \omega_0) \Rightarrow \omega_1 \Rightarrow \dots \Rightarrow \omega_{n-1} \Rightarrow (\omega_n = \gamma)$) con $\omega_i \in (V_N \cup V_T)^*$ - λ para todo i . La lista $\{\omega_i\}$ es una *prueba* de γ en G .

Tomando en cuenta el ejemplo anterior, Σ genera a $aabb$ ya que

$$\Sigma \Rightarrow A \Rightarrow aAb \Rightarrow aabb.$$

Cabe aclarar que una *forma sentencial* es una serie que puede ser derivada a partir del símbolo inicial S , mientras que una *sentencia* solo contiene símbolos terminales. En el ejemplo de $\Sigma \Rightarrow A \Rightarrow aAb \Rightarrow aabb$, a , aAb y $aabb$ son formas sentenciales, mientras que $aabb$ es una sentencia.

Las producciones pueden tener distintas restricciones para llevarse al cabo. Chomsky divide a las gramáticas en cuatro tipos de acuerdo a las restricciones que se imponen a sus producciones, las cuales son:

Gramáticas de tipo 0

Gramáticas de tipo 1

Gramáticas de tipo 2

Gramáticas de tipo 3

2.2.1.1 Gramáticas de tipo 0.

En las gramáticas de tipo 0, o gramáticas no restringidas, no se limitan sus producciones. Estas gramáticas también reciben el nombre de lenguajes de tipo 0, y las series intermedias pueden expandirse o contraerse. Un ejemplo es la producción $abA \rightarrow aA$, donde desaparece la b en el contexto aA .

2.2.1.2 Gramáticas de tipo 1.

A este tipo de gramáticas también se les da el nombre de gramáticas de contexto sensitivo, y las producciones están restringidas de la forma:

$$\sigma\alpha\tau \rightarrow \sigma\beta\tau$$

donde α pertenece a V_N , σ, τ, β pertenecen a V^* , y $\beta \neq \lambda$. Esto se puede leer como "α puede ser remplazada por β en el contexto σ,τ." Esto implica que:

$$|\sigma\alpha\tau| \leq |\sigma\beta\tau| \quad \text{ó} \quad |\alpha| \leq |\beta|$$

es decir, las series no son contractivas, ya que la segunda serie formada es de igual o mayor longitud²⁷ que la primera serie. Estas gramáticas también reciben el nombre de lenguajes de contexto sensitivo. Un ejemplo de un lenguaje de contexto sensitivo es $aAb \rightarrow abb$, donde $\sigma=a$, $\tau=b$, $\alpha=A$ y $\beta=b$.

2.2.1.3 Gramáticas de tipo 2.

A estas gramáticas también se les llama de contexto libre, y son de la forma:

$$\alpha \rightarrow \beta$$

Donde α pertenece a V_N y β pertenece a V . Nótese que en una producción de esta forma, α puede ser remplazada por β independientemente del contexto en que se encuentre α , pero a diferencia de las gramáticas de tipo 0, las series no pueden expandirse o contraerse, sino que conservan el mismo número de caracteres. Las gramáticas de tipo 2 también son conocidas como lenguajes de contexto libre.

2.2.1.4 Gramáticas de tipo 3.

También se les conoce como gramáticas regulares o de estado finito, y son de la forma:

$$A \rightarrow aB \quad \text{ó} \quad A \rightarrow b$$

donde A, B pertenecen a V_N , y a, b pertenecen a V_T . En estas gramáticas se admite como máximo un símbolo no terminal tanto en el primer como en el segundo miembro de la producción, en el ejemplo $A \rightarrow aB$, A y B son símbolos no terminales. Cuando sucede esto, se dice que la producción es *lineal*. Si el símbolo no terminal ocurre siempre a la derecha de todos los otros símbolos en el segundo miembro de una producción, se llama *producción lineal derecha*. Si por el contrario, se encuentra a la izquierda de los demás símbolos, se llama *producción lineal izquierda*. Una gramática será *lineal derecha* o *lineal izquierda* si todas sus producciones son lineales derechas o izquierdas respectivamente. Por último, un lenguaje se llama *regular* si puede ser creado por una gramática lineal izquierda o derecha.

Cabe mencionar que cada gramática presentada es un subconjunto de la anterior, ya que cada restricción nueva incluye las que le preceden. Así, una gramática de tipo 3 es un subconjunto de las gramáticas

²⁷ La longitud, o tamaño de una serie es el número de símbolos que contiene esta. Por ejemplo, la serie abb tiene longitud 3, y la serie λ tiene longitud 0.

de tipo 2; a su vez, las de tipo 2 son un subconjunto de las gramáticas de tipo 1 y estas son un conjunto de las gramáticas tipo 0.

Aunque la teoría de los lenguajes formales fue la base para crear a los sistemas Lindenmayer, estos guardan diferencias con los lenguajes formales. En el siguiente apartado se resaltarán las semejanzas entre los sistemas Lindenmayer y los lenguajes formales para ubicarlos dentro de este contexto.

2.2.2 Diferencias entre lenguajes formales y sistemas Lindenmayer.

Con los estudios hechos por Noam Chomsky acerca de los lenguajes formales, se dio un periodo de fascinación por la sintaxis, las gramáticas y sus aplicaciones. Los avances en la teoría de los lenguajes formales dieron origen a la idea de usar las series o cadenas de símbolos para describir dibujos. El objetivo inicial fue el reconocer objetos, como letras o cromosomas, mediante la codificación de las imágenes y el posterior análisis de las cadenas resultantes.

En 1968, el biólogo húngaro, Aristid Lindenmayer, introdujo un tipo nuevo de reglas de reescritura llamadas sistemas Lindenmayer (conocido también como *sistemas-L* o *L-systems*, este último por su abreviación en inglés). Este tipo de sistemas fueron concebidos como un modelo formal del crecimiento de las plantas, y sus esfuerzos iniciales se concentraron en la generación automática de imágenes de plantas.

Los *sistemas Lindenmayer* se definen como un conjunto de reglas de producción que especifican una repetida secuencia de transformaciones a un modelo dado (Valdés, 1991:46). A pesar de que estos sistemas se basan en la teoría de lenguajes formales, existen diferencias, por lo que se les considera como un tipo de lenguajes diferentes. Es importante entender que los sistemas Lindenmayer y los lenguajes formales no son lo mismo. Las diferencias son:

1. En los lenguajes formales, se hace una clara distinción entre los símbolos terminales y no terminales. En los sistemas Lindenmayer *no existen símbolos terminales*, ya que las estructuras que representan como hojas tallos y flores están en constante transformación y no existe un órgano terminal. Por ejemplo, una hoja se transforma durante el crecimiento de la planta: cambia de forma, aumenta de tamaño, toma otro color, se marchita, etc., pero no existe una forma terminal, en todo caso, la forma terminal es el conjunto vacío, cuando la hoja se marchita y cae.
2. Debido al primer punto, la definición de alfabeto (V) que agrupaba a los símbolos terminales y no terminales, *solo agrupará a los símbolos no terminales*.

3. Otra diferencia fundamental es el modo en que son aplicadas las producciones, ya que en los lenguajes formales las sustituciones se hacen secuencialmente, mientras que en los sistemas Lindenmayer se aplican *en paralelo*

Supongamos que:

$$V = \{A, B, \Sigma, a, b\}, \quad S = \{\Sigma\}$$

$$P = \{ \Sigma \rightarrow AB \quad \dots (1)$$

$$A \rightarrow aBb \quad \dots (2)$$

$$B \rightarrow b \quad (3)$$

En los lenguajes formales, la producción se llevaría a cabo como:

$$\Sigma \Rightarrow AB \Rightarrow aBbB \Rightarrow aBbb \Rightarrow abbb$$

mientras que en los sistemas Lindenmayer, la producción se llevaría a cabo como:

$$\Sigma \Rightarrow AB \Rightarrow aBbb \Rightarrow abbb$$

Hasta aquí se han revisado las bases de los sistemas Lindenmayer adentrándose a la manera en que se expresan y sus reglas de producción. Teniendo en cuenta las diferencias con los lenguajes formales, se verán los tipos de sistema Lindenmayer y ejemplos de como modelan algunos órganos de las plantas en el siguiente apartado.

2.2.3 Tipos de sistemas Lindenmayer.

Los sistemas Lindenmayer se basan en teorías matemáticas que permiten incluir en un modelo, además de observaciones puramente morfológicas, observaciones genéticas, citológicas y físicas para describir el desarrollo de organismos multicelulares.

En el desarrollo de un organismo vegetal ocurren importantes cambios en células y tejidos, como la acumulación de constituyentes químicos y cambios estructurales que se conocen con el nombre de

“diferenciación”²⁸. Estos cambios pueden ocurrir bajo la represión o la expresión de ciertos genes, procesos que a su vez se ven influidos por la producción y distribución de inductores e inhibidores a través de las células. Los mecanismos de control genético, los linajes celulares, los centros de organización, los campos morfogenéticos y la polaridad, son conceptos que pueden ser incorporados a las tasas de crecimiento y las relaciones alométricas para desarrollar una moderna teoría de desarrollo. Estos aspectos biológicos se pueden aunar a la combinatoria, las teorías algebraicas y los lenguajes formales para formalizar un modelo de desarrollo, que es una de las metas de los sistemas-L.

Un sistema o modelo de desarrollo tiene un alfabeto y un conjunto de producciones que son aplicadas en paralelo a todos los símbolos de la cadena para formar la siguiente cadena en la secuencia de desarrollo. Al alfabeto y las producciones se les conoce como el *esquema de desarrollo*. Además, este sistema especifica uno o más símbolos iniciales, llamados *axiomas*. Una *secuencia de desarrollo* es una serie de cadenas $\omega_0, \omega_1, \omega_2, \dots$ de símbolos del alfabeto tal que ω_0 es el axioma, y para toda i , ω_{i+1} se obtiene de ω_i aplicando las producciones en paralelo a todos los símbolos de ω_i .

Los sistemas Lindenmayer representan los diferentes estadios por los que pasa un organismo vegetal. Para ello, sigue una secuencia de desarrollo utilizando reglas de producción, las que utilizan un alfabeto de símbolos, que, a su vez, representan estructuras del organismo a modelar, y estas producciones, dependiendo de sus características, pertenecen a un tipo diferente de L-system, como se describirá a continuación.

2.2.3.1 Sistemas 0L.

Los sistemas-0L son el conjunto más simple de los sistemas Lindenmayer, los de contexto libre (para abreviar, sistemas-0L, donde el cero significa “cero interacciones”). Prusinkiewicz y Hanan (1989:39), explican que este tipo de sistemas-L imitan un desarrollo unidimensional, como el de un organismo filamentosos, donde la descendencia se da por linajes celulares y es independiente de otros factores como las condiciones de las células vecinas o el clima.

Lindenmayer y Jürgensen (1992:6) explican que los sistemas-0L consisten en un conjunto finito V de símbolos (el alfabeto) para estadios celulares, un conjunto P de reglas de estadios transitorios (producciones) y un conjunto de cadenas α . Las reglas en P son de la forma $\alpha \rightarrow \beta$ donde $\alpha \in V$ y $\beta \in V^*$, y la regla $\alpha \rightarrow \beta$ representa la transformación del estado α de la cadena al estado β .

²⁸ Por *diferenciación* se entiende el proceso de diversificación de las células, tejidos y órganos a partir de una célula o tejido fundamental (Font, 1977:330)



Supongamos que:

$$V = \{a, b, c, \lambda\}, \quad \omega = \{a\}$$

$$P = \{a \rightarrow b \quad \dots (1)$$

$$a \rightarrow bc \quad \dots (2)$$

$$a \rightarrow \lambda \quad \dots (3)$$

Y que a y b se representarán como células  y  respectivamente. Las diferentes reglas de producción aplicadas quedarían como muestra la fig. 10:

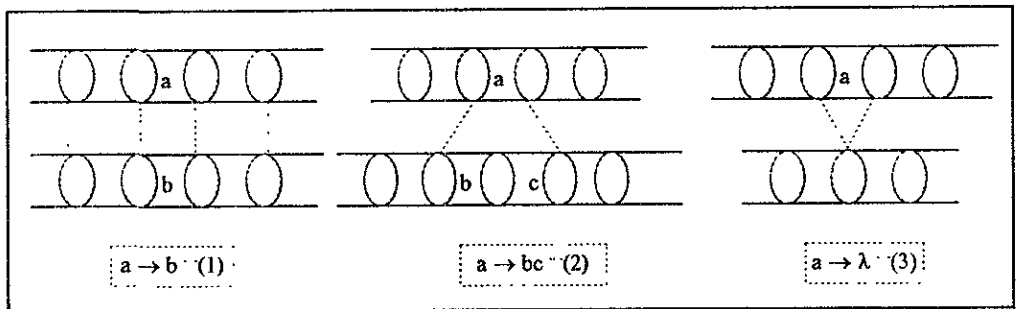


Figura 10. Tres estados transitorios de una célula y sus reglas.

2.2.3.2 Sistemas 0L determinísticos (sistemas-D0L).

Los sistemas-0L se pueden dividir en diferentes clases sin perder la característica de que imitan el desarrollo por imágenes celulares. Lindenmayer y Jürgensen (1992:6-7), determina que, si dada una cadena α , esta tiene reglas de transición sencillas donde un símbolo de la cadena produce un paso de derivación único, entonces esta pertenece a los sistemas llamados sistemas-0L determinísticos o sistemas-D0L.

El siguiente ejemplo modela el desarrollo de la bacteria verde-azul *Anabaena catenula*, donde.

$V=\{a, b\}$, $\omega=\{b\}$, y

$$P=\{\vec{a} \rightarrow \overleftarrow{a} \vec{b} \quad (1)$$

$$\vec{b} \rightarrow \vec{a} \quad (2)$$

$$\overleftarrow{a} \rightarrow \overleftarrow{b} \overleftarrow{a} \quad (3)$$

$$\overleftarrow{b} \rightarrow \overleftarrow{a} \quad (4)$$

y las flechas arriba de a y b son la polaridad de la célula. La secuencia de desarrollo será la siguiente:

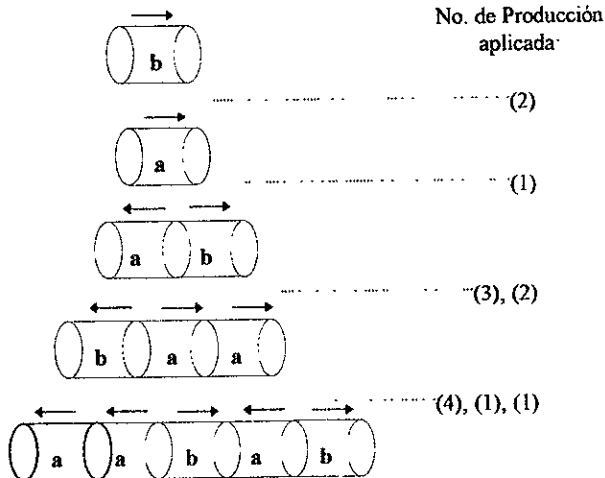


Figura 11. Ejemplo de desarrollo en *Anabaena catenula*

Al comparar este modelo con el de la fig. 10 (sistemas 0L), la célula a tiene tres reglas diferentes de producción, mientras que en este ejemplo, la célula a con cierta polaridad tiene solo una regla de producción. Esta es la característica que hace la diferencia entre los sistemas-0L y los sistemas-D0L.

2.2.3.3 Sistemas-T0L.

Para modelar el desarrollo de plantas cuyo crecimiento depende de factores climáticos se utilizan diferentes conjuntos de reglas de producción correspondientes a diferentes condiciones ambientales. Las reglas

de producción con estas características forman los sistemas-TOL (Donde la T hace referencia a las tablas de decisión que se utilizan), otra clase de sistemas-OL.

Herman (1975:112) explica que los sistemas TOL constan de un alfabeto V , un conjunto finito de tablas de decisión \mathcal{P} , compuestas a su vez, por un conjunto finito de reglas de producción P , y un símbolo inicial o axioma ω . Dentro de la secuencia de derivación se puede cambiar de una tabla a otra. En el caso de las plantas que cambian de un crecimiento vegetativo a la floración dependiendo de las horas luz, como en la Nochebuena, (*Euphorbia pulcherrima*), se pueden utilizar tablas como muestra la figura 12:



Figura 12. Tablas de decisión para la floración de una planta dependiendo de las horas-luz.

Donde a , y A son ápices, H es una hoja, I es un internodo, y F es la flor. En este caso, se utilizaría la primer tabla mientras las horas luz no fueran las requeridas para que la planta de flores, ya que el ápice a se transforma en una parte vegetativa $I[H]a$; en caso contrario, la planta daría flores y se utilizaría la segunda tabla donde el ápice A de la parte vegetativa $I[H]A$ se transforma en una flor, F . Prusinkiewicz y Hanan (1990:66), previenen que el uso de este tipo de sistemas-L es solo una solución parcial al modelar plantas cuyo crecimiento está influenciados por las condiciones ambientales, y además es necesario elegir una tabla adecuada.

2.2.3.4 Sistemas-OL estocásticos.

Se dice que un sistema OL es *estocástico* si las reglas de producción tienen una cierta probabilidad de que sucedan Prusinkiewicz (1989:60), los define formalmente como una cuádrupla $\{V, \omega, P, \pi\}$, donde V es el vocabulario o alfabeto, ω el axioma, P el conjunto de producciones, y π es la distribución de probabilidad, es decir, la probabilidad de que una producción se lleve a cabo, en escala de 0 a 1. Supongamos que las flores de cierta planta puedan ser rojas, moradas y blancas, y la probabilidad de que sea de un color dado es de $1/3$

Entonces:

$$V = \{B, F_b, F_r, F_m\}, \quad \omega = \{B\}$$

$$p = \overset{0.33}{B} \rightarrow F_b \quad \dots \quad (1)$$

$$\overset{0.33}{B} \rightarrow F_r \quad \dots \quad (2)$$

$$\overset{0.33}{B} \rightarrow F_m \quad \dots \quad (3)$$

Donde B es un botón, F_b son flores blancas, F_r son flores rojas y F_m son flores moradas. Cada producción puede ser seleccionada con la misma probabilidad de 0.33.

2.2.3.5 Sistemas-IL.

En este tipo de sistemas-L se modelan las plantas que responden a interacciones celulares, y sistemas-IL significa “*sistemas-L* de contexto sensitivo”, [por su abreviación en inglés Prusinkiewicz explica que las producciones son de la forma $\delta < \alpha > \gamma \rightarrow \beta$, donde la letra α puede producir β si y solo si α es precedido por δ y seguido por γ . Las letras δ y γ forman el contexto izquierdo y derecho de α en esta producción.

Los sistemas-IL se dividen en los sistemas-2L y en los sistemas-1L. Los sistemas-2L tienen dos contextos, uno izquierdo y uno derecho, de allí su abreviación. Los sistemas-1L, como su nombre lo indica, tienen un solo contexto, que puede ser de la forma $\delta < \alpha \rightarrow \beta$ ó $\alpha > \gamma \rightarrow \beta$.

Un ejemplo de como se aplican estos sistemas es la difusión de una hormona a lo largo de células filamentosas. Supongamos que

$$V = \{a, b\}, \quad \omega = \{baaaaaa\}$$

$$P = \{b < a \rightarrow b\}$$

Donde ω es el estado inicial del filamento, a es una célula con una concentración baja de hormona y b es una célula con una concentración excedente. El proceso de difusión se dará de la siguiente manera:

baaaaaa

bbaaaaa

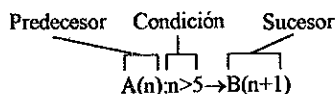
bbbaaaa

bbbbaaaa

...

2.2.3.6 Sistemas 0L y IL paramétricos

Este tipo de sistemas trabajan con letras parametrizadas²⁹. Los sistemas parametrizados se definen como una cuádrupla $\{V, \Sigma, \omega, P\}$ donde V es el vocabulario, Σ es el conjunto de parámetros formales (números reales \mathbb{R}), ω es el axioma, y P es el conjunto de producciones (Prusinkiewicz y Hanan, 1990 185; Prusinkiewicz y Lindenmayer, 1990:41-42). En estos sistemas, los símbolos “:” y “→” sirven para separar los tres componentes de una producción: el *predecesor*, la *condición*, y el *sucesor*



Esta producción se entiende como “A se describirá como B si n es mayor de 5”. Una producción se puede aplicar si:

1. La letra en la cadena que se está derivando y en el predecesor de la producción es la misma.
2. El número de parámetros formales en la cadena es igual al número de parámetros formales en la producción predecesora y
3. La condición evaluada es “verdadera” en el valor de los parámetros actuales.

Este tipo de sistemas se puede dar también en los sistemas-IL. Un ejemplo es el desarrollo de *Anabaena catenula*. Esta bacteria verde-azul forma un filamento sin ramificaciones con dos tipos de células: vegetativas y heterocistos. Comúnmente, las células vegetativas se dividen y producen dos células vegetativas hijas, pero en algunos casos las células se diferencian en heterocistos. La forma en que se distribuyen es un número relativamente constante de células separadas por heterocistos, y la distribución de estos últimos está regulada por compuestos de nitrógeno generados por los heterocistos, transportados de célula a célula por el filamento hasta que decrece en las células vegetativas. Cuando la concentración de estos compuestos nitrogenados llega a un nivel específico en las células vegetativas, estas se diferencian en heterocistos

Para hacer el modelo, las células son representadas mediante módulos $F(s,t,c)$ donde s es la longitud de la célula, la cual puede llegar al umbral en que se diferencia en heterocisto (3.9); t es el tipo de célula (0-heterocisto, 1 y 2 células vegetativas); y c representa la concentración de compuestos de nitrógeno, la cual

²⁹ El término letras parametrizadas se refiere a que cada letra del vocabulario utilizado (V) se acompaña de un elemento que la caracteriza, en este caso un número.

puede ser alta (900), o estar en el umbral (0.4) para que la célula se diferencie en un heterocisto. Las reglas de producción quedan como sigue:

$$\begin{array}{l}
 \omega: F(0,0,900) F(4,1,900) F(0,0,900) \quad \text{-----} \quad \text{Axioma} \\
 p_1: F(s,t,c): t=1 \text{ y } s \geq 6 \rightarrow F(\frac{2}{3}s, 2, c) F(\frac{1}{3}s, 1, c) \\
 p_2: F(s,t,c): t=2 \text{ y } s \geq 6 \rightarrow F(\frac{1}{3}s, 2, c) F(\frac{2}{3}s, 1, c) \\
 p_3: F(h,i,k) < F(s,t,c) > F(o,p,r): s > 3.9 \text{ ó } c > 0.4 \rightarrow F(s+0.1, t, c+(0.25*(k+r-(3*c)))) \\
 p_4: F(h,i,k) < F(s,t,c) > F(o,p,r): s \leq 3.9 \text{ ó } c \leq 0.4 \rightarrow F(0,0,900) \\
 p_5: F(h,i,k) < F(0,0,900) > F(o,p,r) \rightarrow H(1) \quad \text{-----} \quad \text{DOL} \\
 p_6: H(s): s < 3 \rightarrow H(s*1.1) \quad \text{-----} \quad \text{OL paramétrico}
 \end{array}
 \left. \begin{array}{l} \\ \\ \\ \\ \\ \\ \end{array} \right\} \text{-----} \quad \text{II. paramétrico}$$

Las dos primeras producciones describen la división de células vegetativas tipo 1 y 2. La producción 1 dice que si una célula es tipo 1 y su longitud es mayor o igual a 6, entonces se convertirá en 2 células:

- Una de tipo 2, con longitud $\frac{2}{3}$ de la que tenía originalmente y la misma concentración, y
- Otra de tipo 1, a un tercio del tamaño original y con la misma concentración.

La producción 2 dice que si una célula de tipo 2 tiene tamaño mayor o igual a 6, entonces se convertirá en 2 células:

- Una de tipo 2, tamaño un tercio del original con la misma concentración, y
- otra de tipo 1, con $\frac{2}{3}$ del tamaño original y la misma concentración.

La producción 3 muestra el proceso de transportación y decremento de los compuestos de nitrógeno. Ella dice que si existe una célula con tamaño mayor de 3.9 o concentración mayor de 0.4 que además se encuentre entre dos células, entonces se convertirá en una célula del mismo tipo, con tamaño igual al que tenía más 0.1, y su concentración será igual a la suma de las concentraciones de las dos células vecinas inmediatas menos 3 veces la concentración de la célula en cuestión. Esto se multiplica por 0.25, y al resultado se le suma la concentración que tenía originalmente la célula.

La producción 4 describe la diferenciación de una célula vegetativa a un heterocisto, que se convierte como tal en p_5 . La producción 4 dice que si existe una célula con tamaño menor o igual a 3.9 o concentración menor o igual a 0.4 y que además se encuentre en medio de dos células, entonces se diferenciará en una célula de tipo 0, con tamaño 0 y concentración de 900. En la producción 5, este tipo de célula (tipo 0, tamaño 0 y concentración 900) se convierte en un heterocisto ($H(1)$) siempre y cuando se encuentre entre dos células.

La última producción (p_6) describe el desarrollo del heterocisto, ya que cada vez que se encuentra uno de ellos crecerá a una tasa del 0.1 %.

De manera gráfica y abreviada, porque sólo se muestran los pasos mas importantes, se vería como se muestra en la figura 13:

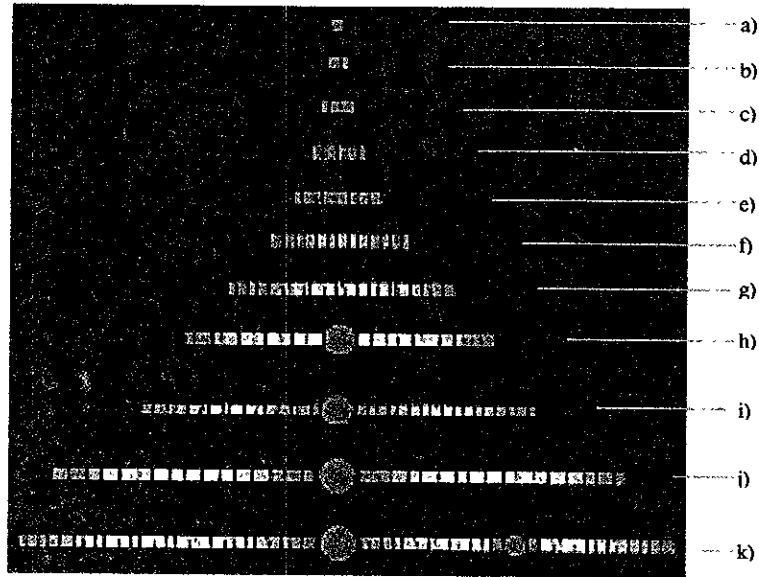


Figura 13. Ejemplo de la acción de hormonas en el crecimiento de *Anabaena catenula*.

Donde las células vegetativas son los cuadros de diferente tamaño y su color varía según la concentración de compuestos nitrogenados, y los heterocistos son las células que aparecen como círculos.

Analicemos la figura 13 con mayor detenimiento para entender como se llega a la representación gráfica mediante las 6 reglas de producción.

La célula inicial (señalada por el inciso a) en la figura 13) es el axioma o punto del cual se parte:

$$F(0,0,900) \quad F(4,1,900) \quad F(0,0,900)$$

donde la única célula visible es la de en medio, ya que las otras dos tienen longitud cero. De las 6 reglas de producción que rigen su crecimiento, la única que se puede aplicar es la tercera (P_3), ya que la célula de en medio ($F(4,1,900)$) cumple con las condiciones de ser una célula con tamaño mayor de 3 9 y concentración mayor de 0.4. La primera aplicación de la regla P_3 generará:

$$\begin{array}{c} F(0,0,900) \quad F(4,1,900) \quad F(0,0,900) \\ \downarrow P_3 \\ F(0,0,900) \quad F(4.1,1,675.00) \quad F(0,0,900) \end{array}$$

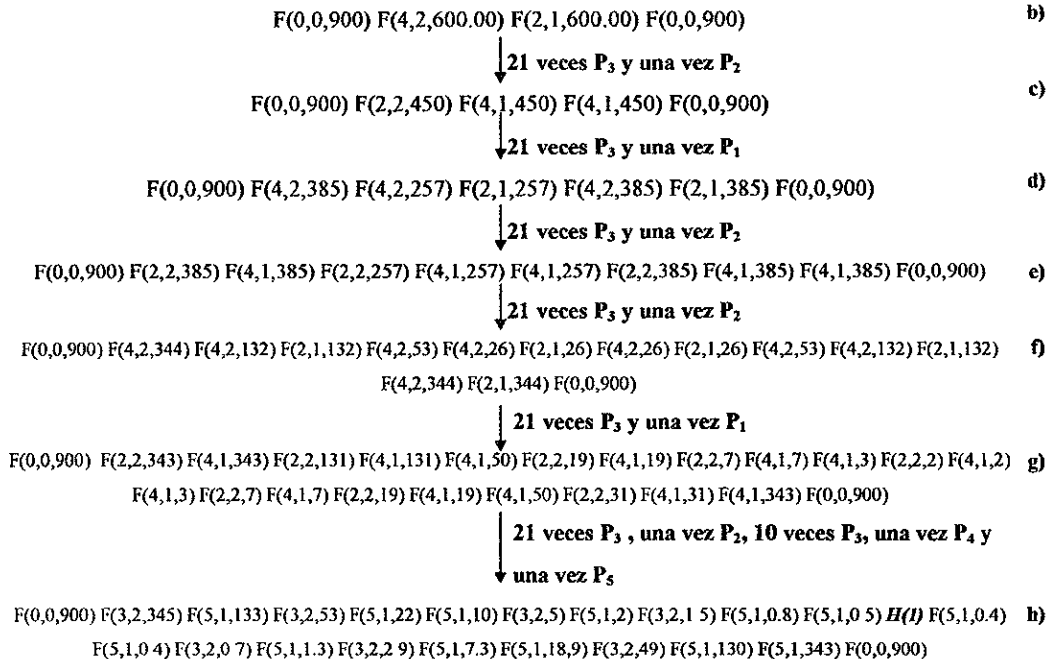
Si se siguieran aplicando la regla 3 consecutivamente, la secuencia generada sería:

$$\begin{array}{c} F(0,0,900) \quad F(4.1,1,675.00) \quad F(0,0,900) \\ \downarrow P_3 \\ F(0,0,900) \quad F(4.2,1,618.75) \quad F(0,0,900) \\ \downarrow P_3 \\ F(0,0,900) \quad F(4.3,1,604.68) \quad F(0,0,900) \\ \downarrow P_3 \\ F(0,0,900) \quad F(4.4,1,601.17) \quad F(0,0,900) \\ \downarrow P_3 \\ F(0,0,900) \quad F(4.5,1,600.29) \quad F(0,0,900) \\ \downarrow P_3 \\ \dots \\ \downarrow P_3 \\ F(0,0,900) \quad F(6,1,600.00) \quad F(0,0,900) \end{array} \quad \text{(Producción 3 aplicada 16 veces más)}$$

Al llegar a este paso, ya se puede aplicar la producción 2 porque la célula de en medio es de tipo 1 y de longitud igual a 6, por lo que:

$$\begin{array}{c} F(0,0,900) \quad F(6,1,600.00) \quad F(0,0,900) \\ \downarrow P_2 \\ F(0,0,900) \quad F(4,2,600.00) \quad F(2,1,600.00) \quad F(0,0,900) \end{array}$$

que resulta ser el inciso b) de la figura 13. Partiendo ahora de esta segunda figura (inciso b) de la figura 13) las subsecuentes figuras que aparecen y las reglas de producción aplicadas son las siguientes:



En esta última producción, el $H(1)$ denota el heterocisto que se forma y se nota en la figura 13. Los incisos y), j) y k), son muy largos para anotarlos en este espacio.

En resumen, los sistemas Lindenmayer se basan en la teoría revisada en este capítulo. Con el desarrollo de la computación, los sistemas Lindenmayer son capaces de mostrar plantas con la computadora en dos y tres dimensiones. Para entender como se dio este proceso, es necesario conocer las bases de la creación de software especializado y la historia de como los sistemas Lindenmayer utilizaron e integraron como herramienta útil la computadora, lo que se verá en el siguiente capítulo

CAPITULO III : MODELACIÓN DE LOS PATRONES DE DESARROLLO DE LAS PLANTAS UTILIZANDO LA COMPUTADORA.

Uno de los inventos que ha revolucionado nuestra vida, es la computadora. Desde su invención³⁰ en los años 40's, hasta nuestros días, la computadora ha pasado a ser una herramienta útil para cualquier profesionista, ya que con ella podemos hacer un sinnúmero de actividades, como almacenar información, procesar y analizar datos con diferentes fines (como en los análisis estadísticos), y graficar

Las gráficas por computadora³¹ datan de los primeros tiempos de la computadora, cuando los sistemas eran capaces de producir dibujos simples en impresoras de línea. Los científicos no tardaron en reconocer las ventajas de convertir grandes cantidades de datos en imágenes significativas y fácilmente comprensibles. Sin embargo, el uso generalizado de las gráficas por computadora se dio a finales de los setenta. Desde entonces, la graficación por computadora se ha convertido en parte integral de los negocios, la ingeniería, la medicina, las bellas artes y la ciencia³².

También Lindenmayer se dio cuenta de que, mostrar el desarrollo de las plantas mediante gráficas en computadora haría de más utilidad a los sistemas Lindenmayer, ya que estos se podrían representar mediante dibujos en dos y tres dimensiones y darles diferentes texturas, colores, matices para que se parecieran más a las plantas reales y así compararlos con estas. Además, la computadora podría aplicar las diferentes producciones en un sistema-L rápidamente, y posteriormente mandar una gráfica que muestre cómo se desarrolla la planta dependiendo del lenguaje utilizado. En este capítulo se relata cómo se enlazan los sistemas-L con la

³⁰ Existen algunas controversias entre el año en que se inventó la primera computadora, aunque todos los autores coinciden que se dio en los años 40's. Villaseñor menciona que fue en el año de 1947, Atelsek dice que fue en 1948 con la computadora llamada ENIAC, la cual podía resolver complejos cálculos necesarios para desarrollar la bomba de hidrógeno. Nuncio señala que esta computadora fue creada en 1945, y dentro de la historia de la computadora menciona antes a tres máquinas que antecedieron a la ENIAC: la Z1 (1938), que fue la primer calculadora electrónica con sistema binario; la MARK 1 (1943), que fue la primer calculadora electromecánica de tipo universal y podía multiplicar dos cifras de 23 dígitos cada una en tres minutos; y la COLOSSUS (1943), que fue una máquina criptográfica de alta velocidad, la cual podía descifrar las claves utilizadas por el ejército alemán en la segunda guerra mundial (Villaseñor y Murguía, 1992:130; Atelsek, 1993:1-6; y Nuncio, 1991:34-37) Este último autor ofrece una amplia cronología de la historia de la computadora)

³¹ El término graficación por computadora hace referencia a la representación de datos mediante magnitudes geométricas o figuras utilizando la computadora.

³² Dentro de las aplicaciones de la graficación por computadora se puede mencionar el diseño asistido por computadora, el cual comprende cualquier actividad en que una computadora interviene en el diseño de un producto. Otra son los sistemas de diseño gráfico y de pintura, con los cuales estudiantes y profesionistas del arte crean dibujos realistas o abstractos. (Para más información consulte a Berger, 1991:7-12)

computación; los tipos de hardware y software utilizados y potencialmente utilizables, y se mostrarán ejemplos de como se ha utilizado esta herramienta y su potencial en la investigación biológica.

3.1. Historia del enlace entre los sistemas Lindenmayer y la computación.

Prusinkiewicz y Hanan (1989:6) mencionan que en colaboración con científicos computacionales intentaron mostrar el desarrollo de las plantas vía computadora. En primer lugar, trataron de que las letras del alfabeto utilizado en los sistemas-L fueran representadas gráficamente como rectas largas o cortas. Posteriormente se tomaron en cuenta aspectos geométricos, como la longitud de los segmentos utilizados y los ángulos.

Prusinkiewicz se dedicó a desarrollar un programa que representara el crecimiento de las plantas basado en un *lenguaje de programación*³³: Logo³⁴. En este lenguaje, se pueden dibujar líneas por medio de los movimientos de una tortuga imaginaria, la cual se representa como un triángulo en la pantalla. La tortuga puede avanzar y retroceder, o bien girar a la derecha y a la izquierda.

La idea general del programa creado por Prusinkiewicz es la siguiente:

Se le llama un *estado* de la tortuga al lugar en que se encuentra sobre un plano. Este estado se define como una tripleta (x, y, α) , donde las coordenadas (x, y) representan la posición de la tortuga sobre el plano, y el ángulo α se interpreta como la dirección en la cual la tortuga se desplaza. Dado el tamaño de paso d y el ángulo de incremento δ , la tortuga puede responder a comandos representados por los siguientes símbolos:

- F La tortuga se mueve un paso de longitud d . El estado de la tortuga cambia a (x', y', α') , donde $x' = x + d \cos(\alpha)$ y $y' = y + d \sin(\alpha)$. Se traza una línea entre los puntos (x, y) y (x', y') .
- f La tortuga se mueve un paso de longitud d sin trazar línea.
- + La tortuga da un giro a la izquierda con ángulo δ . El siguiente estado de la tortuga es $(x, y, \alpha + \delta)$. La orientación positiva de los ángulos se da en contra de las agujas del reloj.
- La tortuga da un giro a la derecha con ángulo δ . El siguiente estado de la tortuga es $(x, y, \alpha - \delta)$.

³³ Un lenguaje de programación es un conjunto de caracteres, símbolos y reglas que permiten escribir las instrucciones que se le dan a una computadora.

³⁴ Logo fue creado en 1968 especialmente para introducir a los niños en la computación y enseñarles a pensar y expresar sus ideas mediante algoritmo que se pueda programar en un lenguaje sencillo y fácil de comprender. A este lenguaje se le incluyó un módulo denominado Turtle Graphics, donde una tortuga imaginaria (representada por un pequeño triángulo en la pantalla de la computadora) se puede mover hacia un extremo u otro de la pantalla. (Si se desea información general sobre este lenguaje consulte a Nuncio, 1991:107-108; para información más descriptiva del lenguaje de Logo vea a Moncho, 1986:32-47)

Para comprender mejor estos símbolos utilizados en el programa desarrollado por Prusinkiewicz, véase el ejemplo en la figura 14.

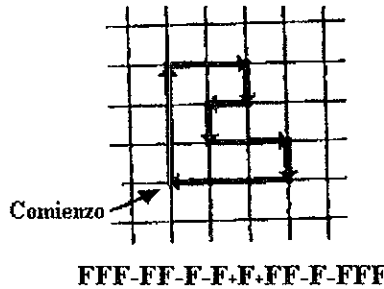


Figura 14. Interpretación de una cadena de símbolos.
El ángulo de incremento δ es igual a 90° .

Para graficar la ramificación de un árbol, dos símbolos se introdujeron para delimitar la ramificación. La tortuga los interpreta de la manera siguiente.

- [La tortuga adquiere un nuevo estado, mientras que el actual es guardado en la memoria. El nuevo estado de la tortuga contiene la posición y orientación de la tortuga, y posiblemente otros atributos como el color o el ancho de línea.
-] La tortuga regresa al estado anterior (el que se guardó al abrirse el corchete). Aunque no se dibujan líneas, el estado general de la tortuga cambia.

Un ejemplo de la utilización de estos símbolos se presenta en la fig. 15.

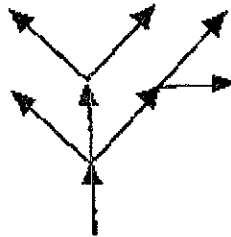


Figura 15. Ramificación en un árbol representado por la cadena $F[+F][-F[-F]F]F[+F][-F]$

Es importante conocer las bases utilizadas por Prusinkiewicz para hacer su primer programa, debido a que pueden servir como referencia en la creación de otros programas que utilicen a los sistemas Lindenmayer para simular el crecimiento de las plantas

Para modelar en tercera dimensión, Albelson y DiSessa³⁵ propusieron que la tortuga fuera representada en el espacio por tres vectores (H,L,V), donde:

H Representa la posición de la tortuga hacia adelante o hacia atrás del plano.

L Representa la posición de la tortuga hacia arriba o hacia abajo del plano.

V Representa la posición de la tortuga hacia la izquierda o la derecha del plano.

Estos vectores se representan en la figura 16, y una ejemplo de plantas creadas con estas bases se muestra en la fig. 17.

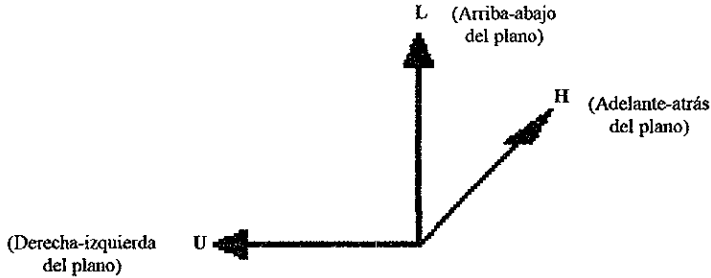


Figura 16. Representación de los tres vectores propuestos por Albelson y diSessa para representar a los sistemas Lindenmayer en tres dimensiones.



Figura 17. Estructura tridimensional de un arbusto.

³⁵ Citados por Prusinkiewicz, y Lindenmayer, 1990:19

Tomando como base estas ideas, desde 1986 a la fecha, Prusinkiewicz y sus colaboradores han desarrollado varios paquetes que pueden mostrar imágenes de plantas muy similares a las reales, como se verá en el siguiente apartado

3.2. Tipos de hardware y software utilizados y potencialmente utilizables en los sistemas Lindenmayer.

3.2.1 Hardware

Una de las principales diferencias entre los lenguajes formales y los sistemas Lindenmayer es el modo en que son aplicadas las producciones, ya que en los lenguajes formales se aplican secuencialmente y en los sistemas-L se aplican paralelamente.

Para aplicar estas producciones paralelas, los sistemas Lindenmayer pueden hacer uso de 2 tipos de *hardware*³⁶

- a) Computadoras con procesamiento simple (o de un procesador), y
- b) Computadoras con procesamiento en paralelo (o que tienen más de un procesador).

Una máquina está compuesta de varias partes, pero la parte fundamental que diferencia a una máquina con procesamiento simple de una con procesamiento paralelo es el número de procesadores que contiene. Los procesadores se encargan de coleccionar la información, calcular los resultados de esta información y pasarlos a memoria para que salgan. Cuando una máquina posee un procesador, solo puede hacer operaciones secuenciales, es decir, una tras otra. Cuando una computadora posee más de un procesador, puede hacer operaciones secuenciales o paralelas, es decir, varias a la vez, y para ello necesita software especial. Para entender mejor estos conceptos, imaginemos una escena clásica de una familia después de comer, donde las tareas (operaciones) a realizar son: limpiar la mesa, lavar los platos, secarlos y guardarlos. Los integrantes de la familia son cuatro: papá, mamá, hijo e hija. Los "procesadores" será el número de ayudantes para realizar las 4 tareas:

³⁶ El *hardware* está representado por los dispositivos electrónicos, eléctricos y mecánicos que se emplean para procesar datos.

Modelo secuencial usando un procesador.

Siguiendo el ejemplo, podemos representarlo de la siguiente manera:

Después de comer:

- Papá se lleva a los hijos al parque.
- Mamá limpia la mesa, lava los platos, los seca y los guarda

Modelo secuencial usando cuatro procesadores.

Después de comer:

- Papá limpia la mesa. Cuando termina:
- Mamá lava los platos. Cuando termina:
- Hija seca los platos. Cuando termina:
- Hijo guarda los platos.

Modelo paralelo usando cuatro procesadores.

Después de comer:

- Papá limpia la mesa y le va llevando los platos sucios a mamá;
- Mamá va recibiendo los platos sucios, los lava y se los va dando a hija;
- Hija va secando los platos limpios y se los va entregando a hijo;
- Hijo los va guardando.

El paralelismo implica que se van haciendo varias cosas al mismo tiempo.

3.2.2 Software

Los tipos de *software*³⁷ que se han utilizado se puede clasificar en dos:

- a) Lenguajes de programación.
- b) Software especializado.

3.2.2.1 Lenguajes de programación.**3.2.2.1.1 LOGO.**

Este fue el primer lenguaje utilizado para graficar a los sistemas Lindenmayer, como se vio anteriormente. Logo fue creado en 1968, y su nombre deriva de la raíz griega que significa *palabra*,

³⁷El *software* son las instrucciones y datos relacionados, almacenados en formato electrónico, que dirigen a la computadora en la ejecución de cierta tarea.

pensamiento. Fue creado especialmente para introducir a los niños en la computación y enseñarles a pensar y expresar sus ideas en un algoritmo que se pueda programar en un lenguaje sencillo y fácil de comprender. A este lenguaje se le incluyó un paquete denominado *Turtle Graphics*, donde una tortuga imaginaria (representada por un pequeño triángulo en la pantalla de la computadora) se puede mover hacia un extremo u otro de la pantalla. Logo es fácil de usar, aunque no puede manejar grandes volúmenes de datos. Logo es uno de los paquetes considerados como introductores para quien desea aprender a programar.

3.2.2.1.2 BASIC.

Su nombre es la abreviación de *Beginner's All purpose Symbolic Instruction Code*, y fue desarrollado en 1965. Basic llegó a convertirse en uno de los lenguajes de programación más utilizados en el mundo por su facilidad de uso, ya que estaba dirigido a estudiantes que quisieran introducirse al mundo de la programación y posteriormente quisieran aprender otro lenguaje de programación más complejo como FORTRAN. Hoy en día, se usa en forma común tanto para aplicaciones científicas como empresariales, e inclusive en juegos de video. Dentro de las variantes de Basic se encuentran QuickBasic y VisualBasic, de Microsoft. QuickBasic ofrece una interface conveniente para el usuario, que ayuda a simplificar y agilizar el proceso de desarrollo. VisualBasic se utiliza para generar aplicaciones en ambiente Windows. Basic es adecuado para aprender o realizar pequeños programas via la computadora, pero no para desarrollar sistemas de procesamiento de información. Utilizando este lenguaje de programación, un profesor de biología en Princeton, Henry S. Horn (citado por Robertson, 1989:73), desarrolló un programa llamado "Realtree" (Plantas reales), donde se pueden crear plantas en dos dimensiones. Horn se basó en el programa realizado por Prusinkiewicz en Logo y lo trasladó hasta Basic. "Realtree" está dirigido a estudiantes de primaria, quienes pueden crear árboles en la computadora mediante el programa.

3.2.2.1.3 FORTRAN.

Este programa se desarrolló en 1954, y su nombre deriva de las palabras en inglés *Formula Translator*. Antes de su creación los programas eran escritos por un reducido número de personas que conocían el funcionamiento de la computadora y como programarla, pero al aparecer este lenguaje de programación, que incluía un manual de referencia, un mayor número de personas tuvo acceso a la computadora y a crear sus propios programas. Poco a poco, profesionistas como los ingenieros y los matemáticos empezaron a utilizar la computadora como una herramienta de fácil acceso, sin tener que utilizar los servicios de un programador. Desde entonces, ha sido el lenguaje científico más popular.

3.2.2.1.4 Lenguaje de programación C.

El nombre de este lenguaje se debe a que fue el resultado de una serie de tres pasos para escribir un lenguaje de programación. El primero fue BCPL, escrito por Martin Richards; el segundo fue B, producido por Ken Thompson; y el tercero fue C, creado por Dennis Ritchie en 1972. A partir de 1980, y después de haber aparecido un artículo en una revista orientada a aficionados a las computadoras personales, "Dr Dobb's Journal", se empezó a utilizar con éxito y se volvió popular. Las ventajas que han contribuido a su popularidad son las siguientes:

1. Ofrece un extenso conjunto de facilidades para manejar una amplia gama de aplicaciones,
2. Maneja todos los tipos de organización de datos,
3. Tiene un complejo conjunto de operadores y un moderno control de estructuras,
4. Maneja una biblioteca de rutinas que facilita el manejo de datos de entrada y salida,
5. Es eficaz, compacto, y
6. Tiene un alto grado de transportabilidad, es decir, que se puede procesar con relativa facilidad en diferentes tipos de computadoras

De hecho, la mayor parte de los paquetes especializados que se dan a continuación se han escrito en C, como el caso de programa Pfg desarrollado por Przemyslaw Prusinkiewicz y James Hanan, que ocupa un espacio de 16k

3.2.2.2 Software especializado.

3.2.2.2.1 Pfg. (*Plant and fractal generator*)

Este programa, desarrollado por Przemyslaw Prusinkiewicz y James Hanan, contiene un conjunto de parámetros y de archivos con estructuras de plantas predefinidas para un sistema-L dado. Este paquete se basa en el lenguaje Logo. Puede ser usado para experimentar con el modelo interactivamente y presentar las secuencias de su desarrollo. Las imágenes obtenidas se pueden imprimir en impresora láser, o guardarlas para darle otro formato posterior.

3.2.2.2.2 Spiral (*Modeling program for phyllotactic patterns*).

Fue desarrollado por D.R. Fowler y sirve para modelar organismos con patrones filotácticos espirales. El usuario puede cambiar entre un modelo plano y un modelo cilíndrico, y modificar sus parámetros. En este programa se pueden utilizar sistemas-L paramétricos.

3.2.2.2.3 Ise (*Interactive surface editor*)

Desarrollado por J Hanan permite editar las estructuras prediseñadas en *Pfg* y en *Spiral* dependiendo de las necesidades del usuario.

3.2.2.2.4 Mapl (*Modeling program for cellular structures*)

Este programa, creado por F.D Fracchia, permite generar estructuras de organismos vegetales en dos y tres dimensiones a nivel celular. Los modelos generados pueden verse directamente en la pantalla o se pueden guardar para darles otro formato.

3.2.2.2.5 Panel (*Control panel manager*)

Panel es un programa desarrollado por L. Mercer y A. Snider que permite crear un panel de control que contenga reglas y botones de acuerdo a la configuración del archivo proveniente del usuario. Este programa puede ser usado para controlar parámetros usados en *Pfg*, *Spiral* o *Mapl*.

3.2.2.2.6 Rayshade (*Ray tracer*)

Creado por C. Kolb, de la Universidad de Yale, este programa crea imágenes de plantas con elementos geométricos primitivos como rectas, triángulos, cilindros, polígonos y esferas. Las plantas generadas pueden verse en 2 y 3 dimensiones.

3.2.2.2.7 CELIA (*Celular Linear Interactive Array simulator*)

Este programa, desarrollado por Prusinkiewicz, es el más utilizado hasta la fecha para crear plantas en 2 y 3 dimensiones utilizando computadoras paralelas.

3.2.2.2.8 AMAP

Este software está basado en la clasificación de Hallé y Oldeman³⁸ quienes reducen a 23 las variedades de ramificación de más de 100,000 especies de árboles. Incluye una biblioteca de órganos (diferentes tipos de hojas, flores y frutos) y una paleta de colores. Además, se le puede especificar diferentes parámetros como edad de la planta, ángulo de inserción de las ramas, y probabilidad de que una rama caiga o no del árbol.

Como se puede observar, existen diversos paquetes a utilizar para simular el crecimiento de las plantas utilizando los sistemas Lindenmayer, dependiendo si se quiere crear un programa propio o utilizar uno ya desarrollado y que cumpla nuestros objetivos.

³⁸ Citados por Bell, A. D., 1991-288. Se resume en una tabla esta clasificación en el capítulo 1 de este trabajo.

Nuncio (1991:196) comenta que desde el año de 1983, la incorporación del uso de las computadoras en la oficina, el hogar y la escuela llevaba un retraso de 5 años a comparación de lo que sucedía en Estados Unidos. De igual manera, la brecha tecnológica en cuanto a la creación de software en el país se acentúa con respecto a los E. U. y hace más distante la cultura informática y sus aplicaciones entre los dos países. De hecho, a finales de los años 70's. en Estados Unidos ya existía en todas las universidades una sólida cultura en computación, mientras que en México fue hasta los noventa que se empezó a dar esto. También el precio de las computadoras ha sido otro factor que ha retrasado el acercamiento a estas, ya que hasta 1985 empezaron a venderse computadoras a precios accesibles.

3.3. Ejemplos

Una planta virtual no es solo una imagen generada por computadora. Más que una imagen es un modelo en el sentido de que es una representación de una planta en un instante particular de su crecimiento.

Una planta virtual no se restringe a como se verá una planta, sino que incorpora medidas o procesos hipotéticos los cuales influyen en la arquitectura de la planta, como movimiento de metabolitos, hormonas, defensas químicas, patógenos y pesticidas que pueden ser simulados en relación de su morfogénesis y ser vistos con colores falsos o símbolos. También se pueden simular procesos externos independientes como la llegada del sol, patógenos, daño de pesticidas o el movimiento de insectos en plantas.

Johnson (1996:118), menciona que se puede modelar el orden de apertura de las flores en una planta utilizando los sistemas-L. Para modelar una planta en la cual aparecen los botones de abajo hacia arriba, y florece de arriba hacia abajo, lo cual se da por una hormona que es secretada en la raíz y que se dirige al ápice y las flores abren debido a una segunda hormona que se desplaza del ápice a la raíz.

Utilizando software que permita simular el desarrollo de las plantas, como el mostrado en la figura 18, Robertson (1989) reporta que Prusinkiewicz ha simulado su desarrollo dependiendo de la acción de ciertas hormonas y el efecto del CO₂ en el crecimiento de raíces. También menciona que los programas de generación de plantas hasta ahora desarrollados pueden ser usados por los biólogos, agrónomos, botánicos y paleontólogos, y dentro de las partes que modelan están el crecimiento de raíces, árboles, plantas herbáceas y enredaderas.

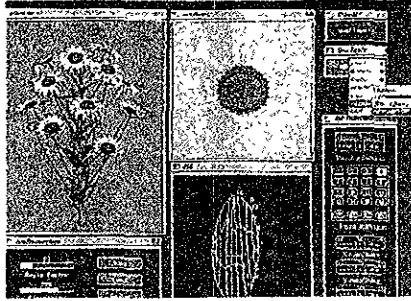


Figura 18. Aspecto del paquete llamado Isc.

CAPITULO IV: MÉTODO

4.1 Revisión de bibliografía.

Se buscó bibliografía en diversas bibliotecas y hemerotecas. No se encontró mucha información referente a los sistemas Lindenmayer, por lo que se realizó una búsqueda en Internet en dos etapas: en la primera se utilizaron las palabras claves de “Biología y cómputo”, “Modelación y crecimiento vegetal” y “Sistemas L”. Mucha de la información obtenida no tenían nada que ver con los temas delimitados, por lo que se realizó una segunda búsqueda en Internet a la dirección: <http://www.hotbot.com> donde se dio la palabra clave “Sistemas Lindenmayer” y se encontraron más de 1094 referencias diferentes, tanto de direcciones en que se trataba este tema como de bibliografía impresa o en proceso de impresión. Algunos de los artículos que aparecieron en esta búsqueda no estaban completos, por lo que se recurrió a buscar las revistas citadas en hemerotecas. Cabe mencionar que mucha información no se encontró en el país.

Posteriormente se procedió a la lectura de la información para seleccionar la más adecuada a los temas delimitados. Esta información se clasificó por temas en diferentes folders, los cuales llevaban el tema en cuestión marcado en la lengüeta.

La información se sintetizó en fichas de trabajo para facilitar su posterior lectura y análisis, las que se clasificaron dependiendo del tema que trataran. A partir de estas fichas se desarrollaron los capítulos I, II y III del presente escrito.

La exhaustiva búsqueda bibliográfica referente a los sistemas Lindenmayer y su aplicación en la fase experimental, se realizó con el fin de obtener un conocimiento general acerca del uso de estos. Se encontró que esta herramienta sirve para modelar el crecimiento de organismos vegetales utilizando la computadora, con la cual se puede predecir, por ejemplo, el rendimiento de un cultivo dependiendo de los niveles de fertilización, o del tipo de suelo; así como para mostrar el desarrollo de una planta y, poder llegar a proponer teorías acerca de los mecanismos que influyen en el crecimiento

También se encontró que los sistemas Lindenmayer están actualmente en desarrollo y se están trabajando con buenos resultados en otros países del mundo. De acuerdo a la búsqueda realizada, se encontró que en México no se utilizan a pesar del amplio potencial que tienen en la investigación biológica, posiblemente porque:

- 1) Se tiene poca información respecto a los sistemas Lindenmayer, o
- 2) En México no se cuenta con la infraestructura necesaria (software y hardware) para usar los paquetes o para crearlos.

Estas situaciones nos llevaron a dividir el trabajo en 3 fases, las cuales fueron.

1. Revisión bibliográfica, para establecer y desarrollar el marco teórico y referencial del trabajo, así como para conocer y analizar los fundamentos teóricos de los sistemas Lindenmayer.
2. Elaboración de programas que modelen el crecimiento de los vegetales utilizando los sistemas Lindenmayer y la computadora, para determinar que tan fácil es crear un programa utilizando la infraestructura disponible, y
3. Observación y reconocimiento de patrones de crecimiento en plantas, para determinar que tipo de plantas se pueden modelar

Para ello se comenzó a buscar información sobre qué eran los sistemas Lindenmayer, que tipos de sistemas-L existían y como era que generaban modelos de plantas. Haciendo una revisión más profunda de la bibliografía, su análisis y sintetización, se encontró que solo se comenta su aplicabilidad, pero no se mencionan los supuestos con los que se elaboran, por lo que consideró que era necesario conocer el funcionamiento de la herramienta para decidir si es útil y en que medida en la investigación biológica.

Posteriormente se procedió a desglosar y entender la gramática de los sistemas Lindenmayer, por lo que se planteó revisar varios ejemplos de interpretación de los caracteres y las reglas de producción de los sistemas Lindenmayer. Se hicieron ejercicios que se analizaron profundamente hasta lograr un entendimiento pleno de las gramáticas y las reglas de producción de los sistemas-L. Esto nos llevó a conocer mas a fondo la herramienta: los tipos de sistemas-L que existen y como se diferencian unos de otros.

La revisión de información que describían los tipos de sistemas Lindenmayer y su uso fue difícil de entender por que se presentaban diversos formalismos matemáticos, esto se leyendo la información y discutiéndola para aclarar dudas. Como el tema siguió presentando dificultades, se recurrió a buscar información referente a la teoría sobre la que se concibieron los sistemas-L: las gramáticas Chomsky. Este tema fue más accesible porque ya se tenían conocimientos acerca de como funciona el idioma utilizado por los humanos y por tanto de como las gramáticas Chomsky explican cualquier lenguaje. El entender las gramáticas Chomsky fue el punto de partida para retomar los diferentes tipos de sistemas Lindenmayer, y sus diferencias con las gramáticas Chomsky.

De esta manera se comprendió mejor la herramienta, en la que por último se analizaron los ejemplos nuevamente y se verificó que, teniendo una buena fundamentación de la notación usada y de como se generaba los modelos, quedaba claro cómo los autores habían manejado la herramienta.

Pero allí no quedó la búsqueda de información con respecto a los sistemas Lindenmayer, ya que con ellos se generan modelos del crecimiento de plantas y no se tenía un conocimiento claro de que son los modelos: si existían diferentes tipos, que se podía esperar de ellos, como saber si un modelo es válido y en donde caían los sistemas Lindenmayer. Para conocer estos temas nuevamente se recurrió a la bibliografía referente a modelos, tipos de modelos, validez, ventajas y desventajas, y así poder ubicar en este contexto a los sistemas Lindenmayer. Esta información sirvió para constatar que los modelos creados con los sistemas-L son adecuados para modelar el crecimiento de organismos vegetales.

Conjuntamente a la búsqueda de información referente a los sistemas-L, se estaba trabajando sobre como modela la herramienta la forma de los organismos vegetales en su crecimiento, es decir, su morfología. Esto nos llevó a plantear cuestiones importantes tales como:

- a) ¿Existen partes de una planta que tengan una forma constante? Responder a esta pregunta fue fundamental, porque los sistemas Lindenmayer representan las partes de una planta con letras constantes.
- b) ¿Existe un comportamiento predecible en la colocación de las partes de una planta? Esta respuesta nos haría mención de que tan real es el funcionamiento de las reglas de producción de los sistemas-L.
- c) ¿Qué factores afectan el crecimiento de las plantas? Este tema fue el mas importante, porque cualquier modelo que represente a la naturaleza debe tomar en cuenta los mismos factores que afectan a la planta en la realidad

Una vez aclarada la forma en que funcionan los sistemas-L, se procedió a afinar la parte de cómputo que es la herramienta adecuada en su optimización. Para ello se requería conocer como se enlazan estas dos herramientas y los requerimientos de hardware y software para este trabajo.

4.2 Elaboración de programas que modelen el crecimiento de los vegetales utilizando los sistemas Lindenmayer y la computadora.

Dentro de la información bibliográfica buscada, se encontró que el software utilizado para modelar el *crecimiento vegetal utilizando los sistemas Lindenmayer* no se encontraba en México, por lo que, o se buscaba en los Estados Unidos o Canadá, que es donde más se trabaja el tema, o se generaba un software utilizando un

lenguaje de programación. Se procedió a elaborar un software que modelara el crecimiento vegetal utilizando los sistemas Lindenmayer mediante los siguientes pasos:

4.2.1 Determinación del tipo de programas a elaborar.

El programa a crear debía tener las siguientes características:

- 1) **Mostrar el crecimiento de un organismo vegetal utilizando los sistemas Lindenmayer.** El organismo vegetal debía estar reportado en la bibliografía y su crecimiento ser claramente explicado en la fuente bibliográfica. Para esto se eligió el crecimiento del alga roja *Callithamnion roseum*, reportada por Lindenmayer y Jürgensen (1992:8-9), ya que cumplía con estas características. Posteriormente se tomó un ejemplo reportado por Chien y Jürgensen (1992:223-226) referente al desarrollo de una planta hipotética, esto con el fin de explorar las posibilidades de mostrar órganos como hojas, flores y frutos en dos dimensiones en computadora.
- 2) **Crear un programa utilizando un lenguaje de programación fácil de manejar por una persona no experta en el área de computación.** Para cumplir con este objetivo, se seleccionó un lenguaje de programación el cual fue Visual Basic, que además de ser de fácil manejo está en ambiente Windows y ya se tenían conocimientos sobre como programar y utilizar el lenguaje.
- 3) **Crear un programa utilizando una computadora adecuada.** Se utilizó para este fin una computadora Pentium, con procesamiento simple.

4.2.2 Elaboración de los programas.

Cabe aclarar que, aunque al inicio sólo se pensó en 2 programas, al finalizar el trabajo se tuvieron 4. Esto se debió a que en el primer programa elaborado del alga roja *Callithamnion roseum* se realizaron dos versiones. La primera versión abarcó la descripción del alga, y sirvió para explorar la dificultad de elaborar un programa teniendo en cuenta solo lo que se quería que el programa mostrara. La segunda versión del programa Alga se realizó con los conocimientos adquiridos en el análisis de la información referente a como Prusinkiewicz y sus colaboradores crearon los primeros programas que muestran el crecimiento vegetal utilizando los sistemas Lindenmayer.

El cuarto programa, denominado *Pruebagra*, se creó para cubrir la necesidad de mostrar otras formas de plantas utilizando una gramática similar a la utilizada en el programa Alga versión 2, ya que en esta se presenta el crecimiento del alga, pero no a otra forma de planta.

La secuencia de elaboración de estos fué la siguiente:

- a) Primera versión del programa Alga.
- b) Segunda versión del programa Alga
- c) Programa planta.
- d) Programa Pruebagra.

4.3 Descripción de la elaboración de programas.

4.3.1 Primera versión del programa Alga: *Alga1*.

En la elaboración del programa se siguieron los siguientes pasos:

1) Análisis del ejemplo a programar.

Se analizó como se desarrolla el Alga roja *Callithamnion roseum* para determinar que debía mostrar el programa a crear. Según Lindenmayer³⁹, una parte del desarrollo del talo del alga sigue las siguientes reglas de producción:

$$V = \{a, b, c\}, \quad \omega = \{b\}$$

$$P = a \rightarrow c[b]a \quad (1)$$

$$b \rightarrow a \quad (2)$$

$$c \rightarrow c \quad (3)$$

Donde:

ω es el axioma o punto de partida en el crecimiento del alga.

$P(1), P(2), P(3)$ son las reglas de producción.

a y b representan células apicales.

c representa una célula basal.

$[]$ y \downarrow significan puntos de ramificación.

El crecimiento del alga en una gramática se representa en la Figura 19.

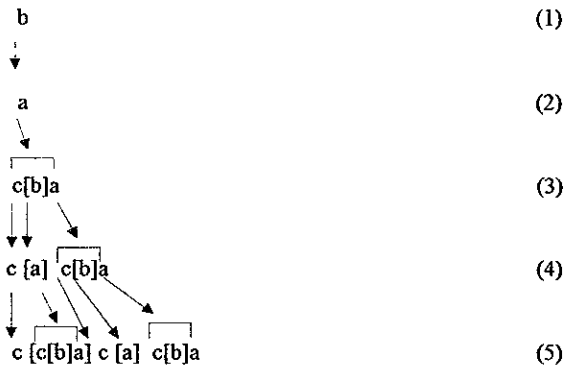


Figura 19. Gramática del crecimiento del alga roja *Callithamnion roseum*.

³⁹ Para mayor información véase a Lindenmayer y Jürgensen, 1992: 8-9

Donde (1) es el primer paso determinado por el axioma. En el siguiente paso (2) se aplica la regla de producción P_2 y b se transforma en a . En (3), se aplica la regla de producción P_1 y a se transforma en $c/b/a$. En (4) se aplican las tres reglas de producción, donde la c se queda como c , la b se transforma en a y la a en $c/b/a$. En (4) y (5) también se aplican las 3 reglas de producción. Gráficamente esto se muestra en la figura 20.

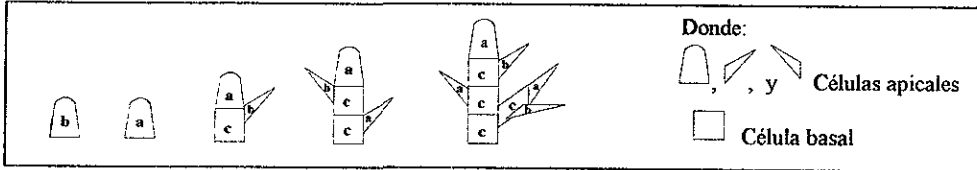


Figura 20. Crecimiento del alga *Callithamnion roseum*.

El programa debía mostrar este desarrollo en 2 dimensiones usando líneas para cada una de las células, ya que es la forma más sencilla de mostrar gráficos por computadora. El crecimiento del alga quedaría como muestra la figura 21:

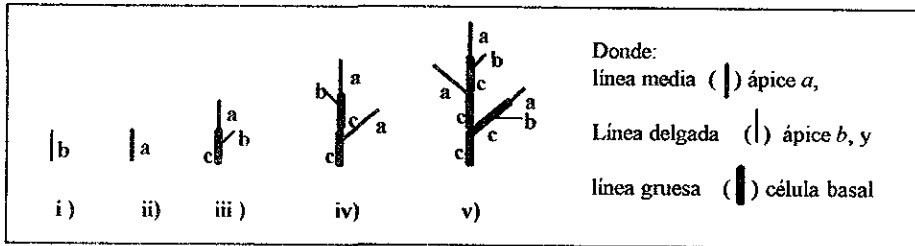


Figura 21. Crecimiento del alga *Callithamnion roseum* en dos dimensiones.

Una vez establecida la forma como se vería en la computadora, se procedió a diseñar el programa algal.

2) Diseño del programa a desarrollar.

Se determinaron las metas a satisfacer por el programa. La interface del usuario debía mostrar el crecimiento del alga en 2 dimensiones a una etapa dada por el propio usuario, mediante un botón de comando. Para esto, se diseñó la interface como muestra la figura 22.

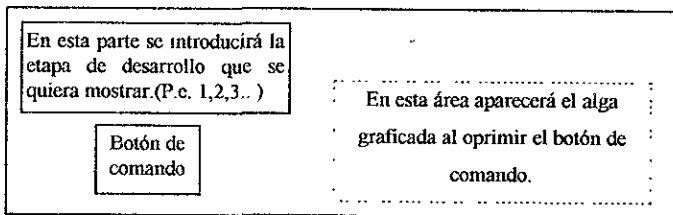


Figura 22. Interface de usuario para el programa Algal

3) Análisis del programa a crear.

El programa se dividió en tres partes para analizarlo:

- Datos de entrada.** Esta información comprendió el tipo de datos a introducir antes de correr el programa, esto es, ver en que etapa de crecimiento se desea ver el alga
- Procesamiento de la información.** Comprende los análisis matemáticos y/o lógicos contenidos en el programa con los datos de entrada. En el caso del programa alga, una vez capturados los datos de entrada, el programa genera el crecimiento gráfico del alga.
- Salida de datos.** Aquí se diseñó la salida de los resultados del programa después de su procesamiento.

4) Diseño de la lógica del programa

Se diseñaron los procesos del programa en papel. Primero, se debía analizar el desarrollo del alga y encontrar la forma de graficación en Visual Basic. Se partió de la figura 21 inciso iv y se realizaron dos pasos más, numerando las letras para hallar un patrón y poder utilizar estas letras numeradas como variables que almacenan un número dentro del programa de Visual Basic, y así poder graficar las diferentes líneas que componen el cuerpo del alga. En estas variables se guardarían las coordenadas x , y de las puntas terminales del alga graficada como muestra la figura 23:

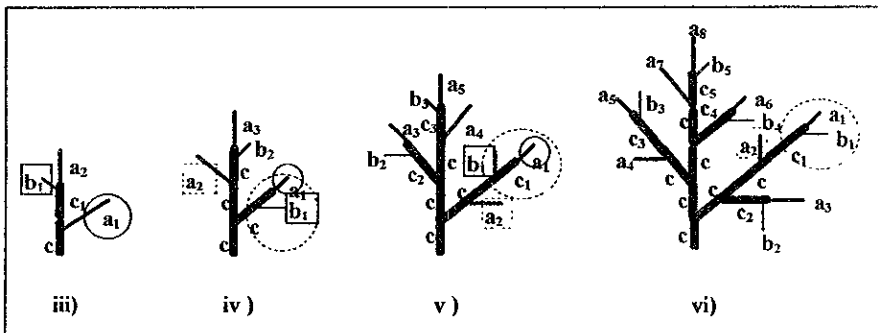


Figura 23. Crecimiento del alga *Callithamnion roseum* en dos dimensiones con células numeradas.

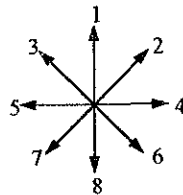
Para hallar un patrón, obsérvese en la figura 23 incisos *iii*, *iv*, *v* y *vi* que las a_1 (círculos en línea continua) generan siempre a_1 , b_1 , y c_1 (círculos en línea discontinua), según las reglas de producción P_1 , P_2 y P_3 . Del mismo modo, las b_1 (cuadrados en línea continua) forman a_2 (cuadrados en línea discontinua), y las b_2 forman a_4 , etc

Observando estos patrones se puede llegar a un arreglo, el cual se describe en el cuadro 3:

Cuadro 3. Arreglo para el crecimiento del alga *Callithamnion roseum*.

La letra:	Genera:	Y la letra:	Genera:
a_1	a_1, b_1, c_1	b_1	a_2
a_2	a_3, b_2, c_2	b_2	a_4
a_3	a_5, b_3, c_3	b_3	a_7
a_4	a_6, b_4, c_4	b_4	a_9
a_5	a_8, b_5, c_5	b_5	a_{12}
.	.	.	.
.	.	.	.
.	.	.	.
a_n	Si $n/3=0$ entonces: $a_{((n/3) \times 3)}, b_n, c_n$ Si $n/3=1$ entonces: $a_{((n/3) \times 3)+1}, b_n, c_n$ Si $n/3=2$ entonces: $a_{((n/3) \times 3)+3}, b_n, c_n$	b_n	$a_{(\text{Ent}((n/2)+1) \times 2) + (\text{Ent}((n/2)-0.5) \times 3)}$ *

Una vez determinadas las relaciones entre las variables a utilizar y como se generan unas de otras, se procedió a determinar la manera en que se debían trazar las líneas con un ángulo de 45° en el ambiente gráfico de Visual Basic. Para ello se determinó que podían seguir una de ocho direcciones diferentes



Tomando en cuenta la dirección y la letra correspondiente (si se graficaba una a, una b o una c) se determinaron las ecuaciones de cada línea. Se da un ejemplo para la línea de tipo 1:

Si la letra es c entonces.

$$\text{line}(x, (y-0.5))-(x, (y+0.5))$$

Si la letra es b y se debe trazar a la derecha entonces:

$$\text{line}(x, (y+0.5))-(x+\sqrt{0.125}, ((y+0.5)+\sqrt{0.125})), \text{ el tipo de línea generada es 2}$$

Si la letra es b y se debe trazar a la izquierda entonces:

$$\text{line}(x, (y+0.5))-(x-\sqrt{0.125}, ((y+0.5)+\sqrt{0.125})), \text{ el tipo de línea generada es 3}$$

* Ent() se refiere a la parte entera del resultado que arroje la operación dentro del paréntesis.

Si la letra es a entonces:

line $(x, (y+0.5))-(x,(y+1))$, el tipo de línea generada es l

Cabe mencionar que cada dos ramas generadas sobre el mismo tallo, una debe ir a la izquierda y otra a la derecha. Para lograr tal efecto se generó otra variable que toma los valores de +1 y -1 alternativamente, el primer valor indica que la rama se grafica a la derecha y el segundo a la izquierda, del eje principal.

Determinadas las variables y el tipo de líneas a utilizar, se conjuntaron ambos aspectos y se procedió a la codificación del programa.

5) Codificación del programa

El diseño detallado del programa se trasladó al lenguaje de programación utilizado, en este caso Visual Basic, el cual se muestra en el anexo I.

6) Prueba y depuración del programa

Se procedió a ejecutar el programa y se depuraron los errores de sintaxis y lógicos que se encontraron para el adecuado funcionamiento del programa.

4.3.2 Segunda versión del programa Alga: Alga2.

1) Análisis del ejemplo a programar.

Retomando el ejemplo del alga roja *Callithamnion roseum*, se consideró como generar las cadenas de caracteres de acuerdo a como funcionan los sistemas Lindenmayer, para posteriormente interpretar la cadena generada y así graficar el alga. Si se mostraran las cadenas según el crecimiento del alga, la tercera estructura del desarrollo del alga se puede escribir como $c[b]a$, la cuarta como $c[a]c[b]a$ y la quinta como $c[c[b]a]c[a]c[b]a$ como muestra la figura 24:

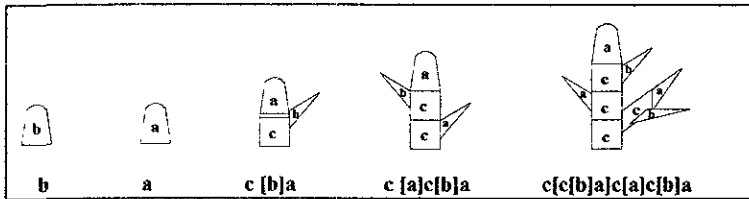


Figura 24. Cadena de caracteres para cada etapa del crecimiento del alga *Callithamnion roseum*.

Al respecto, Lindenmayer y Jürgensen (1992:9) comenta que para escribir la estructura n-ésima del desarrollo de esta alga se puede utilizar la siguiente fórmula recursiva:

$$S_n = c[S_{n-2}]S_{n-1}$$

donde se necesita saber como están estructuradas las dos cadenas anteriores para escribir la cadena n-ésima.

Por otro lado, para graficar la manera como se ramifica el alga se tomó como base la propuesta hecha de manera gráfica por Prusinkiewicz. Dentro del programa, se necesitaba trazar una línea de un punto x,y en el plano, a otro punto al que llamaremos x_1,y_1 utilizando un cierto ángulo lo cual representa una dificultad por que puede ser una x,y cualquiera. Además, esta ramificación se podría dar a la izquierda o a la derecha del punto x,y . Prusinkiewicz propone la utilización de las siguientes fórmulas para mostrar la ramificación de una línea:

$$\begin{aligned} x_1 &= x + d \cos \alpha \\ y_1 &= y + d \operatorname{sen} \alpha \end{aligned}$$

donde:
 α es el ángulo que tiene la línea que se va a trazar con respecto al ángulo que tiene la línea previamente trazada, y
 d es la longitud de la línea a trazar.

Prusinkiewicz también propone el uso de los signos:

- Para señalar que la ramificación se da a la derecha del punto x,y . El nuevo ángulo que tendría la línea trazada sería $\alpha - \delta$, donde δ es el ángulo de ramificación respecto al punto x,y .
- + Para señalar que la ramificación se da a la izquierda del punto x,y . El nuevo ángulo que tendría la línea trazada sería $\alpha + \delta$, donde δ es el ángulo de ramificación respecto al punto x,y .

Estos signos tienen una razón de ser. Veamos un ejemplo de graficación usando las ecuaciones y los signos propuestos por Prusinkiewicz

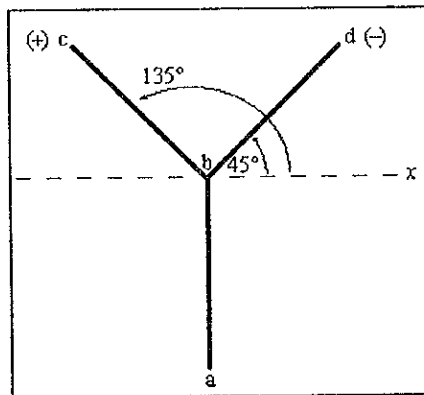


Figura 25. Ejemplo de graficación utilizando las bases propuestas por Prusinkiewicz.

En la figura 25, la línea ab tiene un ángulo de 90° ($\alpha=90^\circ$), y el punto b sería el equivalente al punto x,y . El ángulo de ramificación supongamos que es de 45° ($\delta=45^\circ$) y la ramificación puede ser a la derecha (línea bd) o a la izquierda (línea bc). Como el ángulo con que se va a graficar la línea es respecto al eje de las x

(línea punteada), si la ramificación es a la derecha se restarían 45° (es decir que, $\alpha - \delta = \alpha$, $90^\circ - 45^\circ = 45^\circ$) mientras que si es a la izquierda, el ángulo de ramificación aumenta (es decir que $\alpha + \delta = \alpha$; $90^\circ + 45^\circ = 135^\circ$) con lo que las ecuaciones serían:

$$x_1 = x + \delta \cos \alpha$$

$$y_1 = y + \delta \sin \alpha$$

con $\alpha = 45^\circ$ si la ramificación es a la derecha y

$\alpha = 135^\circ$ si la ramificación es a la izquierda

por lo que la ramificación se dará de la línea $b(x, y)$ a la línea d o $c(x_1, y_1)$.

A diferencia de la primera versión, este programa debe, por una parte, generar una cadena de caracteres a partir de un estadio de crecimiento del alga dado por el usuario, y por otra parte, leer la cadena de caracteres y de allí graficar el crecimiento del alga.

En la primera parte del programa, identificada con el nombre de "Mostrar cadena de caracteres", el usuario puede elegir el estadio de desarrollo que quiera ver. Un botón de comando generara la cadena de caracteres, y existe un lugar donde esta se muestra (figura 26).

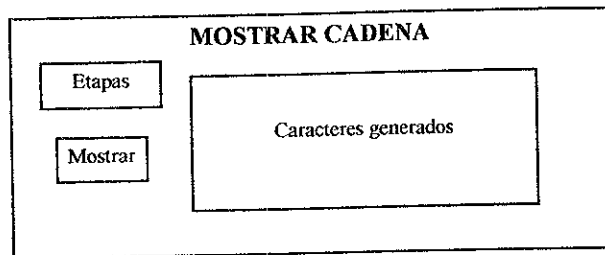


Figura 26. Interface de usuario para el programa Alga2 en la creación de cadenas.

Posteriormente se analiza cada uno de los caracteres y se grafica el alga al oprimir un segundo botón de comando. Esto se muestra en la figura 27:

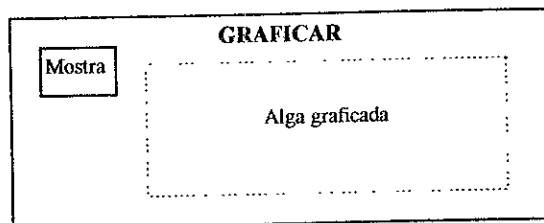


Figura 27. Interface de usuario para la graficación del alga.

Así, el programa se dividió en dos partes:

a) Mostrar cadena de caracteres.

Datos de entrada: El número del estadio que el usuario quiera ver

Procesamiento de la información: Al oprimir el botón de comando, la computadora crea la cadena de caracteres.

Datos de salida: Muestra la cadena de caracteres.

b) Graficar.

Datos de entrada: Es la cadena de caracteres generada en la etapa “mostrar cadena de caracteres”.

Procesamiento de la información: Mediante el botón de comando se analiza toda la cadena y se grafica el alga.

Datos de salida: Dibuja el crecimiento del alga en dos dimensiones.

La lógica del programa considera también la fórmula recursiva planteada por Lindenmayer y Jurgensen (1992:9):

$$S_n = c\{S_{n-2}\}S_{n-1}$$

Es decir, que el cuarto estadio del desarrollo del alga utilizando la fórmula queda como:

$$S_2 = c\{S_1\}S_1$$

$$S_4 = c\{c\{b\}a\}c\{a\}c\{b\}a$$

Para mostrar hacia que lado se ramifica, se adicionaron los signos + y - para graficar las ramas a la izquierda y a la derecha del eje principal, basándose en lo reportado por Prusinkiewicz⁴⁰ y por Cornell (1994:204,213). La cadena entonces queda:

$$S_2 = c\{-b\}a$$

$$S_3 = c\{-a\}c\{+b\}a$$

$$S_4 = c\left[\underbrace{-c\{-b\}a}_{S_2} \right] c\left[\underbrace{+a\}c\{-b\}a}_{S_3} \right]$$

De acuerdo a la fórmula usada anteriormente, el segundo miembro (S_3) del cuarto estadio cambia sus signos. El programa debe ser capaz de cambiar estos signos y crear cadenas de caracteres nuevas dependiendo del estadio de desarrollo elegido por el usuario. El diseño hace lo siguiente

⁴⁰ Véase a Prusinkiewicz y Hanan, 1990: 6-21; y Prusinkiewicz y Hanan, 1992.195.

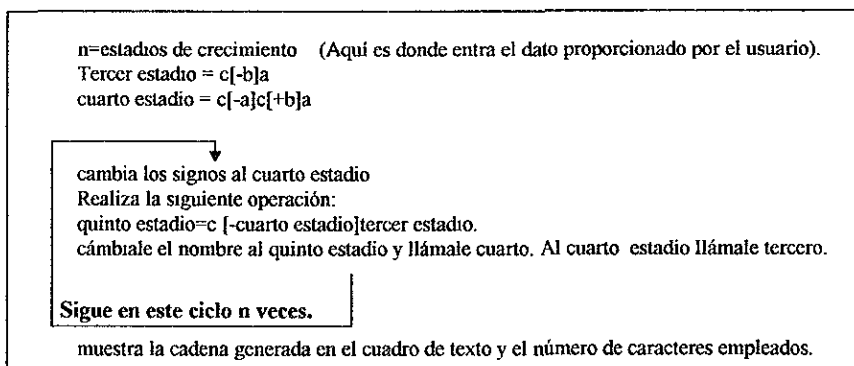


Figura 28. Diseño de la parte "Mostrar cadena de caracteres" en el programa Alga2.

Para la segunda parte (Graficar), el programa analiza la cadena de caracteres generados y grafica el alga conforme lo indica la cadena.

Como primer paso se debe saber cuantos caracteres contiene la cadena, para después analizarlos uno a uno, y dependiendo del tipo de letra con la que se encuentre (a, b o c) es la longitud y color con que debe mostrar una línea. Si se encuentra con un corchete abierto ([]) la computadora debe interpretar que existe una ramificación y preparar el ángulo con el cual va a ramificarse ($\alpha = \alpha + / - \delta$). Si el carácter es un corchete cerrado (]) debe interpretarlo como el final de la ramificación. Los signos de + y - los debe interpretar como ramificaciones a la derecha o a la izquierda respectivamente. Con esta información se procedió a codificar el programa

La primera parte del programa opera como lo muestra la figura 28, y su código se encuentra en el anexo II. Con respecto a la segunda parte, la correspondiente al botón "graficar", cabe aclarar algunas partes del código

El ángulo base para ramificar el alga fue de 90° ($\alpha = 90^\circ$) ya que la primer célula a graficar (una c) tiene la dirección vertical de abajo hacia arriba. Este ángulo se convirtió a radianes, ya que Visual Basic solo acepta radianes y no grados (renglón 11 del código) El ángulo de ramificación (δ) puede ser dado por el usuario en un cuadro de texto destinado a ello (renglón 12) el cual también es convertido a radianes. Los caracteres de la cadena son analizados uno a uno (renglón 18) y dependiendo del tipo de carácter (a, b, c, [,], +, -,) es lo que hace la computadora.

En caso que sea alguna letra (a, b, c) el programa calcula el punto al cual debe graficar la línea (renglones 20-23, 25-28, 29-32 para las letras c, a y b respectivamente) con la longitud y color dados por el usuario (renglones 6 y 7 del código).

Posteriormente se procedió a ejecutar el programa y se depuraron los errores de sintaxis y lógicos que se encontraron para que el programa funcionara adecuadamente

4.3.3 Programa *Planta*.

El segundo ejemplo que se trató de modelar fue el reportado por Chien y Jürgensen (1992:223-226) sobre el crecimiento de una planta artificial, para explorar la forma de mostrar diferentes partes de una planta en la computadora. Siguiendo el desarrollo de la planta, esta crece y le salen hojas al tallo principal, después botones que se transforman en flores. Estas flores se marchitan, caen y en su lugar crecen frutos, como lo muestra en 8 ilustraciones⁴¹ de lo que hace su programa. En este caso no se siguieron reglas de producción ni gramáticas, ya que el programa fue un tanto exploratorio para graficar partes de una planta, como hojas, flores, frutos y tallos por medio de elipses, círculos y líneas. El ejemplo modelado difiere del reportado por Chien en los últimos pasos, ya que al reportado no se le caen las hojas y al desarrollado, sí.

En este programa se tuvieron en cuenta las siguientes consideraciones:

- a) **Datos de entrada.** Los datos de entrada son el número del estadio que desee ver el usuario.
- b) **Procesamiento de los datos.** Mediante un botón de comando, la computadora debe graficar el crecimiento de la planta.
- c) **Datos de salida.** Los datos de salida son la graficación de la planta.

El programa fue diseñado con el objetivo de ser capaz de graficar el desarrollo de una planta dependiendo del número de estadio en que el usuario quiera ver. Para ello, la interface de usuario generada fue como se muestra en la figura 29:

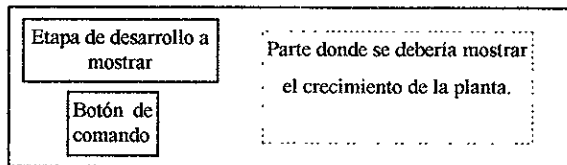


Figura 29. Diseño del programa *Planta*.

⁴¹ Si desea ver estas ilustraciones comentadas vea a Chien y Jürgensen 1992:224-225.

En la lógica del programa se pensó que este debe mostrar el crecimiento de la planta dependiendo del estadio, igual que en el ejemplo de Chien y Jürgensen.

En los primeros 3 estadios, la planta debe crecer y mostrar una rama a cada lado de su eje principal con una hoja. Del estadio 4 al 9, deben aparecer ramas a las cuales posteriormente, del estadio 10 al 25, le crecen botones y flores. Del estadio 26 al 38, los frutos deben reemplazar a las flores y crecer. En los estadios 39 al 41, las hojas se deben volver amarillas, y caerse del estadio 42 al 44. Los frutos caen del estadio 45 al 53, donde termina la vida de la planta. Las ramas se hicieron de líneas, las hojas de elipses al igual que los frutos, y las flores de dos elipses y un círculo. Cada etapa de su desarrollo depende de un número de estadio determinado por el usuario.

Una vez hechas estas consideraciones, se procedió a escribirlo en el lenguaje de Visual Basic. El código del programa se encuentra en el anexo III, y se ejecutó para depurar los errores de sintaxis y lógicos para su adecuado funcionamiento.

4.3.4 Programa *Pruebagra*.

Este programa se desarrolló para graficar formas vegetales de manera diferente al hecho con el programa *alga2* utilizando una gramática creada con los sistemas Lindenmayer para conjuntar la forma de ramificación de un organismo vegetal (como en el ejemplo del alga) y por otra parte los dibujos de partes diferentes a las ramas como en el programa de *Planta*. En la creación de este programa no se tenía un ejemplo a programar, ya que el objetivo era graficar estructuras de plantas sencillas, sin tener un organismo en especial, como estrategia para explorar la facilidad de describir la forma de las plantas.

Para su diseño se tomaron en cuenta las siguientes consideraciones:

- a) **Datos de entrada.** Fue la gramática escrita.
- b) **Procesamiento de los datos.** Mediante un botón de comando, la computadora analiza la gramática escrita y grafica el crecimiento de la planta.
- c) **Datos de salida.** Los datos de salida son la graficación de la planta.

Se consideró que el programa debe contener un espacio en el cual se debe escribir la gramática a seguir. Mediante un botón de comando, la computadora analiza la gramática y la muestra. Esto se puede observar en la figura 30:

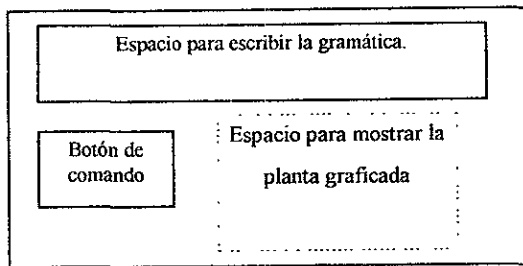


Figura 30. Diseño del programa Pruebagra

El objetivo de este programa es muy similar al programa Alga2, solo que sin la creación automática de la cadena de caracteres, por que esta la debe teclear el usuario.

Además, el programa debe graficar otras figuras geométricas como el círculo y la elipse a las que se les pudiera designar el color y el tamaño

Para su creación, se tomó al programa Alga2 del que se hizo una copia. A la copia se le llamó Pruebagra. Se le quitó el botón de comando que creaba la cadena de caracteres, y los cuadros de texto donde se daban el numero de pasos y el número de caracteres con sus correspondientes etiquetas ya que no se necesitaban. La etiqueta que dice "creación de la cadena de símbolos" se cambió por "escriba la cadena de símbolos"

Posteriormente se adicionó código para que el programa fuera capaz de mostrar un círculo y una elipse.

El código del programa se encuentra en el anexo IV, y es muy similar al del botón "graficar" de la segunda versión del programa Alga. Lo que los hace diferentes son las líneas adicionales que tiene el botón de el programa Pruebagra. Se adicionaron las líneas 42-43 para que al escribir una "d" en la gramática el programa grafique un círculo, y la 44-45 para que el programa reconozca a una "e" como una elipse. Se ejecutó el programa y se depuraron los errores de sintaxis y lógicos que se encontraron para que el programa funcionara adecuadamente.

4.4 Creación de la versión ejecutable de los programas elaborados.

Visual Basic 4.0 genera una versión ejecutable de los programas hechos mediante un asistente de instalación. La función del asistente es detectar los posibles errores del código del programa; convertir los dos archivos de este (con extensiones *.vbp* y *.frm* respectivamente) en un solo archivo ejecutable (con extensión

exe); determina que archivos necesita el programa para ejecutarse en una computadora que no tenga Visual Basic; y compactar estos archivos junto con el archivo ejecutable para que se guarden en discos flexibles.

Los cuatro programas creados se hicieron ejecutables con este asistente. Los programas se instalan como una aplicación de Windows.

4.5 Observación y reconocimiento de patrones de desarrollo en plantas.

Con la finalidad de conocer qué tan fácil era encontrar un patrón de desarrollo en plantas, describirlo usando los sistemas Lindenmayer y graficarlo usando el programa Pruebagra, se observaron varios especímenes en el jardín botánico “De la Cruz Badiano” de la FES Zaragoza, analizando su estructura. Para ello se siguieron las etapas de:

1) Selección de una planta.

Se seleccionó una planta que cumpliera las características de no estar densamente ramificada, no ser muy grande (ser un árbol joven, arbusto o herbácea), ya que se trataba de encontrar un patrón no muy complicado para poder escribirlo y de allí partir hacia otros patrones más difíciles de observar. Los especímenes seleccionados fueron un árbol de ciprés (*Cupressus lindleyi* Kotsch) y una planta de hinojo (*Foeniculum vulgare* Will.)

2) Observación de la planta para encontrar un patrón.

En esta etapa se observó a la planta dividiéndola en sus partes más simples:

- a) *Ramificación.* Se observó si las ramas seguían un patrón con respecto a la rama principal. De allí se observó la ramificación de las ramas terciarias con respecto a las secundarias y así sucesivamente.
- b) *Disposición de las hojas en la rama.* Se observó si las hojas tenían cierta disposición sobre la rama y cual era este
- c) *Tipo de estructura reproductiva.* Se determinó el tipo de inflorescencia o flor que tenían las plantas y en que parte de ellas se encontraba, así como su aspecto y color.

3) Conjunción de los patrones observados para mostrar a la planta en general.

Se unieron las diferentes partes observadas para describir a la planta en general.

4) Escritura de la gramática de la planta.

A partir de los patrones encontrados se propuso una gramática utilizando los sistemas Lindenmayer para describir a la planta observada.

5) Prueba de la gramática en la computadora con el programa Pruebagra.

Se escribió la gramática basada en los sistemas Lindenmayer en el programa "Pruebagra" para ver si la imagen generada se parecía a la real en su estructura básica, ya que el programa muestra a la planta en dos dimensiones.

CAPITULO V: RESULTADOS

Los resultados dependieron de las actividades principales del trabajo, que fueron: búsqueda, recopilación, organización del material bibliográfico y análisis, elaboración de programas que modelen el crecimiento de los vegetales utilizando los sistemas Lindenmayer y la computadora y la observación de patrones de crecimiento en plantas

5.1 Material bibliográfico.

Referente a los sistemas Lindenmayer, se encontró poca bibliografía, de hecho, solo 6 referencias a cerca de los sistemas Lindenmayer (Donovan, 1986; Fu, 1974, Herman y Rozeberg, 1975; Lindenmayer y Jürgensen, 1992; Prusinkiewicz y Hanan, 1989 y Prusinkiewicz y Hanan, 1990), aunadas a otras 9 referencias que explican su aplicación (Johnson, 1996; Kron y Stewart, 1994, Mahoney, 1993; Merril, 1989; Prusinkiewicz y Lindenmayer, 1990, Prusinkiewicz *et al*, 1994, Reeting, 1993; Robertson, 1989 y Valdes, 1991). El tema más frecuente en cuanto a referencias fue el de modelos y sus tipos con 6 referencias (Achinstein, 1987; Lapedes, 1981; Pfaffenberger, 1996; Pidd, 1988, Shannon, 1975 y Tejera, 1994).

Fu (1974 25-46) da una idea muy clara y completa de lo que son los lenguajes formales desarrollados por Chomsky ilustrándolos con ejemplos También explica las diferencias entre los diferentes tipos de gramáticas para que el lector pueda comprenderlas sin ser un experto en el área. Aunque existe información en español, la traducción suele llevar algunos errores debido a que el traductor no conoce a fondo el tema

Las diferencias encontradas entre los *lenguajes formales* y los *sistemas Lindenmayer*, no son establecidas de manera puntual por algún autor De hecho, es preciso conocer como funcionan los sistemas Lindenmayer y los lenguajes formales para poder establecer dichas diferencias mediante una comparación entre uno y otro, como se realizó durante el trabajo

Los diferentes tipos de sistemas Lindenmayer son ampliamente explicados en Lindenmayer y Jurgensen (1992), Prusinkiewicz y Hanan (1989, 1990) y Herman y Rozenberg (1975). La información contenida en estas citas parece repetitiva, ya que difiere muy poco entre ellas. A pesar de esto, la cita más completa es la de Prusinkiewicz y Hanan (1990). En ella se aprecia que, los sistemas Lindenmayer son una herramienta teórica por su origen y que no dependen de la computadora necesariamente para su aplicación y uso. También se puede ver que esta herramienta genera modelos descriptivos, y que pueden tener una parte matemática, una parte lógica y una más probabilística.

La mayor parte de la información fue difícil encontrarla, debido a que pertenece al área de cómputo. A través de la revisión de esta información se observó que la herramienta se está utilizando más por expertos en computación que por expertos en botánica o desarrollo vegetal, lo cual es un problema por que los modelos generados no están siendo aplicados en la investigación biológica, esto es, que el modelo se genera de datos biológicos, se apega a la realidad, pero no le sirve al experto en desarrollo vegetal por que el modelo no fue creado con ese propósito. Lo que hacen los expertos en computación es recabar datos de una planta, crear el modelo y luego decir que el modelo es bueno y que necesitan que un investigador biológico tome el modelo y lo utilice, ya que el experto en computación generalmente sabe muy poco de biología.

Otros autores más osados, han tratado de utilizar su modelo generado, y han llegado a conclusiones tan obvias que, desde el punto de vista biológico no necesitaban ninguna clase de experimento para llegar a esas conclusiones, como es el caso de un investigador que concluyó que “la naturaleza tiende a la ley del mínimo esfuerzo”.

En el tema destinado a los modelos y tipos de modelos, se encontraron términos no muy claros que necesitaban de buscar otras definiciones diferentes a la que daba el autor, como el caso de una “modelación” y una “simulación”. Tejera (1994:1177) maneja como sinónimos a la modelación y a la simulación, mientras que Shannon (1975:4) establece una clara diferencia entre modelar, que es definida como la creación de un modelo, y simular, que es el utilizar el modelo creado. No se encontró dificultad para entender el tema tratado, ya que la consulta de los diferentes autores fue enriquecedora, por que todos se complementaban.

En cuanto a los temas buscados de biología, existe gran cantidad de bibliografía. En el tema acerca de la existencia de patrones en vegetales, se encontraron más de 14 referencias de las cuales 7 fueron las más completas y referenciadas (Bell, 1991; Greulanch y Adams, 1990, Jones, 1988; Lyndon, 1990, Newman y Boles, 1992; Prusinkiewicz y Hanan, 1989 y Steeves y Sussex, 1991); en el tema de los factores que afectan el crecimiento vegetal, se encontraron 8 referencias de las cuales 3 fueron las más completas y por tanto referenciadas (Chien y Jürgensen, 1992; Fuller, 1974 y Nultsch, 1975)

En la revisión y análisis de este tema, se encontraron algunos términos que se podían confundir. Tal es el caso de la morfología y la morfogénesis, conceptos que Robbins (1976:301) define como sinónimos. Lawrence (1989:326) no establece una distinción clara entre morfología y morfogénesis. Se logró conceptualizar qué era morfología y como se relacionaba con la morfogénesis utilizando las definiciones dadas por Font (1977:735) y Parquer (1985:234) respectivamente, ya que fueron las más completas y claras. De la

bibliografía consultada referente al tema del surgimiento de patrones en vegetales, uno de los mejores libros fue el de Lyndon, ya que en él se ven los diferentes patrones desde un punto de vista microscópico.

Respecto al tema de los factores que afectan el crecimiento vegetal, se encontró una amplia polémica acerca de que tipo de factor afecta más: si el genético o el ambiental. Fuller (1974:201) afirma que los factores genéticos juegan un papel dominante en la forma que tendrá la planta, más que los factores ambientales que la rodean; Chien y Jürgensen (1992:213-229) afirman que depende del tipo de planta, si es leñosa o herbácea; y por último, Nultsch (1975:398) menciona que no hay porqué discutir que es mas importante, si los factores genéticos o los factores ambientales, ya que depende del tipo de planta, punto de vista con el cual estamos de acuerdo, ya que, para crear el modelo del desarrollo de una planta, primero se debe estudiar a esta.

En cuanto al uso de la computadora para generar modelos teniendo como herramienta a los sistemas Lindenmayer, se encontró la información necesaria en un total de 7 referencias (Bell, 1991; Berger, 1991; Johnson, 1996; Moncho, 1986; Prusinkiewicz, y Jürgensen 1989; Prusinkiewicz y Lindenmayer, 1990; y Robertson, 1989).

También se encontraron controversias en cuanto a en que año se inventó la primer computadora, ya que Villaseñor (1992:130) menciona que fue en 1947; Atelsek (1993:1-6) que en 1948 y Nuncio (1991:34-37) que en 1945. Aún cuando estos tres autores reporten diferentes fechas, el que presenta una cronología más clara y desarrollada, así como la problemática en México con la introducción y el desarrollo de la computación es Nuncio (1991)

Dentro de la bibliografía analizada, cabe destacar la información que Prusinkiewicz y Hanan (1990) proporcionan para graficar en dos dimensiones (tanto el uso de los símbolos +, -, / y {, como la ecuación trigonométrica), ya que útil para dibujar la ramificación de una planta, independientemente del lenguaje de programación que se utilice, y fue la base para dos de los programas creados (Alga2 y Pruebagra) que se describen más adelante en este mismo capítulo.

Durante el acopio de bibliografía, se encontró que la referente al segundo y tercer tema (Patrones de desarrollo en plantas y factores que afectan el crecimiento de estas, y el uso de la computadora con los sistemas Lindenmayer) fue fácil de encontrar. El primer tema fue más difícil de encontrar, por que existe poca información de trabajos sobre el tema desarrollados en México. Con las búsquedas realizadas en Internet, este problema se solucionó, por que la información encontrada es reciente y a nivel mundial, obteniendo

información de lo que se está desarrollando en otros países y cómo. También uno puede tener acceso a la dirección electrónica de las personas que llevan la vanguardia en investigaciones hechas utilizando los sistemas Lindenmayer, como es el caso del Dr. Pruzinkiewicz, quien actualmente trabaja en Canadá

La clasificación de la información en folders y fichas de trabajo ahorró tiempo porque permitió un análisis sistemático de la información.

5.2 Elaboración de programas que modelen a los organismos vegetales.

Se elaboraron en total 4 programas que permiten ver ejemplos de como se puede modelar el crecimiento vegetal. Estos programas fueron: Alga1, Alga2, Planta y Pruebagra.

5.2.1 Primera versión del programa Alga: Alga1.

El programa Alga1 constó de 2 archivos en su creación y depuración. *Alga1.frm* y *Alga1.vbp*, de 8 y 1 Kb respectivamente. Posteriormente, estos dos archivos se integraron en uno solo al convertir el programa Alga1 en ejecutable, por lo que el archivo *Alga1.exe* ocupa 16 Kb. Junto con los archivos complementarios del programa, el disco de instalación ocupa 1.02 MB, y se puede instalar y ejecutar en cualquier computadora con Windows 95. El programa contiene 237 líneas de instrucciones y la interface de usuario se muestra en la figura 31

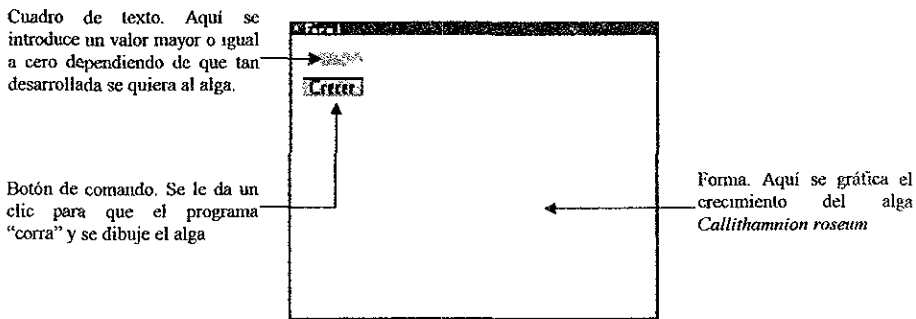


Figura 31. Interface de usuario del programa Alga1

Cuando el programa se ejecuta, se ve el desarrollo del alga dependiendo del estadio que defina el usuario, como se aprecia en la figura 32.

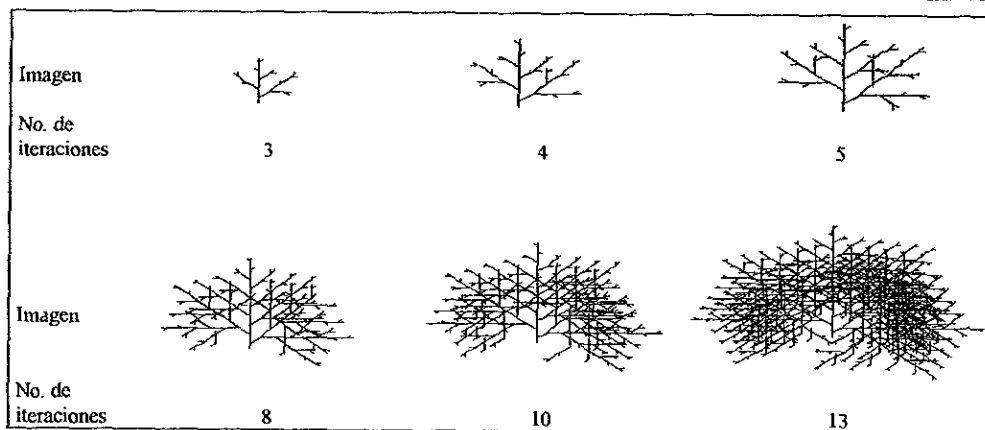


Figura 32. Modelación del desarrollo del alga *Callithamnion roseum* creada con el programa Alga1.

En este primer programa, el alga graficada tiene un ángulo de ramificación de 45° . No se puede manipular el largo de las diferentes células que comprenden al alga ni el color o el ángulo con el que se grafican, por lo que es difícil distinguir en cierto momento qué parte del alga corresponde a que tipo de célula

El programa funciona desde un estadio de crecimiento 0, pero a partir del estadio número 6 el crecimiento del alga se empieza a traslapar siendo cada vez más difícil diferenciar unas ramas de otras por que todas las células son del mismo color; y conforme el número de estadios aumenta, le toma más tiempo a la computadora graficarlos.

Pensando en crear un programa en el cual se pudiera manipular el largo de las células así como su color y ángulo de ramificación, para que se distinguieran los diferentes tipos de células que constituían al alga *Callithamnion roseum*, y que además generara ya una cadena de caracteres de acuerdo a los sistemas Lindenmayer, se creó una segunda versión de Alga1: Alga 2.

5.2.2 Segunda versión de Alga: Alga2

El programa Alga2 constó de 2 archivos en su creación y depuración, llamados Alga2.frm y Alga2.vbp, de 6 y 1 KB respectivamente. Posteriormente, estos dos archivos se convirtieron en uno solo: Alga2.exe, el cual ocupa un espacio de 15 KB.

Junto con los archivos complementarios del programa ocupa un espacio de 1.03 MB en el disco de instalación, y puede ser instalado y ejecutado en cualquier computadora que tenga Windows '95. El programa posee 74 líneas de instrucciones y su aspecto al momento de ejecutarse es el que se muestra en la figura 33:

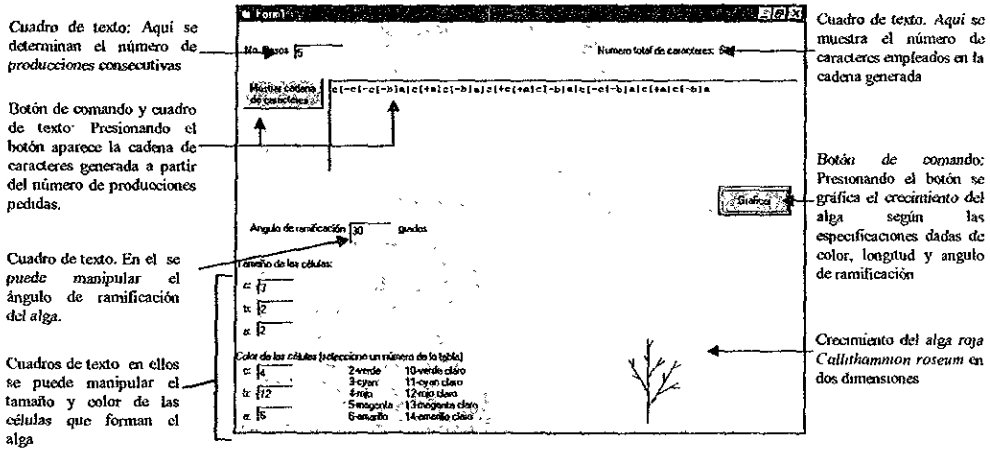


Figura 33. Interface del usuario del programa Alga2.

A diferencia de la primera versión, en esta se puede manipular la longitud de las células que conforman al alga así como su color y su ángulo de ramificación. El primer estadio que muestra el programa corresponde al número 3, sin límite de estadios, aunque empieza a crear cadenas de caracteres más lentamente desde el estadio 14. La graficación no muestra ningún tipo de problema, aunque a más ramas, más difícil distinguir unas de otras. La figura 34 muestra la graficación del alga a diferentes estadios de crecimiento.

Caracteres	c[-c[-b]a]c[+a]c[-b]a	c[-c[-a]c[+b]a]c[+c[+b]a]c[-a]c[+b]a	c[-c[-c[-b]a]c[+a]c[-b]a]c[+c[+a]c[-b]a]c[-c[-b]a]c[+a]c[-b]a
(No. de caracteres)	(21)	(36)	(61)
Imagen*			
No. de iteraciones	3	4	5
(No. de caracteres)	(271)	(716)	(3046)
Imagen*			
No. de iteraciones	8	10	13

Figura 34. Modelación del desarrollo del alga *Callithamnion roseum* creada con el programa Alga2.

* Las imágenes de esta figura tienen un ángulo de 25° y una longitud de célula de 2

Este programa fue la base para desarrollar el programa Pruebagra, que se describe más adelante en este capítulo, porque muestra la ramificación del alga en dos dimensiones a partir de una gramática generada y basada en los sistemas Lindenmayer.

5.2.3 Programa Planta.

El programa Planta contó de dos archivos en su creación y depuración: Planta frm y Planta vbp, de 8 y 1 KB respectivamente. Posteriormente, estos 2 archivos se convirtieron en un ejecutable. Planta.exe, de 17KB. Junto con los archivos complementarios del programa, ocupa 1.03 MB en el disco de instalación, y puede ser instalado y ejecutado en cualquier computadora con Windows '95. El programa posee 279 líneas de instrucciones. Su aspecto al ejecutarse se muestra en la figura 35.

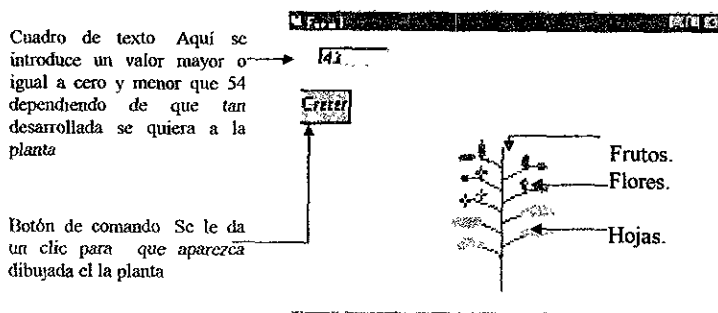


Figura 35. Interface del usuario del programa Planta.

El programa acepta estadios que van del 0 al 53, sin presentar ningún tipo de problema. Si se compara este programa con el creado por Chien y Jürgensen (1992:224-225), y mostrado en las figuras 36 y 37, se puede apreciar que los estadios no corresponden, porque tiene más estadios el programa Planta que el de Chien y Jürgensen, aún cuando las figuras son similares. Otra diferencia es que en el programa de Planta, solo se muestra el desarrollo de una planta, mientras que en el creado por Chien y Jürgensen se muestran tres. Al igual que en el programa creado por Chien y Jürgensen (fig. 36 paso 18 y 21), el programa Planta genera capullos y las flores crecen (fig. 37, paso 14 y 21). En el programa Planta no se caen los pétalos como en el programa de Chien y Jürgensen (fig. 36, paso 24) sino que en su lugar, las flores pasan a formar frutos directamente (fig. 37, paso 26). Por último, estos frutos crecen (fig. 36, pasos 27, 30 y 33) mientras que en el programa Planta estos crecen, las hojas se toman amarillas y caen junto con los frutos (fig. 37, pasos 29, 42 y 50).

Figura 36. Modelación de una planta artificial creada por Chien.

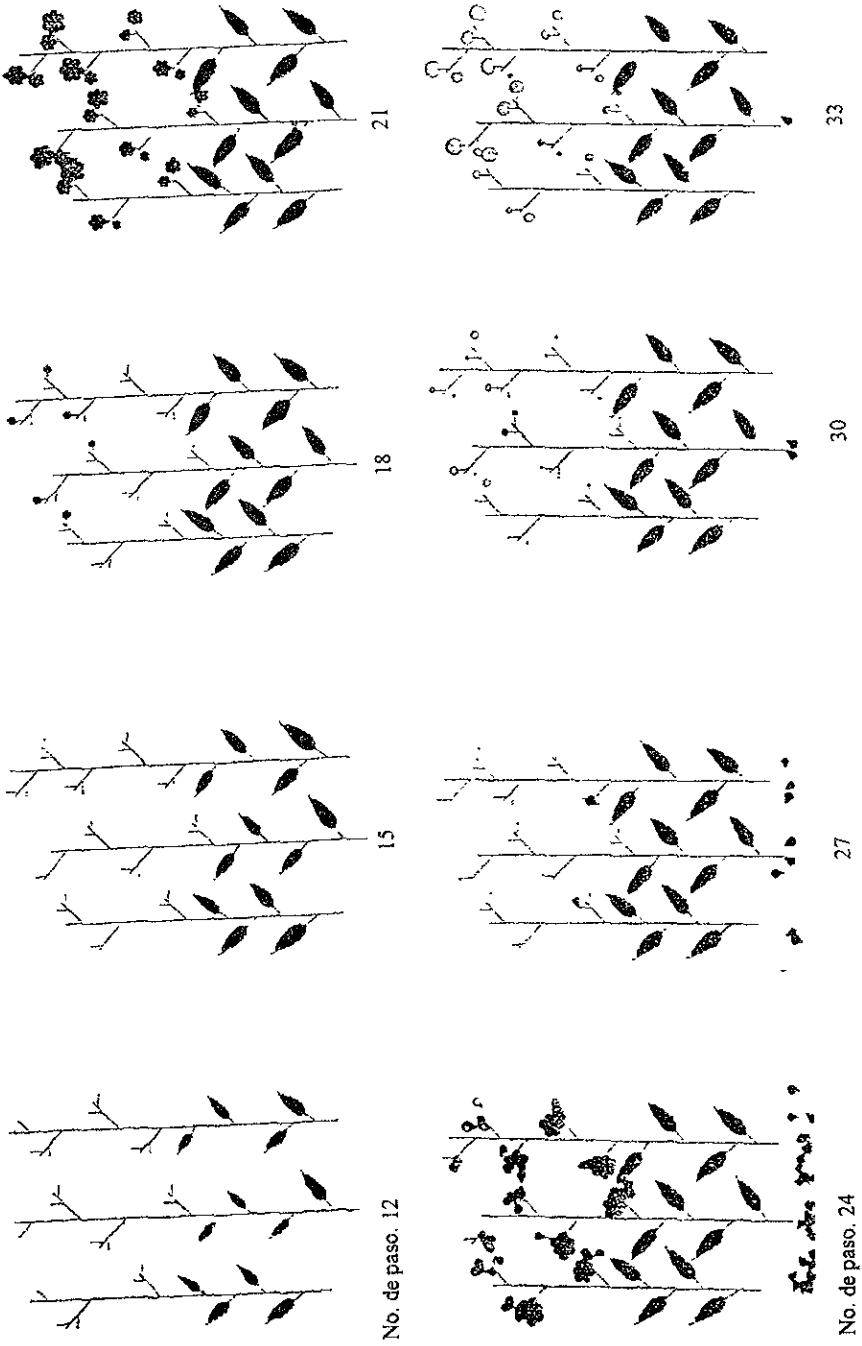
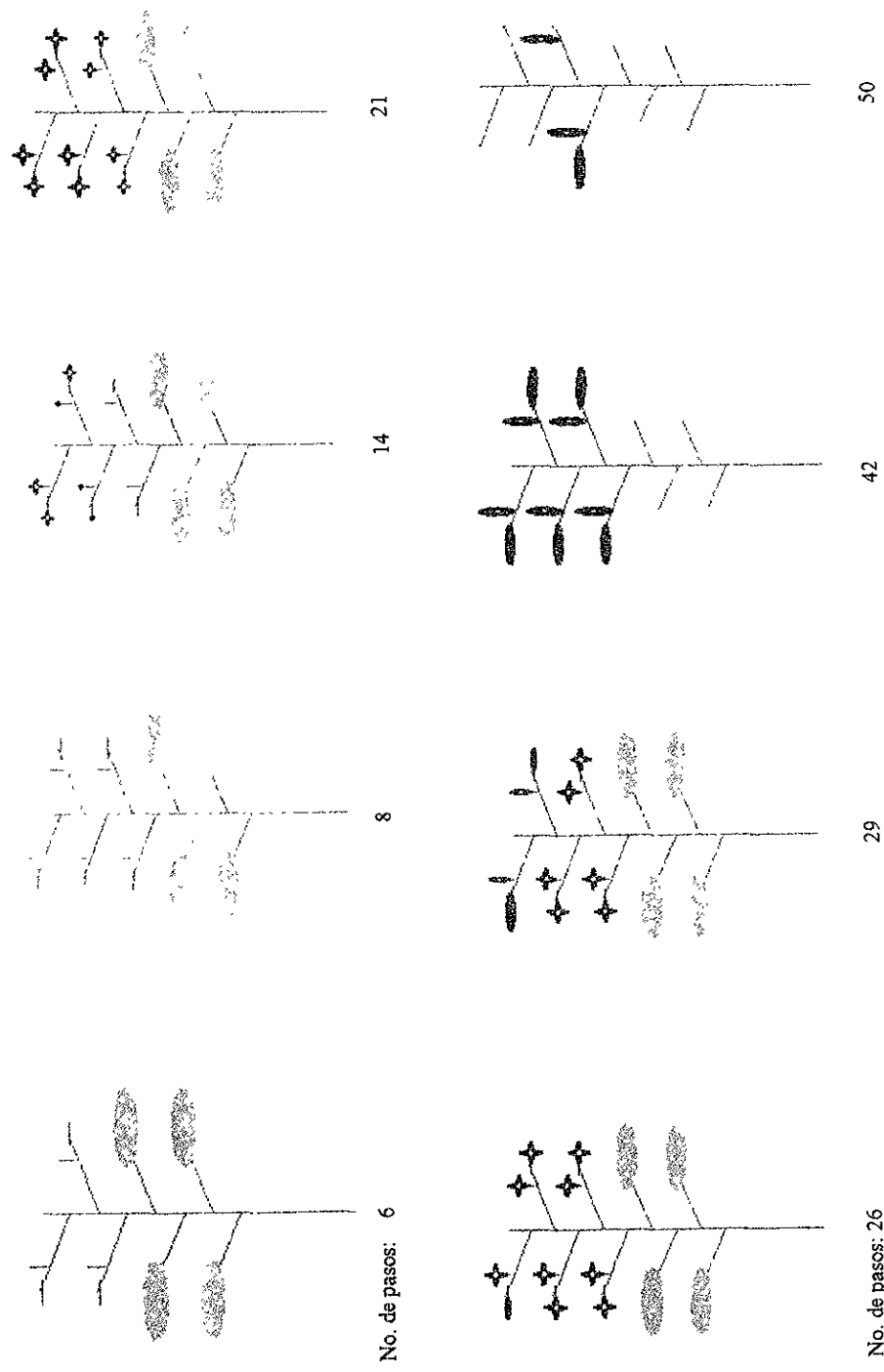


Figura 37. Modelación del crecimiento de una planta artificial con el programa Planta.



Este programa muestra la manera en que se pueden representar diferentes órganos de una planta a partir de figuras sencillas como líneas, óvalos, y círculos. En la creación de programas que modelen el desarrollo de plantas, estas figuras pueden ser utilizadas en primera instancia, para después crear figuras más complejas que describan mejor la parte de la planta que se desea graficar.

5.2.4 Programa Pruebagra.

El programa *pruebagra* constó de dos archivos en su creación y depuración: *Pruebagra.frm* y *Pruebagra.vbp*, de 13 y 1 KB respectivamente. Posteriormente, estos dos archivos se unieron en uno solo llamado *Pruebagra.exe* de 15KB, que junto con los archivos complementarios tienen un espacio de 1.03MB en el disco de instalación, y puede ser instalado y ejecutado en cualquier computadora con Windows '95. El programa posee 54 líneas de instrucciones y su aspecto se muestra en la figura 38.

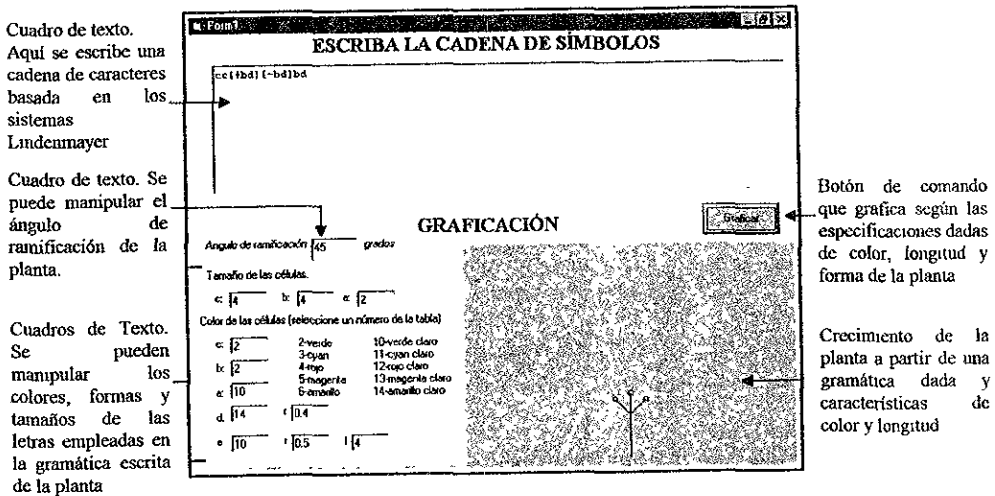


Figura 38. Interface de usuario del programa Pruebagra.

El Programa *pruebagra* acepta el vocabulario conformado por las letras *a*, *b* y *c* para líneas rectas, *d* para un círculo y *e* para una elipse. Al igual que el programa *alga2*, se puede manipular el color, tamaño, ángulo de ramificación de las diferentes letras usadas.

En este programa se puede dar la gramática que describa la forma arquitectural de una planta utilizando la notación de los sistemas Lindenmayer y comprobar que tanto se parece en su forma en dos dimensiones con la planta real. Este es un primer paso hacia la generación de reglas de reescritura en la modelación del desarrollo de plantas utilizando los sistemas-L. Además, este programa es la base para otros

programas que muestren el desarrollo de una planta utilizando reglas de producción que la computadora aplique, muestre la gramática generada y por último grafique la planta deseada dependiendo de factores como el largo de ramas, color y ángulo de ramificación. Dentro de una etapa posterior, los programas deberán utilizar reglas de producción que integren factores internos y externos que modifiquen el aspecto de la planta, así como una graficación de esta *tridimensionalmente*.

5.3 Observación y reconocimiento de patrones de desarrollo en plantas.

Los ejemplares elegidos cumplieron las características de no estar densamente ramificados y no ser muy grande (ser un árbol joven, arbusto o herbácea), para facilitar el encontrar un patrón no muy complicado que se pudiera describir mediante una gramática generada con los sistemas Lindenmayer.

Dentro de las plantas seleccionadas se encuentran un ciprés (*Cupressus lindleyi* Klotzsch) y una planta de hinojo (*Foeniculum vulgare* Will.) De ellos se obtuvo una gramática utilizando los sistemas Lindenmayer y se probó con ayuda del programa Pruebagra.

5.3.1 Árbol de ciprés (*Cupressus lindleyi* Klotzsch).

El árbol de ciprés escogido se muestra la figura 39.



Figura 39. Foto del árbol de ciprés.

Para describir el árbol de ciprés, se definió un vocabulario a utilizar así como las parte que cada letra iba a representar. Pero para definir este vocabulario se analizó las diferentes partes que conforman al árbol. Primero, el árbol tiene un tronco *principal del que se ramifican* ramas secundarias una vez a cada lado como se aprecia en la figura 40:



Figura 40. Ramificación secundaria del árbol de ciprés. a) foto. b) esquema.

Las ramas secundarias se vuelven a ramificar en ramas terciarias, de nuevo, una a cada lado de la rama secundaria como muestra la figura 41.



Figura 41. Ramificación terciaria del árbol de ciprés. a) Foto. b) esquema .

Por último, las ramas terciarias se vuelven a ramificar en otras ramas donde van las hojas, como muestra la figura 42:

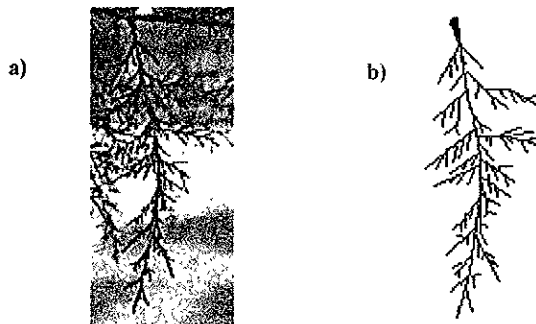


Figura 42. última ramificación del árbol de ciprés. a) foto b) esquema .

El vocabulario que se designó fue el siguiente:

$V = \{a, b, c,], [, +, -\}$

donde: a=Partes más pequeñas que conforman la última ramificación del árbol de ciprés.

b= internodos de las ramas secundarias.

c= internodos del tronco principal.

],[= Indicación de ramificación

+ =ramificación a la izquierda.

- =ramificación a la derecha.

La primer parte que se describió utilizando el vocabulario fue la última ramificación del árbol. Para ello se utilizaron los símbolos de a,], [, +, -, y el esquema de la foto real. La gramática generada es la siguiente:

a[-a[+aa]a[+aa]aa][+aaa]aaa[-aaa][+aa[-aa]a[+a[-a]a[-a]a[-a]a]aa[-a]aa]aa[-a[+aaa[-aa]a[+aa]a[+a a
a]a]aaaa[+a[-a]a]a[-a[+aa]a[+aa]a[+aa]a[+aa]a[+a[+a[-a]a[-a]a[+aa][[-aa]a[+aa]a[-aa][+aa]aa]]a
[+aa]a[-a[+a]a[+a]aa]a[+a[-a]a[-a][+aa]a[-a]aa]a[-a[+aa]a[+aa]a[+a]aaa]a[+a[+a][[-a]a[-a]aa]a[+a[-a]a
]aa]a[-a[+a]a[+a]a[+a]a[+a[-a][+a]a[+a]]a[-a[+a][[-a]aaa][+a[-a]a[-a][+a]a[+a]aa]aa[-a[-a]a[+a]a]aa[-a[
+a]aa][+a[+a][[-a]]a[+aaa]aaa

Esta gramática, graficada en el programa Pruebagra, con una longitud de la línea a de 1 cm, en color verde y con un ángulo de ramificación de 45°, se muestra en la figura 43 inciso c):

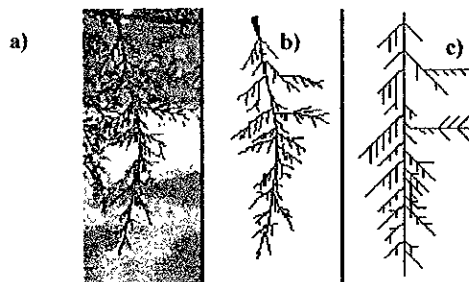


Figura 43. Comparación entre la última ramificación del árbol y su gramática. a) Foto de la ramificación b) esquema de la ramificación. c) aspecto de la rama graficada con el programa Pruebagra.

Una vez que la rama terciaria fue realizada, se procedió a crear la rama secundaria utilizando, además de la gramática de la rama generada, la letra b para los internodos de las ramas secundarias.

La gramática generada fue la siguiente:

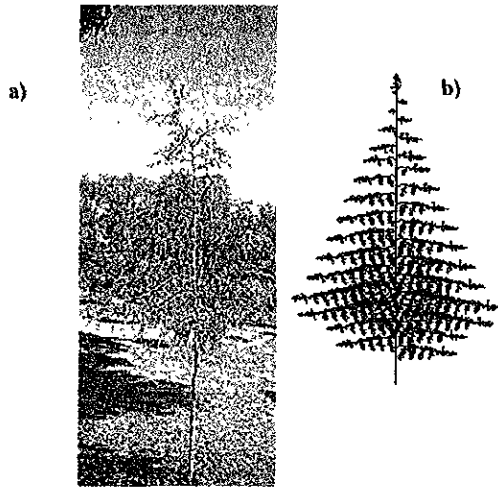


Figura 45. Comparación entre el árbol gramatizado y el real. a) Foto. b) gramática probada en el programa Pruebagra.

5.3.2 Planta de hinojo (*Foeniculum vulgare* Will).

La planta de hinojo se seleccionó por ser una herbácea que no está densamente ramificada, como se puede apreciar en la figura 46:



Figura 46. Planta de hinojo.

Para crear la gramática que describiera la forma que tenía la planta en cuestión, primero se definió un vocabulario a utilizar. Para ello, se observó detenidamente a la planta de hinojo en cuanto a su ramificación.

La planta posee un tallo principal (Fig. 47) en el que, a cada internodo le crece una rama secundaria, una vez de un lado y al siguiente del lado contrario a la rama anterior, como muestra la figura 47:

Tallo principal.



Figura 47. Ramificación de la planta de hinojo.

Analizando la ramificación secundaria de la planta, se puede observar que la ramificación de las hojas se da de manera verticilada, ya que a cada nodo le crecen dos ramas terciarias, como se aprecia en la figura 48:



Figura 48. Ramificación secundaria en la planta de hinojo.

Por último, la ramificación de las hojas con respecto a los tallos terciarios era de manera similar a la ramificación secundaria, como se puede apreciar en la figura 49:



Figura 49. Ramificación terciaria en el tallo de hinojo.

Resumiendo esta información, se definió un tallo principal, tallos secundarios, terciarios y hojas. Para generar la gramática que describiera la forma de esta planta, se designó el siguiente vocabulario:

$V = \{a, b, c, (, +, -\}$

donde:

a=hojas y ramas terciarias, representadas por una línea de 0.8 cm.

b= internodos de las ramas secundarias, representadas por líneas de 4 cm.

c= internodos del tallo principal, representados por líneas de 8cm.

(,) =Indicación de ramificación.

+ = ramificación a la izquierda.

- = ramificación a la derecha.

Al igual que en el ejemplo del árbol de ciprés, primero se describió el tallo terciario y las hojas que tenía. Después, el tallo secundario y por último, como este se ramificaba del tallo principal. La gramática final tiene un total de 1872 caracteres, y se muestra en el anexo V. Su forma, como se ve con el programa *pruebagra* utilizando una longitud de a de 0.8, b de 4 y c de 8 cm., y con un ángulo de ramificación de 50° se muestra en la figura 50 inciso b.



Figura 50. Comparación entre la planta de hinojo real y gramatizada. a) Foto. b) Planta graficada con el programa *Pruebagra*.

Aunque no es muy claro en la foto de la figura 50 a), esta planta tiene una inflorescencia de tipo umbela en la punta del tallo, como se aprecia en la figura 51.



Figura 51. Inflorescencia del hinojo.

Observándola de manera más detallada, es una umbela de umbelas, como se aprecia en la figura 52.



Figura 52. Detalle de la inflorescencia del hinojo.

Para escribir la gramática de la inflorescencia, primero se dibujó un esquema del esqueleto de la inflorescencia como se muestra en la figura 53:

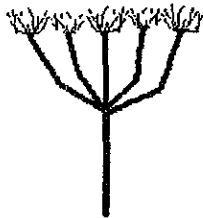


Figura 53. Esquema del esqueleto de la inflorescencia del hinojo.

Después, se designó el vocabulario a utilizar:

$V = \{a, b, c, e, (, +, -\}$

donde:

a=Partes más pequeñas que conforman la última ramificación de la inflorescencia.

b= Partes intermedias que conforman la ramificación de la inflorescencia.

c= Tallo principal de la inflorescencia.

e= flores (en forma de elipses)

(,) =Indicación de ramificación.

+ = ramificación a la izquierda.

- = ramificación a la derecha

El código escrito es el siguiente:

$c[+b[+ae][-ae]ae][-b[+ae][-ae]ae]b[+ae][-ae]ae$

La forma de la inflorescencia quedaría como muestra la figura 54, usando una longitud de a de 2 cm, b de 4 y c de 8 cm., con un ángulo de ramificación de 50°:

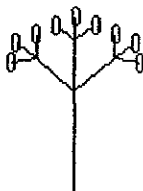


Figura 54. Inflorescencia de hinojo generada con el programa Pruebagra.

Todo lo anterior permite visualizar el potencial de aplicación de los sistemas Lindenmayer en la investigación biológica, ya que son capaces de mostrar la forma arquitectural de las plantas fiel a la realidad, y utilizando tipos de sistemas Lindenmayer mas complejos, como los T0L, 0L estocásticos, los IL, y los paramétricos, es posible modelar el desarrollo de las plantas de manera más completa dependiendo de los factores que afectan la forma que toman en su crecimiento.

CAPITULO VI: DISCUSIÓN

Al revisar la información correspondiente al crecimiento de las plantas, se encontró que este no es azaroso, porque posee características que son constantes en su forma cuando una planta de una especie alcanza cierta etapa de su crecimiento. Por ejemplo una planta de maíz (*Zea mays*) siempre tendrá venación paralela independientemente del estado de crecimiento en que se halle. Tan constantes son estas características en la forma que adquiere una especie, y hasta en un conjunto de especies, que se usan en taxonomía para clasificar y determinar a las plantas y se les conoce como caracteres morfológicos.

También existen caracteres que son predecibles como la filotaxia. Si una planta tiene filotaxia de $\frac{3}{8}$, es muy probable que la siguiente hoja se forme exactamente a 135° de la anterior y así consecutivamente. Otro ejemplo es la ramificación dicotómica, en donde las ramas se van bifurcando a lo largo del crecimiento de las plantas

Tanto los caracteres constantes como predecibles se atribuyen a la información genética de la planta, ya que es esta la que determina la forma que tendrá, de acuerdo a la información analizada. Pero no solo ésta determina la forma que adquirirá una planta, sino también las condiciones ambientales contribuyen a moldear la forma de la planta a través de la plasticidad fenotípica. Una planta herbácea siempre tendrá nodos e internodos, pero la luz puede determinar que tan lejos estarán unos de otros; lo mismo sucede con una planta que cambiando filotaxia al florecer, el clima determina cuando se da ese cambio. Aún cuando diversos autores den diferentes puntos de vista, se deben tener en cuenta los dos tipos de factores, así como determinar el papel que juegan en el crecimiento de la misma

Cabe recalcar tres puntos sobre el crecimiento de las plantas:

- a) Existen caracteres constantes en una planta en cuanto a la forma que tienen;
- b) Existen otros caracteres predecibles, además de los caracteres constantes, que tienen una alta probabilidad de que sigan un comportamiento definido;
- c) La forma que adquiere una planta está definida por su genética y afectada por su medio ambiente

Si una planta tiene caracteres constantes, y/o caracteres predecibles, y si se conoce como afectan a estos caracteres el medio ambiente, se puede generar un herramienta que prediga el crecimiento de la planta en cuestión: un modelo. Esta fue la idea que concibió Aristid Lindenmayer cuando propuso a los sistemas Lindenmayer, que son un modelo lógico-matemático el cual describe el crecimiento de las plantas mediante la

utilización de un lenguaje formal. Dentro de esta herramienta se representan las partes de la planta a modelar por medio de letras y los factores que afectan a la planta, como el clima, mediante variables que se agregan al modelo en su parte lógica y probabilística.

Con los sistemas Lindenmayer se pueden generar modelos que representen el crecimiento de las plantas y que lo predigan, con lo cual se cumple la función de un modelo según lo cita Shanon (1975 7). La información consultada muestra que esta herramienta se está desarrollando desde el punto de vista de cómputo, por lo que hace falta que se utilice y se aplique desde el punto de vista biológico. El principal problema es que, los expertos en computación generan el modelo para ver que tan cerca de la realidad pueden llegar con el software especializado, o en su defecto, crear programas que permitan ver el crecimiento de las plantas de manera más natural. Para sus propósitos, el generar modelos de esta manera cumple los objetivos de ellos, pero no sirve para la investigación biológica. La creación de un modelo no es la parte más útil de los sistemas Lindenmayer en la investigación biológica, sino la utilización de ese modelo, es decir, la *simulación*. Con la simulación, no solo se describe y se predice el comportamiento de una planta, sino que se pueden construir teorías o hipótesis a cerca del comportamiento observado en la planta debido a que un modelo es una representación de la realidad simplificado.

Aquí, cabe resaltar que, todo modelo debe ser creado para simular posteriormente con él y para que sea de utilidad en la investigación biológica.

Además, los modelos generados con los sistemas Lindenmayer poseen ventajas sobre la experimentación directa porque:

- a) Ahorra tiempo, ya que el crecimiento de las plantas puede durar de días a años.
- b) Es replicable.
- c) Es seguro, ya que no representa ningún tipo de peligro para quien lo utiliza.
- d) Es barato. Aquí cabe aclarar que solo se está tomando en cuenta a la teoría de los sistemas Lindenmayer antes de que se unieran a la computadora, ya que esta puede incrementar el costo de utilizar la herramienta a cambio de hacer más fácil y rápida su aplicación.

La simulación y la modelación del crecimiento de los vegetales usando los sistemas Lindenmayer es útil, porque ayuda a muchas áreas biológicas. Quizá el problema ahora es el generar el modelo de una planta. Para crear el modelo de una especie utilizando los sistemas Lindenmayer se necesita conocer primero la herramienta. Entender como funcionan los sistemas Lindenmayer con solo leer uno o dos artículos que

comenten su aplicación es difícil, sobre todo porque hay muchos conceptos que se desconocen o que están fuera del campo biológico. Para entender mejor que son los sistemas Lindenmayer, se recomienda iniciar con el entendimiento de las bases sobre las que se originaron y seguir con la comprensión total de los diferentes sistemas Lindenmayer, para posteriormente entender mejor su aplicación en los diversos artículos que manejan y aplican esta herramienta.

Una vez que se conoce la herramienta, se debe tomar en cuenta los siguientes aspectos para generar el modelo de una planta:

1. **Determinar para qué se quiere generar el modelo.** Este paso es, según mi opinión, el más difícil y básico para utilizar la herramienta. El objetivo de generar un modelo debe ir más allá de predecir o describir el crecimiento de una planta, y evaluar el tiempo y costo de elaborar el modelo. Es necesario desechar toda posibilidad de crearlo con otro tipo de herramientas. Por ejemplo, si se desea modelar el crecimiento de una planta para que, dependiendo del color y aspecto que presente uno pueda saber que nutriente le falta, cabría hacerse la pregunta: ¿por qué no hacer mejor un catálogo con fotos? No es que la herramienta no sea útil para este fin, pero se llevaría mucho más tiempo, dinero y esfuerzo que si se hiciera un catálogo y cubriría el mismo objetivo.

Si esta misma herramienta se utilizara para predecir cuanta producción de fruto dará un cultivo dependiendo de los factores nutrimentales y el clima, cabría preguntarse: ¿porqué no hacerlo solo de manera matemática sin imágenes? En este caso, el componente gráfico de la herramienta quedaría subutilizado.

Si por el contrario, el objetivo es crear un modelo acerca de la forma que puede adoptar un árbol dependiendo de una condición climática como lluvia ácida, y comparar la forma de una muestra representativa de árboles de la misma especie para saber cuanta lluvia ácida existe en la zona, el modelo sería útil. Primero, porque se necesita del componente gráfico y no solo del matemático y probabilístico que posee la herramienta. Segundo, porque existe una gran cantidad de formas que puede tener un árbol aún cuando exista una característica fundamental que exprese una afectación por parte del clima. Y tercero, por que esta variabilidad en la forma del árbol puede ser comparada con una creada por la computadora, que también puede crear una gran cantidad de formas en, relativamente, poco tiempo. Es por estas razones que el determinar para qué se quiere la herramienta es tan importante.

2. **Definir que parte se quiere modelar de la planta** (su ramificación, la formación del fruto, la germinación) **y que parte del crecimiento** (desde que es una semilla hasta que caen los primeros cotiledones, desde que

germina hasta que muere, etc). Es fundamental definir que parte de la planta se va a modelar, para destinar recursos y tiempo acordes al objetivo.

- 3 **Encontrar si existe un patrón de crecimiento en la planta**, sin el cual no se puede generar el modelo. Este paso consistiría de analizar todo el proceso de crecimiento que tiene el organismo, así como sus posibles caracteres predecibles y constantes. Esto dará una idea de qué vocabulario se puede utilizar.

Una vez que se han cubierto estos puntos, el siguiente paso es generar el modelo. Antes de generar dicho modelo, valdría la pena describir a la planta en cuestión en uno o varios momentos de su vida usando una gramática, tal como describí a los tres ejemplos mostrados en este trabajo: la planta de hinojo, su flor y el ciprés. El describir a la planta así ayuda a familiarizarse con las partes de la planta, y con el vocabulario a utilizar en el momento de crear las reglas de producción, lo cual se haría en un paso posterior. Por otro lado, ayuda a encontrar patrones de desarrollo o a deducir que la planta puede o no tenerlos, y así a generar las reglas de producción. Cabe resaltar que la gramática generada de la planta a modelar no es la generación del modelo, sino una prueba para designar vocabulario, y para familiarizarse con la utilización de los signos empleados en los sistemas Lindenmayer. La creación del modelo, además de describir como es la planta en un momento dado, dice porqué se generó de esa manera y bajo qué circunstancias.

Para analizar y poder describir a la planta de hinojo, su flor y al árbol de ciprés, resultó útil observar a la planta desde diversos puntos de vista, es decir, primero fijarse como es su ramificación; luego, en como están dispuestas las hojas alrededor del tallo y por último, si tenía una floración determinada como la panícula en el caso del hinojo.

Después de describir a la planta, se comenzaría a crear el modelo ya en sí. Para ello, no solo se requiere describir a la planta como en una foto sino en una secuencia de fotos y de observar como se van dando los cambios respecto al tiempo en la planta. Por ejemplo, en el caso del hinojo, cada cuanto tiempo va saliendo una rama y creciendo el internodo del tallo principal; como se van desarrollando las hojas respecto del tiempo; cuando florece, fructifica y muere.

Este paso entraña quizá la mayor dificultad de elaborar un modelo: generar las respectivas reglas de producción. Lo más conveniente es que las reglas de producción se creen en base al tipo de sistema Lindenmayer más simple: los sistemas OL. De esta manera se generará la estructura básica de las reglas de producción.

El siguiente paso es incorporar los factores que afectan el crecimiento de la planta, como el clima o la luz que perciben. Para crear estas reglas de producción se deben tener datos de la planta en cuestión de cuánto influye el factor climático sobre cierta estructura de la planta. Se puede usar los sistemas TOL o los 0L parametrizados, dependiendo si las condiciones que afectan el desarrollo de la planta son externos (en este caso se puede utilizar un TOL) o internos (para usar un 0L parametrizado).

Si la probabilidad de que una parte de la planta se desarrolle de cierta manera, por ejemplo, que el ápice de una hoja termine en un número par o impar de foliolos, y no depende de factores externos porque la probabilidad registrada es casi 0.5, se utilizarán los sistemas 0L estocásticos.

Cabe destacar que al crear un modelo de desarrollo de una planta no solo se necesita estudiar una planta sino un conjunto de plantas de las cuales se recabarán datos y se generarán promedios, porque el objetivo desde el punto de vista biológico no es generar el modelo de un individuo, sino de una especie bajo ciertas circunstancias.

La creación de un modelo debe ser seguida por su depuración. Difícilmente un modelo quedará perfecto a la primera vez que se intente hacer. Se deberá corregir su diseño y compararse con la realidad hasta que se ajuste lo más posible a esta.

Una vez generado el modelo, ¿cómo se sabe qué tan válido es? De acuerdo a los puntos propuestos por Achinstein, (1987,8), el modelo debe cumplir con 2 características:

1. **Ser Útil.** ¿Qué tan útil es el modelo generado? Depende del propósito para el cual fue concebido, y por eso es que es tan importante determinar para qué se quiere generar el modelo. Desde el punto de vista computacional, los modelos generados con los sistemas Lindenmayer han probado ser buenos ya que mediante ellos se han creado nuevas formas de representar gráficamente, por medio de la computadora, el crecimiento de los organismos vegetales. Desde el punto de vista biológico, también han mostrado que pueden ser útiles para áreas como la paleontología, la agronomía, la morfología y la botánica.
2. **Ser representativo.** El crecimiento de las plantas tiene patrones definidos, muchos de los cuales ya tienen su clasificación y son conocidos. Los factores que afectan al crecimiento y a la forma de una planta pueden entrar como variables dentro de las reglas de producción. Los modelos creados se acercan tanto a la realidad que parecen imágenes reales. Se puede afirmar que los modelos generados con los sistemas Lindenmayer son altamente representativos.

En la creación del modelo y en su depuración, la computadora es una gran ayuda. Aunque los sistemas Lindenmayer pueden ser desarrollados y aplicados teóricamente (es decir, que las imágenes se pueden crear en papel), el uso de la computadora permite que la modelación y la simulación sea más rápida, y más representativa debido al uso de gráficos.

Existe software especializado para los sistemas Lindenmayer, desarrollados por Przemyslaw Prusinkiewicz y sus colaboradores, pero desgraciadamente solo se pueden conseguir en E.U. y se deben utilizar con una computadora de procesamiento en paralelo, la cual es difícil conseguir. Otra opción es que estos paquetes se creen. Las ventajas de crear un paquete es, por un lado, el que se pueden usar varios lenguajes de programación como FORTRAN, BASIC, y C, utilizando una computadora con procesamiento simple. Por otro lado, el programa estará hecho de acuerdo a las necesidades del propio creador del programa, y en caso de necesitar una modificación, solo el desarrollador del programa sabrá como funciona y por tanto como modificarlo, como sucedió con el programa creado bajo el nombre de Pruebagra. Por ejemplo, en el caso de la escritura de la gramática del hinojo se necesitaron mas letras para representar a la planta en cuestión, como la f, d y c. El programa se pudo modificar de acuerdo a lo que necesitaba porque se conocía el programa

Pero no es necesario que el biólogo aprenda a programar. Se puede crear un grupo multidisciplinario entre biólogos y un programadores, y para que ambos profesionistas se entiendan debe existir un lenguaje común: Los sistemas Lindenmayer. Si ambos profesionistas conocen esta herramienta, el biólogo no necesitará de conocer el área de cómputo, así como el programador no necesita saber de biología para crear un programa que modele el crecimiento vegetal.

Si por el contrario, el experto en crecimiento vegetal quiere además crear sus propios programas, debe tener también conocimientos de cómo se hicieron los primeros paquetes destinados a mostrar gráficos de plantas usando los sistemas Lindenmayer. La fórmula propuesta por Prusinkiewicz y sus colaboradores fue de gran ayuda al crear los programas Alga2 y Pruebagra mostrados en este trabajo, y solucionó el problema de trazar una línea independientemente del punto en el que se originara, problema que se encontró al crear la primer versión del programa alga (alga1). Si desea crear programas en tercera dimensión, es además necesario que conozca los fundamentos proporcionados por Alberson y Disessa⁴², los cuales pueden dar una idea de como generar la tercera dimensión en la pantalla de la computadora.

⁴² Citados por Prusinkiewicz y Lindenmayer, 1990:19

Otro aspecto a explotar es la forma en que aparecen los diferentes órganos en una planta. En el caso del programa Planta, estos órganos fueron representados usando figuras primitivas como círculos, elipses y líneas. Se pueden generar dibujos de partes de plantas que puedan ser pegados dentro del proceso de graficación. Otra opción es crear diferentes aspectos de tallos, de hojas, de flores, de frutos, etc. y generar una base de datos para diferentes formas de partes de una planta.

En cuanto al software a utilizar, va a depender de la capacidad y la necesidad del usuario. Los programas se pueden generar con varios lenguajes de programación como FORTRAN, C, Visual Basic, u otros.

El trabajar con Visual Basic dentro de este trabajo fue fácil, pero respondió a que ya se tenían conocimientos del programa y de su uso. Una desventaja de utilizar Visual Basic en la creación de los programas fue al volverlos ejecutables usando el asistente de instalación. Primero, por que para cada programa se necesitó un disco flexible. Segundo, por que a pesar de necesitar los mismos archivos, no se puede crear una serie de archivos que se necesiten para ejecutar los programas y por otro lado un disco con los 4 programas. Y por último, que estos programas solo se pueden instalar y ejecutar en Windows 95' y no en versiones anteriores.

CAPITULO VII : CONCLUSIONES

- ☉ Existen dos factores importante para que los sistemas Lindenmayer no se utilicen aún en México. la falta de difusión en el área biológica, y la nula existencia de software especializado y disponible. La mayor parte de la información existente respecto a los sistemas-L, sus fundamentos y su aplicación pertenece al área computacional, por lo que es difícil que expertos en el área biológica los conozcan. Aunque no existen programas especializados que utilice a los sistemas Lindenmayer en México, se pueden conseguir en Estados Unidos o se pueden crear en algún lenguaje de programación. Esta última opción tiene la ventaja de que se ajusta a las necesidades del programador, pero la desventaja de que el usuario tiene que aprender a programar.
- ☉ Los sistemas Lindenmayer pueden generar modelos del desarrollo de las plantas fieles a la realidad, debido a que consideran los dos grandes factores que lo afectan: el genético y el ambiental. Para crear un modelo de desarrollo de una planta utilizando esta herramienta, el vegetal debe poseer caracteres constantes y/o predecibles a lo largo de su crecimiento.
- ☉ El principal factor a tomar en cuenta al usar los sistemas Lindenmayer, es el para qué se usa. El establecimiento de la meta final debe contemplar objetivos que vayan más allá de la simple descripción, además de que esta herramienta debe ser usada completamente, es decir, sin subutilizar alguna de sus componentes, sea gráfica, lógica o matemática.
- ☉ La descripción de una planta usando gramáticas basadas en los sistemas Lindenmayer dentro de los primeros pasos de la modelación, ayuda a familiarizarse con la ella, con sus diferentes partes y a dilucidar patrones recurrentes así como sus posibles reglas de producción.
- ☉ Al generar los programas de Alga1, Alga2, y Planta, se corroboró que se puede crear software que modele el desarrollo de organismos vegetales en dos dimensiones utilizando la notación de los sistemas Lindenmayer, computadoras con procesamiento simple, y lenguajes de programación como Visual Basic.
- ☉ El programa Pruebagra constituye la base para desarrollar otros programas que además de graficar plantas en dos dimensiones a partir de una gramática que describa su forma arquitectural, tomen en cuenta reglas de producción que contengan las variables externas e internas que afectan la forma de la planta, y muestren el desarrollo de esta en tres dimensiones.

- ⊗ Los sistemas Lindenmayer tienen un alto potencial de aplicación en la investigación biológica, tanto para predecir el desarrollo de organismos vegetales como para entender como afectan las diversas variables a la forma que toma la planta.

BIBLIOGRAFÍA

1. Achával, M G., *Visual Basic 3.0*, Ed. Ventura, 1a. ed., México, 1994, Pp. 7-146
2. Achinstein, P., *Los modelos teóricos*, Ed. UNAM, 1a. ed., México, 1987, Pp. 1-27
3. Atelsek, J., *All about computers*, Ed. Ziff-Davis Press, 1a. ed., E. U. A., 1993, Pp 1-6
4. Backer, J.J.W.; Allen, G E., *Biología e investigación científica*, Ed. Fondo de Cultura Interamericana, 1a. ed., México, 1970, Pp. 453-470
5. Balcázar, J L., *Programación metódica*, Ed. McGraw-Hill, 1a. ed., España, 1993, Pp 1-6
6. Ball, G., Nociones actuales acerca de la sistemática y la clasificación de los insectos, en: Taxonomía biológica, comp. Jorge Llorente B. e Isolda Luna, Serie Texto Científico Universitario, Ed. UNAM-Fondo de Cultura Económica, 1a. ed., México, 1994, Pp. 39-52
7. Bell, A. D., *Plant form. An illustrated guide to flowering plant morphology*. Ed. Oxford University Press, 1a. ed., E.U., 1991, Pp. 26-27, 34-35, 140-143, 217-229, 280-315.
8. Berger, M , *Graficación por computadora con PASCAL*, Ed Addison-Wesley Iberoamericana, 1a ed , E U.A , 1991, Pp. 7-13
9. Berntson, G M., *Modeling root architecture: are there tradeoffs between efficiency and potential of resource acquisition?*, New Phytologist, 1994, Vol. 127, No. 3, Pp. 483-493
10. Boudreaux, J. C.; Hamill, B. W.; Jernigan, R , *The role of language in problem solving 2*, Ed North-Holland, 1a. ed., Holanda, 1987, Pp. 161-319
11. Brown, D.; Rothery, P., *Models in Biology: mathematics, statistics and computing*, Ed John Wiley and Sons, 1a. ed , Inglaterra, 1993, Pp 1-9, 53-55
12. Cain, A. J., *Las especies animales y su evolución*, Ed. Nueva Colección Labor. 1a. ed , España, 1970, Pp. 9-25
13. Chang, C.C , Keisler, H J., *Model theory*, Serie: studies in logic and the foundations of mathematics, Vol. 73, Ed. North-Holland, 3ra. ed , Holanda, 1991, Pp. 1-37
14. Chien ; Jürgensen, *Parametrized L-systems for modeling: Potentials and limitations*, en: Lindenmayer systems, comp. Rozenberg, G., y Salomaa, A., Ed. Springer-Verlag, 1a. ed., E.U., 1992, pp.213-229
15. Cornell, G , *Manual de Visual Basic 3.0 para Windows*, Serie McGraw-Hill de Informática, Ed McGraw-Hill, 1a. ed., España, 1994, Pp. 204, 213.
16. De la Torre V., E., Navarro de A., R., *Metodología de la investigación*, Ed. McGraw-Hill, 1a. ed., México, 1988, Pp. 8-95.
17. Donovan, J.J., *Programación de sistemas*, Ed. El Ateneo, 2da. ed., Buenos Aires, 1986, Pp. 214-247.

18. Espasa-Calpe, *Enciclopedia Universal Ilustrada Europeo-Americana*, Ed. Espasa-Calpe, España, 1973, Pp. 393-399, 678-680
19. Fitter A. H.; Stickland, T. R.; Harvey M. L.; Wilson, G. W., *Architectural analysis of plant root systems*.
1. *Architectural correlates of exploitation efficiency*, en: New Phytologist, 1991, Vol. 118, No.3, Pp. 375-382
20. Fitter A. H., Stickland, T. R., *Architectural analysis of plant root systems*. 2. *Influence of nutrient supply on architecture in contrasting plant species*, en: New Phytologist, 1994, Vol. 118, No.3, Pp. 383-389
21. Font Q., P. Dr., *Diccionario de botánica*, Ed. Labor, 1a ed., 6ta. reimp., España, 1977, Pp. 330, 735
22. Fu, K. S., *Syntactic methods in pattern recognition*, Ed. Academic Press, 1a. ed., E.U.A., 1974, Pp.25-46
23. Fuller, H. J., *Botánica*, Ed. Interamericana, 5ta. ed., México, 1974, Pp. 1-5, 199-215.
24. Ghyka, M., *The geometry of art and life*, Ed. Dover Publications, 1a. ed., E. U., 1977, Pp. 7-19
25. Greulach, V. A.; Adams, J. E., *Las plantas. Introducción a la botánica moderna*, Ed. Limusa, 1a. ed., 4ta. reimp., México, 1990, Pp. 424-425.
26. Hearn, D.; Baker, M P., *Gráficas por computadora*, Ed. Prentice-Hall Hispanoamericana, 1a. ed., México, 1995, Pp.87-641.
27. Herman G. T; Rozeberg G., *Developmental systems and languages*, Ed. North-Holland Publishing Company, 1a ed., Inglaterra, 1975, 363 pp.
28. Hernández S, R., Fernández C., C, Baptista L., P., *Metodología de la investigación*, Ed. McGraw-Hill, 1a ed., México, 1994, 505 pp.
29. Heyman, M. S., *La esencia de Visual Basic 4*, Ed. Prentice Hall Hispanoamericana, 1996, México, Pp.1-7
30. Jean, R. V., *Phyllotaxis: a reappraisal*, en: Canadian Journal Botanique, 1989, No. 67, Pp. 3103-3107
31. Johnson, R. C., *Biomodelling takes to the Web*, en: Electronic Engineering Times, 1996, No. 922, P. 118.
32. Jones, S. B., *Sistemática vegetal*, Ed McGraw-Hill, 1ra ed., México, 1988, Pp 87-89, 232-270
33. Koleff, O., P., *Los modelos sobre filotaxia: ¿Abstracción matemática o realidad Vegetal?*, en: Tópicos de Investigación y Posgrado, 1992, Vol. 2, No. 3, Pp. 26-33
34. Kron, P.; Stewart, S. C., *Variabilities in the expression of a rhizome architectural model in a natural population of *Iris versicolor* (Iridaceae)*, en: American Journal of Botany, 1994, Vol. 81, No. 9, Pp.1128-1138.
35. Kruse, R. L., *Estructura de datos y diseño de programas*, Ed. Prentice-Hall Hispanoamericana, 1a. ed., México, 1988, Pp. 43-86
36. Langton, C. G.; Taylor, C.; Farmer, J. D.; Rasmussen, S., *Artificial life II*, Ed. Addison-Wesley Publishing Company, 1a. ed. E.U., 1991, Pp 3-23, 465-483

37. Lapedes, D.N., *Dictionary of scientific and technical terms*, Ed. McGraw-Hill, 2da ed., E.U., 1978, Pp. 1032, 1465
38. Lapedes, D N , *Diccionario de términos científicos y técnicos*, Ed. Marcombo Boixareu Editores, España, 1981, Pp. 1335-1336, 1870.
39. Larousse, *El gran saber Larousse*, Ed. Larousse, Vol. 8, Biología General, México, 1989, Pp. 40-41
40. Lawrence, E., *Henderson's dictionary of biological terms*, Ed. John Wiley & Sons, 10a. ed., E.U., 1989, Pp. 134, 326
41. Lindenmayer A.; Jürgensen H., *Grammars of development: discrete-state models for growth, differentiation and gene expression in modular organisms*, en: Lindenmayer systems, comp. Rozenberg, G., y Salomaa, A., Ed. Springer-Verlag, 1a. ed., E.U., 1992, Pp.3-21
42. Long, L , *Introducción a las computadoras y al procesamiento de información*, Ed. Prentice-Hall Hispanoamericana, 4ta. ed., México, 1995, Pp. 294-312.
43. Lyndon, L.F., *Plant development. The cellular basis*, Ed. Unwin Hyman, 1a. ed., Inglaterra, 1990, Pp. 42-293.
44. Mahoney, D. P., *Watching things grow*, en: Computer Graphics World, 1993, Vol. 16, No. 2, P. 40
45. Martin, J., y Odell, J.J., *Análisis y diseño orientado a objetos*, Ed. Prentice-Hall Hispanoamericana, 1a. ed., México, 1994.
46. Merril, E. K., *Structure and development of terminal bud scales in green ash*, en: Canadian Journal of Botany, 1989, No. 68, Pp. 12-20
47. Moncho M., J., *Programación informática. Lenguajes y Aplicaciones*, Biblioteca Santillana de consulta. Modulo Matemáticas e Informática, Vol. 6, Ed. Santillana, 1a. ed., España, 1986, Pp. 32-47
48. Murray J. D., *Mathematical biology*, Ed. Springer, 2da. ed., E.U., 1993, 767 pp.
49. Newman, R.; Boles, M., *Universal patterns, The golden relationship: art, math & nature*, Ed. Pythagorean press, 2da. ed., E.U., 1992, Pp. 169-196
50. Nultsch, W., *Botánica General*, Ed. Omega, España, 1975, 417 pp
51. Nuncio L., R., *Historia y Perspectivas de la Programación. Fundamentos de informática*, Ed. Trillas, 1a. ed., México, 1991, Pp. 34-37, 107-108
52. Oborny, B., *Growth rules in clonal plants and environmental predictability - a simulation study*, en: Journal of Ecology, 1994, Vol 82, Pp. 341-351
53. Parker, S. P., *McGraw-Hill dictionary of biology*, Ed. McGraw-Hill, 3ra. ed., E.U., 1985, P. 234, 239
54. Perez G., F.; Martínez L. J. B., *Introducción a la fisiología vegetal*, Ed. Mundi-Prensa, 1a. ed. España, 1994, Pp.137-141
55. Pfaffenberger, B., *Diccionario para usuarios de computadoras e internet*, Ed. Prentice-Hall Hispanoamericana, 6ta. ed., México, 1996, P. 330

56. Pidd, M., *Computer simulation in management science*, Ed. John Wiley & sons, 2da. ed., Inglaterra, 1988, P. 3-13
57. Prusinkiewicz, P.; Hanan, J., *Lindenmayer systems, fractals and plants*, Colección Lecture Notes in Biomathematics, No. 79, Ed. Springer-Verlag, 1a. ed., Alemania, 1989, Pp3-9, 11-22, 23-54, 81-85.
58. Prusinkiewicz, P.; Hanan, J., *L-systems: from formalism to programming languages* en Lindenmayer systems, comp. Rozenberg, G., y Salomaa, A., Ed. Springer-Verlag, 1a. ed., E.U., 1992, Pp 194-211
59. Prusinkiewicz, P.; Lindenmayer, A., *The algorithmic beauty of plants*, Ed. Springer-Verlag, 1a. ed., E.U., 1990, 228 pp.
60. Prusinkiewicz, P.; Hammel, M. S.; Mjolsness, E., *Animation of plant development*, en: Computer graphics proceedings. Annual Conference Series, 1993, Pp. 351-360
61. Prusinkiewicz, P.; Hanan, J., *Visualization of botanical structures and processes using parametric L-systems*, en: Scientific visualization and graph simulation, comp. Thalmann, D., Ed. John wiley and sons, Inglaterra, 1990, 264 pp
62. Prusinkiewicz, P. W., Remphrey, W. R., Davidson, C. G.; Hammel, M. S., *Modeling the architecture of expanding Fraxinus pennsylvanica shoots using L-systems*, en: Canadian Journal of Botany, 1994, Vol 72, Pp.701-714.
63. Radford, A. E.; Dickson, W. C.; Massey, J.R.; Bell, R.C., *Vascular plant systematics*, Ed. Harper & Row, 1a. ed., E.U., 1974, Pp. 83-166.
64. Reader's Digest, *Gran diccionario enciclopédico ilustrado*, Ed. Reader's Digest, 27 ed., 1986, México, Pp. 3471, 3530
65. Retting, M., *Virtual reality and artificial life*, en. AI Expert, 1993, Vol. 8, No. 8, P.15
66. Rada, R., *AI Languages should express gradualness*, en: The role of language in problem solving 1, comp. Jernigan, R., Hamill, B. W. y Weintraub, D.M., Ed. North-Holland, Holanda, 1985, Pp 61-78
67. Riquelme A., G. M. L., *Informática y métodos de diseño de productos informáticos computacionales*, Tesis de Maestría, PESTyC, IPN, México, 1995, Pp. 11-18
68. Robertson, B., *Technology in bloom: researchers cultivate plant-generation programs and a potential new market*, en: Computer graphics world, 1989, Vol. 12, No. 1, P.73
69. Robbins, W.W., *Botánica*, Ed. Limusa, 1a. ed., México, 1976, Pp. 301
70. Shannon, R. E., *Systems simulation. The art and science*, Ed. Prentice-Hall, 1a. ed., E.U., 1975, Pp.2,4,7
71. Steeves, T. A.; Sussex, I.M., *Patterns in plant development*, Ed. Cambridge University Press, 2da. ed., E.U., 1991, Pp. 114
72. Tejera, H.G., *Diccionario enciclopédico de informática*, Tomo 2, Ed. Grupo editorial Iberoamérica, México, 1994, Pp. 820-821, 1177

73. Tyree, M.; Ewers, F. W., *The hydraulic architecture of trees and other woody plants*, en: New Phytologist, 1991, Vol. 119, No. 3, Pp. 345-360.
74. Valdes, R., *What is biocomputing? Biologically-inspired approaches to creating software*, en: Dr. Dobb's Journal, 1991, Vol 16, No.4, P.46
75. Viladui, C., *Proposiciones informáticas al problema de la clasificación biológica*, en: Mundo Científico, Vol.11, No.117, Pp.936-943
76. Villaseñor, J L.; Murguía R., M., *La computadora en la identificación botánica*, en: Ciencia y Desarrollo, 1992, Vol.18, No.104, Pp.130-137
77. Weier, T.E., Stocking, C.R.; Barbour, M.G., *Botánica. Biblioteca científica y tecnológica*, Vol. 1, Ed. Noriega, 5ta. ed, México, 1992, P.4
78. Williams, S. A. *Programming models for parallel systems*. Ed. Wiley. 1a ed Inglaterra. 1990. 170 pp

DIRECCIONES EN INTERNET

http://www.csu.edu.au/ci/vol2/f_soddel/node6.html, 24Jun97

http://www-hppool.cs.uni-magdeburg.de/Software/documents/MuPAD_HTML/DEMO/PRIM/tree.html,
02Oct95

<http://www.bauwesen.tu-cottbus.de/Pool/Software/MuPAD/DEMO/PRIM/tree.htm>, 16sep95.

http://www.hlt.uni-duisburg.de/MuPAD_HTML/DEMO/PRIM/grass.html, 06sep95

<http://www.softlab.ece.ntua.gr/faq/fractal/faq-doc-14.html>, 04Jun9484.

<http://www.cpsc.ucalgary.ca>, 08may97

<http://www.ctpm.uq.edu.au.ipivis>, 15ene97

ANEXO I: CÓDIGO DEL PROGRAMA "ALGA 1"

```

Private Sub Command1_Click()
Scale (-15, 25)-(-15, -6)
Cls
m = Text1.Text
o = 0
n = 1
For i = 0 To m
c = o + n
o = n
n = c
Next i
lo = 3 * n
ReDim ax(lo), ay(lo), at(lo), al(lo), bx(lo), By(lo),
bt(lo), bl(lo) As Currency
For i = 1 To n
ax(i) = 0
ay(i) = 0
Next i
For i = 1 To n
bx(i) = 0
By(i) = 0
Next i

o = 0
p = 1
z = ((0.125) ^ 0.5)
pa = 0
pe = 1
at(1) = 1
al(1) = 1
For i = 1 To n
For j = i To 1 Step -1
cx = bx(j)
cy = By(j)
ct = bt(j)
cl = bl(j)
Select Case at(j)
Case Is = 1
x = ax(j)
y = ay(j)
Line (x, y)-(x, (y + 0.5)), QBColor(2)
Line (x, (y + 0.5))-(x, (y + 1)), QBColor(2)
xa = x
ya = y + 1
If al(j) = 1 Then
Line (x, (y + 0.5))-(x + z, (y + 0.5 + z)), QBColor(2)
xb = (x + z)
yb = (y + 0.5 + z)
tb = 2
Else
Line (x, (y + 0.5))-(x - z, (y + 0.5 + z)),
QBColor(2)
xb = x - z
yb = y + 0.5 + z
tb = 3
End If
Case Is = 2
x = ax(j)
y = ay(j)
Line (x, y)-(x + z, (y + z)), QBColor(2)
Line ((x + z), (y + z))-(x + (2 * z), (y + 0.5 + z)),
QBColor(2)
xa = x + (2 * z)
ya = y + 0.5 + z
If al(j) = 1 Then
Line ((x + z), (y + z))-(x + z + 0.5, (y + z)),
QBColor(2)
xb = x + z + 0.5
yb = y + z
tb = 4
Else
Line ((x + z), (y + z))-(x + z, (y + z + 0.5)),
QBColor(2)
xb = x + z
yb = y + z + 0.5
tb = 1
End If
Case Is = 3
x = ax(j)
y = ay(j)
Line (x, y)-(x - z, (y + z)), QBColor(2)
Line ((x - z), (y + z))-(x - (2 * z), (y + (2 * z))),
QBColor(2)
xa = x - (2 * z)
ya = y + (2 * z)
If al(j) = 1 Then
Line ((x - z), (y + z))-(x - 0.5 - z, (y + z)),
QBColor(2)
xb = x - 0.5 - z
yb = y + z
tb = 5
Else

```

```

Line ((x - z), (y + z))-((x - z), (y + z + 0.5)), Line(x, y)-((x + z), (y - z)), QBColor(2)
QBColor(2) Line((x + z), (y - z))-((x + (2 * z)), (y - (2 * z))),
  xb = x - z QBColor(2)
  yb = y + z + 0.5 xa = x + (2 * z)
  tb = 1 ya = y - (2 * z)
End If If al(j) = 1 Then
Case Is = 4 Line((x + z), (y - z))-((x + z), (y - z - 0.5)),
  x = ax(j) QBColor(2)
  y = ay(j) xb = x + z
  Line(x, y)-((x + 0.5), y), QBColor(2) yb = y - z - 0.5
  Line((x + 0.5), y)-((x + 1), y), QBColor(2) tb = 8
  xa = x + 1 Else
  ya = y Line((x + z), (y - z))-((x + z + 0.5), (y - z)),
  If al(j) = 1 Then QBColor(2)
    Line((x + 0.5), y)-((x + 0.5 + z), (y - z)), xb = x + z + 0.5
    QBColor(2) yb = y - z
    xb = x + 0.5 + z tb = 4
    yb = y - z
    tb = 6 End If
  Else Case Is = 7
    Line((x + 0.5), y)-((x + 0.5 + z), (y + z)), x = ax(j)
    QBColor(2) y = ay(j)
    Line(x, y)-((x - z), (y - z)), QBColor(2)
    Line((x - z), (y - z))-((x - (2 * z)), (y - (2 * z))), Line(x, y)-((x - z), (y - z)), QBColor(2)
    QBColor(2) xa = x - (2 * z)
    ya = y - (2 * z)
    If al(j) = 1 Then
      Line((x - z), (y - z))-((x - z), (y - z - 0.5)),
      QBColor(2)
      xb = x - z
      yb = y - z - 0.5
      tb = 8
    Else
      Line((x - z), (y - z))-((x - z - 0.5), (y - z)),
      QBColor(2)
      xb = x - z - 0.5
      yb = y - z
      tb = 5
    End If
  Case Is = 8
    x = ax(j)
    y = ay(j)
    Line(x, y)-((x - z), (y - 0.5)), QBColor(2)
    Line(x, (y - 0.5))-((x, (y + 1))), QBColor(2)
    xa = x
    ya = y + 1
    If al(j) = 1 Then
      Line(x, (y - 0.5))-((x - z), (y - 0.5 - z)),
      QBColor(2)

```

```

yb = y - 0.5 - z
tb = 7
Else
  Line (x, (y - 0.5))-(x + z), (y - 0.5 - z)),
QBColor(2)
  xb = x + z
  yb = y - 0.5 - z
  tb = 6
  End If
End Select
Select Case w
Case Is = 0
  r = (j / 3) * 5
  ax(r) = xa
  ay(r) = ya
  at(r) = at(j)
  al(r) = (-1 * al(j))
Case Is = 1
  r = (((j - 1) / 3) * 5) + 1
  ax(r) = xa
  ay(r) = ya
  at(r) = at(j)
  al(r) = (-1 * al(j))
Case Is = 2
  r = (((j - 2) / 3) * 5) + 3
  ax(r) = xa
  ay(r) = ya
  at(r) = at(j)
  al(r) = (-1 * al(j))
End Select
bx(j) = xb
By(j) = yb
bt(j) = tb
bl(j) = l
Dim f, g, h As Integer
f = Int(j / 2) + 1
g = Int((j / 2) - 0.5)
h = (f * 2) + (g * 3)
ax(h) = cx
ay(h) = cy
at(h) = ct
al(h) = cl
Next j
pi = pa + pe
pa = pe
pe = pi
i = pe - 1
Next i

```

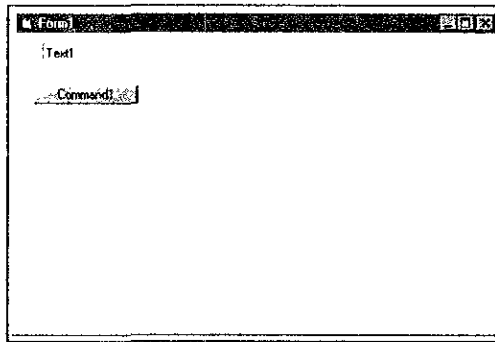


Figura I. Interface de usuario usada en el programa Alga1

ANEXO II: CÓDIGO DEL PROGRAMA “ALGA 2”

Código para el botón “Mostrar cadena de caracteres”.

```
Private Sub Command1_Click()
n = Text1.Text
a = "c[-b]a"
b = "c[-a]c[+b]a"
For i = 3 To n Step 1
d = b
p = Len(d)
For j = 1 To p
m = Mid$(d, j, 1)
Select Case m
Case Is = "+"
d = Mid$(d, 1, j - 1) + "-" + Mid$(d, j + 1, p)
Case Is = "-"
d = Mid$(d, 1, j - 1) + "+" + Mid$(d, j + 1, p)
End Select
Next j
c = "c[-" + a + "]" + d
a = 0
a = b
b = 0
b = c
Next i
Text2.Text = c
Text3.Text = Len(c)
End Sub
```

Código para el botón “Graficar”

```
Private Sub Command2_Click()
Scale (-80, 80)-(30, -2)
Cls
ReDim Tr(3), Col(3)
For i = 0 To 2
Tr(i) = Text5(i).Text
Col(i) = Text6(i).Text
Next i
CurrentX = 0
CurrentY = 0
angulo = (90 * 3.1416) / 180
angram = (Text4.Text * 3.1416) / 180
n caracteres = Text3.Text
cadena = Text2.Text
quedo = 0
```

```
ReDim quedox(20), quedoy(20), quedoangulo(20)
```

```
For i = 1 To n caracteres
caracter = Mid$(cadena, i, 1)
Select Case caracter

Case Is = "c"
X1 = CurrentX + (Tr(0) * Cos(angulo))
Y1 = CurrentY + (Tr(0) * Sin(angulo))
Line (CurrentX, CurrentY)-(X1, Y1),
QBColor(Col(0))

Case Is = "a"
X1 = CurrentX + (Tr(1) * Cos(angulo))
Y1 = CurrentY + (Tr(1) * Sin(angulo))
Line (CurrentX, CurrentY)-(X1, Y1),
QBColor(Col(1))

Case Is = "b"
X1 = CurrentX + (Tr(2) * Cos(angulo))
Y1 = CurrentY + (Tr(2) * Sin(angulo))
Line (CurrentX, CurrentY)-(X1, Y1),
QBColor(Col(2))

Case Is = "["
quedox(quedo) = CurrentX
quedoy(quedo) = CurrentY
quedoangulo(quedo) = angulo
quedo = quedo + 1
Case Is = "]"
quedo = quedo - 1
CurrentX = quedox(quedo)
CurrentY = quedoy(quedo)
angulo = quedoangulo(quedo)

Case Is = "+"
angulo = angulo + angram

Case Is = "-"
angulo = angulo - angram

End Select
Next i

End Sub
```

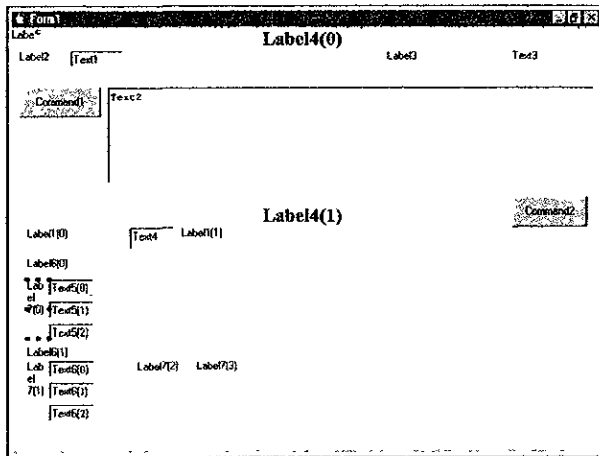


Figura II. Interface del usuario utilizada en el programa Alga2

ANEXO III: CÓDIGO DEL PROGRAMA “PLANTA”

```

Private Sub Command1_Click()
Scale (-4, 20)-(4, -5)
Cls
ReDim ax(1 To 10), ay(1 To 10)
Dim n, a As Long
a = Text1.Text
If a > -1 And a < 54 Then
Text2.Text = ""
n = a
Cls
Line (0, 0)-(0, -4), QBColor(2)
If a > 3 Then
n = 3
End If
'Traza ramas y hojas verdes
For i = 0 To n
Line (0, i)-(0, (i + 1)), QBColor(2)
If 1 = i Mod 2 Then
Line (0, i)-(0.5, (i + 1)), QBColor(2)
For j = 0 To 0.2 Step 0.001
Circle (0.65, (i + 1.06)), j, QBColor(10), , , 0.3
Next j
Else
Line (0, i)-(-0.5, (i + 1)), QBColor(2)
For j = 0 To 0.2 Step 0.001
Circle (-0.65, (i + 1.06)), j, QBColor(10), , , 0.3
Next j
End If
For k = 1 To 100000
Next k
Next i
n = a
If a > 8 Then
n = 8
End If
'Traza ramas para flor
If a > 3 Then
For i = 4 To n
Line (0, i)-(0, (i + 1)), QBColor(2)
If 1 = i Mod 2 Then
Line (0, i)-(0.6, (i + 1.1)), QBColor(2)
Line (0.5, (i + 1))-(-0.7, (i + 1)), QBColor(2)
Line (0.4, (i + 0.9))-(-0.4, (i + 1.5)), QBColor(2)
x = i - 3
ax(2 * x) = 0.7
ay(2 * x) = i + 1
ax((2 * x) - 1) = 0.4
ay((2 * x) - 1) = i + 1.5
Else
Line (0, i)-(-0.6, (i + 1.1)), QBColor(2)
Line (-0.5, (i + 1))-(-0.7, (i + 1)), QBColor(2)
Line (-0.4, (i + 0.9))-(-0.4, (i + 1.5)), QBColor(2)
x = i - 3
ax(2 * x) = -0.7
ay(2 * x) = i + 1
ax((2 * x) - 1) = -0.4
ay((2 * x) - 1) = i + 1.5
'Las hojas crecen
For k = 0 To 3
If 1 = k Mod 2 Then
For j = 0.2 To 0.3 Step 0.001
Circle (0.65, (k + 1.06)), j, QBColor(10), , , 0.3
Next j
Else
For j = 0.2 To 0.3 Step 0.001
Circle (-0.65, (k + 1.06)), j, QBColor(10), , , 0.3
Next j
End If
Next k
End If
For k = 1 To 100000
Next k
Next i
End If
'traza capullos
n = a
If a > 18 Then
n = 18
End If
If a > 8 Then
m = 19 - n
For i = 10 To m Step -1
w = ax(i)
x = ay(i)
For j = 0 To 0.025 Step 0.001
Circle (w, x), j, QBColor(4)
Next j
Next i
'traza flores
For i = 10 To (m + 3) Step -1
w = ax(i)
x = ay(i)
For j = 0.03 To 0.07 Step 0.001

```

```

Circle (w, x), j, QBColor(4), , , 0.25
Circle (w, x), j, QBColor(4), , , 4
  Next j
  For j = 0 To 0.01 Step 0.001
    Circle (w, x), j, QBColor(14)
  Next j
Next i
' Las flores crecen
For i = 10 To (m + 7) Step -1
  w = ax(i)
  x = ay(i)
  For j = 0.03 To 0.1 Step 0.001
    Circle (w, x), j, QBColor(4), , , 0.25
    Circle (w, x), j, QBColor(4), , , 4
  Next j
  For j = 0 To 0.015 Step 0.001
    Circle (w, x), j, QBColor(14)
  Next j
Next i
End If
'Flores siguen brotando
If a > 18 Then
  m = 22 - a
  If a > 21 Then
    m = 1
  End If
  For i = 3 To m Step -1
    w = ax(i)
    x = ay(i)
  For j = 0.03 To 0.07 Step 0.001
    Circle (w, x), j, QBColor(4), , , 0.25
    Circle (w, x), j, QBColor(4), , , 4
  Next j
  For j = 0 To 0.01 Step 0.001
    Circle (w, x), j, QBColor(14)
  Next j
Next i
'Flores siguen creciendo
For i = 7 To (m + 4) Step -1
  w = ax(i)
  x = ay(i)
  For j = 0.03 To 0.1 Step 0.001
    Circle (w, x), j, QBColor(4), , , 0.25
    Circle (w, x), j, QBColor(4), , , 4
  Next j
  For j = 0 To 0.015 Step 0.001
    Circle (w, x), j, QBColor(14)
  Next j
Next i
End If
End If
'Flores siguen creciendo
If a > 21 Then
  m = 26 - a
  If a > 25 Then
    m = 1
  End If
  For i = 4 To m Step -1
    w = ax(i)
    x = ay(i)
    For j = 0.03 To 0.1 Step 0.001
      Circle (w, x), j, QBColor(4), , , 0.25
      Circle (w, x), j, QBColor(4), , , 4
    Next j
    For j = 0 To 0.015 Step 0.001
      Circle (w, x), j, QBColor(14)
    Next j
  Next i
End If
'flores caen
If a > 25 Then
  m = 36 - a
  If a > 35 Then
    m = 1
  End If
  For i = 10 To m Step -1
    w = ax(i)
    x = ay(i)
    For j = 0.03 To 0.1 Step 0.001
      Circle (w, x), j, QBColor(15), , , 0.25
      Circle (w, x), j, QBColor(15), , , 4
    Next j
  'brotan frutos
  If 1 = i Mod 2 Then
    For j = 0 To 0.1 Step 0.001
      Circle (w, x), j, QBColor(5), , , 4
    Next j
  Else
    For j = 0 To 0.1 Step 0.001
      Circle (w, x), j, QBColor(5), , , 0.25
    Next j
  End If
  Next i
'Los frutos crecen
For i = 10 To (m + 3) Step -1
  w = ax(i)
  x = ay(i)
  If 1 = i Mod 2 Then
    For j = 0 To 0.17 Step 0.001

```



```

Circle (w, x), j, QBColor(5), , , 4
Next j
Else
  For j = 0 To 0.17 Step 0.001
    Circle (w, x), j, QBColor(5), , , 0.25
    Next j
  End If
Next i
End If
If a > 35 Then
  m = 39 - a
  If a > 38 Then
    m = 1
  End If
  For i = 3 To m Step -1
    w = ax(i)
    x = ay(i)
    If l = i Mod 2 Then
      For j = 0 To 0.17 Step 0.001
        Circle (w, x), j, QBColor(5), , , 4
        Next j
      Else
        For j = 0 To 0.17 Step 0.001
          Circle (w, x), j, QBColor(5), , , 0.25
          Next j
        End If
      Next i
    End If
  ' Hojas se ponen amarillas
  If a > 38 Then
    m = a - 38
    If a > 41 Then
      m = 3
    End If
    For k = 0 To m
      If l = k Mod 2 Then
        For j = 0 To 0.3 Step 0.001
          Circle (0.65, (k + 1.06)), j, QBColor(14), , , 0.3
          Next j
        Else
          For j = 0 To 0.3 Step 0.001
            Circle (-0.65, (k + 1.06)), j, QBColor(14), , , 0.3
            Next j
          End If
        Next k
      End If
    'Hojas se caen
    If a > 41 Then
      m = a - 41
      If a > 44 Then
        m = 3
      End If
      For k = 0 To m
        If l = k Mod 2 Then
          For j = 0 To 0.3 Step 0.001
            Circle (0.65, (k + 1.06)), j, QBColor(15), , , 0.3
            Next j
          Else
            For j = 0 To 0.3 Step 0.001
              Circle (-0.65, (k + 1.06)), j, QBColor(15), , , 0.3
              Next j
            End If
          Next k
        End If
      'Frutos se caen
      If a > 44 Then
        m = 54 - a
      If a > 53 Then
        m = 1
      End If
      For i = 10 To m Step -1
        w = ax(i)
        x = ay(i)
        If l = i Mod 2 Then
          For j = 0 To 0.17 Step 0.001
            Circle (w, x), j, QBColor(15), , , 4
            Next j
          Else
            For j = 0 To 0.17 Step 0.001
              Circle (w, x), j, QBColor(15), , , 0.25
              Next j
            End If
          Next i
        End If
      Else
        Text2.Text = "El número debe estar entre 0 y 53!!!"
      End If
    End Sub

```

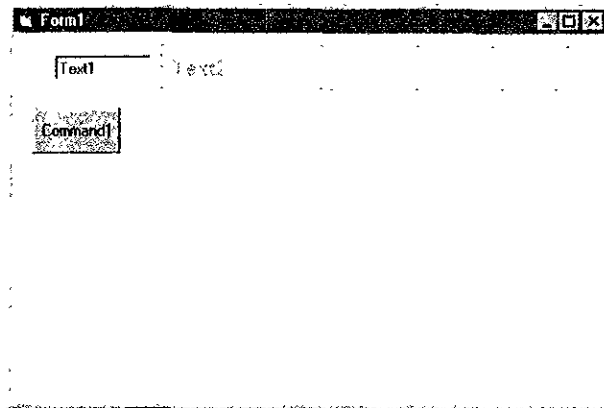


Figura III. Interface del usuario utilizada en el programa Planta

ANEXO IV. CÓDIGO DEL PROGRAMA PRUEBAGRA

```

Private Sub Command2_Click()
Scale (-80, 80)-(30, -2)
Cls
ReDim Tr(3), col(5)
For i = 0 To 2
Tr(i) = Text5(i).Text
Next i
For i = 0 To 4
col(i) = Text6(i).Text
Next i
CurrentX = 0
CurrentY = 0
angulo = (90 * 3.1416) / 180
angram = (Text4.Text * 3.1416) / 180
n caracteres = Len(Text2.Text)
r = text7.Text
re = Text8.Text
f = text1.Text
cadena = Text2.Text
quedo = 0
ReDim quedox(60), quedoy(60), quedoangulo(40)
For i = 1 To n caracteres
caracter = Mid$(cadena, i, 1)
Select Case caracter

Case Is = "c"
X1 = CurrentX + (Tr(0) * Cos(angulo))
Y1 = CurrentY + (Tr(0) * Sin(angulo))
Line (CurrentX, CurrentY)-(X1, Y1),
QBColor(col(0))

Case Is = "a"
X1 = CurrentX + (Tr(1) * Cos(angulo))
Y1 = CurrentY + (Tr(1) * Sin(angulo))
Line (CurrentX, CurrentY)-(X1, Y1),
QBColor(col(1))
Case Is = "b"
X1 = CurrentX + (Tr(2) * Cos(angulo))
Y1 = CurrentY + (Tr(2) * Sin(angulo))
Line (CurrentX, CurrentY)-(X1, Y1),
QBColor(col(2))
Case Is = "d"
Circle (CurrentX, CurrentY), r, QBColor(col(3))
quedox(quedo) = CurrentX
quedoy(quedo) = CurrentY
quedoangulo(quedo) = angulo
quedo = quedo + 1
Case Is = "]"
quedo = quedo - 1
CurrentX = quedox(quedo)
CurrentY = quedoy(quedo)
angulo = quedoangulo(quedo)
Case Is = "+"
angulo = angulo + angram
Case Is = "e"
Circle(CurrentX, CurrentY), re, QBColor(col(4)), f
Case Is = "["
Case Is = "-"
angulo = angulo - angram
End Select
Next i
End Sub

```

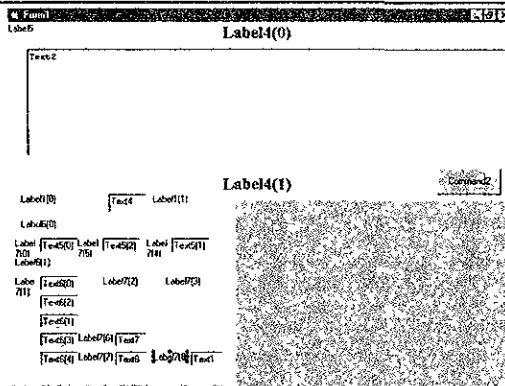


Figura IV. Interface de usuario utilizado en el programa Pruebasgra

