Sistema Web para la generación de filogenias en base a caracteres homólogos

LÓPEZ, Benito, AYALA, Joel, LUGO, Oziel y ZARCO, Alfonso

B. López, J. Ayala, O. Lugo y A. Zarco

Universidad Autónoma del estado de México, Centro Universitario Texcoco, Alumno de la maestría en Ciencias de la Computación; Av. Jardín Zumpango s/n Fracc. El Tejocote, Texcoco, Estado de México.
b.samuellopez7@gmail.com

F. Pérez, D. Sepúlveda, R. Salazar, D. Sepúlveda (eds.) Ciencias Matemáticas aplicadas a la Agronomía. Handbook T-I.-

Abstract

When we talk about phylogenetic trees is important to know the best Techniques to find and obtain better cladograms based on Homologous characters. For this reason, another technique to reconstruct cladograms is been proposed. We propose an algorithm belonging to the field of conglomerates that is call LinkAge simple and another one that is call Hennig used in a classic way to reconstruct phylogenetic trees.

3 Introducción

La reconstrucción de relaciones ancestrales es uno de los aspectos más importantes en el estudio de la evolución de las especies dentro de la entomología, la reconstrucción se basa en características específicas y la forma en que participan para lograr este fin. Las características se pueden trabajar de dos formas: podemos trabajar en base a datos representados como secuencias de ADN; o con características homologas que son representadas en la matriz de características como ceros (ausencia) y unos (presencia). Para cumplir con los objetivos de este trabajo se han tomado como base las características homologas. Mediante el análisis filogenético puede medirse la similaridad entre un conjunto de especies, cuales son menos parecidos entre si y poder conocer cuales caracteres son los que aportan más información. Es por esta razón que las técnicas para la reconstrucción se dividen en: basados en distancias, y basados en caracteres por criterio de optimización.

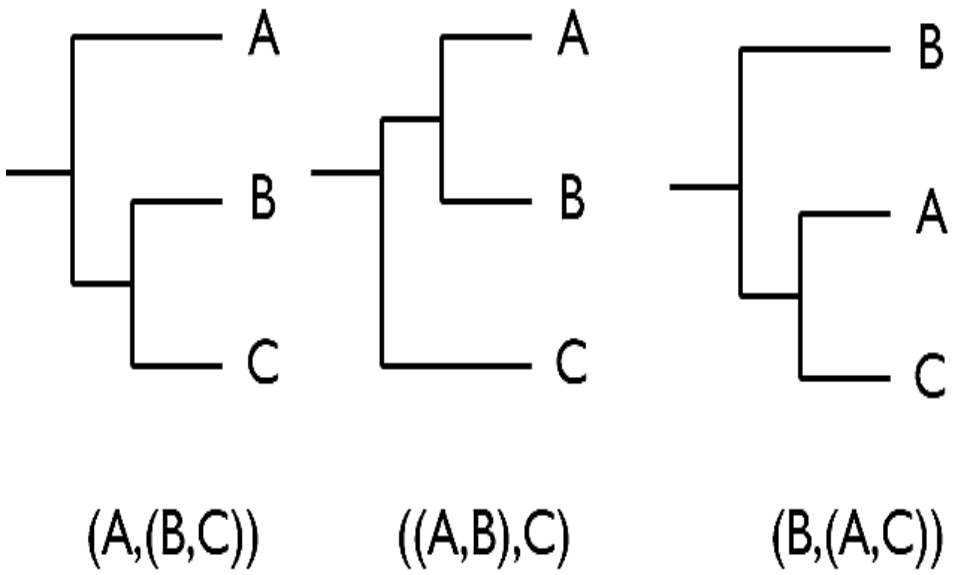
El número de árboles filogenéticos que van a resultar dentro de un análisis filogenético puede ser calculado en base al número de características que participaran en este proceso. Es por esto que se deben crear métodos basados en heurísticas que nos ayudaran a acotar el número de posibles soluciones. En este trabajo se presenta una aplicación web para la reconstrucción de árboles filogenéticos, que a diferencia de otras aplicaciones, ésta es gratuita y está disponible en la red para apoyo y consulta. Se programaron dos algoritmos para el desarrollo de árboles filogenéticos: El algoritmo de Hennig (Lipscomb, 1998) (siendo éste aún un clásico en el mundo de la Entomología) y el algoritmo por conglomerados (una propuesta reciente en el cual muestra un algoritmo menos complejo de programar con resultados similares) (Lopez Razo, Ayala de la Vega, Lugo Espinoza, & Napoles Romero , 2016). Se empleó el lenguaje Java para el back end (toda la parte algorítmica y de graficación). El front end fue desarrollado en lenguaje HTML y con CSS. Para la interface con Java se utilizó JavaScript. Para la recolección de los datos se empleó el lenguaje PHP.

3.1 Marco teórico

3.1.1 Intratabilidad

Como se mencionó en la introducción, el número de árboles filogenéticos depende completamente en el número de especies seleccionadas por participar. Por ejemplo, si hablamos de 3 especies (A, B, C), pueden existir 3 árboles con raíz y uno sin raíz (ver Figura 3).

Figura 3 Árboles con 3 taxones



Fuente: (Tato Gomez , 2011)

Según (Rodriguez Catalán, 2001) el número posible de árboles enraizados para n taxones puede ser calculada en base a la ecuación 3:

$$N_r = \frac{(2n-3)!}{2^{n-2}(n-2)!} \qquad \text{Para } n>2 \tag{3}$$

Donde:

Nr es la cantidad de árboles con raíz.
n es el número de taxones.

Y el número de árboles sin raíz se puede calcular mediante la ecuación 3.1:

$$Nu = \frac{(2n-5)!}{2^{n-3}(n-3)!} \qquad \text{Para } n>2 \tag{3.1}$$

Donde:

Nu es la cantidad de árboles sin raíz.
n es el número de taxones que utiliza.

El número de posibles árboles enraizados para n taxones es igual al de los árboles sin raíz para n-1 taxones. Ambos números se incrementan a medida que n aumenta. De este modo, a partir de 12 especies se vuelve difícil cuantificar el número de árboles con y sin raíz que se pudieran obtener (debido a que es un problema intratable ya que el calcular todos los posibles árboles tiene un costo computacional temporal muy elevado). Por ejemplo, un año tiene 31 536 000 segundos, un procesador Pentium IV ejecuta $4'000{,}000$ instrucciones por segundo, por lo que aproximadamente ejecuta $126\,144x10^9$ instrucciones por año.

Suponiendo que en cada instrucción se realiza un árbol, y apoyándose de la tabla 3, para 20 especies se tardaría 65 011 380 años en mostrar todos los árboles y para 30 especies tardaría $3.925x10^{25}$ años (Tato Gomez , 2011).

Tabla 3 Cantidad de árboles dependiendo del número de especies

Especies	Numero de arboles
1	1
2	1
3	3
4	15
5	105
6	945
7	10395
8	135135
9	2027035
10	34459425
20	8200794532637891559375
30	49518x10^38

Fuente: (Tato Gomez , 2011)

De esta forma, cuando n es grande, el experto no puede analizar todos los árboles generados, ya que sólo uno de esos árboles representa correctamente la verdadera relación evolutiva. Por lo tanto, se utilizan heurísticas que permitan generar árboles cercanos al árbol correcto.

3.1.2 Hennig

La argumentación de Hennig considera la información de cada carácter, uno a la vez. Es decir, se detecta la presencia o ausencia de dicho carácter en cada uno de los grupos seleccionados. Un ejemplo de ello se ve a continuación en la Figura 3.1.

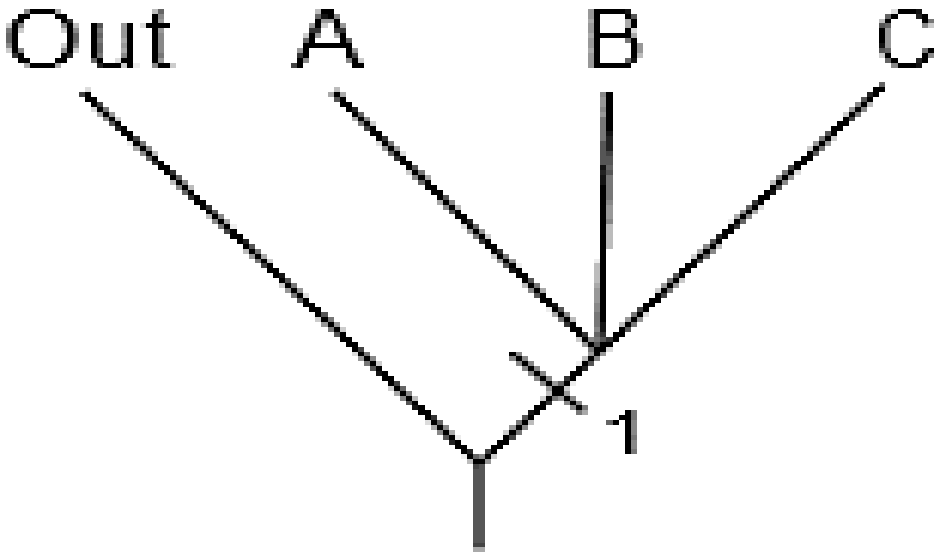
Figura 3.1 Matriz de Datos Hennig

	Características				
	1	2	3	4	5
Outgroup	0	0	0	0	0
A	1	0	0	0	1
B	1	1	0	1	0
C	1	0	1	1	0

Fuente: (Lipscomb, 1998)

1.- El carácter 1 une los taxas (grupos) A, B y C porque ellos comparten el carácter apomorfico 1 (ver Figura 3.2).

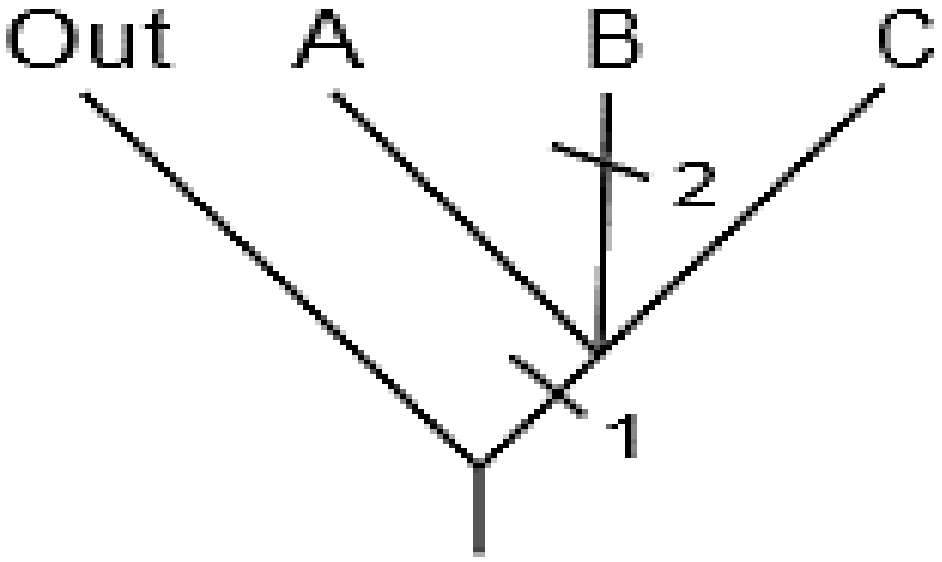
Figura 3.2 Árbol con el carácter 1



Fuente: (Lipscomb, 1998)

2.- Carácter 2 – el carácter derivado es encontrado solo en el taxón B, y no provee mucha información sobre las relaciones entre taxas (ver Figura 3.3).

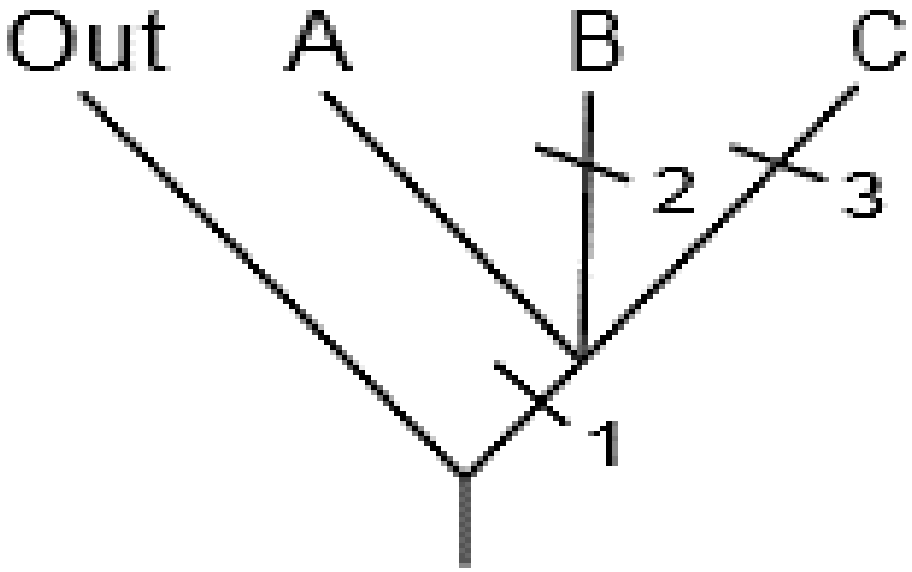
Figura 3.3 Árbol con el carácter 2



Fuente: (Lipscomb, 1998)

3.- Carácter 3 - el carácter derivado es autopomorifico para el grupo C (ver Figura 3.4).

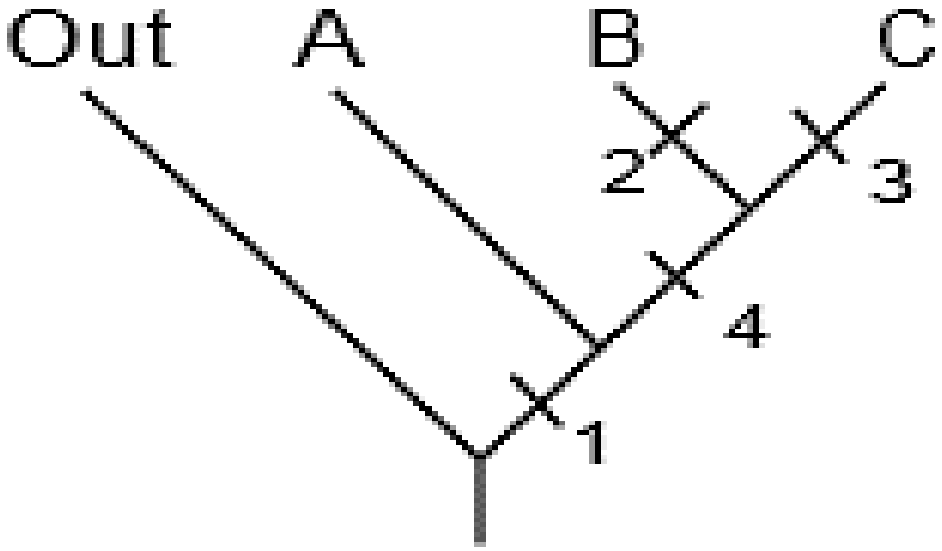
Figura 3.4 Árbol con el carácter 3



Fuente: (Lipscomb, 1998)

4.- Carácter 4 – el carácter derivado es sinapomorfico y une los taxas A y B (ver Figura 3.5)

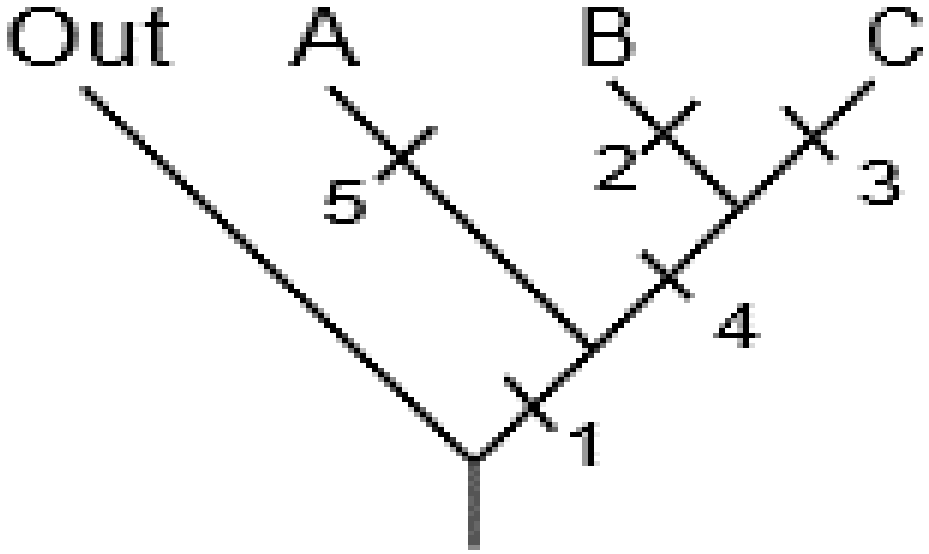
Figura 3.5 Árbol con el carácter 4



Fuente: (Lipscomb, 1998)

5.- Carácter 5 – El carácter derivado es un antropomórfico para el taxón A (ver Figura 3.6)

Figura 3.6 Árbol con el carácter 5



Fuente: (Lipscomb, 1998)

Las matrices de datos reales raramente son así de simples. Sin embargo, el concepto es el mismo.

3.1.3 Conglomerados

El Análisis Clúster o Análisis de Conglomerados, es una técnica estadística multivariante que busca agrupar o separar elementos o variables tratando de lograr la máxima homogeneidad en cada grupo y la mayor diferencia entre los grupos. Existen muchas técnicas para el uso de conglomerados sin embargo nos basaremos en los algoritmos jerárquicos acumulativos (forman grupos haciendo conglomerados cada vez más grandes), aunque no son los únicos posibles (Terrádez Gurrea).

Para poder unir variables o individuos es necesario tener algunas medidas numéricas que caractericen las relaciones entre las variables o los individuos. Cada medida refleja una asociación en un sentido particular y es necesario elegir una medida apropiada dependiendo del problema que se esté tratando.

Como cualquier algoritmo, es conveniente identificar los pasos que se requieren para efectuar el análisis. Los pasos dentro del análisis de conglomerados son:

1. Elección de variables.
2. Elección de las medidas de asociación.
3. Elección de la técnica de clúster.

Dentro del análisis filogenético que a continuación se presenta, la elección de variables ha sido determinada por un experto que avala la veracidad de la matriz de datos. La medida de asociación será calculada en base a la diferencia de valores que tendrán cada una de las características homólogas con respecto al conjunto de taxones que forman la matriz de datos. Estas diferencias serán calculadas y representadas en una matriz de distancias.

Algoritmo:

- Se comienza con una matriz con n taxones (matriz de datos) y con una matriz n×n de distancias $\Delta = (\delta ij)$ simétrica y con ceros en la diagonal.
- Se busca en la matriz de disimilaridades los grupos que tengan menor distancia entre si (el par de grupos más próximos). Sean U y V los grupos más próximos, y d (UV) su distancia.
- Se unen los grupos U y V, y se etiqueta el nuevo grupo como (UV). Se actualiza la matriz de disimilaridades, de la siguiente forma:

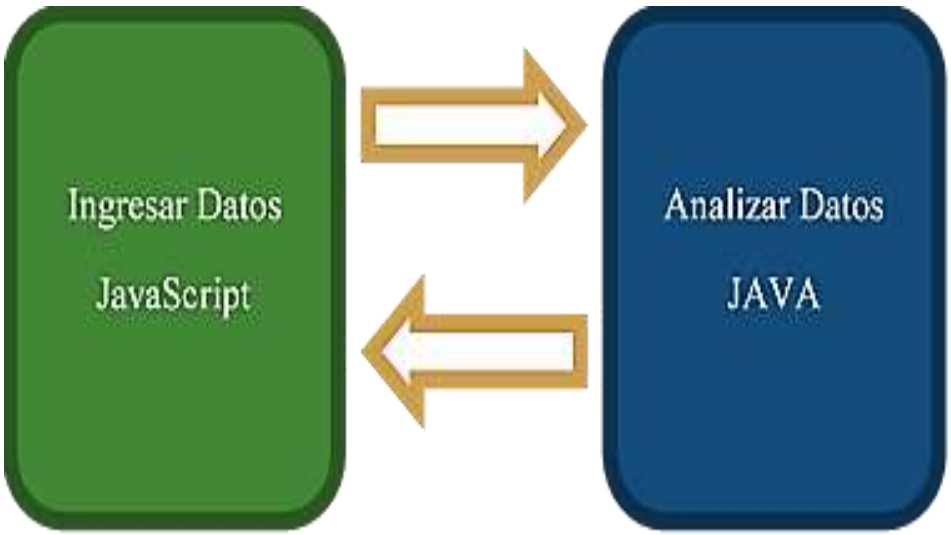
a) Se borran las filas y columnas correspondientes a los grupos U y V.
b) Se añade una fila y una columna con las distancias entre el grupo (UV) y los grupos restantes.

Repetir los pasos 2 y 3, n − 1 veces. Al final, todas las unidades estarán incluidas en un único grupo y las etiquetas de los grupos que se han unido, así como las distancias con las que se unieron (Hernández, 2011).

3.2 Materiales y métodos

La aplicación realiza el análisis de datos mediante la ejecución de un script de JavaScript instanciado por el usuario. JavaScript Inicialmente fue desarrollado por la empresa Netscape en 1995 con el nombre de LiveScript. Posteriormente pasó a llamarse JavaScript quizás tratando de aprovechar que Java era un lenguaje de programación de gran popularidad y que un nombre similar podía hacer que el nuevo lenguaje fuera atractivo. JavaScript, a diferencia de Java, se ejecuta directamente en el navegador y es por esto que nos da una forma más eficiente de manipulación de datos. En la figura 3.7, que se muestra a continuación, se observan las dos principales tareas a realizar con JavaScript y la interacción entre sí.

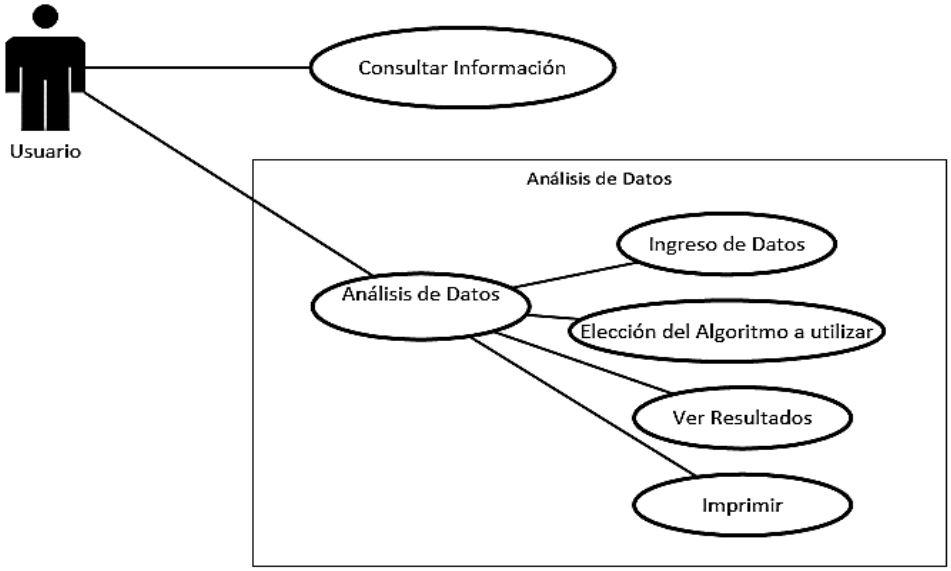
Figura 3.7 Uso principal de Java y JavaScript



El sistema se desarrolló utilizando la plataforma IDE Netbeans 8.1 y se realizaron pruebas utilizando el navegador Google Chrome. Sin embargo, está disponible para los diferentes navegadores como Mozila Firefox o Microsoft Edge.

La aplicación Web supone que el usuario cuenta con un grupo de estudio particular para poder ingresar dicha información dentro de una matriz (considerada matriz de datos). La aplicación no está acotada a un grupo determinado de especies ni a un conjunto específico de datos, es por esto que los datos a analizar deben ser validados para poder obtener resultados aceptables (ver figura 3.8). Dichas acciones son observables en la figura 10 UML (Booch & Rumbaugh).

Figura 3.8 Diagrama UML de casos de uso



Una vez que los datos han sido introducidos, el algoritmo escogido entra en acción. En la figura 3.9 se muestra el seudocódigo del algoritmo de Hennig.

Figura 3.9 Seudocódigo Hennig

```
Algoritmo sin_titulo
    Obtener matriz de Datos
    Recorrer matriz de datos
    Buscar característica con mayor incidencia
    Copiar columna en nueva matriz
    Terminar de recorrer matriz de datos
    Recorrer matriz de columnas ordenadas
    Buscar grupo (fila) con mayor número de características
    Copiar fila a nueva matriz
    Terminar de recorrer matriz ordenada por columnas
    Recorrer matriz ordenada por columnas y filas
    Identificar que grupos se unen con cada característica
    Guardar grupos en Array
    Terminar de recorrer matriz ordenada por columnas y filas
    Recorrer array de grupos
    Si array[i] = array[i+1] Entonces
        Graficar solo una vez dicho grupo
            SinoGraficar ambos grupos
            Terminar de recorrer array
        acciones_por_falso
    Fin Si
FinAlgoritmo
```

Como se observa en la figura 3.10, el algoritmo programado requiere tres ciclos. Cada ciclo contiene anidados dos ciclos para poder recorrer la matriz (A (n^3) + B (n^3) +C (n^3)). Por lo que la complejidad es 𝒪 (n^3) Como se observa en la figura 3.10, Para recorrer la matriz se requieren dos ciclos anidados, por lo que el algoritmo tiene una complejidad 𝒪 (n^3).

Figura 3.10 Seudocódigo Simple Linkage

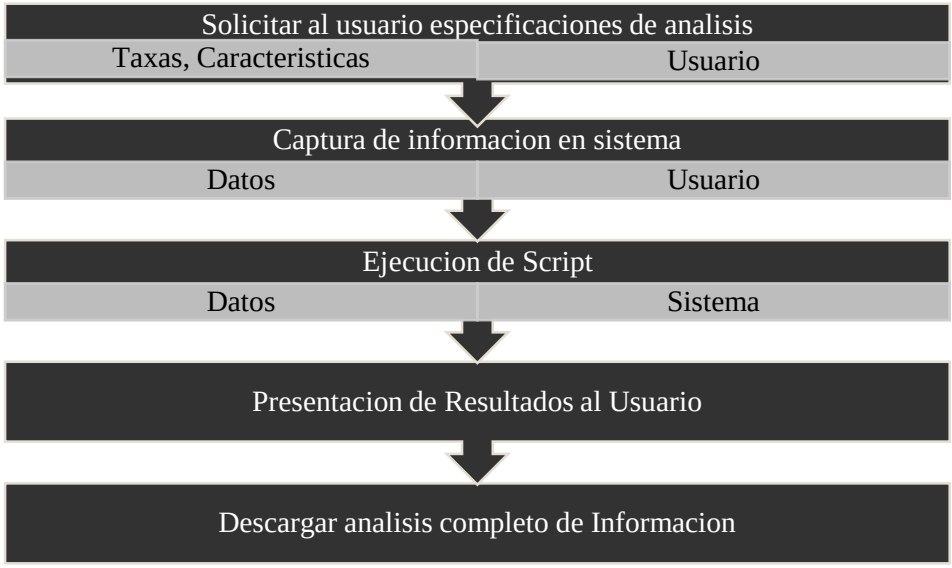
```
Algoritmo sin_titulo
    Leer matriz de datos
    Mientras expresion_logica Hacer
+       Calcular matriz de distancias
+       recorrer matriz de distancias [filas][columnas]
+       min=Buscar el valor mínimo()
+       if matriz de distancias [filas][columnas] es igual min
        Grupo1 = fila
        Grupo2 =columna
+       Termina if
+       Termina ciclo
+       Unir grupo1 y grupo2
+       Actualizar la matriz de datos
        |
+       While (matriz de datos > 2)
    Fin Mientras

FinAlgoritmo
```

Dentro del concepto de complejidad, es observable que ambos poseen una complejidad de 𝒪 (n^3), sin embargo son completamente diferentes al momento de ser programados, especialmente en las funciones de graficación. Simple Linkage vincula un grupo a la vez en cada iteración del análisis mientras que Hennig agrupa o separa especies en base a cada carácter que participa dentro de las iteraciones correspondientes.

Esto produce que Hennig sea mucho más compleja su programación tanto en la algorítmica como en su graficación

El funcionamiento general del sistema web está basado con la idea de que el usuario ya cuenta con la matriz de datos que se quiere trabajar. Teniendo la matriz de datos la forma en que se trabaja se muestra en la Figura 3.11.

Figura 3.11 Funcionamiento general del Sistema WEB

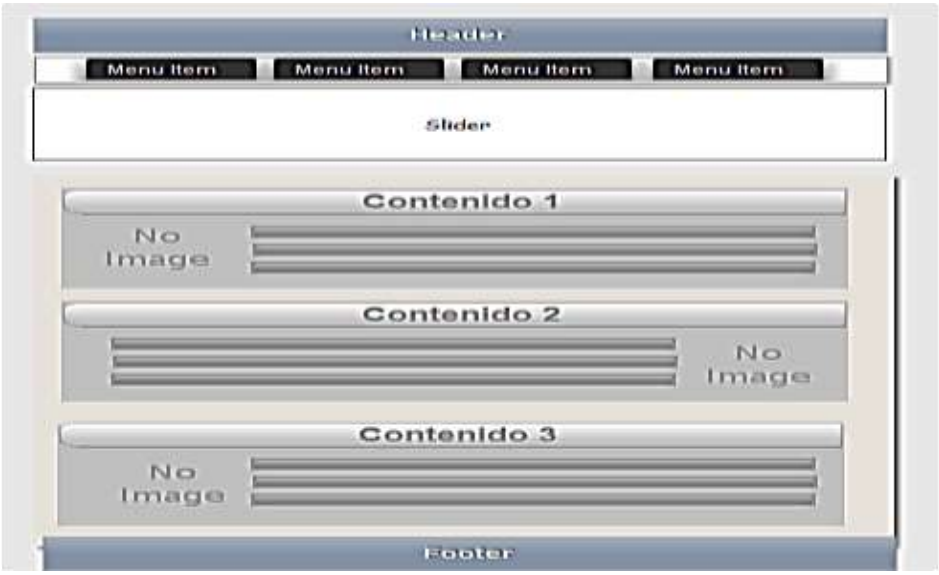


Una vez terminado el análisis, el reporte final de los datos podrá ser analizado por el usuario sin necesidad de tener que ingresar al sistema nuevamente ya que toda la información final será presentada en un documento con formato PDF.

3.3 Diseño del Sitio Web

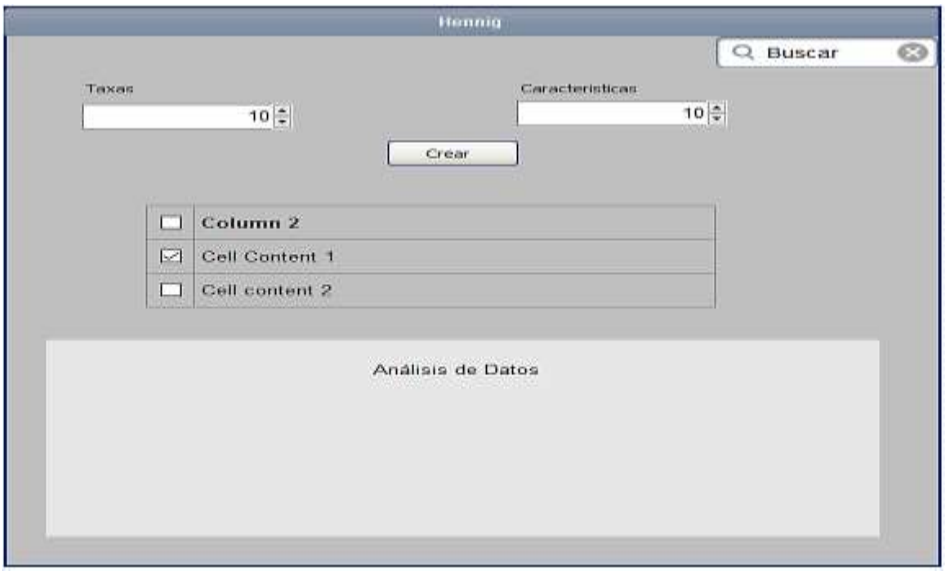
El maquetado de la página principal del Sitio Web se muestra en la figura 3.12. Dentro de la página principal se busca tener una distribución óptima y un diseño que permita la facilidad en la navegación.

Figura 3.12 Maquetado Principal del Sitio Web



Se cuenta con una página exclusiva para poder introducir los datos de la matriz de características. El maquetado de la página para el análisis puede verse en la Figura 3.13. Dentro de esta página se puede indicar el número de especies, características y especificar los estados que presentan las características en cada especie. Una vez ingresada toda la información podemos esperar a que los resultados sean desplegados y poder obtener el informe final de las características y los cladogramas.

Figura 3.13 Página de Análisis



3.4 Conclusiones

El avance tecnológico y el acceso a medios electrónicos y digitales han incrementado la posibilidad de compartir información y conocimiento al poder desarrollar aplicaciones específicas para darle solución a un problema real. Los sitios Web son una poderosa herramienta ya que nos brindan acceso a dichos sitios Web únicamente teniendo una conexión a internet y un navegador web actualizado. Dentro del campo entomológico, el estudio de las relaciones ancestrales entre especies es un pilar importante al momento de querer hacer un análisis de un grupo en común, y es por esta razón que la aplicación descrita anteriormente le permite al usuario reconstruir dichas relaciones de una forma sencilla y con una interfaz fácil para el usuario. La complejidad de los algoritmos mencionados relativamente es similar ya que ambos poseen una complejidad de n^3. Sin embargo, en la programación de los algoritmos y su graficación, simple linkage permite una programación mucho más ágil.

3.5 Referencias

Booch, G., & Rumbaugh, J. (s.f.). *El lenguaje Unificado de Modelado.* Recuperado el 7 de Julio de 2016, de elvex.ugr.es: http://elvex.ugr.es/decsai/java/pdf/3E-UML.pdf

Hernández, S. d. (2011). *Análisis de Conglomerados.* Madrid, España: Universidad Autonoma de Madrid.

Lipscomb, D. (1998). *Basics of Cladistic Analysis.* Washington D. C.: George Washington University.

Lopez Razo, B. S., Ayala de la Vega, J., Lugo Espinoza, O., & Napoles Romero , J. (2016). Cluster Analisys as a methodology within Phylogenetc Systematics to Construct Phylogenetic Trees. *International Journal of Modern Engineering Research*, 15.

Rodriguez Catalán, P. (Septiembre de 2001). *Anàlisis Filogenèticos*. Recuperado el 05 de Septiembre de 2015, de academia.edu: http://www.academia.edu/3578130/AN%C3%81LISIS_FILOGEN%C3%89TICOS

Tato Gomez , A. (2011). *Grupo de Bioinformática de la Facultad de Matemáticas*. Recuperado el 04 de Octubre de 2015, de http://mathgene.usc.es/cursoverano/cv2005/materiales/filogenia/filogenia1.pdf

Terrádez Gurrea, M. (s.f.). *http://www.uoc.edu/in3/emath/docs/Cluster.pdf*. Recuperado el 07 de Julio de 2016, de http://www.uoc.edu/in3/emath/docs/Cluster.pdf.