



UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MÉXICO

---

---



Centro Universitario UAEM Texcoco

Métricas de similitud mediante *String Matching Comparison* en identificación de plantas para su clasificación en el Instituto de Biología de la UNAM

# T E S I S

QUE PARA OBTENER EL GRADO DE  
Maestro en Ciencias de la Computación

PRESENTA:

Ricardo Rodrigo Juárez Hernández

Tutor Académico:

M. en CCA. José Sergio Ruiz Castilla

Tutores Adjuntos:

Dr. Jair Cervantes Canales

Dr. Farid García Lamount

Diciembre, 2015

**Métricas de similitud mediante *String Matching Comparison* en identificación de plantas para su clasificación en el Instituto de Biología de la UNAM.**

por

Ricardo Rodrigo Juárez Hernández

Tesis presentada para obtener el grado de

Maestro en Ciencias de la Computación

en la

Universidad Autónoma del Estado de México

Diciembre, 2015

## **Agradecimientos**

Con todo el cariño a mis padres, mis hermanos y mis seres queridos por su apoyo en los tiempos difíciles, y en todo momento han estado a mi lado dándome apoyo y fortaleza.

También a mis maestros y amigos, por impulsarme a ser mejor cada día en todos los aspectos de mi vida, por compartir sus conocimientos y momentos de alegría conmigo.

A todos ellos les doy gracias.

# Índice de figuras

Figura 1 Demostración de la computadora en la botánica (Munguía- Romero, 1992) ....	5
Figura 2 Estándar de desarrollo para bases de datos (Zarembo, Teilans, Rausis, & Buls, 2014).....	6
Figura 3 Ejemplo de cadenas concatenadas con anotaciones (Apostolico et al., 2007)..	9
Figura 4 Ejemplo de árbol p-suffix (Barker, 1996).....	13
Figura 5 Representación de comportamiento mediante Jaccard (Rahman et al., 2012).	18
Figura 6 Métricas de similitud en forma de valores numéricos (Lin, 1998).....	22
Figura 7 Plantilla para captura de requerimientos.....	27
Figura 8 Requerimientos funcionales SIE.....	29
Figura 9 Requerimientos no funcionales SIE.....	30
Figura 10 Administrador, dar de alta planta nueva.....	30
Figura 11 Administrador, consulta de una planta.....	31
Figura 12 Usuario, consulta vía Web por características.....	31
Figura 13 Usuario, consulta por nombre vía Web.....	32
Figura 14 Secuencia de una consulta vía Web.....	32
Figura 15 Diagrama de clases de la aplicación Taxón 2.....	33
Figura 16 Diagramas de clase de clase, familia, género y especie de la aplicación Taxón 2.....	33
Figura 17 Diagrama de secuencia Taxón 2.....	34
Figura 18 Secuencia de una consulta vía Web.....	35
Figura 19 Arquitectura de la aplicación (Elaboración propia).....	35
Figura 20 Árbol para la clasificación de una planta.....	37
Figura 21 Árbol de decisiones (Elaboración propia).....	45
Figura 22 Representación interna de un nuevo registro.....	45

Figura 23 Formación del vector descriptivo.....	46
Figura 24 Concatenación de características.....	46
Figura 25 Características en forma de casillas. ....	46
Figura 26 Transposición de las características seleccionadas (Elaboración propia). ....	47
Figura 27 Resultados de similitud con la Base de Datos.....	48
Figura 28 Convención de datos usada para las métricas de similitud. ....	49
Figura 29 Prototipo Página Principal Taxón 2. ....	50
Figura 30 Autenticación Taxón 2.....	50
Figura 31 Prototipo para registrar una nueva planta.....	51
Figura 32 Datos Taxonómicos.....	52
Figura 33 Datos Morfológicos.....	52
Figura 34 Datos de Fenología.....	53
Figura 35 Mensaje de Registro Guardado. ....	53
Figura 36 Muestra de Posibles Menús.....	54
Figura 37 Formas de una Planta. ....	54
Figura 38 Distribución de Plantas en México. ....	55
Figura 39 Distribución del listado de características.....	55
Figura 40 Despliegue de posibles resultados.....	56
Figura 41 Prototipo aplicación para administrador Taxón 2. ....	57
Figura 42 Pantalla Web para consulta. ....	59
Figura 43 Demostración MVC Aplicación Taxón 2. ....	60
Figura 44 Familia Liliopsida, conjunto de vectores descriptivos.....	60
Figura 45 Ejemplo de una consulta por familia en la aplicación Taxón 2. ....	61
Figura 46 Consulta por clase vía Web en base a características. ....	61
Figura 47 Ejemplo de listado de posibles resultados.....	62

Figura 48 Características de la clase Liliopsida. ....	62
Figura 49 Mapeo completo de la clase Magnoliopsida. ....	63
Figura 50 Fracción del mapeo de la familia Asteraceae.....	64
Figura 51 Distancia Levenstein en la familia Orchidaceae. ....	65
Figura 52 Resultados de métrica SMC a todas las familias. ....	71
Figura 53 Resultados de métrica Jaccard a todas las familias.....	72
Figura 54 Resultados de métrica Rogger & Tanimoto a todas las familias. ....	72
Figura 55 Resultado de la métrica SMC a solo 10 familias. ....	73
Figura 56 Resultado de la métrica Jaccard a solo 10 familias.....	73

# Índice de tablas

Tabla 1 Ejemplo de pseudocódigo programación dinámica (García, 2007). .....	11
Tabla 2 Operaciones de autómata fuzzy intuicionista (Ravi et al., 2013). .....	12
Tabla 3 Ventajas y desventajas de métricas de similitud existentes Primera parte. ....	24
Tabla 4 Plantas en México agrupadas por clase, familia y especie. Primera parte. ....	38
Tabla 5 Resultados de métricas en todas las colecciones de plantas. Primera parte. ....	66

# Índice de formulas

( 1 ) Función de discernibilidad.....	8
( 2 ) Métrica de similitud coseno modificada.....	19
( 3 ) SimCosine.....	20
( 4 ) SimJaccard.....	20
( 5 ) SimOverLap.....	20
( 6 ) SimLevenshtein.....	20
( 7 ) SMCij.....	48
( 8 ) Jaccard(X1, X2).....	49
( 9 ) Rogger & Tanimoto.....	49



## Tabla de contenido

Introducción.....	1
1.1 Planteamiento del problema .....	1
1.2 Justificación .....	2
1.3 Objetivos.....	3
1.4 Hipótesis .....	3
Revisión bibliográfica y estado del arte .....	4
2.1 Revisión Bibliográfica.....	4
2.2 Estado del Arte .....	24
Metodología de desarrollo de la aplicación.....	26
3.1 Proceso Unificado Racional (RUP).....	26
3.2 Reglas del Negocio.....	27
3.3 Requerimientos.....	27
3.4 Análisis y Diseño.....	30
3.4.1 Casos de Uso .....	30
3.4.2 Diagrama de Actividades .....	32
3.4.3 Diagrama de Clases .....	33
3.4.4 Diagrama de Secuencia .....	34
3.4.5 Arquitectura del Sistema .....	35
3.4.6 Descripción del funcionamiento cliente/servidor.....	36
3.4.7 Diseño de la Base de Datos .....	37
3.4.8 Diseño de la Interfaz.....	45
3.4.9 Consulta Nueva Planta via <i>Web</i> .....	46
3.4.10 Selección de las métricas de similitud.....	48
3.4.11 Propuesta del diseño de la interfaz de usuario: Prototipo de navegación.....	50
3.5 Implementación .....	56
3.5.1 Aplicación prototipo administrador.....	56
3.5.2 Aplicación prototipo usuario, consulta vía <i>Web</i> .....	58
3.6 Prueba .....	59
3.6.1 Aplicación prototipo administrador.....	59
3.6.1 Aplicación prototipo Administrador.....	61
Resultados.....	65
4.1 Ventana de Corrida.....	65
4.2 Tablas de Resultados .....	65
4.3 Grafica de Resultados.....	71
4.4 Descripción de los resultados .....	74
Conclusiones.....	75
Trabajos futuros.....	76

## Resumen

México es uno de los países más ricos en su flora, teniendo cerca de 30 000 especies diferentes y su territorio posee gran diversidad de grupos taxonómicos.

La identificación taxonómica se define como el proceso de nombrar o catalogar un espécimen dentro de un sistema de clasificación previa, la cual comprende tres actividades principales que son: clasificación (asignar un grupo), nomenclatura (asignar un nombre) e identificación (asignar una ubicación).

La clasificación de una planta en México es por clase, familia, género y especie.

En el Instituto de la UNAM se tiene un “museo de plantas”, que es la más completa nivel nacional. Toda planta almacenada primero es “curada”, esto es clasificarla correctamente para su almacenamiento. Este proceso es manual por el especialista en el ramo que es el Dr. José Luís Villaseñor Ríos, teniendo una gran cantidad de trabajo atrasado por escasez de personal, tiempo y recursos.

Anteriormente este Instituto desarrolló una herramienta para ayudar en esta tarea de clasificación llamada *GENCOMEX: a Computerized Key to Identify the Genera of Asteraceae Of México* en 1998, por lo cual, se actualizo al crear una nueva aplicación de clasificación llamada Taxón 2 programada en Java mediante el diseño MVC. En dicha aplicación, el usuario marca las características que posee la planta formando una cadena de 0's (cuando no posee la característica) y 1's (si tiene la característica) formando un vector descriptivo.

En el ramo de la inteligencia artificial una de sus principales aplicaciones es la de clasificación. Tales clasificaciones se pueden realizar en base a un patron que son rasgos dispuestos en vectores descriptivos. Una de las multiples herramientas para clasificar basadas en vectores son las métricas de similitud, para el caso de la búsqueda aproximada (vectores diferentes). Algunas de las aplicaciones de dichas métricas son en: búsqueda de ADN, reconocimiento de patrones, procesamiento de señales, comparación de archivos, corrección de texto, minería de datos y consulta de bases de datos. Para este caso, se aplicaron las métricas *String Matching Comparison (SMC)*, Jaccard y Roger&Tanimoto, las cuales son herramientas diseñadas específicamente para comparar patrones entre cadenas binarias.

Teniendo como punto de referencia la distancia Levenshtein (número de inserciones, sustituciones, eliminaciones o transposiciones para duplicar la cadena original), se aplicaron las 3 metricas a todas las familias proporcionadas por la UNAM para su clasificación tomando un patron base al azar. En el caso de búsqueda aproximada SMC resultado la técnica más eficiente, en segundo lugar Jaccard, y por último Rogger&Tanimotor con la menor eficiencia. En el caso de búsqueda exacta (vectores iguales) las 3 metricas arrojaron el mismo grado de similitud del 100 %.

## Abstract

México is one of the richest flora countries, having approximately 23 000 different species and his territory has great taxonomic groups diversities.

The taxonomic identification it's define as the process of naming o catalog a specimen into an earlier classification system, which comprehends three main activities: classification (assign a group), nomenclature (assign a name) and identification (assign a location).

A plant classification in México is by class, family, gender and species.

At the Instituto de la UNAM it has a "plants museum" wich is the most complete at nacional level. Every plant stored firstly is "curada", meaning the correct classification for his storage. This process is manual by the specialist at the field Dr. José Luis Villaseñor Rios, having a great amount of work due the lack of personal, time and resources.

Previously this institute developed a computational tool to help the plant classification named *GENCOMEX, a Computarized Key to Identify the Genera of Asteraceae of México* in 1998, therefore, it's been updated by creating a new classification application name Taxon 2 programmed in Java using the MVC desing. In this application, the user checks the plant characteristics forming a string of 0's (absence of characteristic) and 1's (have that characteristic) forming a descriptive vector.

At the field of atificial intelligence one of the main applications is the classification. These classifications can be made by patrons wich are descriptive vectors features. One of the multiple tools for vector based classification are similarity metrics, for the case for approximate search (different vectors). Some of these metrics applications are: DNA search, patterns recognition, signals processing, files matching, text correction, data minning and data bases querys. In this case, the metrics *String Matching Comparison (SMC)*, *Jaccard* and *Roger&Tanimoto* have been applied, wich are specificly design to compare binary strings.

Having the distance Levenshtein (number of insertions, substitutions, removes or transpositions to duplicate the original string) as reference, the three metrics have been applied to all he families given by the UNAM for his classification taking a random patron to compare. At the case of approximate search SMC result to be most efficient, in second place Jaccard, and finally Roger&Tanimoto with the lowest efficiency. For exact search (equal vectors) the three metrics give the same grade of similarity of 100 %.

# Capítulo I

## Introducción

### 1.1 Planteamiento del problema

México es uno de los países más ricos en diversidad. Su flora incluye cerca de 30 000 especies diferentes y en su territorio se centra una gran diversidad de grupos taxonómicos, por ejemplo, dentro de éstas familias se encuentra la *Asteracea*, que consiste en 1400 géneros y cerca de 23 000 especies (K, 1994). En México esta planta se clasifica en 310 – 387 géneros y 2400 – 3000 especies, dependiendo de la autoridad taxonómica consultada (Villaseñor, 1993).

La identificación taxonómica se define como el proceso de nombrar o catalogar un espécimen dentro de un sistema de clasificación previa. Comprende tres actividades principales que son: clasificación, nomenclatura e identificación (Munguía-Romero, 1992). En primer lugar, se da la clasificación que es el proceso de asignar un espécimen o grupo de especímenes dentro de una categoría taxonómica, o sea un taxón. Posteriormente, en la nomenclatura se le asigna un nombre a las diferentes categorías acorde al *Código Internacional de Nomenclatura Botánica* (Villaseñor, 1998). Por último, en la identificación se determina la ubicación de un espécimen dentro de este sistema de clasificación internacional botánico.

En el Instituto de la UNAM se tiene un “museo de plantas”, que es la más completa de México. Por otra parte, toda planta almacenada primero es “curada”; esto es, clasificarla correctamente para su almacenamiento. Cabe mencionar que si una planta es repetida se almacena guardando la fecha y lugar donde se recolectó. Este proceso es manual por el especialista en el ramo, el Dr. José Luíz Villaseñor Ríos.

Debido a que, el proceso es manual, se tiene una gran cantidad de trabajo atrasado por escasez de personal, tiempo y recursos. Anteriormente, este Instituto desarrolló una herramienta para ayudar en esta tarea de clasificación llamada *GENCOMEX: a Computerized Key to Identify the Genera of Asteraceae Of México* (Villaseñor, 1998), desarrollada en Pascal, por lo cual lleva un rezago (Rubio et al., 2013) ésta aplicación de 15 años aproximadamente.

El uso de la computadora en la identificación biológica puede considerarse en dos niveles: como auxiliar en la elaboración automatizada de claves para identificación, o como un medio en si para llevar a cabo la identificación (Munguía-Romero, 1992).

Para identificar una planta taxonómicamente es necesario conocer a la familia que pertenece (Munguía-Romero, 1992). Por lo tanto, para unas cuantas plantas es fácil diagnosticar su familia, pero para la mayoría de las plantas es una tarea difícil ya que requiere tiempo y conocimientos en el ramo.

## 1.2 Justificación

Debido a que, el proceso de identificación y clasificación en el Instituto Botánico de la UNAM es manual, se tiene una gran cantidad de trabajo rezagado por el alto grado de complejidad y tiempo necesario para este proceso, aunado la falta de personal y/o recursos para agilizar el trabajo pendiente. Es por esto que, se propone reemplazar el sistema antiguo por una nueva aplicación, reutilizando el trabajo ya realizado, y además, dejando una base computacional para poder clasificar: clase, familia, género y especie, dado que el anterior solo permite clasificar directamente la familia a la que la planta pertenece.

Los beneficios que conlleva un sistema taxonómico de plantas serán:

- Simplificar la entrada de datos de clasificación de las plantas
- Disminuir el tiempo del proceso de clasificación
- Agilizar el proceso de identificación.
- Usar la aplicación en diferentes dispositivos y aplicaciones.

Los sectores beneficiados con esta aplicación de manera directa e inmediata son:

- Agilizar el proceso de clasificación de todas las plantas almacenadas sin clasificar en el Instituto de Botánica de la UNAM, dando como resultado: la actualización del “museo de plantas”, y también, investigaciones y publicaciones derivado de la posibilidad de encontrar nueva información en las plantas almacenadas.
- La aplicación puede clasificar una planta por clase, familia, género y especie, permitiendo compartir información especializada entre la UNAM y el Colegio de Postgraduados en el ámbito académico, y también, permitiendo el uso de esta herramienta en el área docente en ambas universidades.
- Las empresas que desarrollan medicinas nuevas, necesitan el nombre correcto de las plantas con las que están trabajando para poder obtener la patente por razones de seguridad, por lo cual con esta aplicación pueden conocer el nombre correcto y al grupo al que pertenece dicha planta, agilizando muchos procesos administrativos y evitando desinformación por parte de terceros no especializados en el ramo.
- En las aduanas se tiene un listado de plantas provenientes de otros países las cuales su acceso es prohibido, ya sea por ilegalidad o por razones de ecosistema. Debido a que el personal en turno no tiene conocimientos de clasificación taxonómica de plantas, esta aplicación agilizará el proceso para permitir o negar su acceso.
- Por último, al público en general que desee buscar la información y clasificación de una planta. Para las personas con escasos conocimientos puede buscar por

clase y familia. Para las personas con conocimientos especializados en el ramo, pueden realizar la búsqueda hasta género y especie.

### **1.3 Objetivos**

Objetivo general

Aplicar métricas de similitud en la identificación de especies de plantas, usando *String Matching Comparison* en la identificación para la clasificación de plantas del Instituto Botánico de la UNAM.

Objetivos particulares

- Actualizar la herramienta *GENCOMEX* con tecnología Java JRE, mediante la creación de una aplicación para la identificación de plantas a partir de sus características para determinar: clase, familia, género y especie.
- Crear una aplicación de consulta vía *Web* para identificar y clasificar una planta en base a la información capturada y validada anteriormente.

### **1.4 Hipótesis**

Desarrollo de una aplicación en Java usando Modelo Vista Controlador (MVC) para la clasificación de plantas en base a sus características utilizando métricas de similitud.

# Capítulo II

## Revisión bibliográfica y estado del arte

A continuación se presenta un conjunto de investigaciones realizadas que muestran un panorama general en la rama de la inteligencia artificial con respecto a patrones representados en forma de cadenas de texto, la base teórica matemática desarrollada en las que se fundamenta, algunas de las métricas de similitud desarrolladas hasta este momento, casos reales optimizados mediante métricas, y finalmente, medidas de similitud desarrolladas específicamente para cadenas binarias.

También se integro como primer artículo el cual argumenta la importancia de la computadora especialmente en la botánica, ya sea para en investigaciones interdisciplinarias, así como en la identificación taxonómica.

Finalmente en el estado del arte muestra las aportaciones o áreas de investigación de cada artículo al trabajo de esta tesis.

### 2.1 Revisión Bibliográfica

En el trabajo de investigación la computadora en la identificación botánica *La era digital Ciencia y Desarrollo* (Villaseñor et al., 1998) muestran el uso que tiene especialmente en la botánica.

Su objetivo es discutir la importancia de la identificación biológica (Munguía-Romero, 1992), así como el papel de la computadora como herramienta de ayuda para el especialista en esta área.

La metodología que se llevó a cabo fue:

- Primeramente definir o contextualizar la identificación botánica.
- Acto seguido proceden a definir que es una clave de identificación y su proceso.
- Después vincula el proceso de identificación con la computadora como herramienta de apoyo.
- Finalmente demuestran un ejemplo de este proceso, véase la Figura 1.



Figura 1 Demostración de la computadora en la botánica (Munguía- Romero, 1992)

Los resultados aportados son una identificación más sencilla y versátil en este proceso largo de la identificación taxonómica.

Su aportación radica en crear software efectivo a todo un nivel en la jerarquía de taxones, subfamilias, géneros, especies, etc, así como generar *software* altamente especializado a través de equipos interdisciplinarios.

En el trabajo de investigación *A Consensus Algorithm for Approximate String Matching ASM* (Rubio et al., 2013) hablan sobre la importancia del ASM, aplicaciones, componentes y sus desventajas debido al número de falsos positivos que se reportan debido a las técnicas Hamming o Levensthein.

Su objetivo fue proponer un algoritmo ASM eficiente con bajo porcentaje de falsos positivo, el cual no se basa en métricas Hamming o Levensthein.

La metodología que llevan a cabo es la siguiente:

- Primero usan como base el algoritmo Baeza-Yates y Perleberg utilizando solo la substracción.
- Posteriormente realizan un censo de los posibles símbolos en la cadena a buscar con un máximo de k inserciones o eliminaciones.
- Finalmente reducen los falsos positivos preservando solamente la posición original donde se detectó el patrón.

Los resultados obtenidos muestran que tanto para  $k = 3, 6$  y  $10$  para los verdaderos positivos y falsos positivos sin procesar y post-procesando con el algoritmo planteado.

La relación de verdaderos positivos en ambas técnicas es muy similar y el costo computacional con el post-procesamiento es de 1% - 6%. La reducción de falsos positivos en  $k = 3$  por mencionar un caso es de 1317.94 a 11.67.



Concluyen proponiendo un nuevo método basado en utilizar el método de Baeza-Yates & Petesberg para buscar errores, inserciones o supresiones, y posteriormente, una fase post-procesamiento para reducir el número de falsos positivos (Rubio et al., 2013).

Esta investigación usa patrones en representadas en forma de cadena de texto, y por otra parte, el uso de la métrica *ASM* para obtener grados de similaridad entre patrones.

En el trabajo de investigación de *Assessment of Name Based Algorithms for Land Administration Ontology Matching* (Zaremba et al., 2014) hablan sobre el problema de interoperabilidad entre bases de datos de administración de tierras, y también, el tiempo que se invierte por parte de los expertos para determinar qué tan parecidas o diferentes son en sus metadatos.

Su objetivo es crear una ontología de dominio que pueda ser usada para relacionar bases de datos relacionales de administración de tierras.

La metodología que se llevó a cabo fue la siguiente:

- Primero definieron el estándar de *software* ISO 9152-2012 el cual permite combinar los recursos de las bases de datos de manera coherente, ver Figura 2.
- Posteriormente desarrollaron una ontología de las bases de datos llamada Latvia, focalizándose solamente en las 248 clases que se encontraron.
- Acto seguido, pre-procesaron la información por normalización (reducir las cadenas en un formato Standard) y *tokens*, el cual incrementa hasta en un 20% la eficiencia de los resultados (Zaremba et al., 2014).
- Finalmente comparan los resultados con las siguientes métricas: distancia Levenshtein, Jaro-Winkler, algoritmo MongeElkan y con la subcadena común más larga.

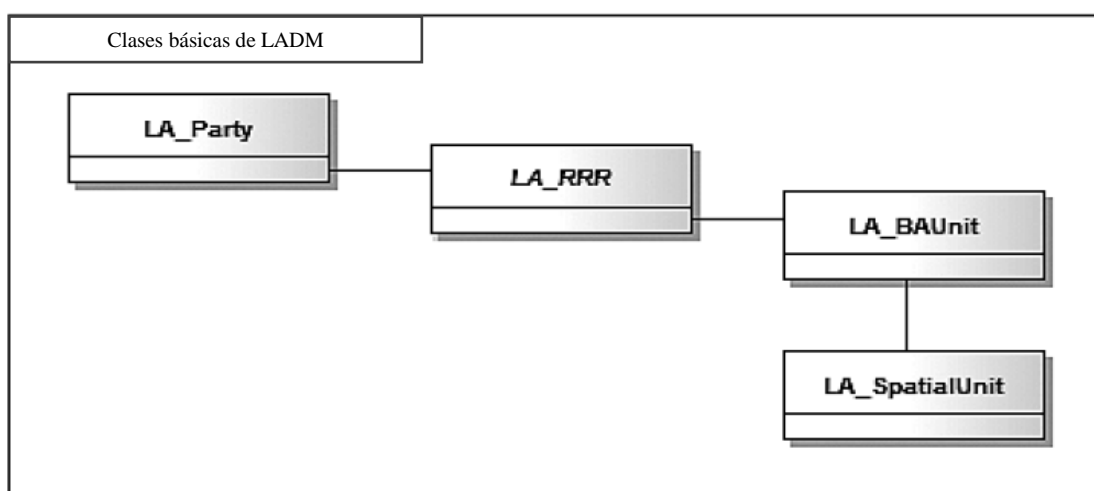


Figura 2 Estándar de desarrollo para bases de datos (Zaremba, Teilans, Rausis, & Buls, 2014).

Los resultados obtenidos fueron que, el pre-procesamiento permite quitar detalles innecesarios. Los resultados obtenidos de cada métrica es un número entre los rangos de 0.0 para cuando los datos son diferentes y 1.0 cuando los datos son iguales. La métrica Jaro-Winkler mostró la mayor cercanía de relación entre datos.

Concluyen que para relacionar bases de datos se siga el estándar ISO 9152-2012 para el diseño, usar la métrica Jaro-Winkler para crear las ontologías entre bases de datos para agilizar su interoperabilidad. También remarcan, la desventaja de que no funciona en bases de datos estructuradas con bases de datos con entidades abstractas (Zarembko et al., 2014).

Esta investigación aplica las métricas Levenshtein, Jaro-Winkler, algoritmo MongeElkan para determinar grado de similaridad entre bases de datos al analizar los archivos DTD's o esquemas, debido a que estos archivos son cadenas de texto estructuradas para facilitar la interoperabilidad.

En el trabajo de investigación *Instance based Matching using Regular Expression* (Mehdi et al., 2012) abordan el problema encontrar de correspondencias entre dos esquemas diferentes tanto en estructura como formato.

El objetivo es desarrollar un completo y automático proceso de comparación basado en expresiones de instancias.

La metodología usada fue la siguiente:

- Primero usan como entradas un esquema fuente y un esquema objetivo.
- Posteriormente las expresiones regulares son creadas del esquema fuente, acto seguido se realiza un proceso de identificación donde se genera un *token* por cada atributo excepto de las cadenas. En la segunda fase, utilizan las expresiones generadas en la primera fase y las comparan directamente con el esquema objetivo incluidos los valores de las expresiones regulares fuente.
- Finalmente, utilizaron como prueba los conjunto de datos de los restaurantes Zagat and Fooder, y por otra parte, utilizaron como medición la notación de verdaderos positivos (TP), falsos positivos (FP), verdaderos negativos (TN) y falsos negativos (FN).

Los resultados obtenidos de 10 experimentos realizados de selección al azar muestran que el promedio de las expresiones regulares tiene un 98% de eficiencia quedando por encima de las demás técnicas utilizadas.

Concluyen que este método no necesita de aprendizaje o técnicas de similitud para funcionar, también representa el primer intento usando expresiones regulares para correspondencia entre atributos y esquemas, y finalmente, que el método basado en expresiones regulares es mejor que otras muy conocidas técnicas de comparación (Mehdi et al., 2012).

Esta investigación hace uso de cadenas como fuente para la identificación de patrones.

En el trabajo de investigación *Funciones de similitud sobre cadenas de texto: Una comparación basada en la naturaleza de los datos* (Amón et al., 2010) abordan la problemática del duplicado de información o tuplas en las bases de datos.

El objetivo es evaluar el desempeño de las medidas de similitud: Levenshtein, Brecha Afín, Smith-Waterman, Jaro, Jaro-Winkler, Bi-grams, Tri-grams, Monge-Elkan y SoftTF-IDF mediante la métrica discernibilidad en variaciones textuales presentes en las representaciones de una misma entidad.

La metodología fue la siguiente:

- Primero describen las funciones de similitud sobre cadenas de texto que son: Levenshtein (conjunto mínimo de operaciones de edición para transformar una cadena A en B y/o viceversa), Brecha afín ( usada para evaluar similitud entre cadenas que usan abreviaturas o la omisión de *tokens*), Smith-Waterman (usada para identificar cadenas equivalentes con prefijos/sufijos, que al no tener valor semántico son descartados), Jaro (medida de similitud permitiendo solo la transposición), Jaro\_Winkler (variante de similitud de cadenas que comparten algún prefijo), Q-grams (se usa cuando dos cadenas son muy similares).
- Posteriormente describen las funciones de similitud basadas en *tokens* que son: Monge-Elkan (máxima similitud promedio entre una pareja [a1, b1]), y TF\_IDF (definida por el ángulo que forman sus vectores respectivos descriptivos).
- Acto seguido evalúan las funciones de similitud sobre cadenas de texto mediante la función de discernibilidad (Amón et al., 2010), la cual determina la eficacia de las funciones de similitud con el objeto a comparar. Véase la Fórmula 1.

$\frac{C1}{C1 + C2} +  T \text{ óptimo max} - T \text{ óptimo min}  + \frac{C2}{C1 + C2} \left( \frac{F \text{ max}}{2 Q } \right)$	( 1 ) Función de discernibilidad
-------------------------------------------------------------------------------------------------------------------------------------	----------------------------------

Acto seguido, se hace la prueba paramétrica de Kruskal-Wallis debido que todas las métricas arrojan resultados variables entre sí. La prueba paramétrica arrojó un nivel de confianza del 95 %, debido a las diferencias estadísticas entre las medianas.

Concluyen que algunas funciones tienden a fallar en la presencia de variaciones textuales, por otra parte en cada caso realizado demostraron cual métrica es mejor en casos específicos, y finalmente, el análisis arrojó la función de similitud Brecha Afín con los mejores resultados en las pruebas con problemas de abreviaturas (Amón et al., 2010).

Esta investigación hace una recopilación de métricas de similitud, su funcionamiento aplicadas para conocer grados de similaridad entre diferentes cadenas de texto.

En el trabajo de investigación *Parameterized matching with mismatches* (Apostolico et al., 2007) tratan el problema de encontrar todas las ocurrencias de un patrón en un texto.

Su objetivo es estudiar variantes de aproximacion con errores controlados en un texto.

La metodología es la siguiente:

- Proporcionan definiciones y propiedades tales como longitud de cadena, concatenación, alfabeto, complejidad algorítmica y búsqueda parametrizada.
- Posteriormente, concatenan las cadenas a comparar para reducir el tiempo de procesamiento.
- Finalmente adicionan notaciones para su procesamiento, vease Figura 3.

	1		3						8		10		12		14		17		19	20						
...21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44...			
1	0	0	1	1	1	1	1	0	0	1	1	0	0	1	1	1	0	0	1	0	0	0	0	1		
	a	a	a	b	b	b	b	b	a	a	a	a	b	b	b	a	a	b	a	a	a	b				
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22				
	1			4					9				13			16		18	19			22				
	1		3	4				8	9	10		12	13	14		16	17	18	19	20		22				

**Figura 3 Ejemplo de cadenas concatenadas con anotaciones (Apostolico et al., 2007).**

Los resultados obtenidos son: propiedad de fusión, propiedad de vecino. Por otra parte, en caso de no haber errores en búsqueda: el número de  $S(k)$  patrones buscados solo cambiará patrón por unidad y viceversa. En caso de existir errores el número de saltos será de  $R_p \times R_t$  por cada error.

Las conclusiones a las que llegan son: tiempo de búsqueda más corta y alfabeto generalizado no solamente para búsquedas parametrizadas (Apostólico et al., 2007).

Esta investigación defini algunas bases y propiedades en la busqueda de patrones en cadenas de texto.

En el trabajo de *Métricas de Similitud para Búsqueda Aproximada* (García, 2007) aborda la búsqueda aproximada para encontrar el número de apariciones de un determinado patrón en un texto permitiendo un cierto número de diferencias o errores.

El objetivo del trabajo es mostrar el estado del arte de la búsqueda aproximada desde el punto de vista de las distancias entre las cadenas de caracteres de texto.

La metodología utilizada es la siguiente:

- Primero definen lo que es un alfabeto, cadena de caracteres, patrón, texto, errores, alineamiento, transcripción, distancia o métrica de similitud, búsqueda exacta, búsqueda aproximada, búsqueda probabilística de proximidad y operaciones de bits.
- Acto describe las siguientes operaciones en búsqueda de texto que son: inserción, eliminación, sustitución y transposición.
- Posteriormente, describe las distancias recopiladas que son: distancia Levensthein (búsqueda mínima entre dos cadenas), Edición por Bloques (hace referencia a distancia Levensthein permitiendo adiciones, inserciones y eliminaciones entre subcadenas y no por caracteres individuales), Hamming (permite reemplazos en cadenas de igual longitud), Episodio (no es simétrica y permite la posibilidad de la existencia de una distancia infinita), Indel (solo permite inserciones y eliminaciones, muy parecida a Levensthein), Q-Grams (entre más similares sean dos cadenas, más subcadenas compartiran), SAD y MAD (utilizadas cuando la distancia del texto y del patrón son iguales), Damerau (distancia Levenshtein permitiendo la transposición), Segmento Afín (donde tienen una penalización menor en abreviaturas), Smith-Waterman (se ignoran los prefijos y sufijos de ciertas cadenas tratadas como Segmento Afín), Jaro (computa el i-esimo carácter patrón con el j-esimo carácter a comparar) y Markov (utiliza información oculta en los patrones y en los textos).
- Finalmente enuncia los algoritmos para agilizar la búsqueda aproximada que son: paralelización por bits, y programación dinámica vease Tabla 1.

**Tabla 1 Ejemplo de pseudocódigo programación dinámica (García, 2007).**

LPDSEARCH(p, t, k)	
Donde: p = vector renglon con valores preguardados t = vector columna con valores preguardados k = valor a comparar D = matriz donde se realiza la búsqueda MIN = funcion que se manda a llamar que recibe como argumento una matriz y devuelve el valor más pequeño	
Algoritmo	
1	cuenta todos los renglones de la matriz D
2	cuenta todos las columnas de la matriz D
3	ordenar los valores de la matriz de la siguiente manera:  for i = 1 hasta todas las columnas for j = 1 hasta todos los renglones si $p_i = t_j$ entonces $D_{i,j} = D_{i-1,j-1}$ si no $min = \text{MIN}(D_{i-1,j}, D_{i,j-1}, D_{i-1,j-1})$ $D_{i,j} = min + 1$
4	por cada columna de la matriz D si $D_{i,j} \leq k$ entonces imprime ese valor

Por la parte de resultados, el autor describe las siguientes aplicaciones de la búsqueda aproximada: Bio-informática (búsqueda de ADN), reconocimiento de patrones (cadenas de texto), procesamiento de señales (señales de audio), comparación de archivos (bits o bytes), corrección de texto (reemplazar una palabra por otra), minería de datos (programación de *firmwares*) y bases de datos (consultas mediante el método de vecino próximo).

Las conclusiones son: presenta una revisión de distancias, métricas y algoritmos utilizados en búsqueda aproximada, y también, encuentra una necesidad de algoritmos más flexibles sin aumentar la complejidad del algoritmo (García, 2007).

Esta investigación prove definiciones y operaciones entre cadenas de texto, también el dar un panorama más profundo de medidas de similitud y su funcionamiento, y finalmente dar a conocer áreas de aplicación de dichas métricas.

En el trabajo de investigación *Intuitionistic Fuzzy Automaton for Approximate String Matching* (Ravi et al., 2013) estudian el caso particular de *String Matching* permitiendo errores, borrado, inserciones y reemplazar caracteres.

El objetivo es proponer un sistema basado en lógica difusa para la evaluación de dos cadenas similares.

La metodología utilizada es la siguiente:

- Primero definen un alfabeto en donde las funciones representan un rango de números reales entre 0-1.
- Posteriormente definen un lenguaje regular de lógica difusa (FRL).
- Acto seguido, crean un lenguaje de lógica difusa intuicionista (IFL) en base al lenguaje creado, en segundo lugar definen un autómata determinístico finito, luego crean un autómata basado en lógica difusa en base al autómata determinístico.
- Finalmente, hacen un autómata basado en lógica difusa intuicionista con las operaciones de comparación inserción o eliminación, véase la Tabla 2.

**Tabla 2 Operaciones de autómata fuzzy intuicionista (Ravi et al., 2013).**

Observed string	Operation	Transition	Value	Effect
$x_1 \cdots x_{k-1} x x_{k+1} \cdots x_m$	No error $c_{aa}^k(x = a)$	$\tau(q, a, p),$ $\gamma(q, a, p)$	(1,0)	$x_1 \cdots x_{k-1} a x_{k+1} \cdots x_m$
$x_1 \cdots x_{k-1} x x_{k+1} \cdots x_m$	Insertion $i_a^k$	$\tau(q, \epsilon, p),$ $\gamma(q, \epsilon, p)$	$(\tau_{i_a^{qp}}, \gamma_{i_a^{qp}})$	$x_1 \cdots x_{k-1} a x x_{k+1} \cdots x_m$
$x_1 \cdots x_{k-1} x x_{k+1} \cdots x_m$	Substitution $c_{xa}^k(x \neq a)$	$\tau(q, x, p),$ $\gamma(q, x, p)$	$(\tau_{c_{xa}^{qp}}, \gamma_{c_{xa}^{qp}})$	$x_1 \cdots x_{k-1} a x_{k+1} \cdots x_m$
$x_1 \cdots x_{k-1} x x_{k+1} \cdots x_m$	Deletion $d_x^k$	$\tau(q, x, q),$ $\gamma(q, x, q)$	$(\tau_{d_x^q}, \gamma_{d_x^q})$	$x_1 \cdots x_{k-1} x_{k+1} \cdots x_m$

El procedimiento para los resultados fue: definen el algoritmo para comparar dos cadenas, posteriormente ejecutan en algoritmo obteniendo valores de similaridad y disimilaridad (0.65 y 0.3 respectivamente), y también que por dada transición de distancia = 1.

Concluyen que el modelo propuesto puede generalizarse para cada cadena de texto que contenga algún patron, también que es un modelo basado en estadística, y finalmente, provee un robusto modelo para imprecisiones en comparación a los modelos convencionales (Ravi et al., 2013).

Esta investigación usa cadenas de texto que forman patrones auxiliándose de lógica difusa.

En el trabajo de investigación *Parameterized Pattern Matching: Algorithms and Applications* (Barker, 1996) tratan el problema de copiar código para arreglar errores o añadir nuevas utilidades en un sistema de software puede originar más errores de los originales.

El objetivo es crear un método para evitar trabajar código más del necesario, tiempo y finalmente, no crear duplicados en el proceso.

La metodología utilizada fue la siguiente:

- Primero definen p-strings (cadenas patrón), p-matches (patrones a comparar) y árbol p-sufijos, ver Figura 4.
- Posteriormente crean un árbol p-sufijos aumentado con enlaces de sufijos debido a: la construcción lineal de tiempo de McCreight's para árboles con sufijos, propiedad de prefijo común, propiedad contextual derecha de distintas restricciones.
- Acto seguido, reordenan el árbol de sufijos aumentado debido a propiedad principal de no decremento, y teniendo en cuenta: a) enlaces de sufijos son definidos en términos de la estructura del árbol, el cual cambia conforme los vértices se agregan, b) enlaces malos no apuntan exactamente hacia donde se necesita, c) revisar las entradas.
- Finalmente buscan ocurrencias en un patrón.

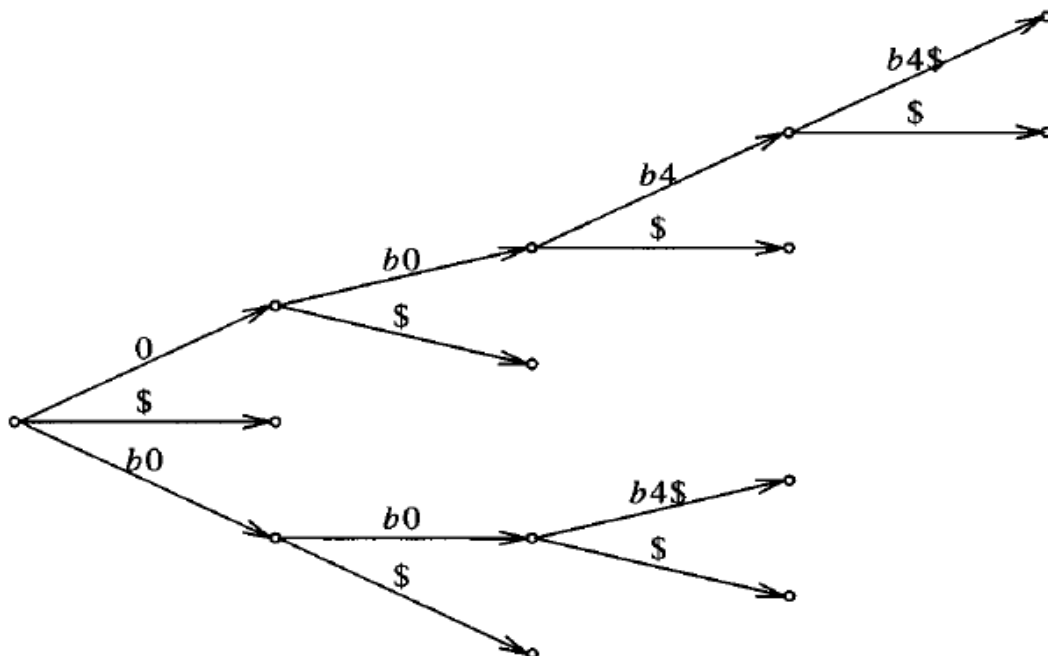


Figura 4 Ejemplo de árbol p-suffix (Barker, 1996).



Por la parte de resultados se implementó en Lenguaje C, sistema operativo UNIX, procesador R3000, 40 Mhz, caché nivel 1 de 64Kb. Se ignoraron los espacios en blanco y comentarios, los identificadores se representaron como *tokens* y se creó una tabla *hash* para obtener una representación numérica, cada línea se transforma al encontrar un patrón con el símbolo P sumando las P totales de esa línea, las P's son marcadores para las posiciones en cada línea, finalmente se accede a las representaciones a través de las tablas hash creadas. En el código utilizado, el análisis léxico de 1 085 432 líneas tomó un total de 190 segundos para la reconstrucción de los patrones y tablas *hash* correspondientes. En la construcción del árbol que representa el código tomó 100 segundos, tomando un total de 400 segundos en reconstruir el código.

Concluye que el algoritmo propuesto incrementa la velocidad en 2 con 30 líneas de código, en un factor de 8 de 15 líneas. También que la sencillez del árbol puede generalizarse para más patrones tipo cadena y un programador puede buscar errores en código utilizando este método.

Esta investigación encuentra patrones repetitivos código de programación en formato de texto.

En el trabajo de investigación *Approximating Edit Distance Efficiently* (Bar-Yossef et al., 2004) demuestran la eficiencia de una medida fundamental de similaridad entre dos cadenas llamada distancia Levenshtein, la cual es el mínimo de inserciones, supresiones y sustituciones de caracteres necesarios para transformar una cadena a otro.

El objetivo del trabajo es presentar una mayor eficiencia en algoritmos que utilizan distancia Levenshtein tanto en espacio como tiempo.

La metodología utilizada fue la siguiente:

- Primero esquematizan los algoritmos en compresión de procedimientos y reconstrucción del procedimiento para construir un diagrama de cada una cadena sin que se comparen entre ellos.
- Posteriormente aplican el algoritmo por Cole and Hariharan para lograr un tiempo de ejecución quasi-lineal.
- Finalmente, acoplan el algoritmo para cadenas no repetitivas y cadenas arbitrarios.

Por la parte de resultados obtuvieron que los algoritmos obtenidos se aplican a problemas que utilizan distancia Levenshtein, demostrando los tiempo más cortos, también utilizan múltiples aristas simultáneamente, el cual en esencia es comparación de patrones (Bar-Yossef et al., 2004).

Las conclusiones a las que llegan son al reducir la distancia Hamming a distancia Levenshtein y por ahora el modelo propuesto no se pudo generalizar a más distancias (Bar-Yossef et al., 2004).

Esta investigación profundiza en la distancia Levenshtein y sus propiedades.

En el trabajo de investigación *A Comparison of String Metrics for Matching Name and Records* (Cohen et al., 2003) formulan entidades de comparación como un problema de clasificación entre pares.

El objetivo es revisar los experimentos anteriores de comparación en cadenas individuales, y posteriormente discutir los experimentos realizados con su *toolkit* de comparación de objetos estructurado.

La metodología utilizada es la siguiente:

- Primero describen las funciones que tiene *SecondString* y su funcionamiento, y por otra parte, la posibilidad de agregar funciones definidas por el usuario.
- Posteriormente implementan funciones de distancia como Jaro, Jaro-Winkler, *tokens*, similaridad coseno y Jaccard.
- Acto seguido prueban las medidas con las cadenas: *animal*, *bird1*, *bird2*, *bird3*, *bird4*, *business*, *game*, *park*, *fodorZagrat*, *ucdFolks* y *census*.
- Finalmente crean un *multiwrapper* de manera que de una instancia se generen varias subcadenas.

En la parte de resultados realizaron los experimentos con los conjuntos de entrenamiento de Census y Cora, en Census algunos *tokens* se insertó la letra capital “T” y agrupados en grupos. La distancia Levenshtein fue la peor en desempeño en promedio a las demás. En Census TFIDF tiene un desempeño más bajo que Jaccard, ninguno de los métodos basados en *tokens* tuvieron un buen desempeño en Census, el método híbrido Jaro-Winkler fue el que tuvo el mejor desempeño en Census a comparación de las demás métricas usadas.

Concluyen que las distancias de las cadenas aunque son comúnmente útiles no son buenas al comparar entidades o estructuras no triviales, también muestran resultados con las diferentes medidas hoy en día utilizadas y el mejor método encontrado es la distancia Levenshtein modificada por Winkler en base a la distancia de Jaro (Cohen et al., 2003).

En este trabajo se comparan los patrones en forma de text en bases de datos.

En el trabajo de investigación *Structural off-line handwriting carácter recognition using approximate subgraph matching and levenshtein disntace* (Wira et al. ,2015) tratan el problema del reconocimiento de letras escritas a mano en imágenes.

El objetivo es proponer y comparar los resultados mediante 2 métodos que son: *subgraph matching* y *edit distance* para el reconocimiento de letras a mano.

La metodología utilizada fue la siguiente:

- Primero utilizaron *Otsu's global thresholding* para convertir una imagen de una letra escrita a mano a una imagen binaria. Acto seguido, se hizo la letra más delgada mediante el algoritmo de Zhang & Suen.
- Posteriormente, se procedió a la extracción de características el cual fue: primero la extracción de las curvas mediante el método de análisis de componentes conectados, segundo la representación de las características en una cadena.
- Tercero la representación de una letra grafica en forma de cadena.
- Finalmente, procedieron a procesar las cadenas generadas por los métodos: a) *subgraph matching* en la cual la gráfica se dividió usando expresiones regulares y mapeada en vértices y esquinas, b) *edit distance*.

Por la parte de resultados utilizaron la base de datos ETL-1 de *Electrotechnical Laboratory National Institute of Advanced Industrial Science and Technology (AIST)* la cual posee imágenes de números, alfabeto, caracteres especiales, caracteres *katakana* japonés, adquiridas de 1445 escritores, el primer experimento, se usó un 10% de entrenamiento de la base de datos el cual demostró que la distancia Levenshtein fue superior en *subgraph matching* por más de 10% de eficiencia y menor costo computacional, el segundo experimento se probó solamente con distancia Levenshtein y se redujo el conjunto de entrenamiento al 5 % el cual bajo solo un 5 % hacia un total de 77 %. Los resultados finales muestran una competitiva eficiencia contra *Hidden Markov Model* y otros métodos comunes utilizados.

Concluyen que proponen una técnica mejorada y basada en cadenas de texto sin depender de técnicas de normalización, también proponen un modelo basados en cadenas de texto para representar curvas, y finalmente, demuestran que la distancia Levenshtein tiene una mejor eficiencia de reconocimiento que la técnica de *subgraph matching* (Wira et al., 2015).

En este trabajo usan las cadenas de texto para representar patrones gráficos acorde a un estándar definido por los investigadores.

En el trabajo de investigación *On the similarity metric and the distance metric* (Chen et al., 2009) comentan que cuando una medida de disimilitud es utilizada, se usa normalmente una métrica de distancia. Sin embargo, cuando una medida de similitud es utilizada, no hay un requerimiento formal para su uso (Chen et al., 2009).

El objetivo es dar tres contribuciones que son: dar una definición formal de métrica, mostrar la relación entre métrica de similitud con métrica de distancia, y finalmente, presentar soluciones generales para normalizar métricas de similitud y distancia.

La metodología utilizada fue la siguiente:

- Primero definen métrica de distancia y métrica de similitud, posteriormente definen la métrica de distancia normalizada ( $d(x, y) \leq 1$ ), y también, la métrica de similitud normalizada ( $|s(x, y)| \leq 1$ ).

- Posteriormente establecen la relación entre la métricas de similaridad y distancia al probar que: sea la métrica de distancia  $e^{-d(x,y)}$  es una métrica de similitud normalizada  $1 - e^{-d(x,y)}$ .
- Acto seguido definen formalmente la métrica de similaridad normalizada.
- Finalmente definen la métrica de distancia normalizada.

Por la parte de resultados aplican las métricas normalizadas de forma general a los casos de: distancia gráfica, distancia de atributos tipo árbol, complejidad de Kolmogorov y en minería de datos.

Concluyen que al proveer una definición formal de la métrica de similaridad, también muestran la relación entre las métricas de similaridad y distancia, proveen una formula general para normalización de métricas de similitud (Chen et al., 2009).

En este trabajo se proveen de fundamentos matemáticos para los patrones en cadenas de texto y métricas de similitud.

En el trabajo de investigación *Jaccard Index based Availability Prediction in Enterprise Grids* (Rahman et al., 2012) tratan el problema de predecir la viabilidad de las máquinas de escritorio o sistemas tipo Grid.

El objetivo es describir e implementar una metodología para predecir la viabilidad de grandes recursos de una empresa, basado en el monitoreo de datos recolectado de varias empresas, inclusive en ambientes muy cambiantes con alto grado de eficiencia.

La metodología utilizada fue la siguiente:

- Primero se seleccionó los modelos a trabajar que son: 51 662 computadoras de la red corporativa de Microsoft, y por otra parte, 321 nodos de las instalaciones de investigación de PlanetLab.
- Posteriormente se propuso el índice de Jaccard usando el algoritmo de aprendizaje lento para la predicción.
- Acto seguido, se transformó el comportamiento histórico obtenido a caracteres binarios. Finalmente, se incorporó la fase de votos para poder realizar una sola predicción.
- La Figura 5 muestra el proceso de comportamiento histórico representado en 0's y 1's.

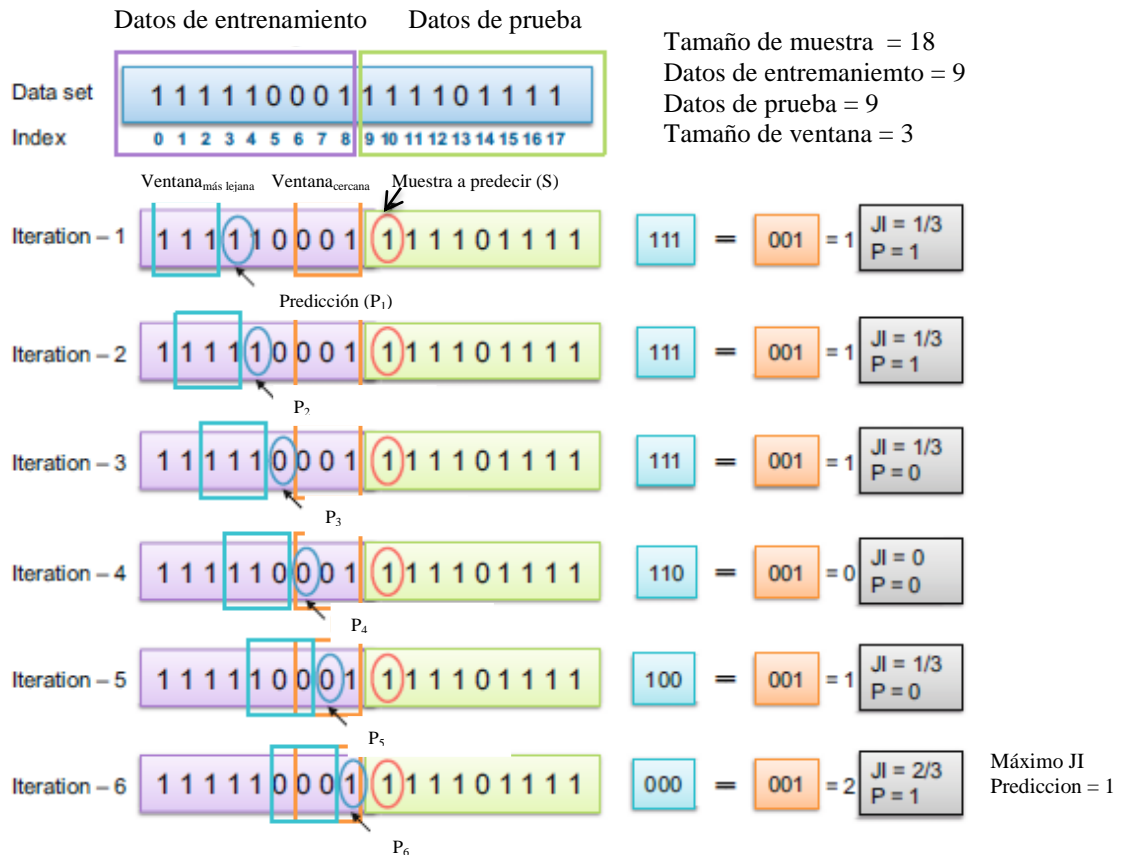


Figura 5 Representación de comportamiento mediante Jaccard (Rahman et al., 2012).

Por la parte de resultados recabaron la información de los centros mencionados durante dos semanas los cuales fueron utilizados como fase de entrenamiento, posteriormente durante tres semanas evaluaron la eficiencia de la evaluación en periodos de una hora, también utilizaron otras tres técnicas de predicción (*K-Nearest Neighbors*, *Naive Bayes* y *Hybrid predictor*) para comparar resultados.

Concluyen que los experimentos realizados muestran que el índice de Jaccard tiene una eficiencia de 99.74 % para PlanetLab y 96.87 % para Microsoft, reduciendo la complejidad computacional comparada contra las otras técnicas utilizadas (Rahman et al., 2012).

En este trabajo usan cadenas de texto binarias para representar patrones y predecir comportamientos en grupos de trabajo.

En el trabajo de investigación de *Personalized Tag Recommendations via Tagging and Content-based Similarity Metrics* (Byde et al., 2007) tratan el problema de la recomendación de etiquetas en sitios de tipo social.

El objetivo es realizar recomendaciones basado en dos diferentes métricas que son por etiquetado y por contenido.

La metodología utilizada fue la siguiente:

- Primero se seleccionó un conjunto de usuarios con sus respectivos *tags* de cada uno de la base del.icio.us.
- Posteriormente se seleccionó la métrica Métrica de similitud coseno modificada (Byde et al., 2007) ver Fórmula 2, la cual es una variante de la métrica coseno.
- Finalmente, se aplicó dicha métrica a los datos seleccionados una muestra de 200 usuarios recientes y las *url*'s etiquetadas fueron en promedio 150 por usuario.

$\sin(u, u') = \frac{u * u'}{\sqrt{(u * u)(u' * u')}}$	<b>( 2 ) Métrica de similitud coseno modificada.</b>
--------------------------------------------------------	------------------------------------------------------

Por la parte de resultados mostraron que el método basado en contenido fue mejor que al basado por etiquetas, como por ejemplo en cobertura el método de etiquetado obtuvo una eficiencia de 63.1 % mientras que basado por contenido obtuvo un 93.1 %. Por otra parte ambos métodos en el resto de las categorías mostraron un comportamiento similar.

Concluyen que proponen un método novedoso para reconocimiento de semántica en etiquetas basándose en la métrica de similaridad, tanto para contenido como para *urls*. El método provee recomendaciones personalizadas para los usuarios. El método basado en etiquetado es más ligero en implementación pero es menos eficiente que el basado por contenido (Byde et al., 2007).

En este trabajo usan los patrones de texto usados en texto o etiquetas para reconocer y proponer nuevos contenidos de interés a los usuarios.

En el trabajo de investigación *Simpack: A Generic Java Library for Similarity Measures in Ontologies* (Bernstein et al., 2006) tratan el problema de la determinación semántica a través de algoritmos en un ontología.

El objetivo del trabajo es presentar un marco de trabajo de métricas de similitud adaptadas a grupos de conceptos llamado SimPack, y por otra parte, el proveer una extensa evaluación empírica contra las métricas estándar.

La metodología fue la siguiente:

- Primero definen formalmente a un conjunto de conceptos, propiedades e individuos representada como un árbol con raíz, etiquetado y sin orden.
- Posteriormente, definen la métrica de similitud en los grupos previamente definidos en base al número conceptos que tienen una relación entre ellos.
- Acto seguido, analizan las métricas de similitud existentes así como sus ventajas y desventajas.

- En cuarto lugar comentan las métricas de similitud usadas con vectores de igual longitud basados en cadenas que son: SimCosine (Bernstein et al., 2006), SimJaccard (Bernstein et al., 2006), SimOverLap (Bernstein et al., 2006) y SimLevenshtein (Bernstein et al., 2006), ver Fórmulas 3, 4, 5 y 6 respectivamente.
- Posteriormente deciden añadir las métricas de similitud antes mencionadas al marco de trabajo agregando  $sim_{tfidf}$  para métricas de texto completo.
- Finalmente, aplican las métricas a la base de datos PH seleccionado 40 y 13 pares de conceptos relacionados y entendibles.

$Sim_{cosine}(x, y) = \frac{X*Y}{\ X\ _2 * \ Y\ _2}$	( 3 ) <b>SimCosine</b>
$Sim_{jaccard}(x, y) = \frac{X*Y}{\ X\ _2 + \ Y\ _2 - (X*Y)}$	( 4 ) <b>SimJaccard</b>
$Sim_{overlap}(x, y) = \frac{X*Y}{\min(\ X\ _2, \ Y\ _2)}$	( 5 ) <b>SimOverLap</b>
$Sim_{overlap}(x, y) = \frac{x\ form(X,Y)}{x\ form_{wc}(X,Y)}$	( 6 ) <b>SimLevenshtein</b>

Por la parte de resultados obtuvieron que las medidas poseen una correlación no mejor ni peor tal como lo haría una persona, también obtuvieron que las medidas y sujetos están agrupados claramente en conjuntos claramente definidos, por otra parte se necesitan implementar cada métrica de similaridad en base al estilo de asignación, también las agrupaciones mostraron que se forman debido a la composición de los conceptos o debido a su naturaleza de procedencia.

Concluyen que se debe estudiar más a fondo las técnicas de *clustering*, minería de datos, desambiguación de semántica, traducción, comparación automática de esquemas de bases de datos y comparación de objetos simples, por otra parte la relación de conceptos con las antologías de los algoritmos se parece mucho a como lo haría una persona, también descubrieron que, algoritmo basado en muestras tiene una superior eficiencia que los basados en datos reportados por ellos mismos, finalmente comentan que este trabajo técnico teórico explora como las personas entienden la similitud en la agrupación de términos por categorías (Bernstein et al., 2006).

En esta investigación usan las métricas de similitud acorde al tipo de datos que están utilizando.

En el trabajo de investigación *Using Similarity Metrics for Terminology Recognition* (Butters et al., 2008) tratan el problema de reconocimiento de terminología donde un término se extrae, y en consecuencia, se asocia a uno o más posibles términos oficiales al aplicarse métricas de similaridad en cadenas de texto.

El objetivo del trabajo es presentar un método para el reconocimiento de terminología usando un umbral definido.

La metodología utilizada fue:

- Primero se seleccionó la base de datos de prueba conformada por una lista de 298 términos hecha a base de reportes, y también, por otra lista oficial de 513 términos de componentes de ingeniería.
- En segundo lugar, se seleccionaron las mejores métricas de similitud combinando cada métrica de cadenas haciendo un total de 298 de prueba.
- Posteriormente, de los resultados obtenidos de las pruebas con métricas, se seleccionó la métrica Levenshtein pues permite mayor eficiencia aun con cadenas sin relación entre ellos.
- Acto seguido, se procedió a aplicar un umbral dinámico sensible tanto al lugar como al ruido de la curva, después se procedió a evaluar el funcionamiento del umbral dinámico al compararlo con una lista de 298 cadenas de prueba.
- Finalmente se graficó el comportamiento de los resultados.

Por la parte de resultados se aplicó la métrica a 298 términos previamente seleccionados comparados a una lista de 513, también solo 112 términos tenían la similitud suficiente y 186 no tenían relación alguna.

Concluyen que el seleccionar métricas que funcionen con ruido incrementa la efectividad en la eficiencia, también que el acotar a solo los 5 resultados mejores es un buen estándar de selección de posibles resultados, y finalmente que aplicar la metodología *Terminology Recognition* puede ser aplicada virtualmente a cualquier área donde se comparen cadenas de texto (Butters et al., 2008).

En esta investigación usan terminos como patrones extraidos en reportes de texco como patrones a comparar en bases de datos.

En el trabajo de investigación *An Information-Theoretic Definition of Similarity* (Lin, 1998) trata el problema de que las medidas de similitud son solo aplicadas en un caso específico, o por otra parte, son asumidas a un particular modelo de dominio.

El objetivo del trabajo es presentar una definición de similaridad que sea universal y justificada teóricamente.

La metodología utilizada fue la siguiente:

- Primero describe de manera corta todas las creencias de una métrica de similaridad, las justifica matemáticamente y propone un teorema.



- Acto seguido prueba que la similaridad da como resultados valores numéricos derivado de su cálculo, ver Figura 6.

$\text{sim}(\text{excellent}, \text{good}) = \frac{2 \times \log P(\text{excellent} \vee \text{good})}{\log P(\text{excellent}) + \log P(\text{good})}$ $\text{sim}(\text{excellent}, \text{good}) = \frac{2 \times \log(0.05 + 0.1)}{\log 0.05 + \log 0.1} = 0.72$
$\text{sim}(\text{good}, \text{average}) = \frac{2 \times \log P(\text{good} \vee \text{average})}{\log P(\text{average}) + \log P(\text{good})}$ $\text{sim}(\text{good}, \text{average}) = \frac{2 \times \log(0.1 + 0.5)}{\log 0.1 + \log 0.5} = 0.34$
$\text{sim}(\text{excellent}, \text{average}) = \frac{2 \times \log P(\text{excellent} \vee \text{good} \vee \text{average})}{\log P(\text{excellent}) + \log P(\text{average})}$ $\text{sim}(\text{excellent}, \text{average}) = \frac{2 \times \log(0.05 + 0.1 + 0.5)}{\log 0.05 + \log 0.5} = 0.23$
$\text{sim}(\text{good}, \text{bad}) = \frac{2 \times \log P(\text{good} \vee \text{average} \vee \text{bad})}{\log P(\text{good}) + \log P(\text{bad})}$ $\text{sim}(\text{good}, \text{bad}) = \frac{2 \times \log(0.1 + 0.5 + 0.2)}{\log 0.1 + \log 0.2} = 0.11$

**Figura 6 Métricas de similitud en forma de valores numéricos (Lin, 1998).**

Por la parte de resultados aplica la universalidad de tres métricas de similitud que son *edit distance* ( $\text{Sim}_{\text{edit}}$ ), *trigrams* ( $\text{Sim}_{\text{tri}}$ ) y probabilidad *trigrams* ( $\text{Sim}$ ) en las cadenas, por ejemplo el primer renglón de la tabla muestra que la métrica  $\text{Sim}$  obtuvo la mayor similitud 92 %. Acto seguido, prueba las métricas en similaridad en triada de palabras acorde a su distribución en un texto extrayendo, la prueba se realizó analizaron un documento de 22 millones de palabras del *Wall Street Journal and San Jose Mercury*, el proceso tomó 72 horas con un procesador Pentium 2 con 80 MB en Ram, derivado del análisis se extrajeron cerca de 14 millones tripletas con relación dependiente entre ellas. También se probaron las métricas de similitud  $\text{Sim}_{\text{Hill,Coast}}$ ,  $\text{Sim}_{\text{Miller\&Charles}}$ ,  $\text{Sim}_{\text{Resnik}}$ ,  $\text{Sim}_{\text{Wu\&Palmer}}$  y  $\text{Sim}$  en la Taxonomía refiriéndose específicamente a la similaridad entre dos conceptos, los resultados mostraron la métrica  $\text{Miller\&Charles}$  mostró un comportamiento mejor que el resto de las métricas, y también, mostró una mejor correlación con el juicio humano en correlación. Finalmente, realiza una comparación entre las diferentes métricas de similitud más usadas, por ejemplo la métrica  $\text{Sim}$  posee: incremento, decremento, desigualdad triangular ( $\text{dist}(A,C) \leq \text{dist}(A,B) + \text{dist}(A,C)$ ), aseveración seis (peso ponderado entre dos diferentes perspectivas), máximo valor = 1, similaridad semántica, similaridad de palabras y valores ordinales.

Concluye que, la similaridad es un concepto importante y fundamental en la inteligencia artificial, también presenta una definición universal de métrica de similitud, y finalmente, demuestra que la universalidad de la definición es demostrada en diferentes áreas donde se aplican métricas de similitud (Lin, 1998).

Por último en este trabajo propone nuevamente bases matemáticas para validar a un patrón en forma de texto, dentro de sus áreas de trabajo se destaca su posible uso en la taxonomía, y finalmente, el uso de las matrices de similaridad en el campo de la inteligencia artificial.

## 2.2 Estado del Arte

Tabla 3 Ventajas y desventajas de métricas de similitud existentes primera parte.

Algoritmo o métrica	Ventajas	Desventajas
<i>Approximate String Matching (ASM)</i>	Reduce el número de falsos positivos	Solo toma en cuenta inserciones o supresiones
Levenshtein	Busqueda mínima entre 2 cadenas	Se necesitan cadenas simétricas y acotadas
Jaro-Winkler	Mide la similitud que comparten algún prefijo	Su funcionamiento óptimo se limita a cadenas cortas tales como nombres
Jaro	Computa el i-esimo caracter patron con el j-esimo caracter a comparar	Permite solo la transposición
Q-gram	Se usa cuando dos cadenas son muy similares	No es eficiente con cadenas grandes diferencias entre ellas.
Expresiones regulares	Utilizada para la búsqueda de patrones de cadenas de caracteres u operaciones de sustituciones	Se necesita un alfabeto previamente definido
Brecha Afin	Evalua similitud entre cadenas que usan abreviaturas o la omisión de identificadores	Bajo costo de penalización en la inserción o eliminación
Monge-Elkan	Mide la máxima similitud entre promedio entre parejas	No es útil para cadenas grandes.
Cosine	Se usa cuando los dos vectores a comparar forman un ángulo entre ellos	No se puede aplicar esta métrica cuando no existe algun ángulo entre cadenas, o también, el ángulo resultante debe ser positivo
Smith-Waterman	Identifica cadenas equivalentes entre prefijos/sufijos	Solo limitado a parejas de cadenas
Edición por bloques	Permite las operaciones de la métrica Levenshtein pero en subcadenas y no caracteres individuales	Necesita incluir al algoritmo para la extracción de identificadores, los cuales son algoritmos NP-Completos
Hamming	Permite reemplazos con cadenas de igual longitud	No puede aplicar cuando las cadenas a comparar son de longitud diferente

**Tabla 3 Ventajas y desventajas de metricas de similitud existentes segunda parte.**

Algoritmo o metrica	Ventajas	Desventajas
Episodio	Permite la posibilidad de una distancia infinita y no es simétrica	Costo computacional alto
Indel (LCS)	Parecida a la métrica Levenshtein pero solo permite inserciones y eliminaciones	Limitante en el uso de operaciones entre caracteres
SAD y MAD	Se usan cuando la distancia del patron y texto son iguales	Baja eficiencia cuando la distancia entre patron y texto es diferente
Damerau	Distancia Levenshtein permitiendo la transposición	Considera solo errores que pueden ser solucionados con una operación de edición solamente
Markov	Utiliza información oculta en los patrones y texto	Penalización entre cadenas no es constante
Jaccard	Aplicable a cadenas con valores binarios	No toma en cuenta cuando ambas cadenas que se comparan no poseen característica alguna en dicha posición
<i>Edit distance</i>	Distancia Levenshtein normalizada	Se limita a operaciones de edición básicas en texto
SMC	Métrica Levenshtein diseñada para valores binarios	Se necesitan cadenas simétricas y acotadas
Rogger & Tanimoto	Variación de la métrica d Jaccard para valores binarios	Castiga con un valor doble a las diferencias entre ambas cadenas

# Capítulo III

## Metodología de desarrollo de la aplicación

El presente capítulo muestra y describe a detalle el orden que se siguió para el desarrollo de la aplicación. Dicho proceso de forma general fue el siguiente:

- Decisión de la metodología de software a seguir.
- Comprender y validar los requerimientos (reglas del negocio) que necesitaba el usuario.
- Construir la documentación necesaria de cada proceso de desarrollo.
- Decidir la arquitectura de software a implementar.
- Organización y eficiencia en el uso de la información en las aplicaciones a desarrollar.
- Decidir las métricas de similitud a implementar.
- Desarrollo de prototipos.
- Validación de los requerimientos en base a prototipos.
- Puestas en marcha de las aplicaciones.

Cabe mencionar que durante el proceso de análisis, el Dr. Villaseñor comentó que la información para clasificar géneros y especies no se tiene completa hasta este momento de manera formal, en capítulo 4 en resultados se detalla esta cuestión esta parte. Derivado de esta cuestión, se planeó la aplicación para poder clasificar en todos los niveles requeridos, y por otra parte, se desarrolló una estructura semi-automática para agilizar la aplicación de consulta vía *Web*.

### 3.1 Proceso Unificado Racional (RUP)

Para el proceso de desarrollo de software se escogió la metodología RUP (*Rational Unified Process*) por las siguientes razones:

- Es una metodología estándar para el análisis, diseño, implementación y documentación en sistemas orientados a objetos.
- Es una metodología RMC (*Rational Method Composer*) la cual es adaptable al contexto y necesidades de cada organización.

Las fases que se implementaron para la nueva aplicación fueron:

- Reglas del Negocio
- Requerimientos
- Análisis y Diseño
- Implementación
- Testeo

### 3.2 Reglas del Negocio


Para entender las políticas, normas y definiciones se procedió a una serie de entrevistas personales en el Colegio de Postgraduados campus Montecillo. Dichas entrevistas fueron realizadas en la oficina del Dr. Lauro López Mata ubicada en el edificio de Botánica. La duración de cada reunión era de entre hora y media a dos horas.

Derivado de las entrevistas se explicó por parte de los expertos en el ramo:

- La importancia de la identificación taxonómica en México.
- La aplicación desarrollada anteriormente (GENCOMEX).
- Las nuevas necesidades emergentes.

### 3.3 Requerimientos

Posteriormente, se procedió a la toma de requerimientos. La información recolectada se organizó y clasificó. La plantilla utilizada para la obtención de información se muestra a continuación, ver Figura 7.



**GESTION DE REQUERIMIENTOS**  
**NOMBRE DEL SISTEMA: TAXON-02**

Nombre del Formato			
Fecha		Numero del Requerimiento	
Nombre del Analista		Clave	
Instrucciones	Registrar nombres, datos, fechas, tiempos, lugares, etc. Precisar la información cuando no sea clara.		
Nombre del Usuario:			
Área o Departamento:			
Correo Electrónico:			

Figura 7 Plantilla para captura de requerimientos.

La figura esta compuesta de los siguientes campos:

- Nombre del formato.- describe su objetivo o funcion.
- Fecha.- fecha cuando se obtuvo dicho requerimiento.
- Número del requerimiento.- el número de la plantilla.
- Nombre del analista.- persona la cual esta aplicando el documento.
- Clave.- clave del analista en turno.
- Instrucciones.- guías o palabras clave a recordar.
- Nombre del usuario.- nombre de la persona entrevistada.
- Area o departamento.- lugar dentro de la organización donde labora.
- Correo electrónico.- contacto de la persona entrevistada para futuras dudas o aclaraciones.
- La zona gris del la plantilla es el área en donde el analista anota la información que importante durante la entrevista.

Posteriormente, se conjuntaron los requerimientos derivados de la entrevista quedando de la siguiente manera, ver Figura 8 y 9.

SIE-RF Sistema de Identificación de Especies	
<ul style="list-style-type: none"> <li>• Registro datos taxonómicos:</li> </ul>	<p><b>División</b> <b>Clase</b> <b>Orden</b> <b>Familia</b> <b>Género</b> <b>Especie</b> <b>Nombre científico</b> <b>Sinonimia (Lista de nombres)</b></p>
<ul style="list-style-type: none"> <li>• Datos morfológicos</li> </ul>	<p><b>Caracteres (Lista con ceros y unos)</b> <b>Raíz</b> <b>Tallo</b> <b>Hoja</b> <b>Inflorescencia</b> <b>Flor</b> <b>Fruto</b> <b>Semilla</b></p>
<ul style="list-style-type: none"> <li>• Ubicación geográfica</li> </ul>	<p><b>Cuadro</b> <b>Latitud</b> <b>Longitud</b> <b>Altitud</b> <b>Estado (del País)</b></p>
<ul style="list-style-type: none"> <li>• Fecha de recolección</li> </ul>	<p><b>Día de recolección</b> <b>Mes de recolección</b> <b>Año de recolección</b> <b>Fenología (estados)</b></p>
<ul style="list-style-type: none"> <li>• Marcar los caracteres de una nueva especie. Agregar 1 cuando tienen la característica y 0 cuando no lo tiene</li> </ul>	
<ul style="list-style-type: none"> <li>• Registrar y asociar imágenes a una especie previamente dada de alta</li> </ul>	
<ul style="list-style-type: none"> <li>• Identificar una especie marcando en una lista los caracteres de la especie. Se deberán filtrar las especies hasta mostrar 0, 1 ó n especies que contienen los caracteres elegidos.</li> </ul>	

**Figura 8 Requerimientos funcionales SIE**



SIE-RNF Sistema de Identificación de Especies	
	<ul style="list-style-type: none"> <li>• Incorporar un módulo de administración de usuarios. Dos súper-usuarios y usuarios del SIE.</li> </ul>
	<ul style="list-style-type: none"> <li>• Registrar un conteo de usuarios que lleven a cabo procesos de identificación de especies.</li> </ul>
	<ul style="list-style-type: none"> <li>• Integrar una función para generar un reporte y gráficas de procesos de identificación de especies.</li> </ul>
	<ul style="list-style-type: none"> <li>• Agregar una función para que los usuarios puedan enviar quejas y sugerencias de mejoras al SIE. Se deberán almacenar los mensajes para emitir un reporte simple.</li> </ul>

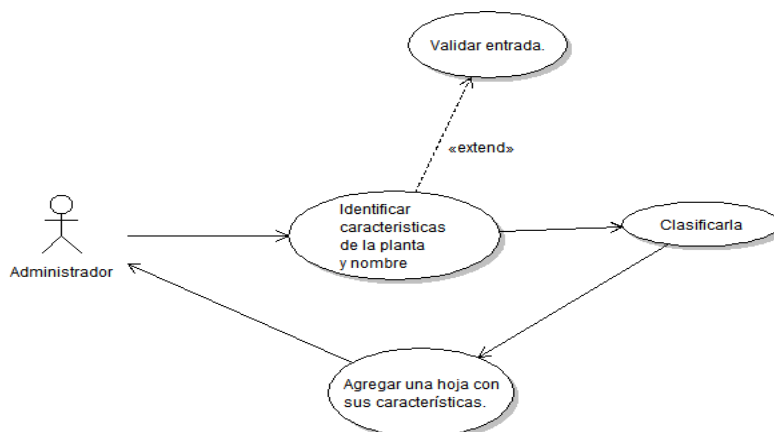
**Figura 9 Requerimientos no funcionales SIE**

### 3.4 Análisis y Diseño

#### 3.4.1 Casos de Uso

A continuación se presentan los casos de uso derivados de los requerimientos primeramente por parte del administrador.

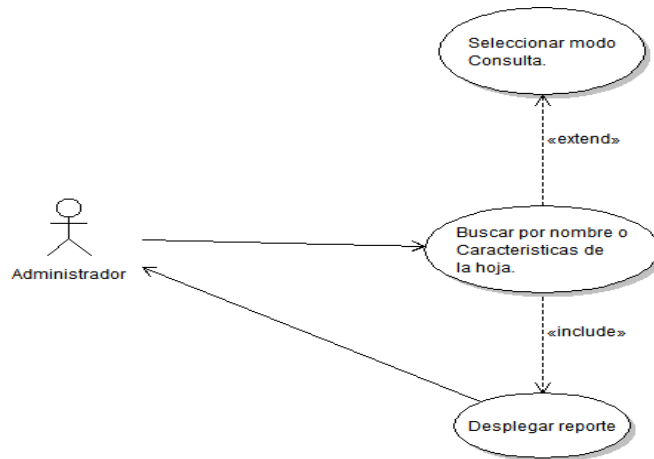
Primero se tiene el caso donde el administrador desea dar de alta una nueva planta, ver Figura 10.



**Figura 10 Administrador, dar de alta planta nueva.**

En este caso, el administrador primero identifica que desea agregar ya sea una clase, familia, género o especie. Posteriormente valida las características que le pertenecen en base a su selección. Acto seguido, verifica las características asignadas a esa planta y guarda el registro. Por último, la aplicación regresa a la pantalla principal para una nueva acción.

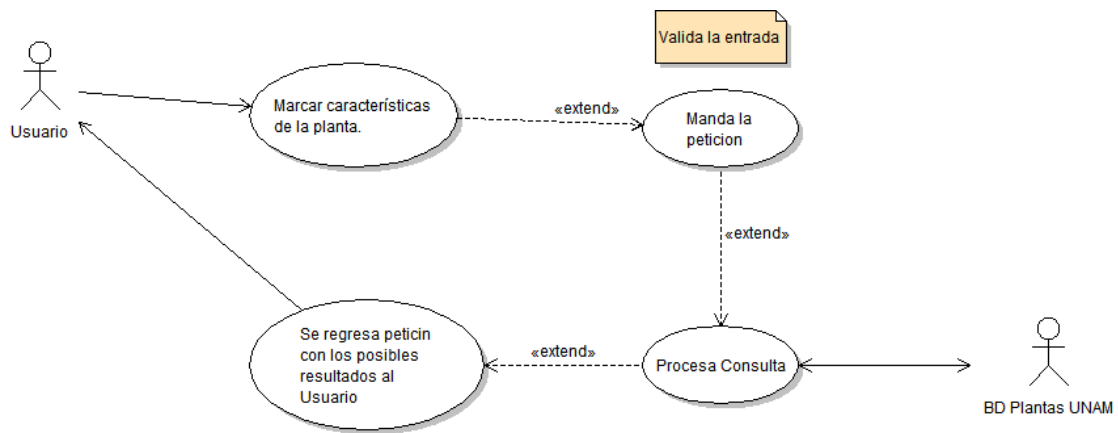
En segundo lugar se tiene el caso cuando el administrador requiere ver la información (consulta) de una planta ya guardada previamente, ver Figura 11.



**Figura 11 Administrador, consulta de una planta.**

En este caso el administrador puede buscar ya sea por nombre o aquellas plantas que posean un conjunto de características. En caso de existir dicho resultado desplegará directamente la planta, su procedencia y que datos están relacionados, de no existir aparecerá un mensaje de que no se encuentra dicha planta con ese nombre. Por la parte de características, el administrador seleccionará las características que desee, posteriormente el sistema despliega los nombres de las plantas y el total de resultados que cumplan las características.

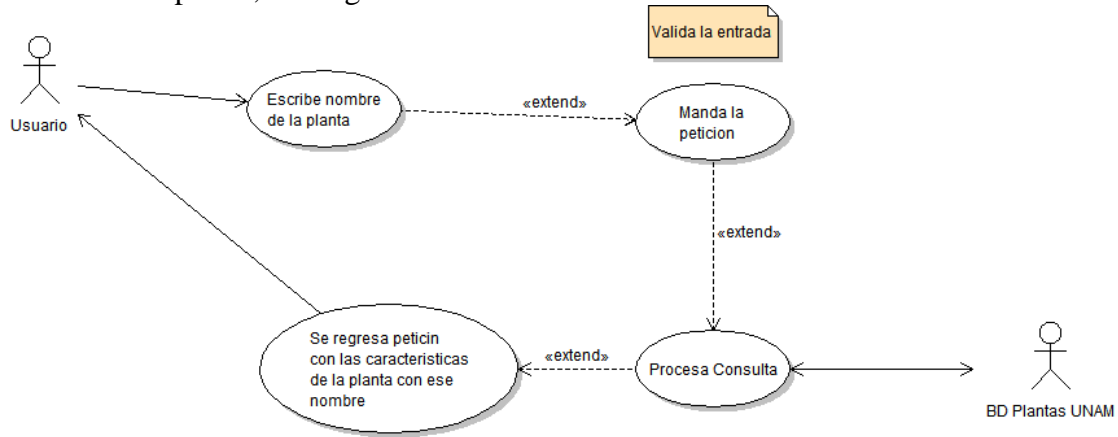
Por otra parte, se tiene el caso cuando un usuario quiere hacer una consulta vía Web por medio de las características de la planta, ver Figura 12.



**Figura 12 Usuario, consulta vía Web por características**

En este caso un usuario vía Web marca las características, la aplicación valida la entrada, se procesa la petición con el servidor y finalmente regresa un listado con los nombres e imágenes de cada planta que cumplan o posean las características marcadas.

Por último, se tiene el caso cuando el usuario requiere la búsqueda directamente por el nombre de la planta, ver Figura 13.

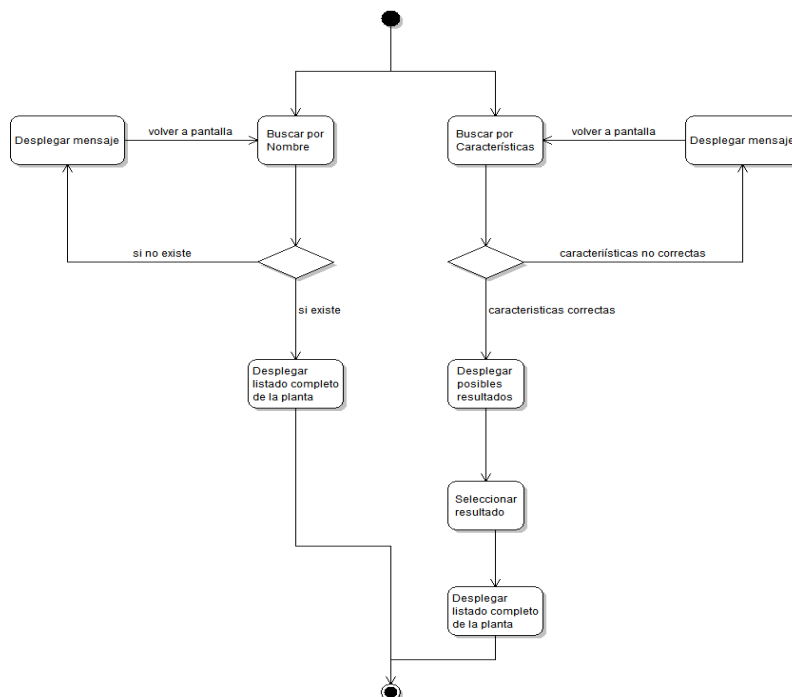


**Figura 13 Usuario, consulta por nombre vía Web.**

En este caso un usuario vía *Web* ingresa directamente el nombre de la planta, la aplicación valida la entrada, se procesa la petición *Web* con el servidor, en caso de existir la planta con ese nombre, la aplicación mostrará toda la información relacionada con esa planta.

### 3.4.2 Diagrama de Actividades

La Figura 14 muestra la secuencia de estados durante una consulta vía *Web*.



**Figura 14 Secuencia de una consulta vía Web.**

En este diagrama se muestra como la aplicación toma una ruta específica en la consulta vía *Web* de cualquier usuario. También, se muestran las actividades que se realizan en

caso de que la información ingresada sea incorrecta, y por ende, las actividades cuando la información sea correcta.

### 3.4.3 Diagrama de Clases

La Figura 15 muestra en UML el diagrama de clases de la aplicación propuesta llamada “Taxón 2”.

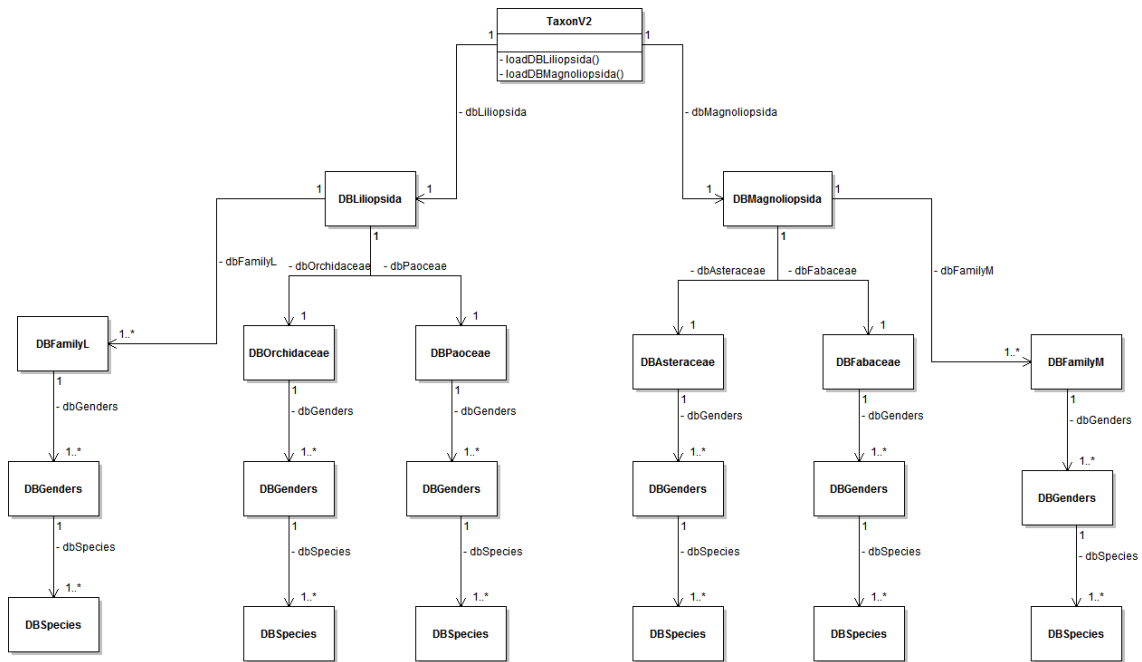


Figura 15 Diagrama de clases de la aplicación Taxón 2.

La estructura muestra que la aplicación posee dos colecciones principales que son la clase Liliopsida y Magnoliopsida. Posteriormente, cada colección se subdivide en otras colecciones cada una. En la clase Liliopsida se tienen a los grupos: Ochidaceae, Poaceae y el resto de las familias de la Liliopsida, de la Magnoliopsida se tienen a los grupos: Asteraceae, Fabaceae y el resto de las familias de la clase Magnoliopsida. Acto seguido, el diagrama muestra que cada familia posee de uno a muchos géneros. Finalmente, cada género puede tener desde una a muchas especies.

La Figura 16 muestra el UML de los objetos por los cuales las colecciones se relacionan.

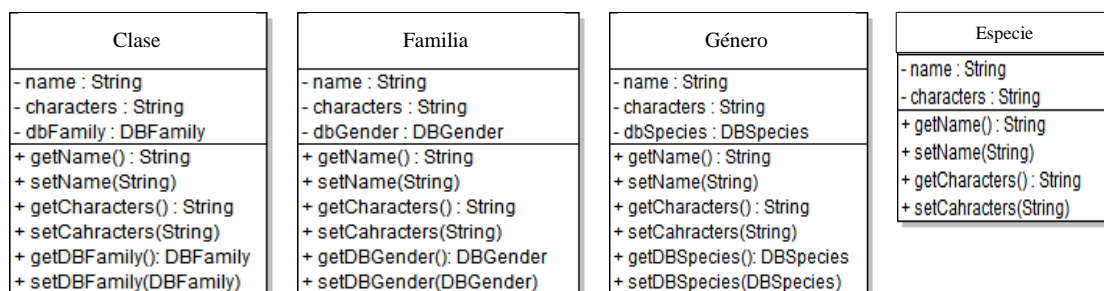


Figura 16 Diagramas de clase de clase, familia, género y especie de la aplicación Taxón 2.

La clase de una planta tiene un nombre, un vector de caracteres y una o varias familias relacionadas con esa clase. La familia de una planta igualmente posee el nombre de la familia, su vector de caracteres y uno o varios géneros al que posee relación. Finalmente, una especie solo tiene nombre y un vector de caracteres que lo describen.

### 3.4.4 Diagrama de Secuencia

La Figura 17 muestra la secuencia de mensajes para la aplicación Taxón 2 ya sea para una consulta, dar de alta o eliminar una planta.

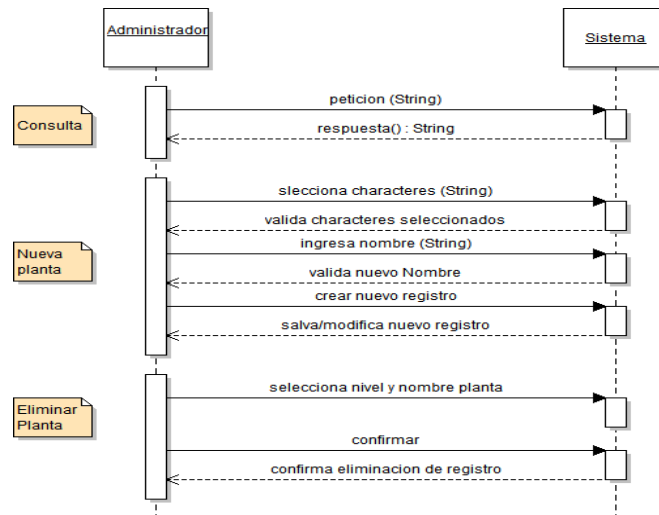


Figura 17 Diagrama de secuencia Taxón 2.

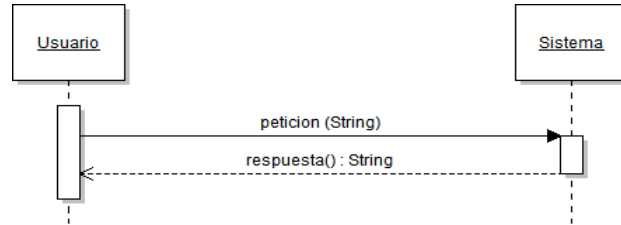
En este diagrama se muestra las tres posibles interacciones que puede tener el administrador de la aplicación.

En primera lugar, se tiene la consulta donde internamente el sistema procesa una cadena ya sea por nombre o un vector de características. Posteriormente, la aplicación procesa la petición y muestra en caso las características de la planta con el nombre ingresado, o en su caso, las posibles plantas que posean las características seleccionadas.

En el caso de una nueva planta, se seleccionan las características que posea la planta, se ingresa el nombre de la planta, si no se cumplen estas condiciones la aplicación despliega un mensaje que se necesitan cumplir ambos requisitos. Posteriormente, una vez validada la entrada de datos se selecciona a que nivel de clasificación se desea guardar ya sea por clase, familia, género o especie. Finalmente, el sistema muestra en pantalla del nuevo registro guardado en caso de no existir esa planta con ese nombre, o modificado en caso de que haya encontrado el nombre de esa planta, solo que ahora se modifica actualizan las nuevas características.

En el tercer caso, de eliminar un registro, se escribe el nombre de la planta a eliminar, acto seguido se selecciona el nivel donde está la planta, finalmente, se confirma la eliminación como medida de seguridad. En caso exitoso, la aplicación desplegará el mensaje de registro borrado exitosamente, en caso de que no exista el nombre de la planta, mostrará que no se eliminó ningún registro.

La Figura 18 muestra el diagrama de secuencia para la consulta de un usuario externo vía *Web*.

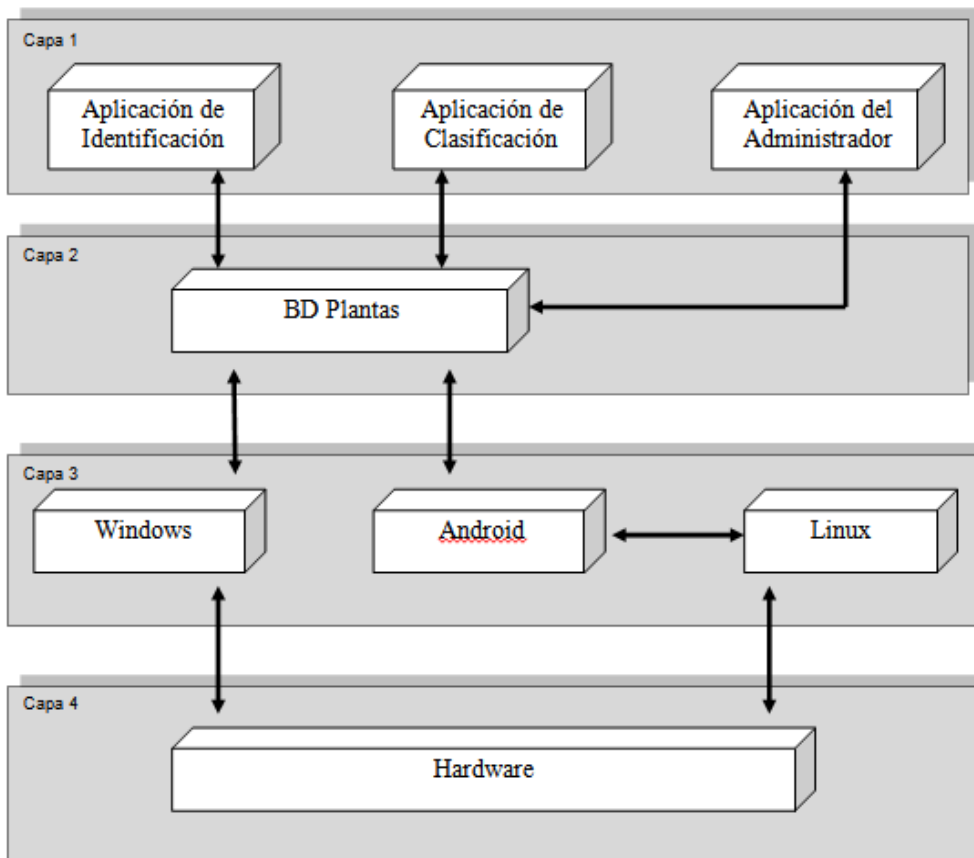


**Figura 18** Secuencia de una consulta vía *Web*.

En este diagrama un usuario vía *Web*, ingresa una petición, internamente la aplicación manda una cadena con la petición, se procesa y regresa una cadena de respuesta. El dispositivo de envío arma el texto mediante el navegador agilizando el proceso de petición/respuesta.

### 3.4.5 Arquitectura del Sistema

En base a los requerimientos obtenidos, se procedió a definir la arquitectura del sistema, ver Figura 19.



**Figura 19** Arquitectura de la aplicación (Elaboración propia).

Esta figura esta compuesta de:

- Aplicación de identificación.- Se ubica en la primera capa que es la superior. Este componente tiene comunicación con las bases de datos de plantas a través de internet.
- Aplicación de clasificación.- Se ubica en la primera capa. Este componente tiene comunicación con las bases de datos de plantas a través de internet al igual que la aplicación de identificación.
- Aplicación del administrador.- Es el último componente de la primera capa. Este módulo tiene comunicación de manera directa con la base de datos de plantas a través de una red interna.
- Base de datos plantas.- Se encuentra ubicada en la segunda capa que es la intermedia de la arquitectura. Esta base de datos tiene comunicación con los componentes aplicación de identificación y clasificación por medio de internet. Con la aplicación del administrador a través de una red interna. Finalmente con el sistema operativo Windows y Android de manera directa.
- Windows.- Ubicado en la tercera capa de la arquitectura. Este tiene comunicación de manera directa con la base de datos plantas y con el componente de *Hardware*.
- Android.- Se encuentra localizado en la tercera capa de la arquitectura. Este tiene comunicación de manera directa con la Base de datos plantas, y con el componente de *Hardware* del dispositivo móvil usándose.
- Linux.- Último componente de la tercera capa. Linux tiene comunicación directa solamente con el Sistema Android y el *Hardware* del dispositivo móvil.
- Hardware.- Esta ubicado en la cuarta capa que es la parte más baja de la arquitectura del Sistema. Este componente solo tiene comunicación de manera directa con los Componentes de Windows y Linux.

### 3.4.6 Descripción del funcionamiento cliente/servidor

Recordando la descripción de la sección de arquitectura del sistema, un usuario (cliente) accede a la página del servidor para identificar alguna planta o sus posibles resultados. El servidor procesa la petición analizando solamente el conjunto de pares ordenados (casilla / valor) marcados por el usuario. Finalmente, la información requerida será una cadena de texto para agilizar el proceso de petición/respuesta por parte del servidor hacia el cliente.

Con el fin de seguir las normas establecidas de calidad de *software*, la página de inicio del servidor estará validada por el W3C para asegurar su correcta construcción. Por otra parte, dado que se tiene previsto que la aplicación tenga también acceso por dispositivos móviles, se propone como formato de respuesta sea en HTML con su respectivo DTD por las siguientes razones:

- DTD es un complemento del HTML
- Es una herramienta independiente de *software* y *hardware* para transportar información.
- DTD es una recomendación del W3C (*World Wide Web Consortium*) que es una comunidad reconocida a nivel mundial dedicada de tiempo completo a desarrollar estándares de calidad en *software*,

El documento DTD resultante estará *bien formado* (sintaxis correcta), también estará *validado*, teniendo un total aproximado de tres MBytes de para desplegar la aplicación web. Ookla Speedtest recientemente realizó un estudio de velocidad de Internet a nivel mundial, este estudio dio como resultado que en México se tiene una velocidad de Internet de descarga de 6.3 Mbps y el doble de subida, en dispositivos móviles de 3.5 Mbps de subida y de descarga cerca del 6.4 Mbps. Dicho lo anterior, una vez teniendo la respuesta lista por el servidor, no tiene que tardar más de un segundo en recibir la respuesta el cliente, el dispositivo procesa la información y se visualiza finalmente.

El servidor se programo en Java siguiendo los estándares de calidad de CMMI y Oracle en codificación, también se genero la documentación de las clases para futuras actualizaciones o modificaciones utilizando la herramienta de *Javadoc* siguiendo el formato que se tiene en el API de *Java Oracle Documentation*.

### 3.4.7 Diseño de la Base de Datos

Derivado de las entrevistas para la obtención de requerimientos, se obtuvo el siguiente árbol jerárquico para la clasificación de una planta como se muestra en la Figura 20.

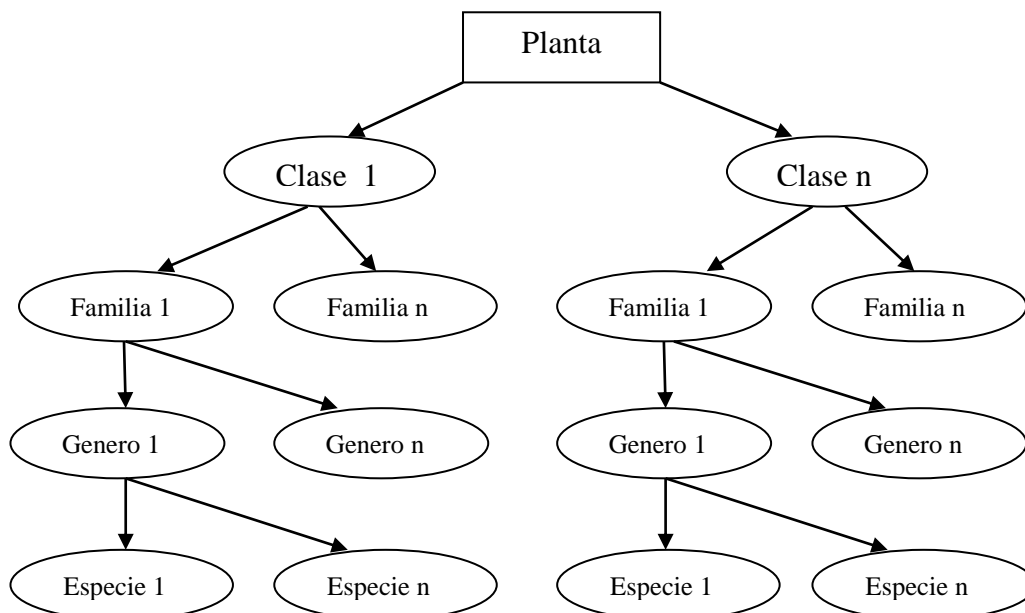


Figura 20 Árbol para la clasificación de una planta.



En esta figura muestra que el primer nivel de clasificación de una planta es por clase, que para el caso de México solo se manejan dos clases que son Liliopsida y Magnoliopsida. En el segundo nivel cada clase posee varias familias. En el tercer nivel cada familia tiene géneros. Y finalmente, un género puede estar conformado de varias especies.

En base a la información recopilada y proporcionada por el Dr. Villaseñor, la Tabla 4 describe las dos clases de plantas existentes en México, así como sus familias y cuantas especies posee de cada una.

**Tabla 4 Plantas en México agrupadas por clase, familia y especie. Primera parte.**

CLASE	FAMILIA	Total Especies
Liliopsida	Alismataceae	22
Liliopsida	Alstroemeriaceae	3
Liliopsida	Amaryllidaceae	114
Liliopsida	Araceae	104
Liliopsida	Asparagaceae	429
Liliopsida	Bromeliaceae	413
Liliopsida	Burmanniaceae	8
Liliopsida	Cannaceae	3
Liliopsida	Commelinaceae	127
Liliopsida	Costaceae	10
Liliopsida	Cyclanthaceae	7
Liliopsida	Cymodoceaceae	2
Liliopsida	Cyperaceae	413
Liliopsida	Dioscoreaceae	75
Liliopsida	Eriocaulaceae	14
Liliopsida	Haemodoraceae	1

**Tabla 4 Plantas en México agrupadas por clase, familia y especie. Segunda parte.**

CLASE	FAMILIA	Total Especies
Liliopsida	Heliconiaceae	20
Liliopsida	Hydrocharitaceae	8
Liliopsida	Hypoxidaceae	11
Liliopsida	Iridaceae	111
Liliopsida	Juncaceae	46
Liliopsida	Juncaginaceae	3
Liliopsida	Liliaceae	24
Liliopsida	Marantaceae	24
Liliopsida	Mayacaceae	1
Liliopsida	Melanthiaceae	36
Liliopsida	Musaceae	1
Liliopsida	<b><i>Orchidaceae</i></b>	1239
Liliopsida	<b><i>Poaceae</i></b>	1037
Liliopsida	Pontederiaceae	13
Liliopsida	Potamogetonaceae	10
Liliopsida	Ruppiaceae	2
Liliopsida	Smilacaceae	20
Liliopsida	Triuridaceae	2
Liliopsida	Typhaceae	4
Liliopsida	Xyridaceae	3
Liliopsida	Zingiberaceae	5
Liliopsida	Zosteraceae	3
Total 38 Familias Liliopsida		

CLASE	FAMILIA	Total Especies
Magnoliopsida	Acanthaceae	377
Magnoliopsida	Actinidiaceae	20
Magnoliopsida	Achariaceae	1
Magnoliopsida	Achatocarpaceae	4
Magnoliopsida	Adoxaceae	26
Magnoliopsida	Aizoaceae	4
Magnoliopsida	Altingiaceae	1
Magnoliopsida	Amaranthaceae	207
Magnoliopsida	Anacampserotaceae	2
Magnoliopsida	Anacardiaceae	67
Magnoliopsida	Annonaceae	53
Magnoliopsida	Apiaceae	206
Magnoliopsida	Apocynaceae	414
Magnoliopsida	Apodanthaceae	4
Magnoliopsida	Aquifoliaceae	20
Magnoliopsida	Araliaceae	35

**Tabla 4 Plantas en México agrupadas por clase, familia y especie. Tercera parte.**

CLASE	FAMILIA	Total Especies
Magnoliopsida	Aristolochiaceae	73
Magnoliopsida	<i>Asteraceae</i>	3018
Magnoliopsida	Balanophoraceae	2
Magnoliopsida	Balsaminaceae	1
Magnoliopsida	Basellaceae	3
Magnoliopsida	Bataceae	1
Magnoliopsida	Begoniaceae	91
Magnoliopsida	Berberidaceae	34
Magnoliopsida	Betulaceae	6
Magnoliopsida	Bignoniaceae	75
Magnoliopsida	Bixaceae	6
Magnoliopsida	Boraginaceae	376
Magnoliopsida	Brassicaceae	206
Magnoliopsida	Brunelliaceae	1
Magnoliopsida	Burseraceae	100
Magnoliopsida	Buxaceae	7
Magnoliopsida	Cabombaceae	3
Magnoliopsida	Cactaceae	669
Magnoliopsida	Calceolariaceae	6
Magnoliopsida	Calophyllaceae	3
Magnoliopsida	Campanulaceae	89
Magnoliopsida	Canellaceae	1
Magnoliopsida	Cannabaceae	11
Magnoliopsida	Capparaceae	31
Magnoliopsida	Caprifoliaceae	51
Magnoliopsida	Caricaceae	9
Magnoliopsida	Caryophyllaceae	116
Magnoliopsida	Celastraceae	100
Magnoliopsida	Ceratophyllaceae	2
Magnoliopsida	Cistaceae	14
Magnoliopsida	Cleomaceae	33
Magnoliopsida	Clethraceae	25
Magnoliopsida	Clusiaceae	18
Magnoliopsida	Combretaceae	14
Magnoliopsida	Connaraceae	9
Magnoliopsida	Convolvulaceae	284
Magnoliopsida	Coriariaceae	1
Magnoliopsida	Cornaceae	5
Magnoliopsida	Crassulaceae	308
Magnoliopsida	Crossosomataceae	5
Magnoliopsida	Cucurbitaceae	150
Magnoliopsida	Cunoniaceae	4

**Tabla 4 Plantas en México agrupadas por clase, familia y especie. Cuarta parte.**

CLASE	FAMILIA	Total Especies
Magnoliopsida	Cyrillaceae	1
Magnoliopsida	Cytinaceae	3
Magnoliopsida	Chloranthaceae	1
Magnoliopsida	Chrysobalanaceae	13
Magnoliopsida	Datisceae	1
Magnoliopsida	Dichapetalaceae	3
Magnoliopsida	Dilleniaceae	7
Magnoliopsida	Dipentodontaceae	2
Magnoliopsida	Droseraceae	2
Magnoliopsida	Ebenaceae	33
Magnoliopsida	Elaeocarpaceae	10
Magnoliopsida	Elatinaceae	4
Magnoliopsida	Ericaceae	95
Magnoliopsida	Erythroxylaceae	10
Magnoliopsida	Euphorbiaceae	731
Magnoliopsida	<b><i>Fabaceae</i></b>	1876
Magnoliopsida	Fagaceae	175
Magnoliopsida	Fouquieriaceae	11
Magnoliopsida	Frankeniaceae	6
Magnoliopsida	Garryaceae	9
Magnoliopsida	Gelsemiaceae	1
Magnoliopsida	Gentianaceae	85
Magnoliopsida	Geraniaceae	45
Magnoliopsida	Gesneriaceae	115
Magnoliopsida	Goodeniaceae	1
Magnoliopsida	Grossulariaceae	23
Magnoliopsida	Guamatelaceae	1
Magnoliopsida	Gunneraceae	3
Magnoliopsida	Haloragaceae	7
Magnoliopsida	Hamamelidaceae	3
Magnoliopsida	Hernandiaceae	8
Magnoliopsida	Hydrangeaceae	32
Magnoliopsida	Hydroleaceae	2
Magnoliopsida	Hypericaceae	27
Magnoliopsida	Icacinaceae	10
Magnoliopsida	Iteaceae	3
Magnoliopsida	Juglandaceae	15
Magnoliopsida	Koerberliniaceae	1
Magnoliopsida	Krameriaceae	9
Magnoliopsida	Lacistemataceae	1
Magnoliopsida	Lamiaceae	578
Magnoliopsida	Lauraceae	140

**Tabla 4 Plantas en México agrupadas por clase, familia y especie. Quinta parte.**

CLASE	FAMILIA	Total Especies
Magnoliopsida	Lecythidaceae	1
Magnoliopsida	Lentibulariaceae	61
Magnoliopsida	Linaceae	25
Magnoliopsida	Linderniaceae	8
Magnoliopsida	Loasaceae	53
Magnoliopsida	Loganiaceae	23
Magnoliopsida	Loranthaceae	54
Magnoliopsida	Lythraceae	111
Magnoliopsida	Magnoliaceae	17
Magnoliopsida	Malpighiaceae	165
Magnoliopsida	Malvaceae	521
Magnoliopsida	Marcgraviaceae	8
Magnoliopsida	Martyniaceae	7
Magnoliopsida	Melastomataceae	202
Magnoliopsida	Meliaceae	27
Magnoliopsida	Menispermaceae	21
Magnoliopsida	Menyanthaceae	3
Magnoliopsida	Mitrastemonaceae	1
Magnoliopsida	Molluginaceae	2
Magnoliopsida	Monimiaceae	8
Magnoliopsida	Montiaceae	13
Magnoliopsida	Moraceae	53
Magnoliopsida	Muntingiaceae	2
Magnoliopsida	Myricaceae	3
Magnoliopsida	Myristicaceae	4
Magnoliopsida	Myrtaceae	118
Magnoliopsida	Nelumbonaceae	1
Magnoliopsida	Nitrariaceae	1
Magnoliopsida	Nyctaginaceae	109
Magnoliopsida	Nymphaeaceae	12
Magnoliopsida	Ochnaceae	13
Magnoliopsida	Olacaceae	3
Magnoliopsida	Oleaceae	45
Magnoliopsida	Onagraceae	171
Magnoliopsida	Opiliaceae	3
Magnoliopsida	Orobanchaceae	169
Magnoliopsida	Oxalidaceae	37
Magnoliopsida	Papaveraceae	46
Magnoliopsida	Passifloraceae	84
Magnoliopsida	Pentaphylacaceae	18
Magnoliopsida	Petenaaceae	1
Magnoliopsida	Phrymaceae	47

**Tabla 4 Plantas en México agrupadas por clase, familia y especie. Sexta parte.**

CLASE	FAMILIA	Total Especies
Magnoliopsida	Phyllanthaceae	45
Magnoliopsida	Phyllonomaceae	1
Magnoliopsida	Phytolaccaceae	12
Magnoliopsida	Picramniaceae	11
Magnoliopsida	Picrodendraceae	1
Magnoliopsida	Piperaceae	242
Magnoliopsida	Plantaginaceae	208
Magnoliopsida	Platanaceae	5
Magnoliopsida	Plocospermataceae	1
Magnoliopsida	Plumbaginaceae	3
Magnoliopsida	Podostemaceae	9
Magnoliopsida	Polemoniaceae	103
Magnoliopsida	Polygalaceae	104
Magnoliopsida	Polygonaceae	156
Magnoliopsida	Portulacaceae	20
Magnoliopsida	Primulaceae	115
Magnoliopsida	Proteaceae	2
Magnoliopsida	Putranjivaceae	3
Magnoliopsida	Ranunculaceae	98
Magnoliopsida	Resedaceae	1
Magnoliopsida	Rhamnaceae	113
Magnoliopsida	Rhizophoraceae	3
Magnoliopsida	Rosaceae	192
Magnoliopsida	Rubiaceae	631
Magnoliopsida	Rutaceae	93
Magnoliopsida	Sabiaceae	12
Magnoliopsida	Salicaceae	79
Magnoliopsida	Santalaceae	97
Magnoliopsida	Sapindaceae	125
Magnoliopsida	Sapotaceae	45
Magnoliopsida	Saururaceae	2
Magnoliopsida	Saxifragaceae	17
Magnoliopsida	Scrophulariaceae	45
Magnoliopsida	Schisandraceae	2
Magnoliopsida	Schlegeliaceae	3
Magnoliopsida	Schoepfiaceae	4
Magnoliopsida	Setchellanthaceae	1
Magnoliopsida	Simaroubaceae	9
Magnoliopsida	Simmondsiaceae	1
Magnoliopsida	Siparunaceae	3
Magnoliopsida	Solanaceae	401
Magnoliopsida	Staphyleaceae	5

**Tabla 4 Plantas en México agrupadas por clase, familia y especie. Séptima parte.**

CLASE	FAMILIA	Total Especies
Magnoliopsida	Stegnospermataceae	3
Magnoliopsida	Styracaceae	14
Magnoliopsida	Surianaceae	4
Magnoliopsida	Symplocaceae	21
Magnoliopsida	Talinaceae	15
Magnoliopsida	Tapisciaceae	1
Magnoliopsida	Theaceae	1
Magnoliopsida	Thymelaeaceae	17
Magnoliopsida	Ticodendraceae	1
Magnoliopsida	Tovariaceae	1
Magnoliopsida	Trigoniaceae	2
Magnoliopsida	Tropaeolaceae	1
Magnoliopsida	Ulmaceae	7
Magnoliopsida	Urticaceae	92
Magnoliopsida	Verbenaceae	161
Magnoliopsida	Violaceae	56
Magnoliopsida	Vitaceae	37
Magnoliopsida	Vochysiaceae	2
Magnoliopsida	Winteraceae	1
Magnoliopsida	Ximeniaceae	3
Magnoliopsida	Zygophyllaceae	31
Total 212 Familias Magnoliopsida		
<b>Cuenta general Especies</b>		<b>21776</b>

El recuadro muestra que el número de familias y las especies de cada familia no son iguales en dimensiones entre ellos.

Debido a lo anterior, la Figura 21 muestra el árbol diseñado para estructurar la información de las plantas.

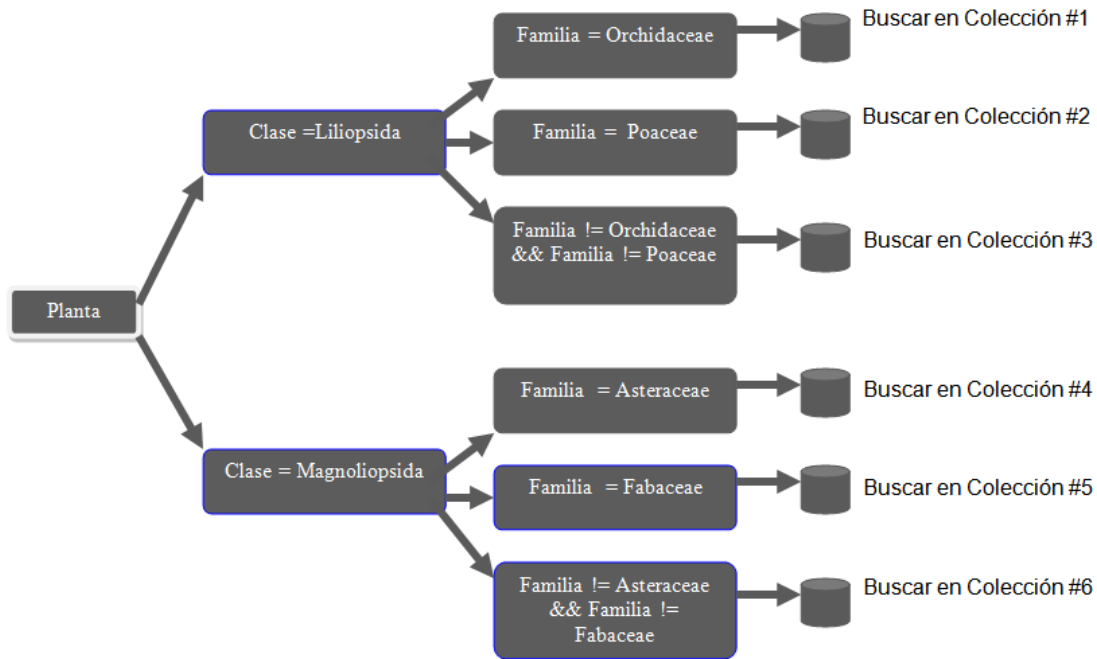


Figura 21 Árbol de decisiones (Elaboración propia).

Dado que la base de datos está fragmentada en seis partes, al consultar una planta se carga la porción de información que se necesita en la RAM, se procesa la consulta, y finalmente, se limpia la memoria utilizada. Este proceso mantiene el mínimo uso de los recursos por parte del servidor.

### 3.4.8 Diseño de la Interfaz

Una planta tiene un conjunto de características que son marcadas por el taxónomo o usuario. La Figura 22 muestra la representación internamente dentro de la aplicación para un nuevo registro.

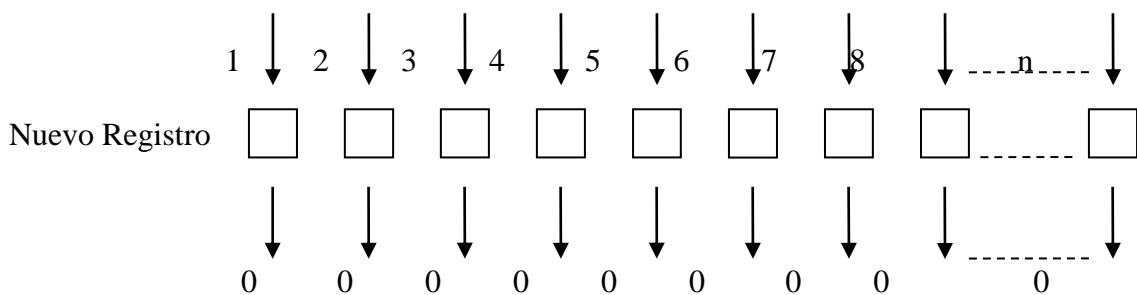


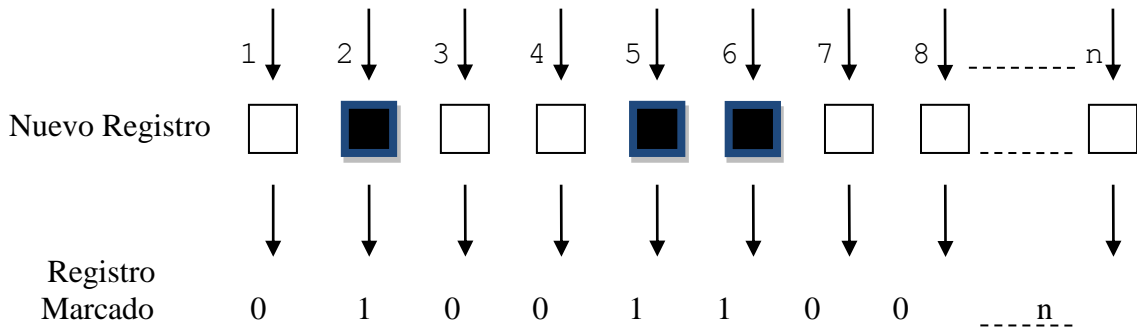
Figura 22 Representación interna de un nuevo registro.

Donde cada casilla en forma de columna representa una característica y al ingresar un nuevo registro todas son 0`s.

Al ingresar las características de una nueva planta en el sistema, internamente la aplicación marcará con 1`s las casillas señaladas por el usuario, dándonos un conjunto de 1`s y 0`s el cual es un vector descriptor de la planta (patrón). Dicho en otras palabras,

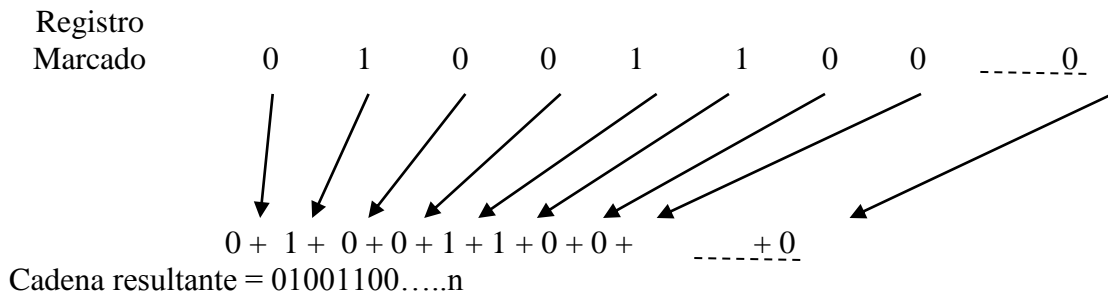


los 1's son las características que posee la planta y los 0's aquellas que no, ver Figura 23.



**Figura 23 Formación del vector descriptivo.**

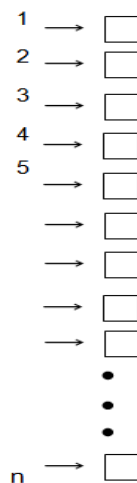
Acto seguido se guardan estas características y se concatenan resultando una cadena representativa de la planta, ver Figura 24.



**Figura 24 Concatenación de características.**

### 3.4.9 Consulta Nueva Planta via Web

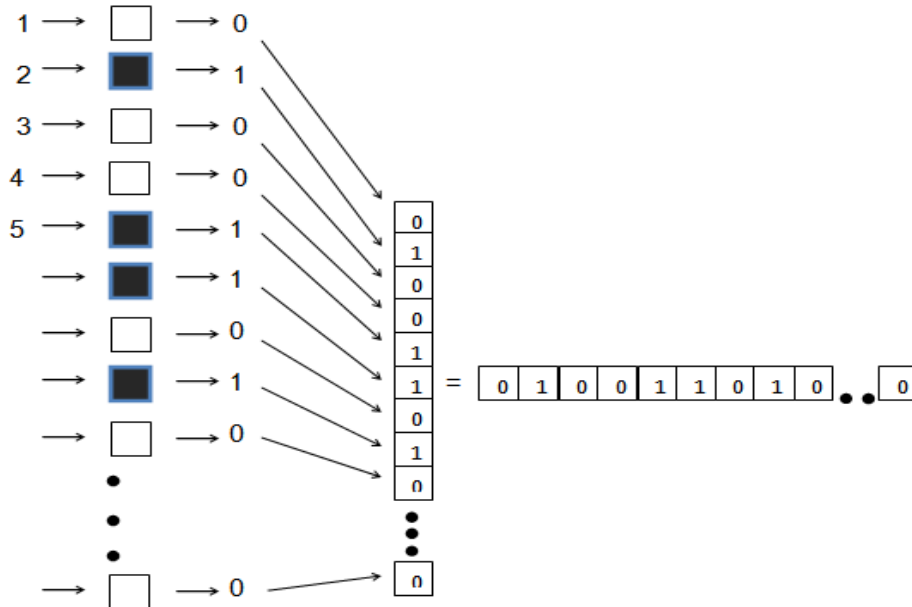
El usuario por medio del navegador accederá a la aplicación del servidor, en respuesta aparecerá en pantalla una página Web donde se mostrará un conjunto de casillas a marcar, ver Figura 25.



**Figura 25 Características en forma de casillas.**

Posteriormente el usuario marcara las casillas que el considere necesarias para identificar la planta que necesite.

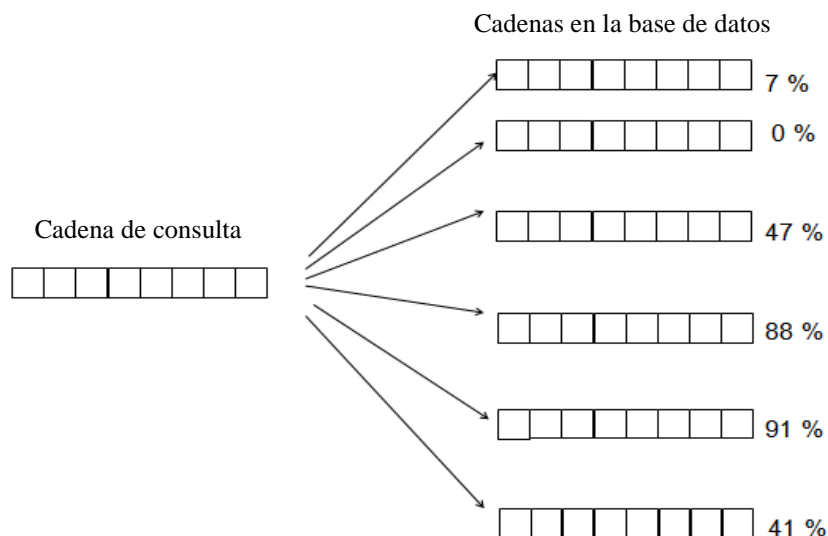
Internamente estas casillas marcadas para su fácil manejo y uso se representarán como 0's y 1's. Cuando se hace la petición de clasificación, todas las casillas no marcadas se convertirán en 0's. Posteriormente este arreglo se transpone de forma matricial y finalmente se concatenan para tener una cadena, ver Figura 26.



**Figura 26 Transposición de las características seleccionadas (Elaboración propia).**

Este proceso tiene varias ventajas que son: mandar la información completa en vez de característica por característica, agilizar el tiempo de petición y respuesta, también al mandar una cadena a través de la red es muy eficiente en vez de mandar grandes cantidades de imágenes o formularios. El servidor recibe una cadena y lo empieza a comparar con los cargados ya en RAM. Y finalmente, el comparar dos cadenas al mismo tiempo es más eficiente que comparar elemento por elemento que en un arreglo.

Sin embargo no todas las cadenas pueden ser idénticas a los que se tiene ya precargados por lo cual, por lo cual se generan posibles resultados, ver Figura 27.



**Figura 27 Resultados de similitud con la Base de Datos.**

La cadena de consulta es un vector descriptivo que lleva la información de las características que posee la planta (patrón) a consultar. Por otra parte, es importante remarcar que ningún vector descriptivo se repite exactamente en toda la base de datos, por lo cual, cada patrón guardado previamente es único. Mediante métricas de similitud con vectores descriptivos binarios, se generaran resultados de comparación entre la cadena de consulta con la base de datos. Los resultados generados son los que se regresaran al usuario que hizo la petición.

### 3.4.10 Selección de las métricas de similitud

Derivado del estado del arte, se seleccionaron las métricas SMC, SimJaccard y Roger&Tanimotto, las cuales están diseñadas específicamente para vectores con valores binarios. Y por otra parte, se tomo como parámetro de referencia la distancia Levenstein para comparar el rendimiento de cada métrica.

La descripción de estos conceptos es la siguiente:

- *Distance*.- es la distancia Levenstein la cual es el mínimo de inserciones, supresiones y substitutiones de caracteres necesarios para transformar una cadena a otro.
- SimSMC.- métrica de similitud *String Matching Comparison*. SMC considera las similitudes 11 y 00 dividida entre las similitudes 11, 00 y las disimilitudes 10 y 01, ver Fórmula 7.

$SMC_{ij} = \frac{a + d}{a + b + c + d}$	(7) SMC <sub>ij</sub>
------------------------------------------	-----------------------

Donde i es la cadena 1 y j es la cadena 2.

- SimJaccard.- métrica de similitud Jaccard, la cual considera solo las similitudes existentes, es decir 11, cuando las dos cadenas contienen la características, dividido entre similitudes 11 y las disimilitudes, pero no incluye similitudes 00, ver Fórmula 8.

$Jaccard(X1, X2) = \frac{a}{a + b + c}$	( 8 ) Jaccard(X1, X2)
-----------------------------------------	-----------------------

Donde X1 es la cadena 1 y X2 es la cadena 2.

- SimRoger&Tanimoto.- métrica de similitud Roger & Tanimoto. Esta métrica divide las similitudes 11 y 00 entre las similitudes 11 y 00, y también, le da más peso al doble de las disimilitudes 10 y 01, ver Fórmula 9.

$Rogger\&Tanimoto = \frac{a + d}{a + d + 2(b + c)}$	( 9 ) Rogger & Tanimoto
-----------------------------------------------------	-------------------------

Las formulas 7, 8 y 9 están diseñadas para medir patrones binarios específicamente, y por otra parte, todas utilizan letras a, b, c, y d, por lo cual se crea la siguiente convención para los datos utilizados, ver Figura 28.

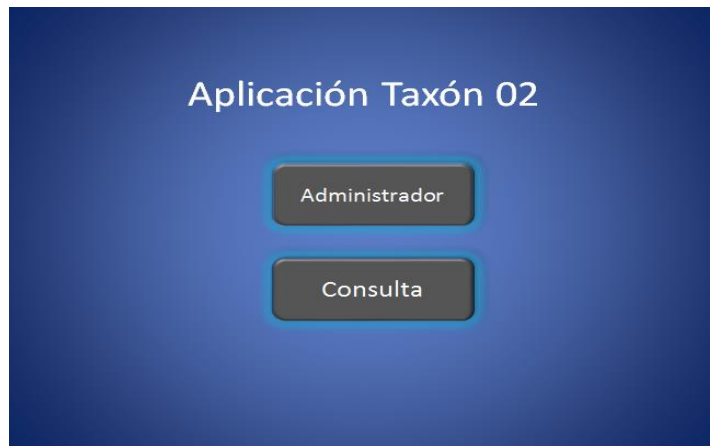
		Cadena 1	
		1	0
Cadena 2	1	11 a	10 b
	0	01 c	00 d

**Figura 28 Convención de datos usada para las métricas de similitud.**

Cuando ambos caracteres de las cadenas a comparar son 1's se asigna una "a", en el caso de que ambos caracteres son 0's se asigna una "d", para el caso cuando se tenga 1 y 0 se asigna la "b", y finalmente, cuando es 0 y 1 se asigna una "c".

### 3.4.11 Propuesta del diseño de la interfaz de usuario: Prototipo de navegación

La Figura 29 muestra la propuesta de pantalla principal para el sistema.



**Figura 29 Prototipo Página Principal Taxón 2.**

Esta figura muestra la ventana con las opciones: una para el administrador de la aplicación, y por otra parte, cuando un usuario normal quiera consultar una hoja.

La Figura 30 muestra el recuadro de autenticación y su palabra clave que el administrador debe ingresar a la aplicación.



**Figura 30 Autenticación Taxón 2.**

La figura la cual está conformada de: título de la ventana, la sección para introducir la contraseña, un botón para ingresar, y finalmente un botón para regresar a la ventana anterior.

La Figura 31 despliega el menú general para dar de alta una nueva planta.

Registrar Nueva Hoja

Datos Taxonómicos

Datos Morfológicos

Fenología

Estado de la República Selección

Ubicación Geográfica Cuadro Latitud Longitud Altitud

Imagen de la Hoja Ruta de Selección

Fecha de Recolección DD/MM/YYYY

Agregar Regresar

**Figura 31 Prototipo para registrar una nueva planta.**

Los componentes de dicha ventana son: título, tres botones para ingresar a las ventanas respectivas de cada opción, un campo tipo lista para seleccionar el estado de la república, cuatro cuadros de texto para ingresar la ubicación geográfica (cuadro, latitud, longitud y altitud), la opción de imagen de la hoja es seleccionar la ruta donde se encuentra la imagen asociada a los datos introducidos, posteriormente un cuadro de texto para ingresar la fecha de recolección con el formato dd/mm/yyyy, y finalmente, en la parte inferior de la ventana se tienen los botones de: agregar el nuevo registro con los datos introducidos o regresar a la ventana anterior.

La Figura 32 muestra los campos ingresar provenientes del menú “Datos Taxonómicos”.

## Registro Datos Taxonómicos

Nombre Científico	Registro Nombre
División	Selección
Clase	Selección
Orden	Selección
Familia	Selección
Genero	Selección
Especie	Selección
Sinonimia	Lista de Sinónimos
Caracteres	Lista de Caracteres

Guardar/Modificar

Regresar

**Figura 32 Datos Taxonómicos.**

Este menú tiene su título en la parte superior, un campo de texto para ingresar el nombre científico correcto de la planta, acto seguido se tienen como lista de selección los campos de división hasta caracteres, un botón para guardar o modificar los datos registrados, y por último, un botón para regresar al menú anterior.

La Figura 33 despliega los datos morfológicos que se deben ingresar de la opción “Datos Morfológicos”.

## Registro Datos Morfológicos

Raíz	Registro Nombre
Tallo	Selección
Hoja	Selección
Inflorescencia	Selección
Flor	Selección
Fruto	Selección
Semilla	Selección

Guardar/Modificar

Regresar

**Figura 33 Datos Morfológicos.**

Esta figura tiene: título de la ventana, un cuadro de texto para ingresar el nombre de la raíz, listas de selección para marcar los datos que posee la planta desde tallo hasta semilla, también posee un botón para guardar o modificar los datos ingresados o marcados, y finalmente un botón para regresar a la ventana anterior.

La Figura 34 despliega los datos propuestos para la opción de Fenología.



The image shows a mobile application interface for 'Fenología'. The title 'Fenología' is centered at the top. Below the title, there are two dropdown menus. The first is labeled 'Mes de Floración' and has a selection box containing 'Selección Mes'. The second is labeled 'Mes de Frutos' and has a selection box containing 'Mes'. Below these are three checkboxes with labels 'Si', 'No', and 'Ambas'. At the bottom of the form, there are two buttons: 'Guardar/Modificar' on the left and 'Regresar' on the right.

**Figura 34 Datos de Fenología.**

Esta figura tiene: título de la ventana, una lista de selección para seleccionar el mes o meses de floración y fruto, posteriormente tiene recuadros de selección para marcar si da flores, si da frutos o ambos casos, posteriormente se tiene el botón para guardar o modificar los datos seleccionados, y finalmente un botón para regresar al menú anterior.

Una vez llenados los campos correctamente, la aplicación despliega la notificación de que la nueva planta fue registrada y guardada correctamente, ver Figura 35.

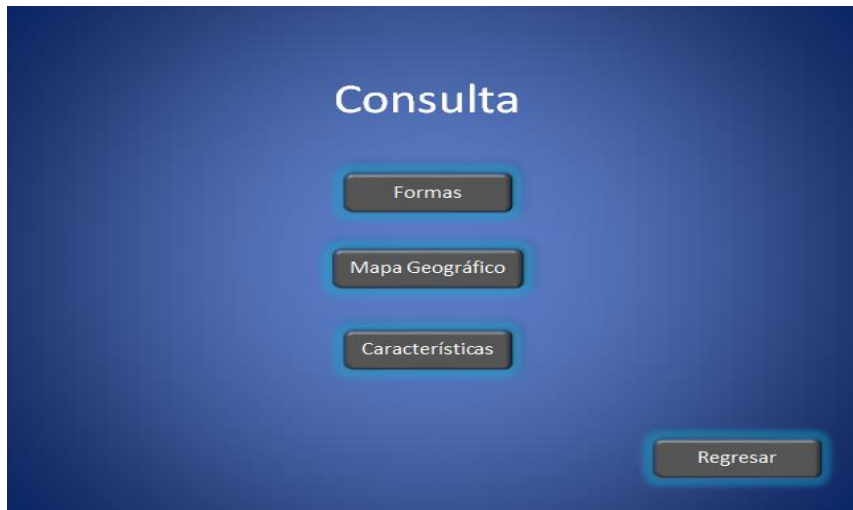


**Figura 35 Mensaje de Registro Guardado.**



Esta figura solo muestra el estatus de que el registro fue guardado correctamente, y por otra parte, el botón para regresar al menú principal.

La Figura 36 muestra los 3 posibles menús para consulta de una planta.



**Figura 36 Muestra de Posibles Menús.**

Cada botón está en ubicado la parte central de la ventana y posee las etiquetas para desplegar el menú respectivo que son en este caso: formas, mapa geográfico y características.

Del menú de consulta la Figura 37 muestra en ejemplo de las posibles formas de hoja de una planta. Esta ventana tiene: título, una imagen que representa la forma de la hoja de la planta, y por otra parte, un botón para regresar al menú anterior de consulta.



**Figura 37 Formas de una Planta.**

También del menú de consulta, la Figura 38 muestra la distribución de las plantas identificadas hasta este momento en la república mexicana.



Figura 38 Distribución de Plantas en México.

En esta ventana se muestra el título del menú de consulta, una imagen con el mapeo de la planta a nivel nacional y botón para regresar al menú de consulta.

Por último, la Figura 39 muestra un menú donde se muestra todas las características de una planta.



Figura 39 Distribución del listado de características.

En esta ventana tiene las listas de selección para marcar las características que identifica en la planta para la consulta, también posee un botón para mostrar la planta que reúna dichas características, y por otra parte, un botón para regresar al menú principal de consulta.

Una vez seleccionadas las características, la Figura 40 muestra la planta(s) que cumplan con dichas características y sus datos respectivos.



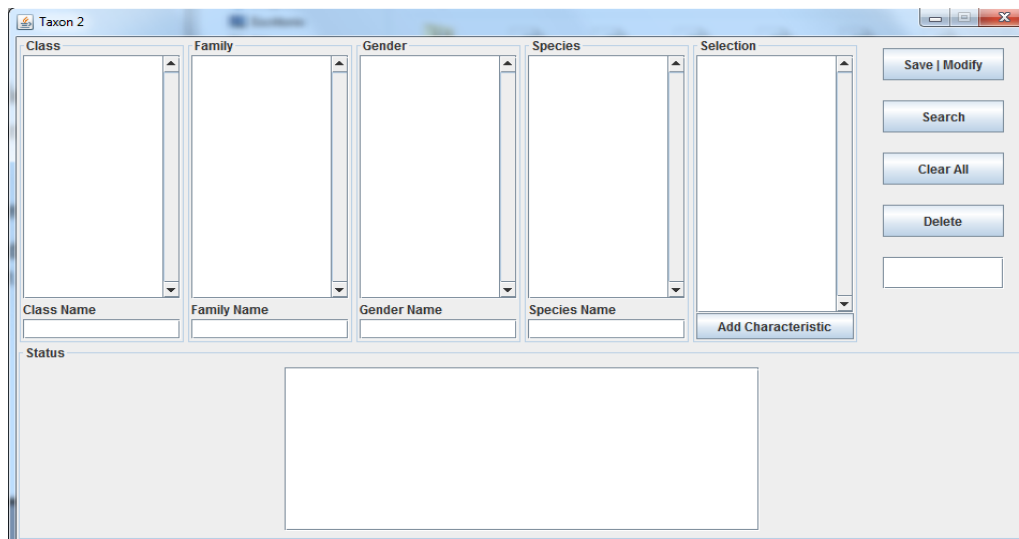
**Figura 40** Despliegue de posibles resultados.

Esta ventana posee: título, en la parte izquierda central una área de texto desplegando las características de la planta, en la parte derecha la imagen asociada a dicha planta, y finalmente, un botón para regresar a hacer una nueva consulta por características.

### **3.5 Implementación**

#### **3.5.1 Aplicación prototipo administrador**

La Figura 41 muestra la interfaz prototipo para el administrador de la aplicación.



**Figura 41 Prototipo aplicación para administrador Taxón 2.**

La aplicación prototipo se encuentra dividido en dos secciones. La parte superior está compuesta de seis paneles dispuestos en forma de columna. Los cuatro pánels de izquierda a derecha están compuestos cada uno de: título del panel, una lista de selección, una etiqueta para identificar el recuadro del nombre, y finalmente, un recuadro de texto.

La función de los primeros cuatro paneles es mostrar el listado de las características definidas para cada clase por medio de claves.

La quinta columna está dispuesta de: una lista que desplegará las características seleccionadas, y también, un botón para poder agregar características a la lista de selección.

La finalidad de la quinta columna es desplegar las características seleccionadas visualmente al usuario. El botón *Add Characteristic* permite añadir más características al listado seleccionado ya sea para añadir/modificar, buscar o eliminar algún registro.

El sexto panel es el de acciones que está compuesto de: un botón para Salvar o Modificar una nuevo registró (planta), un botón de búsqueda, un botón para resetear todos los componentes de la aplicación, un botón para borrar un registro y una caja de texto como medida de seguridad para evitar borrados accidentales.

El botón de *Save/Modify* permite ya sea salvar o modificar un registro en caso de ya existir. Para salvar un nuevo registro se necesita un conjunto de características seleccionadas y un nombre. En caso de que el nombre ya exista, se reemplaza las nuevas características por las anteriores. Para poder crear un nuevo registro es necesario que la base jerárquica al que va depender exista con anterioridad, exceptuando la categoría clase, es decir, para crear o modificar una nueva clase, basta con escribir el nombre y seleccionar las características deseadas y guardar, para el caso de una familia se necesitar escribir el nombre de la clase perteneciente, seleccionar las características del listado de familia e ingresar el nombre de la familia con que se reconocerá, de igual forma para género y especie. Por convención los nombres de las clases, familias y genero solo la primera letra empieza en mayúscula, en el caso de especie el nombre es escrito en minúsculas.

El botón de *Search* permite buscar por nombre directamente, y por otra parte, muestra el listado de plantas que posean las características seleccionadas previamente.

El botón *Clear All* limpia todos los elementos seleccionados de la interfaz, vuelve a su estado original y limpia todas las cajas de texto.

El botón *Delete* funciona en conjunto con la caja de texto ubicada inmediatamente abajo del componente. Su función es eliminar un registro por medio de su nombre directamente perteneciente a un grupo (clase, familia, género o especie), en caso de existir dicho registro. Como medida de seguridad, se debe ingresar la palabra “*DELETE*” en mayúsculas en la caja de texto para eliminar algún registro. Cabe señalar borrar de un registro conlleva la eliminación los registros relacionados a el de manera directa. Por otra parte, para borrar alguna clase, familia o género basta con ingresar el nombre en la caja de texto correspondiente, ya que todos los nombres son únicos en estas categorías. Para borrar una clase se necesita ingresar el género al que pertenece y el nombre de la especie.

La parte inferior de la aplicación está compuesta de un área de texto donde se le informa al usuario de cualquier acción que está realizando de manera correcta o incorrecta. En caso de ser incorrecta se le informa al usuario el error y los lineamientos a seguir para completar la acción correctamente.

Finalmente, en el momento de generar, modificar o eliminar registros, el componente creara y actualizara seis archivos que se utilizaran para la consulta. El funcionamiento de la aplicación es independiente al de consulta.

La aplicación fue desarrollada en Java JDK, diseñada bajo el esquema MVC (Modelo-Vista-Control). Se siguieron los estándares de programación de Icarne y Java Oracle.

### **3.5.2 Aplicación prototipo usuario, consulta vía *Web***

La aplicación para consulta será vía *Web* funciona en base a las seis colecciones generadas por la aplicación del administrador.

La página se divide en cuatro secciones que son: clase, familia, género y especie. Cada sección tiene un recuadro con los iconos, nombre y *checkbox* por característica, también tiene un cuadro de texto en caso de que la búsqueda sea específica por nombre, y finalmente, un botón para mandar la petición de búsqueda. La Figura 42 muestra una parte de la pantalla completa de la página *Web*.



## Identification Plants from México

This site can search a plant by **Class, Family, Gender or Species**.  
The **search** can be done by **name** or by **characteristics selection**.

Class Name

Class Characteristics













<input type="checkbox"/> Hierbas 	<input type="checkbox"/> Arbustos 	<input type="checkbox"/> Arboles 	<input type="button" value="Search Class"/>
<input type="checkbox"/> Eusteles 	<input type="checkbox"/> Atactosteles 	<input type="checkbox"/> Madera producida por el cambium 	
<input type="checkbox"/> Raíz primaria persistente 	<input type="checkbox"/> Raíces adventicias 	<input type="checkbox"/> Nervadura 	
<input type="checkbox"/> Nervadura paralela 	<input type="checkbox"/> Flores tetrámeras o pentámeras 	<input type="checkbox"/> Flores trímeras 	

Figura 42 Pantalla Web para consulta.

Las casillas de selección se pueden activar ya sea marcando su respectivo *checkbox*, o seleccionando el texto de la característica para realizar una consulta. Al seleccionar una casilla, la aplicación marcará la descripción perteneciente con color amarillo dorado como medida visual, si se deselecciona, la casilla volverá a su color normal.

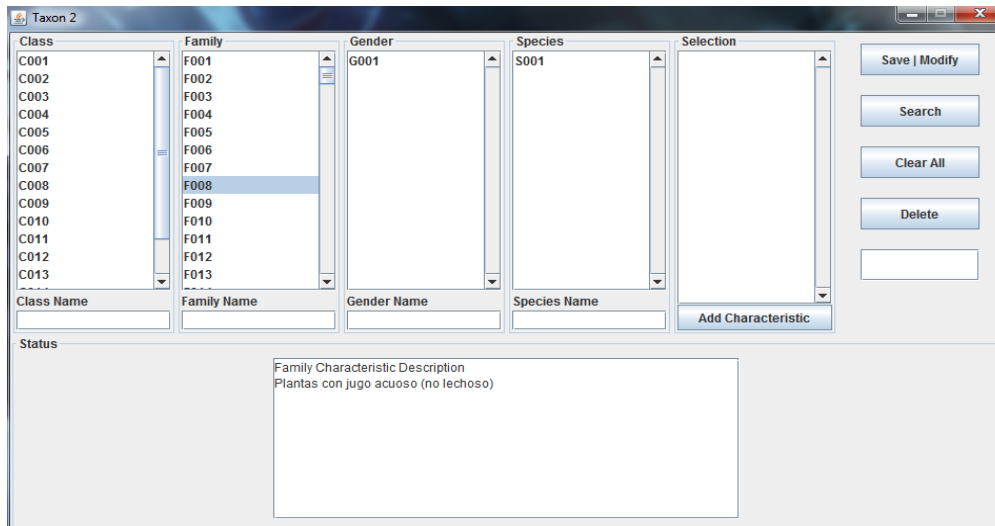
La función del recuadro de texto es buscar un registro específicamente mostrando todo el mapa al que pertenece y las características de cada nivel

La página fue construida utilizando HTML, CSS y JavaScript. Se validó con la herramienta *WDG Html Validator*. Todos los comandos utilizados trabajan en todos los dispositivos móviles. El motor de petición/respuesta es por medio Servlet y las peticiones son por medio de `doPost()`.

### 3.6 Prueba

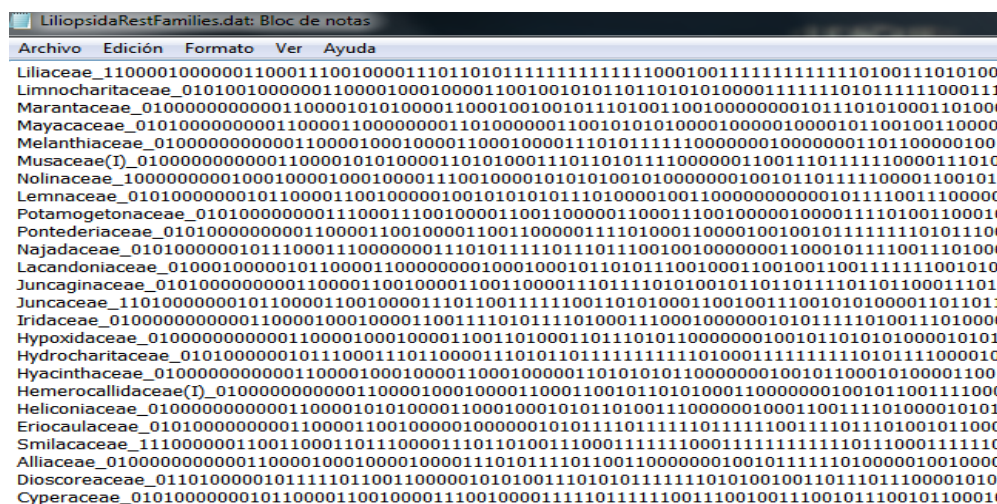
#### 3.6.1 Aplicación prototipo administrador

La Figura 43 muestra como la ventana de estatus muestra la información perteneciente al elemento de la lista seleccionado.



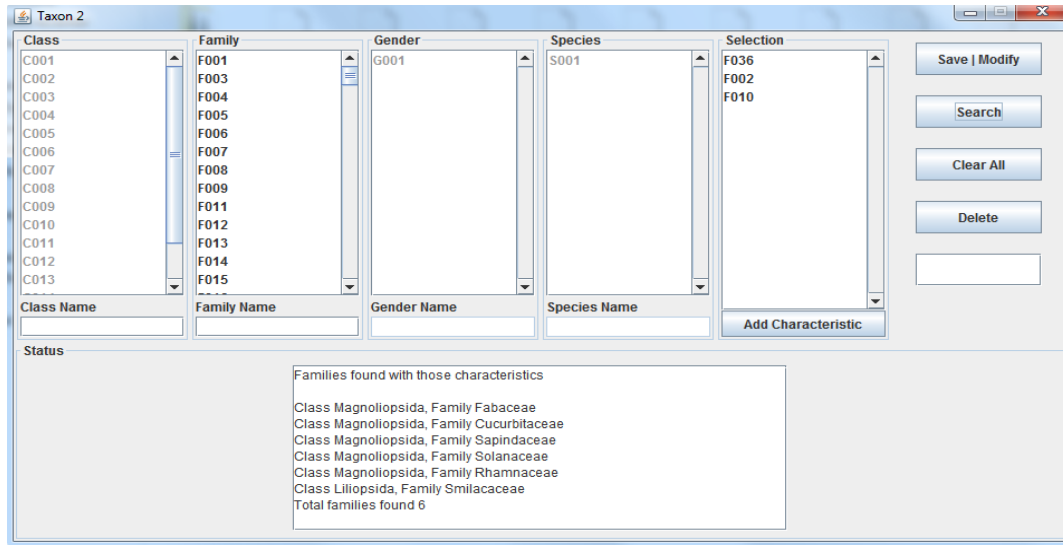
**Figura 43 Demostración MVC Aplicación Taxón 2.**

La Figura 44 muestra parte de la información perteneciente a una de las seis Bases de Datos creadas por la aplicación, y además un conjunto de los vectores descriptivos asociados a este conjunto de familias.



**Figura 44 Familia Liliopsida, conjunto de vectores descriptivos.**

La Figura 45 muestra una consulta por familia en base a un conjunto de características seleccionadas.



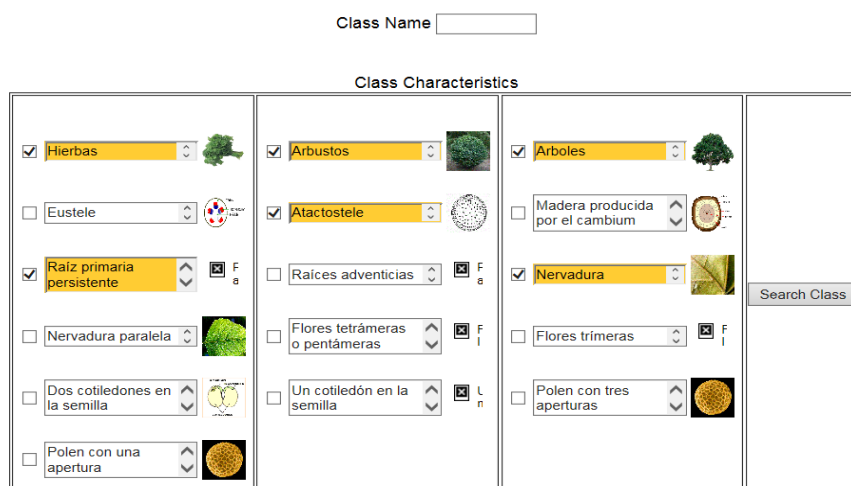
**Figura 45 Ejemplo de una consulta por familia en la aplicación Taxón 2.**

La ventana de estado despliega la consulta realizada con la siguiente información: las familias encontradas con esas características, su clase proveniente de cada una y el número total de resultados.

A medida que se realice la consulta con mayor profundidad, la aplicación mostrará la información de los resultados en base al grupo que se busque.

### 3.6.1 Aplicación prototipo Administrador

La Figura 46 muestra una consulta en la aplicación *Web* por características para usuarios externos.



**Figura 46 Consulta por clase vía Web en base a características.**



Las casillas se marcan ya sea por directa en el *checkBox* o por selección del texto vía *mouse* o *touchPad* en pantalla por dispositivo móvil. El botón de *Search Class* manda la petición al servidor para buscar la clase de planta(s) con las características marcadas. La Figura 47 muestra un listado de posibles resultados por clase.

### Result

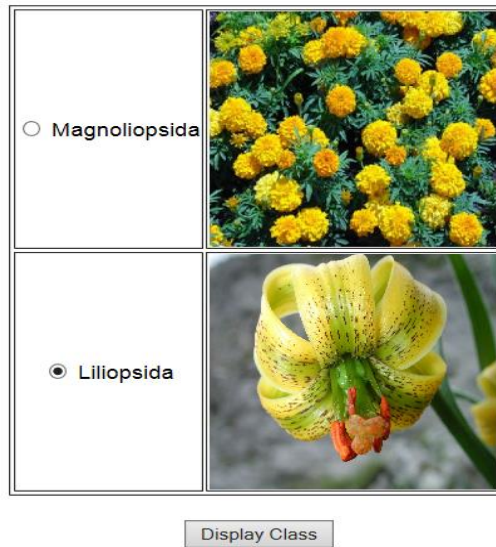


Figura 47 Ejemplo de listado de posibles resultados.

Los resultados de una búsqueda por características se muestran en forma de listado en caso de existir. Cada resultado está conformando por un botón de selección, el nombre a la categoría que le pertenece y una imagen asociada a ese resultado. El botón de *Display Class* despliega la información completa del resultado que se desee mostrar.

La Figura 48 muestra un ejemplo de un resultado específico, ya sea por selección de posible resultado, o también, búsqueda específica en el recuadro de buscar por nombre (*Class Name*).



### Result

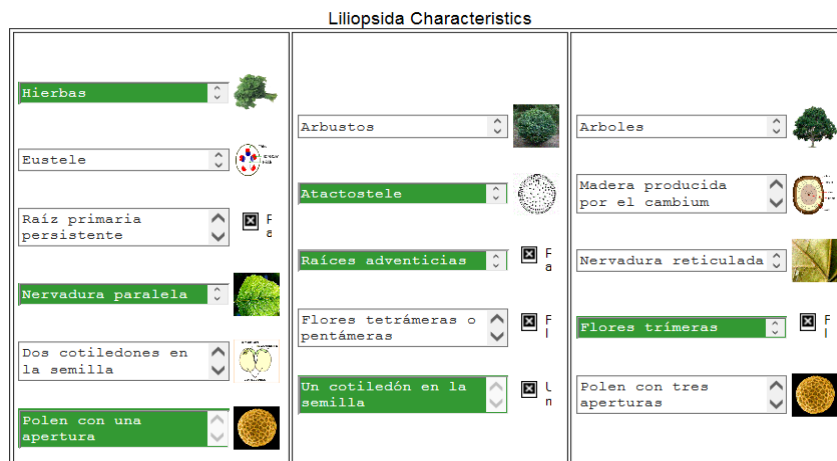


Figura 48 Características de la clase Liliopsida.

















Cada resultado específico está compuesta de: en la parte superior del nombre de la categoría, y en la parte media central, del mapeo completo de las características que la planta posee marcadas en verde.

Las Figuras 49 y 50 muestran el mapeo completo de buscar la familia “Asteraceae”.

**Result**

Class Magnoliopsida

Characteristics

Hierbas 	Arbustos 	Arboles 
Eustele 	Atactostele 	Madera producida por el cambium 
Raíz primaria persistente 	Raíces adventicias 	Nervadura reticulada 
Nervadura paralela 	Flores tetrámeras o pentámeras 	Flores trímeras 
Dos cotiledones en la semilla 	Un cotiledón en la semilla 	Polen con tres aperturas 
Polen con una apertura 		

**Figura 49** Mapeo completo de la clase Magnoliopsida.

Family Asteraceae

Characteristics

Plantas leñosas (árboles o arbustos) <input checked="" type="checkbox"/> F I	Plantas herbáceas (anuales o perennes, <input checked="" type="checkbox"/> F I	Bejucos o plantas escandentes <input checked="" type="checkbox"/> E e
Plantas acuáticas o subacuáticas <input checked="" type="checkbox"/> F I	Plantas epífitas <input checked="" type="checkbox"/> F I	Plantas parásitas o saprófitas <input checked="" type="checkbox"/> F I
Plantas con jugo lechoso (látex) <input checked="" type="checkbox"/> F I	Plantas con jugo acuoso (no lechoso) <input checked="" type="checkbox"/> F I	Plantas aromáticas o resinosas (en corteza, ramas u <input checked="" type="checkbox"/> F I
Plantas con zarcillos <input checked="" type="checkbox"/> F I	Plantas con espinas (en tallos u hojas) <input checked="" type="checkbox"/> F I	Hojas ausentes o reducidas a escamas <input checked="" type="checkbox"/> t c
Estípulas presentes <input checked="" type="checkbox"/> E s	Estípulas ausentes <input checked="" type="checkbox"/> E s	Hojas simples (a veces profundamente partidas, pero no <input checked="" type="checkbox"/> t c
Hojas ternadas o trifoliadas <input checked="" type="checkbox"/> t c	Hojas palmado-compuestas <input checked="" type="checkbox"/> t c	Hojas pinnado-compuestas <input checked="" type="checkbox"/> t c
Hojas opuestas o verticiladas (incluyendo <input checked="" type="checkbox"/> t c	Hojas alternas o basales (incluyendo dísticas) <input checked="" type="checkbox"/> t c	Hojas con la venación invisible o uninervadas <input checked="" type="checkbox"/> t c
Hojas pinnado-nervadas <input checked="" type="checkbox"/> t c	Hojas palmado-nervadas <input checked="" type="checkbox"/> t c	Hojas paralelo-nervadas <input checked="" type="checkbox"/> t c

Figura 50 Fracción del mapeo de la familia Asteraceae.

Para el caso de desplegar un resultado por familia, la página desplegará en primera instancia el nombre de la clase a la que pertenece y las características de dicha clase, acto seguido mostrará en otro cuadro el nombre de la familia buscada en la parte superior, y también el mapeo de las características que posee dicha familia igualmente marcadas en verde.

Cabe mencionar que el Dr. Villaseñor tiene detectadas para clasificar a una clase 16 características, por la parte de familias se consideran 150 características. Actualmente no se tiene la información completa de cuantas y cuáles son las características para clasificar el género y especie de una planta.

# Capítulo V

## Resultados

En este capítulo se muestra los resultados obtenidos de manera gráfica de la distancia Levenshtein aplicado en la base de datos. También se muestran los grados de similitud de las métricas SMC, Jaccard y Roger&Tanimoto en todas las familias en forma de tabla y en forma gráfica. Y finalmente, muestran los resultados de dichas métricas a solo diez familias para un análisis más detallado.

### 4.1 Ventana de Corrida

A continuación se muestra un panorama general de una corrida de comparación de la distancia Levenstein (diagonal principal) en la familia Orchidaceae consigo misma en la siguiente matriz, ver Figura 51.

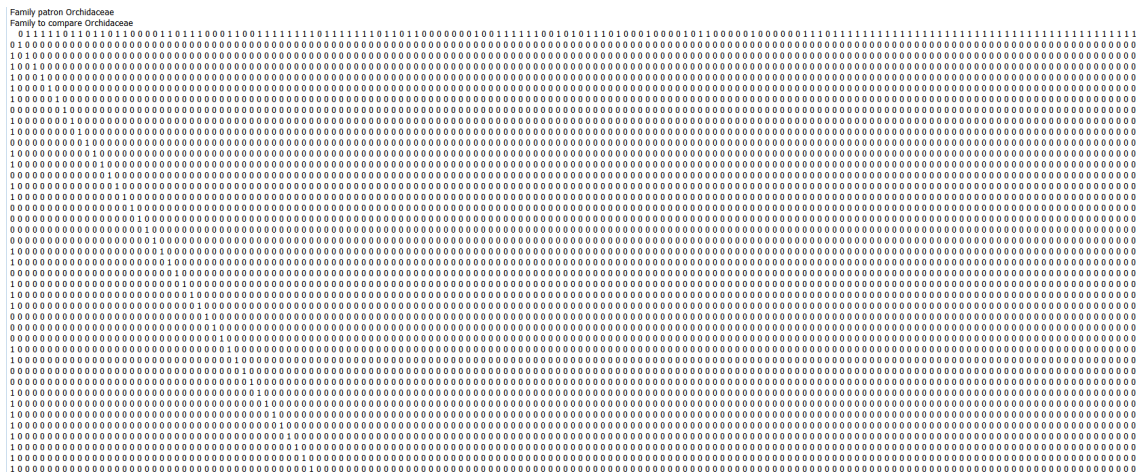


Figura 51 Distancia Levenstein en la familia Orchidaceae.

Esta figura tiene: en el primer renglón el patrón base a comparar, en el segundo renglón la familia contra la que se esta comparando, y finalmente, la matriz parcial de como se forma la distancia levenstein.

### 4.2 Tablas de Resultados

La Tabla 5 muestra los resultados de aplicar las métricas SMC, Jaccard y Roger & Tanimoto a todas las familias de la aplicación Taxón 2.

**Tabla 5 Resultados de métricas en todas las colecciones de plantas. Primera parte.**

Patron	Patron Class	Family	Family Class	Distance	A	B	C	D	SimSMC	SimJaccard	SimRogger & Tanimoto
Orchidaceae	Liliopsida	Elaeocarpaceae	Magnoliopsida	84	40	59	25	26	0,44	0,32258064	0,2820513
Orchidaceae	Liliopsida	Magnoliaceae	Magnoliopsida	73	48	51	22	29	0,51333333	0,3966942	0,34529147
Orchidaceae	Liliopsida	Garryaceae	Magnoliopsida	56	58	41	15	36	0,62666667	0,50877196	0,4563107
Orchidaceae	Liliopsida	Fumariaceae	Magnoliopsida	72	47	52	20	31	0,52	0,394958	0,35135135
Orchidaceae	Liliopsida	Frankeniaceae	Magnoliopsida	85	32	67	18	33	0,43333334	0,2735043	0,27659574
Orchidaceae	Liliopsida	Fouquieriaceae	Magnoliopsida	58	52	47	11	40	0,61333334	0,47272727	0,44230768
Orchidaceae	Liliopsida	Flacourtiaceae	Magnoliopsida	54	69	30	24	27	0,64	0,5609756	0,47058824
Orchidaceae	Liliopsida	Fagaceae	Magnoliopsida	59	64	35	24	27	0,6066667	0,5203252	0,43540668
Orchidaceae	Liliopsida	Fabaceae	Magnoliopsida	51	76	23	28	23	0,66	0,5984252	0,49253732
Orchidaceae	Liliopsida	Euphorbiaceae	Magnoliopsida	56	82	17	39	12	0,62666667	0,5942029	0,4563107
Orchidaceae	Liliopsida	Erythroxylaceae	Magnoliopsida	59	53	46	13	38	0,6066667	0,4732143	0,43540668
Orchidaceae	Liliopsida	Ericaceae	Magnoliopsida	50	72	27	23	28	0,6666667	0,59016395	0,5
Orchidaceae	Liliopsida	Geraniaceae	Magnoliopsida	55	65	34	21	30	0,63333333	0,5416667	0,46341464
Orchidaceae	Liliopsida	Elatinaceae	Magnoliopsida	74	42	57	17	34	0,50666666	0,36206895	0,3392857
Orchidaceae	Liliopsida	Gesneriaceae	Magnoliopsida	47	67	32	15	36	0,68666667	0,5877193	0,52284265
Orchidaceae	Liliopsida	Ebenaceae	Magnoliopsida	66	57	42	24	27	0,56	0,46341464	0,3888889
Orchidaceae	Liliopsida	Droseraceae	Magnoliopsida	83	33	66	17	34	0,44666666	0,28448275	0,28755364
Orchidaceae	Liliopsida	Dipsacaceae(1)	Magnoliopsida	86	30	69	17	34	0,42666668	0,25862068	0,27118644
Orchidaceae	Liliopsida	Dilleniaceae	Magnoliopsida	78	54	45	33	18	0,48	0,4090909	0,31578946
Orchidaceae	Liliopsida	Dichapetalaceae	Magnoliopsida	82	37	62	20	31	0,45333335	0,31092438	0,29310346
Orchidaceae	Liliopsida	Datisceae	Magnoliopsida	89	26	73	16	35	0,40666667	0,22608696	0,25523013
Orchidaceae	Liliopsida	Cyrtillaceae	Magnoliopsida	94	27	72	22	29	0,37333333	0,2231405	0,22950819
Orchidaceae	Liliopsida	Cuscutaceae	Magnoliopsida	44	68	31	13	38	0,70666665	0,60714287	0,5463917
Orchidaceae	Liliopsida	Cunoniaceae	Magnoliopsida	86	40	59	27	24	0,42666668	0,31746033	0,27118644
Orchidaceae	Liliopsida	Cucurbitaceae	Magnoliopsida	41	80	19	22	29	0,7266667	0,661157	0,5706806
Orchidaceae	Liliopsida	Crossosomataceae	Magnoliopsida	88	32	67	21	30	0,41333333	0,26666668	0,26050422
Orchidaceae	Liliopsida	Crassulaceae	Magnoliopsida	53	71	28	25	26	0,64666665	0,57258064	0,47783253
Orchidaceae	Liliopsida	Eremolepidaceae	Magnoliopsida	84	30	69	15	36	0,44	0,2631579	0,2820513
Orchidaceae	Liliopsida	Julianiaceae	Magnoliopsida	84	33	66	18	33	0,44	0,2820513	0,2820513
Orchidaceae	Liliopsida	Lythraceae	Magnoliopsida	48	71	28	20	31	0,68	0,5966387	0,5151515
Orchidaceae	Liliopsida	Loranthaceae	Magnoliopsida	47	74	25	22	29	0,68666667	0,61157024	0,52284265
Orchidaceae	Liliopsida	Loganiaceae	Magnoliopsida	57	60	39	18	33	0,62	0,51282054	0,44927537
Orchidaceae	Liliopsida	Loasaceae	Magnoliopsida	55	69	30	25	26	0,63333333	0,5564516	0,46341464
Orchidaceae	Liliopsida	Linaceae	Magnoliopsida	61	57	42	19	32	0,59333333	0,48305085	0,42180094
Orchidaceae	Liliopsida	Lentibulariaceae	Magnoliopsida	42	72	27	15	36	0,72	0,6315789	0,5625
Orchidaceae	Liliopsida	Lennoaceae	Magnoliopsida	55	55	44	11	40	0,63333333	0,5	0,46341464
Orchidaceae	Liliopsida	Leeaceae	Magnoliopsida	98	1	98	0	51	0,34666666	0,01010101	0,20967741
Orchidaceae	Liliopsida	Lecythidaceae	Magnoliopsida	84	34	65	19	32	0,44	0,2881356	0,2820513
Orchidaceae	Liliopsida	Lauraceae	Magnoliopsida	57	69	30	27	24	0,62	0,54761904	0,44927537
Orchidaceae	Liliopsida	Lamiaceae	Magnoliopsida	50	73	26	24	27	0,6666667	0,5934959	0,5
Orchidaceae	Liliopsida	Gentianaceae	Magnoliopsida	41	76	23	18	33	0,7266667	0,6495727	0,5706806
Orchidaceae	Liliopsida	Krameriaceae	Magnoliopsida	63	54	45	18	33	0,58	0,46153846	0,4084507
Orchidaceae	Liliopsida	Convolvulaceae	Magnoliopsida	50	77	22	28	23	0,6666667	0,6062992	0,5
Orchidaceae	Liliopsida	Juglandaceae	Magnoliopsida	68	55	44	24	27	0,5466667	0,44715446	0,3761468
Orchidaceae	Liliopsida	Illiciaceae	Magnoliopsida	90	29	70	20	31	0,4	0,24369748	0,25
Orchidaceae	Liliopsida	Hydrophyllaceae	Magnoliopsida	45	72	27	18	33	0,7	0,61538464	0,53846157
Orchidaceae	Liliopsida	Hippuridaceae	Magnoliopsida	87	24	75	12	39	0,42	0,21621622	0,2658228
Orchidaceae	Liliopsida	Hippocrateaceae	Magnoliopsida	63	56	43	20	31	0,58	0,47058824	0,4084507
Orchidaceae	Liliopsida	Hippocastanaceae	Magnoliopsida	86	30	69	17	34	0,42666668	0,25862068	0,27118644
Orchidaceae	Liliopsida	Hernandiaceae	Magnoliopsida	73	47	52	21	30	0,51333333	0,39166668	0,34529147

**Tabla 5 Resultados de métricas en todas las colecciones de plantas. Segunda parte.**

Patron	Patron Class	Family	Family Class	Distance	A	B	C	D	SimSMC	SimJaccard	SimRoger & Tanimoto
Orchidaceae	Liliopsida	Hamamelidaceae	Magnoliopsida	74	50	49	25	26	0,50666666	0,4032258	0,3392857
Orchidaceae	Liliopsida	Haloragaceae	Magnoliopsida	70	50	49	21	30	0,53333333	0,41666666	0,36363637
Orchidaceae	Liliopsida	Gunneraceae	Magnoliopsida	79	30	69	10	41	0,47333333	0,27522936	0,31004366
Orchidaceae	Liliopsida	Grossulariaceae	Magnoliopsida	58	65	34	24	27	0,61333334	0,52845526	0,44230768
Orchidaceae	Liliopsida	Goodeniaceae	Magnoliopsida	79	37	62	17	34	0,47333333	0,31896552	0,31004366
Orchidaceae	Liliopsida	Lacistemataceae	Magnoliopsida	79	31	68	11	40	0,47333333	0,28181818	0,31004366
Orchidaceae	Liliopsida	Aristolochiaceae	Magnoliopsida	46	75	24	22	29	0,69333333	0,6198347	0,53061223
Orchidaceae	Liliopsida	Boraginaceae	Magnoliopsida	53	71	28	25	26	0,64666665	0,57258064	0,47783253
Orchidaceae	Liliopsida	Bombacaceae	Magnoliopsida	52	69	30	22	29	0,65333333	0,57024795	0,48514852
Orchidaceae	Liliopsida	Bixaceae	Magnoliopsida	50	64	35	15	36	0,66666667	0,5614035	0,5
Orchidaceae	Liliopsida	Bignoniaceae	Magnoliopsida	40	77	22	18	33	0,73333335	0,6581197	0,57894737
Orchidaceae	Liliopsida	Betulaceae	Magnoliopsida	61	60	39	22	29	0,59333333	0,49586776	0,42180094
Orchidaceae	Liliopsida	Berberidaceae	Magnoliopsida	65	65	34	31	20	0,56666666	0,5	0,39534885
Orchidaceae	Liliopsida	Begoniaceae	Magnoliopsida	44	72	27	17	34	0,70666665	0,62068963	0,5463917
Orchidaceae	Liliopsida	Bataceae	Magnoliopsida	73	41	58	15	36	0,51333333	0,35964912	0,34529147
Orchidaceae	Liliopsida	Basellaceae	Magnoliopsida	60	54	45	15	36	0,6	0,47368422	0,42857143
Orchidaceae	Liliopsida	Balsaminaceae	Magnoliopsida	62	48	51	11	40	0,58666664	0,43636364	0,41509435
Orchidaceae	Liliopsida	Balanophoraceae	Magnoliopsida	80	40	59	21	30	0,46666667	0,33333334	0,3043478
Orchidaceae	Liliopsida	Cornaceae	Magnoliopsida	58	60	39	19	32	0,61333334	0,5084746	0,44230768
Orchidaceae	Liliopsida	Asclepiadaceae	Magnoliopsida	49	71	28	21	30	0,67333335	0,59166664	0,50753766
Orchidaceae	Liliopsida	Buddlejaceae	Magnoliopsida	51	63	36	15	36	0,66	0,55263156	0,49253732
Orchidaceae	Liliopsida	Araliaceae	Magnoliopsida	60	76	23	37	14	0,6	0,5588235	0,42857143
Orchidaceae	Liliopsida	Aquifoliaceae	Magnoliopsida	65	61	38	27	24	0,56666666	0,48412699	0,39534885
Orchidaceae	Liliopsida	Apocynaceae	Magnoliopsida	54	71	28	26	25	0,64	0,568	0,47058824
Orchidaceae	Liliopsida	Apiaceae	Magnoliopsida	45	73	26	19	32	0,7	0,61864406	0,53846157
Orchidaceae	Liliopsida	Annonaceae	Magnoliopsida	61	58	41	20	31	0,59333333	0,48739496	0,42180094
Orchidaceae	Liliopsida	Anacardiaceae	Magnoliopsida	55	74	25	30	21	0,63333333	0,5736434	0,46341464
Orchidaceae	Liliopsida	Amaranthaceae	Magnoliopsida	44	75	24	20	31	0,70666665	0,6302521	0,5463917
Orchidaceae	Liliopsida	Aizoaceae	Magnoliopsida	58	68	31	27	24	0,61333334	0,53968257	0,44230768
Orchidaceae	Liliopsida	Actinidiaceae	Magnoliopsida	62	52	47	15	36	0,58666664	0,45614034	0,41509435
Orchidaceae	Liliopsida	Achatocarpaceae	Magnoliopsida	66	45	54	12	39	0,56	0,4054054	0,3888889
Orchidaceae	Liliopsida	Aceraceae	Magnoliopsida	77	48	51	26	25	0,48666668	0,384	0,3215859
Orchidaceae	Liliopsida	Acanthaceae	Magnoliopsida	38	79	20	18	33	0,74666667	0,6752137	0,59574467
Orchidaceae	Liliopsida	Asteraceae	Magnoliopsida	51	73	26	25	26	0,66	0,58870965	0,49253732
Orchidaceae	Liliopsida	Caricaceae	Magnoliopsida	98	2	97	1	50	0,34666666	0,02	0,20967741
Orchidaceae	Liliopsida	Icacinaceae	Magnoliopsida	87	38	61	26	25	0,42	0,304	0,2658228
Orchidaceae	Liliopsida	Connaraceae	Magnoliopsida	71	47	52	19	32	0,52666664	0,3983051	0,35746607
Orchidaceae	Liliopsida	Combretaceae	Magnoliopsida	67	56	43	24	27	0,55333334	0,45528457	0,3824885
Orchidaceae	Liliopsida	Clusiaceae	Magnoliopsida	50	78	21	29	22	0,66666667	0,609375	0,5
Orchidaceae	Liliopsida	Clethraceae	Magnoliopsida	58	53	46	12	39	0,61333334	0,4774775	0,44230768
Orchidaceae	Liliopsida	Cistaceae	Magnoliopsida	58	64	35	23	28	0,61333334	0,52459013	0,44230768
Orchidaceae	Liliopsida	Chrysobalanaceae	Magnoliopsida	66	56	43	23	28	0,56	0,45901638	0,3888889
Orchidaceae	Liliopsida	Chloranthaceae	Magnoliopsida	71	43	56	15	36	0,52666664	0,37719297	0,35746607
Orchidaceae	Liliopsida	Chenopodiaceae	Magnoliopsida	50	74	25	25	26	0,66666667	0,5967742	0,5
Orchidaceae	Liliopsida	Ceratophyllaceae	Magnoliopsida	75	45	54	21	30	0,5	0,375	0,33333334
Orchidaceae	Liliopsida	Celastraceae	Magnoliopsida	47	75	24	23	28	0,68666667	0,6147541	0,52284265
Orchidaceae	Liliopsida	Cecropiaceae	Magnoliopsida	67	50	49	18	33	0,55333334	0,42735043	0,3824885
Orchidaceae	Liliopsida	Brassicaceae	Magnoliopsida	47	73	26	21	30	0,68666667	0,60833335	0,52284265
Orchidaceae	Liliopsida	Caryophyllaceae	Magnoliopsida	49	74	25	24	27	0,67333335	0,60162604	0,50753766
Orchidaceae	Liliopsida	Brunelliaceae	Magnoliopsida	97	30	69	28	23	0,35333332	0,23622048	0,2145749

**Tabla 5 Resultados de métricas en todas las colecciones de plantas. Tercera parte.**

Patron	Patron Class	Family	Family Class	Distance	A	B	C	D	SimSMC	SimJaccard	SimRoger & Tanimoto
Orchidaceae	Liliopsida	Caprifoliaceae	Magnoliopsida	100	1	98	2	49	0,33333334	0,00990099	0,2
Orchidaceae	Liliopsida	Capparaceae	Magnoliopsida	48	81	18	30	21	0,68	0,627907	0,5151515
Orchidaceae	Liliopsida	Cannabaceae(I)	Magnoliopsida	75	38	61	14	37	0,5	0,33628318	0,33333334
Orchidaceae	Liliopsida	Canellaceae	Magnoliopsida	93	23	76	17	34	0,38	0,19827586	0,2345679
Orchidaceae	Liliopsida	Campanulaceae	Magnoliopsida	42	74	25	17	34	0,72	0,63793105	0,5625
Orchidaceae	Liliopsida	Callitrichaceae	Magnoliopsida	66	49	50	16	35	0,56	0,42608696	0,3888889
Orchidaceae	Liliopsida	Caesalpiniaceae	Magnoliopsida	51	73	26	25	26	0,66	0,58870965	0,49253732
Orchidaceae	Liliopsida	Cactaceae	Magnoliopsida	33	79	20	13	38	0,78	0,70535713	0,6393443
Orchidaceae	Liliopsida	Cabombaceae	Magnoliopsida	90	38	61	29	22	0,4	0,296875	0,25
Orchidaceae	Liliopsida	Buxaceae	Magnoliopsida	82	37	62	20	31	0,45333335	0,31092438	0,29310346
Orchidaceae	Liliopsida	Burseraceae	Magnoliopsida	57	70	29	28	23	0,62	0,5511811	0,44927537
Orchidaceae	Liliopsida	Coriariaceae	Magnoliopsida	82	36	63	19	32	0,45333335	0,30508474	0,29310346
Orchidaceae	Liliopsida	Casuarinaceae(I)	Magnoliopsida	60	50	49	11	40	0,6	0,45454547	0,42857143
Orchidaceae	Liliopsida	Sapindaceae	Magnoliopsida	54	74	25	29	22	0,64	0,578125	0,47058824
Orchidaceae	Liliopsida	Rafflesiaceae	Magnoliopsida	63	53	46	17	34	0,58	0,45689654	0,4084507
Orchidaceae	Liliopsida	Sphenocleaceae(I)	Magnoliopsida	74	33	66	8	43	0,50666666	0,3084112	0,3392857
Orchidaceae	Liliopsida	Solanaceae	Magnoliopsida	42	83	16	26	25	0,72	0,664	0,5625
Orchidaceae	Liliopsida	Simmondsiaceae	Magnoliopsida	88	23	76	12	39	0,41333333	0,2072072	0,26050422
Orchidaceae	Liliopsida	Simaroubaceae	Magnoliopsida	61	66	33	28	23	0,59333333	0,51968503	0,42180094
Orchidaceae	Liliopsida	Setchellanthaceae	Magnoliopsida	80	26	73	7	44	0,46666667	0,24528302	0,3043478
Orchidaceae	Liliopsida	Scrophulariaceae	Magnoliopsida	41	76	23	18	33	0,7266667	0,6495727	0,5706806
Orchidaceae	Liliopsida	Schisandraceae	Magnoliopsida	94	24	75	19	32	0,37333333	0,20338982	0,22950819
Orchidaceae	Liliopsida	Saxifragaceae	Magnoliopsida	61	64	35	26	25	0,59333333	0,512	0,42180094
Orchidaceae	Liliopsida	Sterculiaceae	Magnoliopsida	57	78	21	36	15	0,62	0,57777778	0,44927537
Orchidaceae	Liliopsida	Sapotaceae	Magnoliopsida	60	64	35	25	26	0,6	0,516129	0,42857143
Orchidaceae	Liliopsida	Styracaceae	Magnoliopsida	66	58	41	25	26	0,56	0,46774194	0,3888889
Orchidaceae	Liliopsida	Santalaceae	Magnoliopsida	93	30	69	24	27	0,38	0,24390244	0,2345679
Orchidaceae	Liliopsida	Salicaceae	Magnoliopsida	51	66	33	18	33	0,66	0,5641026	0,49253732
Orchidaceae	Liliopsida	Sabiaceae	Magnoliopsida	67	52	47	20	31	0,55333334	0,4369748	0,3824885
Orchidaceae	Liliopsida	Rutaceae	Magnoliopsida	53	81	18	35	16	0,64666665	0,6044776	0,47783253
Orchidaceae	Liliopsida	Rubiaceae	Magnoliopsida	42	77	22	20	31	0,72	0,64705884	0,5625
Orchidaceae	Liliopsida	Rosaceae	Magnoliopsida	57	76	23	34	17	0,62	0,5714286	0,44927537
Orchidaceae	Liliopsida	Rhizophoraceae	Magnoliopsida	70	50	49	21	30	0,53333336	0,41666666	0,36363637
Orchidaceae	Liliopsida	Rhamnaceae	Magnoliopsida	50	76	23	27	24	0,66666667	0,6031746	0,5
Orchidaceae	Liliopsida	Resedaceae	Magnoliopsida	59	61	38	21	30	0,6066667	0,5083333	0,43540668
Orchidaceae	Liliopsida	Ranunculaceae	Magnoliopsida	53	75	24	29	22	0,64666665	0,5859375	0,47783253
Orchidaceae	Liliopsida	Saururaceae	Magnoliopsida	73	46	53	20	31	0,51333333	0,38655463	0,34529147
Orchidaceae	Liliopsida	Tropaeolaceae	Magnoliopsida	76	39	60	16	35	0,49333334	0,33913043	0,32743362
Orchidaceae	Liliopsida	Zygophyllaceae	Magnoliopsida	55	70	29	26	25	0,63333333	0,56	0,46341464
Orchidaceae	Liliopsida	Winteraceae	Magnoliopsida	81	39	60	21	30	0,46	0,325	0,2987013
Orchidaceae	Liliopsida	Vochysiaceae	Magnoliopsida	76	42	57	19	32	0,49333334	0,3559322	0,32743362
Orchidaceae	Liliopsida	Vitaceae	Magnoliopsida	58	69	30	28	23	0,61333334	0,54330707	0,44230768
Orchidaceae	Liliopsida	Viscaceae	Magnoliopsida	44	71	28	16	35	0,70666665	0,6173913	0,5463917
Orchidaceae	Liliopsida	Violaceae	Magnoliopsida	46	73	26	20	31	0,69333333	0,6134454	0,53061223
Orchidaceae	Liliopsida	Verbenaceae	Magnoliopsida	52	74	25	27	24	0,65333333	0,5873016	0,48514852
Orchidaceae	Liliopsida	Valerianaceae	Magnoliopsida	47	70	29	18	33	0,68666667	0,5982906	0,52284265
Orchidaceae	Liliopsida	Urticaceae	Magnoliopsida	51	71	28	23	28	0,66	0,58196723	0,49253732
Orchidaceae	Liliopsida	Staphyleaceae	Magnoliopsida	76	43	56	20	31	0,49333334	0,36134455	0,32743362
Orchidaceae	Liliopsida	Turneraceae	Magnoliopsida	53	62	37	16	35	0,64666665	0,53913045	0,47783253
Orchidaceae	Liliopsida	Surianaceae	Magnoliopsida	90	28	71	19	32	0,4	0,23728813	0,25

**Tabla 5 Resultados de métricas en todas las colecciones de plantas. Cuarta parte.**

Patron	Patron Class	Family	Family Class	Distance	A	B	C	D	SimSMC	SimJaccard	SimRoger & Tanimoto
Orchidaceae	Liliopsida	Trigoniaceae	Magnoliopsida	86	28	71	15	36	0,42666668	0,24561404	0,27118644
Orchidaceae	Liliopsida	Tovariaceae	Magnoliopsida	80	30	69	11	40	0,46666667	0,27272728	0,3043478
Orchidaceae	Liliopsida	Tiliaceae	Magnoliopsida	50	71	28	22	29	0,66666667	0,58677685	0,5
Orchidaceae	Liliopsida	Ticodendraceae	Magnoliopsida	84	26	73	11	40	0,44	0,23636363	0,2820513
Orchidaceae	Liliopsida	Thymelaeaceae	Magnoliopsida	72	59	40	32	19	0,52	0,45038167	0,35135135
Orchidaceae	Liliopsida	Theophrastaceae	Magnoliopsida	70	49	50	20	31	0,53333336	0,4117647	0,36363637
Orchidaceae	Liliopsida	Theaceae	Magnoliopsida	62	62	37	25	26	0,58666664	0,5	0,41509435
Orchidaceae	Liliopsida	Hydrangeaceae	Magnoliopsida	56	64	35	21	30	0,62666667	0,53333336	0,4563107
Orchidaceae	Liliopsida	Symplocaceae	Magnoliopsida	71	50	49	22	29	0,52666664	0,41322315	0,35746607
Orchidaceae	Liliopsida	Malpighiaceae	Magnoliopsida	51	74	25	26	25	0,66	0,592	0,49253732
Orchidaceae	Liliopsida	Ulmaceae	Magnoliopsida	56	67	32	24	27	0,62666667	0,54471546	0,4563107
Orchidaceae	Liliopsida	Mitrastemonaceae	Magnoliopsida	82	27	72	10	41	0,45333335	0,24770643	0,29310346
Orchidaceae	Liliopsida	Nyssaceae	Magnoliopsida	81	39	60	21	30	0,46	0,325	0,2987013
Orchidaceae	Liliopsida	Nymphaeaceae	Magnoliopsida	52	64	35	17	34	0,65333333	0,55172414	0,48514852
Orchidaceae	Liliopsida	Nyctaginaceae	Magnoliopsida	51	71	28	23	28	0,66	0,58196723	0,49253732
Orchidaceae	Liliopsida	Nelumbonaceae	Magnoliopsida	86	29	70	16	35	0,42666668	0,2521739	0,27118644
Orchidaceae	Liliopsida	Myrtaceae	Magnoliopsida	46	75	24	22	29	0,69333333	0,6198347	0,53061223
Orchidaceae	Liliopsida	Myrsinaceae	Magnoliopsida	56	68	31	25	26	0,62666667	0,5483871	0,4563107
Orchidaceae	Liliopsida	Myristicaceae	Magnoliopsida	87	32	67	20	31	0,42	0,26890758	0,2658228
Orchidaceae	Liliopsida	Myricaceae	Magnoliopsida	76	45	54	22	29	0,49333334	0,37190083	0,32743362
Orchidaceae	Liliopsida	Moringaceae	Magnoliopsida	70	40	59	11	40	0,53333336	0,36363637	0,36363637
Orchidaceae	Liliopsida	Ochnaceae	Magnoliopsida	78	49	50	28	23	0,48	0,38582677	0,31578946
Orchidaceae	Liliopsida	Molluginaceae	Magnoliopsida	65	59	40	25	26	0,56666666	0,47580644	0,39534885
Orchidaceae	Liliopsida	Moraceae	Magnoliopsida	58	71	28	30	21	0,61333334	0,5503876	0,44230768
Orchidaceae	Liliopsida	Mimosaceae	Magnoliopsida	48	72	27	21	30	0,68	0,6	0,5151515
Orchidaceae	Liliopsida	Menyanthaceae	Magnoliopsida	56	52	47	9	42	0,62666667	0,4814815	0,4563107
Orchidaceae	Liliopsida	Menispermaceae	Magnoliopsida	57	70	29	28	23	0,62	0,5511811	0,44927537
Orchidaceae	Liliopsida	Meliaceae	Magnoliopsida	62	71	28	34	17	0,58666664	0,5338346	0,41509435
Orchidaceae	Liliopsida	Melastomataceae	Magnoliopsida	60	67	32	28	23	0,6	0,52755904	0,42857143
Orchidaceae	Liliopsida	Marcgraviaceae	Magnoliopsida	77	41	58	19	32	0,48666668	0,34745762	0,3215859
Orchidaceae	Liliopsida	Malvaceae	Magnoliopsida	53	76	23	30	21	0,64666665	0,58914727	0,47783253
Orchidaceae	Liliopsida	Pyrolaceae	Magnoliopsida	55	54	45	10	41	0,63333333	0,49541286	0,46341464
Orchidaceae	Liliopsida	Tamaricaceae(l)	Magnoliopsida	66	49	50	16	35	0,56	0,42608696	0,3888889
Orchidaceae	Liliopsida	Monotropaceae	Magnoliopsida	52	58	41	11	40	0,65333333	0,5272727	0,48514852
Orchidaceae	Liliopsida	Platanaceae	Magnoliopsida	86	42	57	29	22	0,42666668	0,328125	0,27118644
Orchidaceae	Liliopsida	Punicaceae(l)	Magnoliopsida	67	43	56	11	40	0,55333334	0,3909091	0,3824885
Orchidaceae	Liliopsida	Monimiaceae	Magnoliopsida	89	32	67	22	29	0,40666667	0,2644628	0,25523013
Orchidaceae	Liliopsida	Olacaceae	Magnoliopsida	62	63	36	26	25	0,58666664	0,504	0,41509435
Orchidaceae	Liliopsida	Primulaceae	Magnoliopsida	52	69	30	22	29	0,65333333	0,57024795	0,48514852
Orchidaceae	Liliopsida	Portulacaceae	Magnoliopsida	53	74	25	28	23	0,64666665	0,5826772	0,47783253
Orchidaceae	Liliopsida	Polygonaceae	Magnoliopsida	54	72	27	27	24	0,64	0,5714286	0,47058824
Orchidaceae	Liliopsida	Polygalaceae	Magnoliopsida	53	73	26	27	24	0,64666665	0,5793651	0,47783253
Orchidaceae	Liliopsida	Polemoniaceae	Magnoliopsida	49	69	30	19	32	0,67333335	0,58474576	0,50753766
Orchidaceae	Liliopsida	Plumbaginaceae	Magnoliopsida	49	64	35	14	37	0,67333335	0,5663717	0,50753766
Orchidaceae	Liliopsida	Proteaceae	Magnoliopsida	77	46	53	24	27	0,48666668	0,37398374	0,3215859
Orchidaceae	Liliopsida	Plantaginaceae	Magnoliopsida	47	70	29	18	33	0,68666667	0,5982906	0,52284265
Orchidaceae	Liliopsida	Pittosporaceae	Magnoliopsida	98	1	98	0	51	0,34666666	0,01010101	0,20967741
Orchidaceae	Liliopsida	Orobanchaceae	Magnoliopsida	43	67	32	11	40	0,71333333	0,6090909	0,55440414
Orchidaceae	Liliopsida	Oleaceae	Magnoliopsida	53	68	31	22	29	0,64666665	0,56198347	0,47783253
Orchidaceae	Liliopsida	Onagraceae	Magnoliopsida	43	76	23	20	31	0,71333333	0,6386555	0,55440414



**Tabla 5 Resultados de métricas en todas las colecciones de plantas. Quinta parte.**

Patron	Patron Class	Family	Family Class	Distance	A	B	C	D	SimSMC	Sim.Jaccard	SimRogger & Tanimoto
Orchidaceae	Liliopsida	Podostemaceae	Magnoliopsida	51	63	36	15	36	0,66	0,55263156	0,49253732
Orchidaceae	Liliopsida	Opiliaceae	Magnoliopsida	63	52	47	16	35	0,58	0,45217392	0,4084507
Orchidaceae	Liliopsida	Oxalidaceae	Magnoliopsida	43	75	24	19	32	0,71333333	0,63559324	0,55440414
Orchidaceae	Liliopsida	Papaveraceae	Magnoliopsida	55	71	28	27	24	0,63333333	0,56349206	0,46341464
Orchidaceae	Liliopsida	Piperaceae	Magnoliopsida	50	71	28	22	29	0,66666667	0,58677685	0,5
Orchidaceae	Liliopsida	Passifloraceae	Magnoliopsida	45	76	23	22	29	0,7	0,6280992	0,53846157
Orchidaceae	Liliopsida	Pedaliaceae	Magnoliopsida	48	72	27	21	30	0,68	0,6	0,5151515
Orchidaceae	Liliopsida	Phytolaccaceae	Magnoliopsida	54	75	24	30	21	0,64	0,5813953	0,47058824
Orchidaceae	Liliopsida	Liliaceae	Liliopsida	68	53	46	22	29	0,54666667	0,43801653	0,3761468
Orchidaceae	Liliopsida	Limnocaritaceae	Liliopsida	78	39	60	18	33	0,48	0,33333334	0,31578946
Orchidaceae	Liliopsida	Marantaceae	Liliopsida	50	58	41	9	42	0,66666667	0,537037	0,5
Orchidaceae	Liliopsida	Mayacaceae	Liliopsida	76	30	69	7	44	0,49333334	0,28301886	0,32743362
Orchidaceae	Liliopsida	Melanthiaceae	Liliopsida	42	62	37	5	46	0,72	0,59615386	0,5625
Orchidaceae	Liliopsida	Musaceae(I)	Liliopsida	60	49	50	10	41	0,6	0,44954127	0,42857143
Orchidaceae	Liliopsida	Nolinaceae	Liliopsida	53	59	40	13	38	0,64666665	0,52678573	0,47783253
Orchidaceae	Liliopsida	Orchidaceae	Liliopsida	0	99	0	0	51	1	1	1
Orchidaceae	Liliopsida	Lemnaceae	Liliopsida	48	61	38	10	41	0,68	0,559633	0,5151515
Orchidaceae	Liliopsida	Poaceae	Liliopsida	43	73	26	17	34	0,71333333	0,62931037	0,55440414
Orchidaceae	Liliopsida	Potamogetonaceae	Liliopsida	55	61	38	17	34	0,63333333	0,5258621	0,46341464
Orchidaceae	Liliopsida	Pontederiaceae	Liliopsida	46	68	31	15	36	0,69333333	0,5964912	0,53061223
Orchidaceae	Liliopsida	Najadaceae	Liliopsida	53	63	36	17	34	0,64666665	0,54310346	0,47783253
Orchidaceae	Liliopsida	Lacandoniaceae	Liliopsida	87	28	71	16	35	0,42	0,24347825	0,2658228
Orchidaceae	Liliopsida	Juncaginaceae	Liliopsida	69	52	47	22	29	0,54	0,42975205	0,369863
Orchidaceae	Liliopsida	Juncaceae	Liliopsida	42	70	29	13	38	0,72	0,625	0,5625
Orchidaceae	Liliopsida	Iridaceae	Liliopsida	36	72	27	9	42	0,76	0,66666667	0,61290324
Orchidaceae	Liliopsida	Hypoxidaceae	Liliopsida	45	61	38	7	44	0,7	0,5754717	0,53846157
Orchidaceae	Liliopsida	Hydrocharitaceae	Liliopsida	59	58	41	18	33	0,60666667	0,4957265	0,43540668
Orchidaceae	Liliopsida	Hyacinthaceae	Liliopsida	72	35	64	8	43	0,52	0,3271028	0,35135135
Orchidaceae	Liliopsida	Hemerocallidaceae(I)	Liliopsida	78	28	71	7	44	0,48	0,26415095	0,31578946
Orchidaceae	Liliopsida	Heliconiaceae	Liliopsida	61	47	52	9	42	0,59333333	0,4351852	0,42180094
Orchidaceae	Liliopsida	Eriocaulaceae	Liliopsida	57	55	44	13	38	0,62	0,49107143	0,44927537
Orchidaceae	Liliopsida	Smilacaceae	Liliopsida	47	74	25	22	29	0,68666667	0,61157024	0,52284265
Orchidaceae	Liliopsida	Alliaceae	Liliopsida	40	67	32	8	43	0,73333335	0,62616825	0,57894737
Orchidaceae	Liliopsida	Dioscoreaceae	Liliopsida	41	73	26	15	36	0,72666667	0,6403509	0,5706806
Orchidaceae	Liliopsida	Cyperaceae	Liliopsida	44	71	28	16	35	0,70666665	0,6173913	0,5463917
Orchidaceae	Liliopsida	Haemodoraceae	Liliopsida	69	40	59	10	41	0,54	0,36697248	0,369863
Orchidaceae	Liliopsida	Burmanniaceae	Liliopsida	62	45	54	8	43	0,58666664	0,42056075	0,41509435
Orchidaceae	Liliopsida	Cymodoceaceae	Liliopsida	80	35	64	16	35	0,46666667	0,3043478	0,3043478
Orchidaceae	Liliopsida	Cyclanthaceae	Liliopsida	78	40	59	19	32	0,48	0,33898306	0,31578946
Orchidaceae	Liliopsida	Agavaceae	Liliopsida	33	76	23	10	41	0,78	0,6972477	0,6393443
Orchidaceae	Liliopsida	Alismataceae	Liliopsida	52	67	32	20	31	0,65333333	0,56302524	0,48514852
Orchidaceae	Liliopsida	Aloeaceae(I)	Liliopsida	55	52	47	8	43	0,63333333	0,48598132	0,46341464
Orchidaceae	Liliopsida	Amaryllidaceae	Liliopsida	37	69	30	7	44	0,75333333	0,6509434	0,6042781
Orchidaceae	Liliopsida	Anthericaceae	Liliopsida	43	63	36	7	44	0,71333333	0,5943396	0,55440414
Orchidaceae	Liliopsida	Araceae	Liliopsida	51	81	18	33	18	0,66	0,6136364	0,49253732
Orchidaceae	Liliopsida	Arecaceae	Liliopsida	58	73	26	32	19	0,61333334	0,55725193	0,44230768
Orchidaceae	Liliopsida	Asparagaceae(I)	Liliopsida	46	61	38	8	43	0,69333333	0,57009345	0,53061223
Orchidaceae	Liliopsida	Asphodelaceae(I)	Liliopsida	47	59	40	7	44	0,68666667	0,5566038	0,52284265
Orchidaceae	Liliopsida	Alstroemeriaceae	Liliopsida	46	58	41	5	46	0,69333333	0,5576923	0,53061223
Orchidaceae	Liliopsida	Bromeliaceae	Liliopsida	32	80	19	13	38	0,78666667	0,71428573	0,64835167

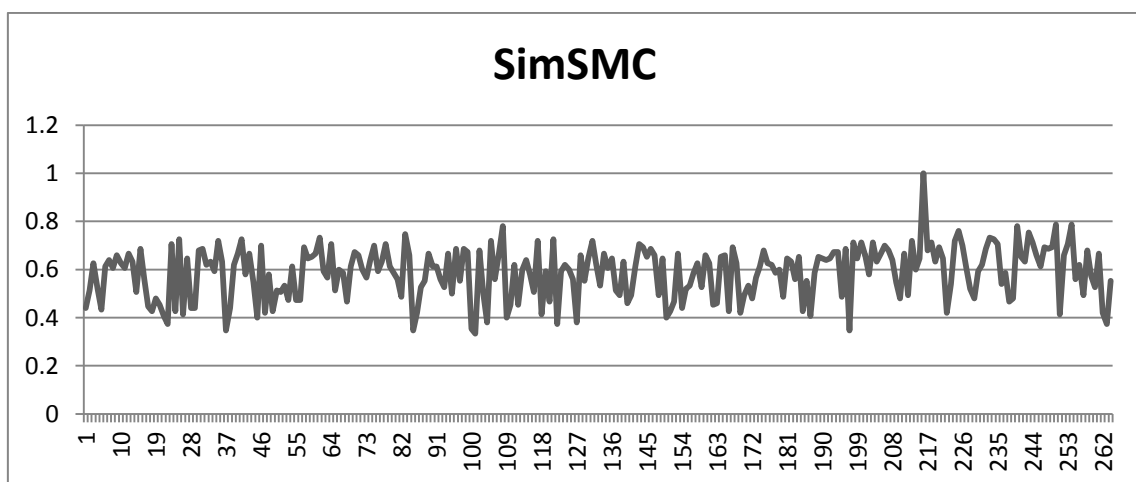
**Tabla 5 Resultados de métricas en todas las colecciones de plantas. Sexta parte.**

Patron	Patron Class	Family	Family Class	Distance	A	B	C	D	SimSMC	SimJaccard	SimRoger & Tanimoto
Orchidaceae	Liliopsida	Sparganiaceae	Liliopsida	88	29	70	18	33	0,41333333	0,24786325	0,26050422
Orchidaceae	Liliopsida	Calochortaceae	Liliopsida	51	56	43	8	43	0,66	0,5233645	0,49253732
Orchidaceae	Liliopsida	Cannaceae	Liliopsida	44	60	39	5	46	0,70666665	0,5769231	0,5463917
Orchidaceae	Liliopsida	Commelinaceae	Liliopsida	32	82	17	15	36	0,78666667	0,71929824	0,64835167
Orchidaceae	Liliopsida	Convallariaceae	Liliopsida	66	45	54	12	39	0,56	0,4054054	0,3888889
Orchidaceae	Liliopsida	Costaceae	Liliopsida	57	49	50	7	44	0,62	0,46226415	0,44927537
Orchidaceae	Liliopsida	Zosteraceae	Liliopsida	76	34	65	11	40	0,49333334	0,3090909	0,32743362
Orchidaceae	Liliopsida	Zingiberaceae	Liliopsida	48	61	38	10	41	0,68	0,559633	0,5151515
Orchidaceae	Liliopsida	Zannichelliaceae	Liliopsida	64	56	43	21	30	0,57333333	0,46666667	0,40186915
Orchidaceae	Liliopsida	Xyridaceae	Liliopsida	71	42	57	14	37	0,52666664	0,37168142	0,35746607
Orchidaceae	Liliopsida	Typhaceae	Liliopsida	50	66	33	17	34	0,66666667	0,5689655	0,5
Orchidaceae	Liliopsida	Triuridaceae	Liliopsida	87	28	71	16	35	0,42	0,24347825	0,2658228
Orchidaceae	Liliopsida	Strelitziaceae	Liliopsida	94	5	94	0	51	0,37333333	0,05050505	0,22950819
Orchidaceae	Liliopsida	Asteliaceae(I)	Liliopsida	67	41	58	9	42	0,55333334	0,37962964	0,3824885

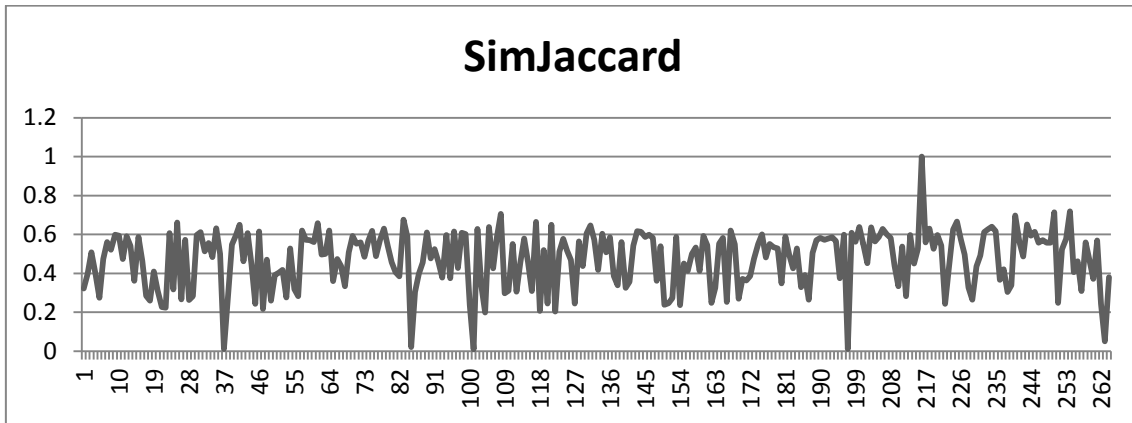
La primera columna de la tabla muestra el nombre de la familia usada como patrón usada a comparar, la segunda columna despliega la clase proveniente de la familia patrón, la tercera columna muestra el nombre de la familia la cual se está comparando, la cuarta columna despliega el nombre de la clase proveniente de la familia a comparar, la quinta columna muestra la distancia Levenstein entre ambas cadenas, sexta, séptima, octava y novena columnas muestran el número de a's, b's, c's y d's encontradas en ambas cadenas, y finalmente, decima, onceava y doceava columnas muestran los resultados de las métricas SMC, Jaccard y Roger & Tanimoto respectivamente.

### 4.3 Grafica de Resultados

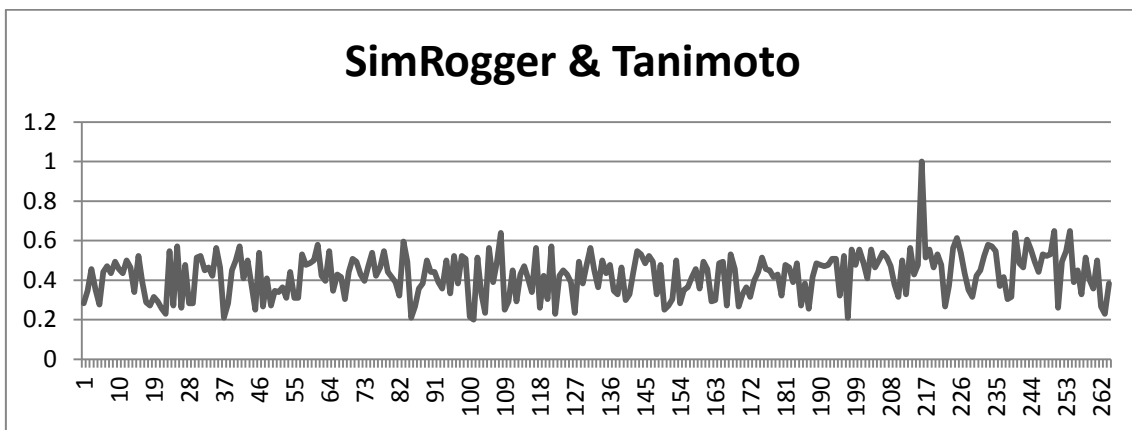
La Figura 52, 53 y 54 muestran el comportamiento de las tres métricas de similitud aplicadas a todas las familias, usando como patron base a la familia Orchidaceae.



**Figura 52 Resultados de métrica SMC a todas las familias.**



**Figura 53 Resultados de métrica Jaccard a todas las familias.**

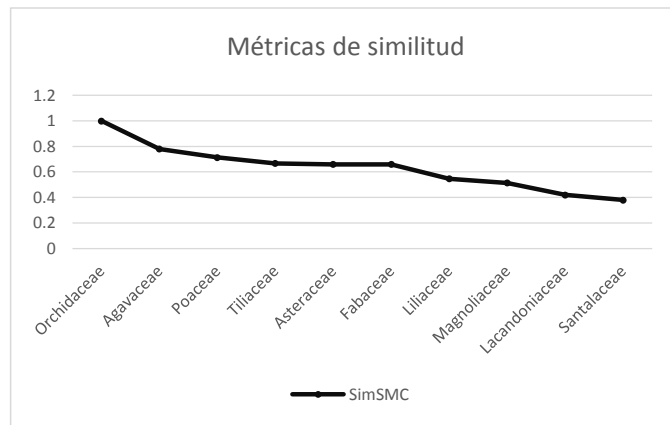


**Figura 54 Resultados de métrica Roger & Tanimoto a todas las familias.**

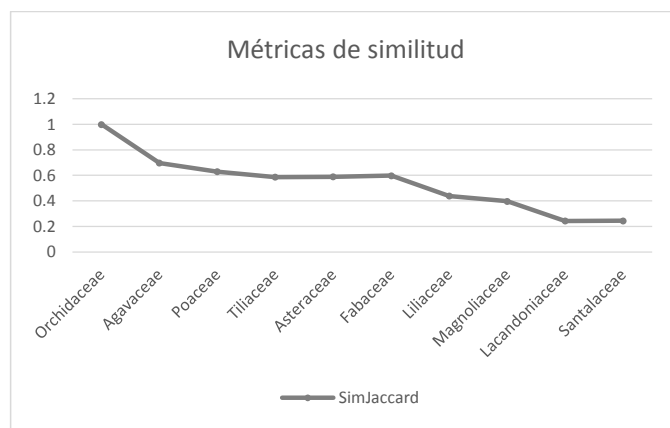
Estas figuras están conforadas de la siguiente manera:

- La parte izquierda esta la escala de similitud del 0-1 (osea 0 a 100 %).
- En la horizontal baja esta el número de familia en intervalos de 10.
- Finalmente en la parte cntral superior la métrica utilizada.

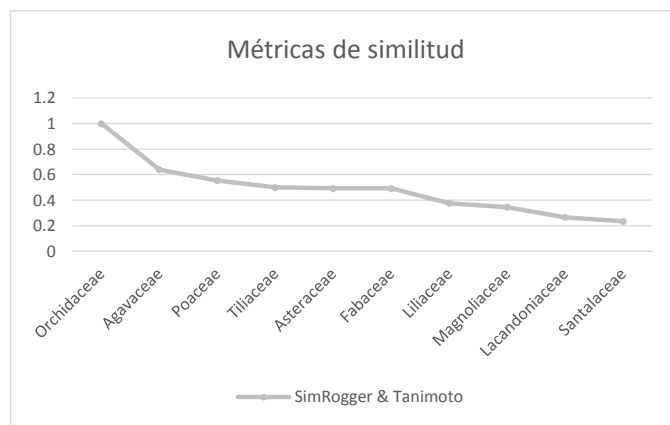
La Figura 55, 56 y 57 muestran el resultado de las SMC, Jaccard y Roger&Tanimoto aplicada a solo 10 familias al azar, distribuidas equitativamente entre las 6 colecciones creadas.



**Figura 55** Resultado de la métrica SMC a solo 10 familias.



**Figura 56** Resultado de la métrica Jaccard a solo 10 familias.



Estas figuras están conforadas de la siguiente manera:

- La parte izquierda esta la escala de similitud del 0-1 (osea 0 a 100 %).
- En la horizontal central esta el nombre de cada familia usada.

- Finalmente en la parte horizontal baja se encuentra el color y nombre distintivo para cada métrica.

#### **4.4 Descripción de los resultados**

En base a los resultados mostrados de las Figuras 56 y 57, las técnicas menos eficiente fue Rogger&Tanimoto, mientras que las más eficiente resultado SMC. Por ejemplo, en la Tabla 12 quinta parte, 16avo renglon la distancia Levenshtein entre los vectores es 0, dando como resultado que las cadenas son iguales y la similitud es 100% en las 3 métricas usadas, lo cual se puede ver gráficamente en el intervalo de las familias 211-221 de la Figura 57, otro ejemplo en el primer renglón la distancia Levenshtein es de 51, las similaridades resultantes fueron SMC con 66.00 %, Jaccard con 55.26 % y Roger&Tanimoto 49.25 %.

En la Figura 58, se percibe claramente los grados de eficiencia de las tres métricas, y también, en la primera comparación las tres métricas coinciden al 100%.

Cabe mencionar que, si el experto en botánica introduce las características completas y correctas logrará una “búsqueda exacta” (García, 2007), no importando la métrica implementada en este caso.

# Capítulo VI

## Conclusiones

Usando la metodología RUP de ingeniería de software, se creó una nueva aplicación prototipo llamada Taxon 2, para actualizar a la herramienta anterior llamada *GENCOMEX*. La nueva herramienta permite clasificar por clase, familia, género y especie, permitiendo una clasificación completa acorde al Dr. Villaseñor. Dicha fue programada en Java siguiendo los estándares de CMMI y Java Oracle en programación. Por otra parte, se usó el modelo MVC para el diseño de la aplicación.

También, se creó la aplicación *Web* para la consulta. Esta aplicación fue diseñada para poder visualizarse en dispositivos móviles, o PC. La selección de los campos puede ser vía *touch* o por selección directa de la casilla. El mapeo resultante es en base a la información capturada por la aplicación del administrador.

Cabe mencionar que, ambas aplicaciones fueron probadas y validadas por los expertos durante el lapso de un mes. La aplicación del administrador ya está instalada y operacional en el Instituto Botánico de la UNAM.

Una ventaja importante de la herramienta propuesta es que el usuario puede consultar la aplicación en un dispositivo móvil debido a la eficiencia de representación, manejo y procesamiento de información, y en trabajo de campo se puede hacer la caracterización y obtener resultados rápidamente.

Para la consulta y en base a las pruebas realizadas de las tres métricas, se comentaron los resultados a los expertos a cargo concluyendo como métrica implementada para el público en general la SMC, solo en el caso de buscar solo una planta. Por otra parte y por petición del Dr. Villaseñor, se implementó la consulta para mostrar todas las plantas que posean las características señaladas en la aplicación del administrador, ver Figura 49.

Se concluye que con los métodos de SMC, Jaccard y Rogger & Tanimoto permiten identificar especies de plantas usando el vector de características con 0's y 1's y que el método más eficiente ha sido SMC.

Es posible apoyar a los alumnos e investigadores de botánica en la identificación de especies al marcar una a una de las características de una planta a identificar con mayor eficacia. Con lo anterior se creó un vector de 0's y 1's y con la aplicación propuesta es posible un resultado exacto, o bien, en caso de que no se incorporen algunas características. La métrica SMC determinará a cuál especie tiene más posibilidad de pertenecer.

## **Trabajos futuros**

Modificar o crear una nueva aplicación taxonómica para clasificar una planta por clase, familia, género y especie, dado que, las características para categorizar a una planta por género y especie, todavía no se tienen completas ni validadas por la autoridad a cargo.

La aplicación solo es un prototipo propuesto en base a la información hasta ahora existente de clase y familia. No se puede asumir que trabaje la aplicación con igual eficiencia, hasta no tener toda la información y realizar las pruebas necesarias con datos reales

## Referencias

- Barker, B. S. (1996). Parameterized Pattern Matching: Algorithms and Applications. *Journal of Computer and System Sciences* 52, 28-42.
- Amón, I., & Jiménez, C. (2010). Funciones de similitud sobre Cadenas de Texto: Una comparación basada en la naturaleza de los Datos. *AIS Electronic Library (AISeL), Association for Information Systems.*, 1-16.
- Apostolico, A., Erdos, P. L., & Lewenstein, M. (2007). Parameterized matching with mismatches. *ELSEVIER, ScienceDirect, Journal of Discrete Algorithms*, 135-140.
- Bar-Yossef, Z., Jayram, T. S., Krauthgamer, R., & Kumar, R. (2004). Approximating Edit Distance Efficiently. *Foundations of Computer Science*, 550-559.
- Bernstein, A., Kaufmann, E., Kiefer, C., & Bürki, C. (2006). Simpack: A Generic Java Library for Similarity Measures in Ontologies. *University of Zurich*, 1-20.
- Butters, J., & Ciravegna, F. (2008). Using Similarity Metrics For Terminology Recognition. *LREC*, 2817-2822.
- Byde, A., Wan, H., & Cayzer, S. (2007). Personalized Tag Recommendations via Tagging and Content-based Similarity Metrics. *In ICWSM*, 1-2.
- Chen, S., Ma, B., & Zhang, K. (2009). On the similarity metric and the distance metric. *ELSEVIER Theoretical Computer Science*, 2365-2376.
- Cohen, W. W., Ravikumar, P., & Fienberg, S. E. (2003). A Comparison of Strietrics for Matching Names and Records. *Kdd workshop on data cleaning and object consolidation*, 73-78.
- García, J. F. (2007). Métricas de Similitud para Búsqueda Aproximada. *Revista de Tecnología, Facultad de Ingeniería de Sistemas, Universidad del Bosque*, 1-11.
- Juárez Hernández, R. R., Ruiz Castilla, J. S., Cervantes Carrillo, J., & García Lamont, F. (2014). Reconocimiento de patrones para la identificación de clase y familia de plantas a partir de sus caracteres. *Cuerpos Académicos de la DES Oriente en busqueda de la implementación de la ciencia y tecnología*, 402-407.
- K, B. (1994). Asteraceae, Cladistics and Clasificación. *Timber Press*.
- Lin, D. (1998). An Information-Theoretic Definition of Similarity. *ICML*, 296-304.
- Mehdi, O. A., Ibrahim, H., & Affendey, L. S. (2012). Instance based Matching using Regular Expression. *ELSEVIER, SciVerse ScienceDirect, Procedia Computer Science*, 688-695.
- Munguía-Romero, J. L. (1992). La computadora en la identificación botánica. *La era digital Ciencia y Desarrollo*.



- Rahman, M., Hassan, M. R., & Buyya, R. (2012). Jaccard Index based Availability Prediction in Enterprise Grids. *ELSEVIER ScienceDirect Procedia Computer Science*, 2707-2716.
- Ravi, K. M., Choubey, A., & Tripathi, K. K. (2013). Intuitionistic Fuzzy Automaton for Approximate String Matching. *ELSEVIER, ScienceDirect, Fuzzy Information and Engineering*, 29-39.
- Rubio, M., Alba, A., Mendez, M., Arce-Santana, E., & Rodriguez-Kessler, M. (2013). A Consensus Algorithm for Approximate String Matching. *ELSEVIER, ScVerse ScienceDirect*, 322-327.
- Villaseñor, J. L. (1993). La familia Asteraceae en México. *Revista de la Sociedad Mexicana de Historia Natural*, 117-124.
- Villaseñor, M. M. (1998). GENCOMEX: a computerized key to identify the genera of asteraceae of México. *Asociación de Biólogos de la Computación AC*.
- Wira Putra, M. E., & Iping, S. S. (2015). Structural off-line handwriting character recognition using approximate subgraph matching and levenstein distance. *ELSEVIER ScienceDirect Procedia Computer Science*, 340-349.
- Zarembo, I., Teilans, A., Rausis, A., & Buls, J. (2014). Assesment of Name Based Algorithms for Land Administration Ontology Matching. *ELSEVIER, ScienceDirect, Procedia Computer Science*, 53-61.
- Bernstein, A., Kaufmann, E., Kiefer, C., & Bürki, C. (2006). Simpack: A Generic Java Library for Similarity Measures in Ontologies. *University of Zurich*, 1-20.
- Butters, J., & Ciravegna, F. (2008). Using Similarity Metrics For Terminology Recognition. *LREC*, 2817-2822.
- Byde, A., Wan, H., & Cayzer, S. (2007). Personalized Tag Recommendations via Tagging and Content-based Similarity Metrics. *In ICWSM*, 1-2.
- Chen, S., Ma, B., & Zhang, K. (2009). On the similarity metric and the distance metric. *ELSEVIER Theoretical Computer Science*, 2365-2376.
- Lin, D. (1998). An Information-Theoretic Definition of Similarity. *ICML*, 296-304.
- Rahman, M., Hassan, M. R., & Buyya, R. (2012). Jaccard Index based Availability Prediction in Enterprise Grids. *ELSEVIER ScienceDirect Procedia Computer Science*, 2707-2716.
- Wira Putra, M. E., & Iping, S. S. (2015). Structural off-line handwriting character recognition using approximate subgraph matching and levenstein distance. *ELSEVIER ScienceDirect Procedia Computer Science*, 340-349.