Administración del conocimiento para la diversidad de plantas medicinales en México: Repositorio Inteligente Badiano S21

T E S I S

Maestría en Ciencias en Sistemas Inteligentes

Instituto Tecnológico y de Estudios Superiores de Monterrey

Por

Lic. David Adán Velázquez Sánchez

Diciembre 2009

Administración del conocimiento para la diversidad de plantas medicinales en México: Repositorio Inteligente Badiano S21

TESIS

Maestría en Ciencias en

Sistemas Inteligentes

Instituto Tecnológico y de Estudios Superiores de Monterrey

Por

Lic. David Adán Velázquez Sánchez

Diciembre 2009

Instituto Tecnológico y de Estudios Superiores de Monterrey

División de Graduados en Mecatrónica y Tecnologías de Información

Los miembros del comité de tesis recomendamos que la presente tesis de David Adán Velázquez Sánchez sea aceptada como requisito parcial para obtener el grado académico de Maestro en Ciencias en:

Sistemas Inteligentes

Comité de tesis:

Dr. José Aldo Díaz Prado
Asesor de la tesis

Dra. Janet Alejandra Gutiérrez Uribe
Sinodal

Dra. Laila Pamela Partida Martínez
Sinodal

Dr. Joaquín Acevedo Mascarúa
Director de Investigación y Posgrado
Escuela de Ingeniería

Diciembre de 2009

Administración del conocimiento para la diversidad de plantas medicinales en México: Repositorio Inteligente Badiano S21

Por

Lic. David Adán Velázquez Sánchez

TESIS

Presentada a la División de Mecatrónica y Tecnologías de Información

Este trabajo es requisito parcial para obtener el grado académico de Maestro en Ciencias en Sistemas Inteligentes

Instituto Tecnológico y de Estudios Superiores de Monterrey

Campus Monterrey

Monterrey, N.L. Diciembre de 2009

Este trabajo se lo dedico con todo mi cariño a mis padres que siempre me han brindado su apoyo y han creído y estado conmigo en las buenas y en las malas. Sin ellos este logro jamás lo hubiera realizado, en especial al apoyo tan grande de parte de mi madre, la cual, siempre fue el más grande aliento en este proceso. Le agradezco a mi novia Alejandra Monserrat que siempre creyó en mí y me dio ánimos en los momentos más difíciles. También le quiero dedicar parte de esfuerzo a mis amigos, los cuales nunca me han dejado de alentar y brindar su apoyo para que logre este objetivo y a ser parte en la formación como persona que este proceso ha brindado a mi vida. ¡Muchas gracias a todos!

Reconocimientos

Deseo externar un sincero agradecimiento a las personas que de alguna forma colaboraron en el desarrollo de esta tesis.

Al Dr. José Aldo Díaz Prado, por su asesoría durante el desarrollo de esta investigación; por su constante apoyo que siempre fue un estímulo para concluir este proyecto.

A mis sinodales, Dra. Laila Pamela Partida Martínez y Dra. Janet Alejandra Gutiérrez Uribe, por sus consejos para mejorar la calidad de esta tesis. Así como también a todo el equipo de doctores de las distintas áreas que trabajo en conjunto para la definición de la taxonomía a utilizar en el proyecto

A mis profesores de la maestría, por su enseñanza y a Doris Juárez, que siempre estuvo allí cuando se necesito de su apoyo.

A mis amigos que siempre me dieron ánimos en los momentos difíciles.

David Adán Velázquez Sánchez

Instituto Tecnológico y de Estudios Superiores de Monterrey
Diciembre 2009

Administración del conocimiento para la diversidad de plantas medicinales en México: Repositorio Inteligente Badiano S21

David Adán Velázquez Sánchez, M.C.
Instituto Tecnológico y de Estudios Superiores de Monterrey, 2009

Asesor de la tesis: Dr. José Aldo Díaz Prado

El conocimiento en México acerca de las propiedades curativas de las plantas mexicanas es extenso, disperso y muy a menudo empírico. Existen libros en los cuales se nombran algunas de las plantas más comunes y sus usos curativos, pero no existe un medio o entidad que trate de regular y acrecentar este conocimiento. Dada la gran diversidad de plantas y sus diferentes usos, se pretende crear una taxonomía de clasificación y estructuración de propiedades, resaltando las obtenidas de publicaciones científicas, que den pauta a la aplicación de algoritmos y técnicas computacionales que permitan la asociación de propiedades y atributos, para el análisis del acervo de plantas medicinales en México, siguiendo un proceso de ingeniería de software que ayude al éxito del proyecto. Con el objetivo de llegar a la administración de la información existente en un medio integrador y publico, el cual lo definimos como un repositorio de administración de conocimiento.

Palabras clave: *Repositorio inteligente, ingeniería de software, clasificación y plantas medicinales de México.*

Índice general

Índice de cuadros

Índice de figuras

Índice de algoritmos

Capítulo 1

Introducción

El estudio y descripción de la flora medicinal mexicana, ha sido un proceso constante a lo largo de la historia del país, en el que se presentan periodos de auténtica consolidación de dicho fenómeno; como resultado del auge inusitado que cobró en algunos momentos ese interés por conocer las propiedades curativas de las plantas, aparecieron obras escritas sobre al respecto, produciéndose abundante bibliografía. Como consecuencia de la época de la colonia, el conquistador español se encuentra no sólo ante el desconocido esplendor de una naturaleza exuberante que lo maravilla por su variedad y riqueza, sino que confronta la realidad cultural de un pueblo indígena cuyo grado de conocimiento sobre recursos naturales era superior[28].

Los estudios sobre las diversas especies de plantas en el pasado, arrojaban resultados en cuanto a la parte visible de las mismas; siendo caso que los estudios más actuales han reflejado que la parte de la raíz también es de suma importancia. Por tal motivo ahora se buscan metabolitos por cada miembro de las plantas como son las flores, frutos y semillas[24]. Este tipo de investigaciones más profundas nos llevan de la mano a realizar proyectos que arrojen una *taxonomía más amplia*, que en conjunto a las técnicas de obtención de datos que se encuentran ocultos en grandes cantidades de información, nos permita estar en posibilidades de lograr avances en el área. Como ejemplo, el encontrar propiedades curativas de plantas que antes no se conocían.

En la actualidad un gran número de investigadores de las áreas de biotecnología, bioquímica y medicina, se dan a la tarea de estudiar plantas mexicanas con propósitos específicos. La mayoría de ellos trabaja de manera aislada y apreciando su conocimiento como un gran tesoro, los únicos casos en los que esta información es compartida es cuando se hacen investigaciones o colaboraciones inter campus, de esta forma el conocimiento sólo queda para el limitado grupo de investigadores que lo creó.

En algunas ocasiones estos investigadores que generaron su conocimiento, no tienen alumnos a los cuales heredar o este o simplemente se han retirado o han cambiado su área de investigación, esto nos lleva a la pérdida de conocimiento previamente generado y por consecuencia a la inversión en vano de recursos.

Algunos de los factores más importantes de esta problemática es la poca disponibilidad de la información, ya que en la mayoría de los casos se requieren citas o reuniones

con los poseedores del conocimiento para poder llegar a la obtención de éste. Otro factor importante es la compartición de este conocimiento con el medio de investigadores o simplemente con los investigadores del mismo campus, el compartir la información se refiere a los datos medulares de la investigación y no simplemente a la noticia.

Actualmente, nuestro país se enfrenta a una crisis socioeconómica sin precedentes, y el desarrollo de la industria fitoquímica puede ser uno de los factores decisivos para fatigar esta crisis[24]. El avance en el área puede llevar a la implementación de una nueva medicina la cual fomente el crecimiento regional, o en el mejor de los casos del país.

1.1. Antecedentes

Existen muchos trabajos de clasificación de plantas, pero en esta tesis nos enfocamos a los trabajos publicados electrónicamente sea cual sea el tipo de sistema de información. En la actualidad existen herramientas de clasificación de plantas en internet, las que son de clasificación general, esto es que no se especializan en propiedades medicinales ni demás características que se desean agregar en este trabajo a la taxonomía de las plantas. Algunos autores de bases de datos creadas para la clasificación y explotación de las plantas son: (Allkin 1998; Allkin & Winfield 1989; Beaman & Regalado 1989; Gómez-Pompa & Nevling 1988; Crosby & Magill 1988; etc.). Estas bases de datos emplearon un simple acercamiento al manejo de la estructura taxonómica de manera jerárquica y la asociación de las entidades de los taxones[36].

La llegada de los repositorios inteligentes a las empresas e instituciones ha venido tomando un crecimiento considerable en los últimos años, debido a que gran parte de las empresas tienen necesidades de información y en ocasiones esta no se encuentra dentro de la empresa o el mismo giro no la produce, quiere decir, que hay que relacionar la información existente con su complemento, el cual puede encontrarse en algún sistema externo, esto para contar con ella de manera completa y poder explotarla de manera adecuada.

El realizar esta tarea de manera manual, en la gran mayoría de los casos es un proceso muy largo y tedioso, que el hacerlo de esta manera puede llevar una mala composición de información y por lo tanto a tomar malas decisiones dentro de las empresas. Por esto, lo ideal sería contar con un proceso de extracción de información, el cual de manera automática vaya en busca de la información necesaria, la limpie y valide para su acoplamiento con el sistema local.

El acceso a la información faltante se puede hacer de manera automática en diferentes medios, siendo los más comunes la búsqueda en la *Web* y la explotación de *Servicios Web* [1] públicos. En nuestro caso particular, se desea realizar un trabajo donde

[1]Los servicios web son frecuentemente solo API[2]'s Web las cuales pueden ser accedidas a través de

áreas como la biología, medicina, química y biotecnología se fusionen con las tecnologías de información para crear un repositorio inteligente llamado Badiano S21. Este proyecto consta principalmente de una solución web sobre la cual se pueda compartir el conocimiento y la información a los estudiantes, maestros, investigadores y el público en general. Se eligió una solución web porque es la forma más fácil y efectiva de lograr nuestra meta, en estos días casi cualquier persona tiene acceso a una computadora con una conexión a internet.

La figura 1.1 muestra los elementos de la solución a un nivel técnico macro.

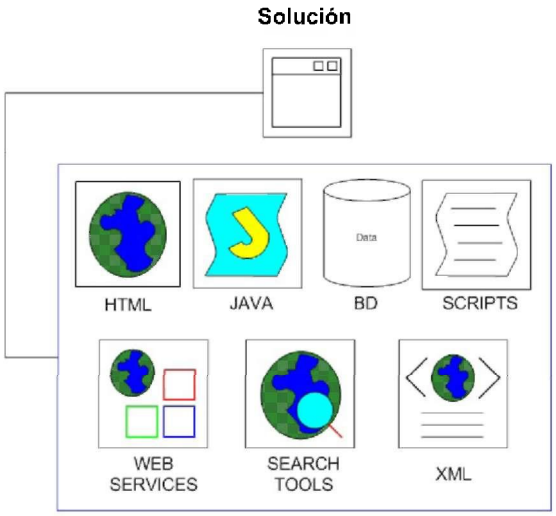

Figura 1.1: Componentes utilizados en la solución.

Con este proyecto se pretende lograr la interacción de los investigadores de las áreas de biotecnología, bioquímica y medicina.

1.2. Definición del problema

En la actualidad existe un gran acervo de información[3] con relación a la taxonomía y propiedades curativas de las plantas mexicanas. Estos acervos se encuentran aislados (solo tienen acceso a ello una cantidad limitada de investigadores) y en peligro de desaparición, dado que los investigadores que los elaboraron fallecieron o se retiraron del medio.

La información existente sobre el dominio de plantas medicinales se encuentra aislada y no está disponible para ser compartida o publicada, además que, esta infor-

toda la red, así como el internet, y las solicitudes hechas son ejecutadas en un sistema remoto. Estos usan mensajes XML que siguen el estándar SOAP los cuales se han hecho muy populares.

[3]Existe un inventario no oficial de casi 4,500 especies.

mación se encuentra clasificada de acuerdo a la forma de trabajo de cada investigador, por lo que es necesario proponer un estándar de clasificación y descripción, para poder compartir y explotar toda la información que en ellas se encuentran.

No existe una documentación de la relación que existe entre las propiedades de las plantas con las características en que se da una determinada variedad en una región .Los puntos anteriores y la falta de un repositorio de información electrónica mexicana (base de datos) que contenga esta para la preservación y publicación de la información a través de las generaciones, son algunas de las causas o puntos críticos que nos llevan a atacar este problema.

Los principales factores que envuelven este problema son:

- *Publicación*: Los estudios hechos a las plantas pueden ser compartido entre la comunidad científica.
- *Preservación*: Los estudios y los frutos de investigadores pueden perdurar en un repositorio de información.
- *Disponibilidad*: El acceso a la información las 24 horas del día, los 365 días del año a través de un sistema web es muy importante para la comunidad científica.
- *Ubicación de plantas*: La posibilidad de poder ubicar en donde se encuentra una cierta planta que cura una cierta enfermedad o las plantas medicinales que se encuentran en una determinada región o estado.

1.3. Objetivos

El objetivo general de este proyecto es crear un medio de difusión y diseminación del conocimiento asociado al acervo de plantas medicinales en México. El cual permita la aplicación de algoritmos y técnicas computacionales para la asociación de propiedades (metabolitos asociados, clasificación taxonómica, usos, etc.) y atributos (Microorganismos, localización geográfica, características de ambiente en el que se desarrollan, etc.) con el fin de analizar este acervo mexicano.

1.3.1. Específicos

Dentro de los objetivos específicos se tiene:

- Crear un medio de publicación de información relacionada a las propiedades medicinales de las plantas.
- Aplicar algoritmos de clasificación para clasificar en base a diferentes atributos y objetivos.

- Crear un repositorio de información que permita compartir el conocimiento almacenado en nuestra base de datos, generar nuevas hipótesis y líneas de investigación.

1.4. Hipótesis

La creación de un repositorio de información con una taxonomía adecuada de las plantas nacionales y la complementación de esta taxonomía con diferentes características asociadas a estas plantas, llevará a la obtención de patrones relacionados entre algunas plantas, y mediante éstos la posible investigación de ellos. Estos patrones se trataran de encontrar aplicando algoritmos de clasificación directamente a este repositorio.

1.4.1. Preguntas de investigación

Las preguntas que desea resolver esta tesis son las siguientes:

- ¿Qué arquitectura computacional es la ideal?
- ¿Cuáles son los factores más relevantes para entender, describir, caracterizar y agrupar las propiedades medicinales de las plantas?
- ¿Qué tipo de relaciones existen entre estos factores?
- ¿Cómo hacer un repositorio de la forma más flexible posible, para que este pueda adaptarse a las nuevas herramientas, descubrimientos científicos y cambios de acuerdo a las necesidades de los investigadores?

1.5. Metodología de trabajo

La construcción de éste repositorio se hará a través de un ciclo iterativo, en el cual se van refinando detalles conforme las iteraciones avanzan[19]. Los siguientes son los pasos que se involucran en el ciclo de vida de este proyecto:

- Definición de equipo de trabajo
- Elección de herramientas y software a utilizar
- Recopilación de requerimientos y definición de taxonomía completa
- Comparativo entre repositorios existentes
- Construcción de componentes reutilizables y estratégicos

- Diagrama de paquetes y clases
- Modelado de datos
- Extracción, transformación y carga (ETL)
- Desarrollo de aplicación final (front-end)
- Desarrollo de reportes
- Puesta a punto y performance
- Aseguramiento de calidad
- Puesta en producción
- Manual de usuario
- Cambios y mejoras

1.6. Alcance

En este proyecto de tesis se pretende cumplir con todos los puntos mencionados en la metodología de trabajo. Dada la naturaleza del proyecto (sistemas de información), se sabe que los cambios y mejoras son comunes, por lo que se brindará el apoyo necesario guiando en la aplicación de estos cambios. Se pretende que el repositorio sea usado por la comunidad de la institución para su consulta y aportación de nuevas investigaciones realizadas.

1.7. Composición

Esta tesis consta de cuatro capítulos en los cuales se presentan los resultados la investigación realizada. El primer capítulo da una introducción acerca del problema a resolver describiendo un poco la forma de la solución, los objetivos a lograr y la metodología de trabajo a utilizar. El segundo explica el origen del nombre del proyecto, además de sentar las bases tecnológicas y metodológicas para poder llevar a cabo esta investigación. El tercero contempla la forma de resolución del problema, una explicación breve de los elementos más importantes de la aplicación y una demostración experimental del alcance del proyecto. Finalmente en el cuarto y último capítulo se expresan las conclusiones de la investigación, las contribuciones realizadas y los trabajos futuros recomendados.

Capítulo 2

Desarrollo teórico

En este capítulo se presenta la historia breve del Códice Badiano, la definición de términos, los lenguajes y herramientas a utilizar, la base de reglas de diseño a seguir, los repositorios inteligentes, y finalmente algunos repositorios existentes en la actualidad que nos servirán como base de estudio.

2.1. Códice Badiano

El *Códice Badiano*[1], conocido también como Códice De la Cruz-Badiano, o por su título en latín *Libellus de medicinalibus indorum herbis* (Libro de las hierbas medicinales de los indios), es un escrito sobre la herbolaria mexica, escrito originalmente en náhuatl por el xochimilca Martín de la Cruz, alumno del Colegio de la Santa Cruz de Tlatelolco en el año 1552. Posteriormente fue traducido al latín por Juan Badiano, también xochimilca y estudiante del Colegio de la Santa Cruz. Otro nombre con que se conoce este códice es "Barberini ", debido a que Francesco Barberini lo poseía durante los primeros años del siglo XVII[12].

La figura 2.1 es una foto del contenido del Códice Badiano.

2.1.1. Contenido

El libro en latín, con material gráfico muy desarrollado, apareció en 1925 en la Biblioteca del Vaticano, después de siglos de aparente pérdida.

El libro sobre herbolaria medicinal mexicana de Martín de la Cruz es un importante legado para botánica y la medicina tradicionales. Todavía en años recientes, su estudio permitió al grupo del doctor José Luis Mateos, en el Instituto Mexicano del Seguro Social, encontrar el principio activo del *cihuapahtli* o *zoapatle*. De la Cruz cita que este vegetal se empleaba para facilitar el parto. Las investigaciones ratificaron que el zoapatle contiene un derivado de la hormona oxitocina (responsable de la contrac-

[1]En honor a este trabajo y retomando este para su publicación de manera informática se ha decidido nombrar a nuestro proyecto informático *Badiano S21*.

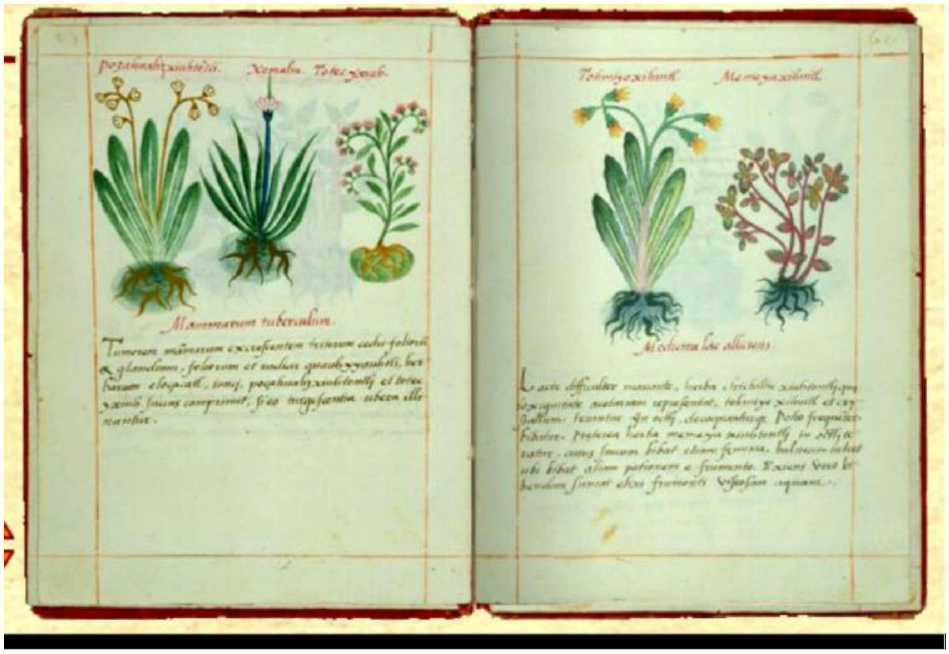

Figura 2.1: Muestra del contenido del Códice.

ción del útero). Toda la sabiduría contenida en este libro fue heredada por los químicos orgánicos no solamente mexicanos de este siglo, que han sobresalido en el terreno internacional con sus investigaciones sobre productos naturales[12].

2.1.2. Historia

Jacobo de Grado, fraile encargado del Convento y Colegio de la Santa Cruz de Tlatelolco, tenía en su posesión el texto creado y traducido para Francisco de Mendoza, hijo de Antonio de Mendoza, primer virrey de Nueva España. Mendoza envió el texto a España, donde fue depositado en la biblioteca real. Es probable que haya permanecido ahí hasta el siglo XVII, cuando apareció en posesión de Diego de Cortavila y Sanabria, farmacéutico de Felipe IV. De Cortavila pasó al cardenal italiano Francesco Barberini, quizá a través de interpósitos propietarios. El Libellus permaneció en la biblioteca de Barberini hasta 1902, cuando ella misma pasó a formar parte de la Biblioteca Vaticana. En 1990, el Papa Juan Pablo II devolvió el códice a México, donde es custodiado por la Biblioteca del Instituto Nacional de Antropología e Historia (INAH), en la Ciudad de México.

El libro, titulado en latín *Libellus de medicinalibus indorum herbis*, fue editado en 1552, cambiando para siempre el mundo de la farmacia, debido a que contenía descripciones de 185 plantas diferentes de América, así como sus usos terapéuticos. El códice Cruz-Badiano originalmente es un libro con dibujos coloreados y algunos en blanco y negro. Se encuentran en folios de 58 hojas (25.2 por 15 cm) y está dividido en

función a los distintos padecimientos de la época.

Don Martin de la Cruz nació en Zacapán, Xochimilco a finales del siglo XV. Estudió en un calmecac y a los 50 años ingresó en el Real Colegio de la Santa Cruz de Tlatelolco (fundado en 1533) donde fungía como curandero y alumno. Su libro original se llama *Amate-Cehuatl-Xihuitl-Pitli* y en este códice aparece él en la portada usando un traje de médico a la usanza indígena.

Don Juan Badiano nació en Chililico, Xochimilco hacia 1484. Se cree estudió en un calmecac, a los 41 años ingresó en el mismo Colegio de Tlatelolco, donde aprendió español, religión y latín, siendo compañero de Martín de la Cruz. Su principal aportación la hizo traduciendo él mismo la obra de Don Martín de la Cruz del náhuatl al latín, así como también aumentando el acervo del herbario indígena, reconociendo su obra y elevándolo a profesor dentro de los indígenas.

Este Códice fue primeramente llevado a España, y actualmente se encuentra en la Biblioteca Vaticana en Roma, Italia. Sigue teniendo gran importancia y vigencia: por mencionar un caso, en 1964 el IMSS mandó publicar ese códice con el fin de estudiar las propiedades de algunas plantas que ahí se mencionan[7].

2.2. Definición de términos

La *taxonomía* es la ciencia y práctica de la clasificación. Los esquemas taxonómicos están compuestos por unidades taxonómicas conocidas como taxas (taxón en singular), o tipos de cosas que están conformadas frecuentemente en una estructura jerárquica. Normalmente están relacionadas por relaciones de sub-tipo y súper-tipo, también llamadas relaciones de padres a hijos.

Un *Data Warehouse* es un repositorio de datos organizados electrónicamente, los cuales están diseñados para facilitar el reporteo y el análisis de información [16].

La *arquitectura computacional* se refiere al diseño de la plataforma computacional, la cual se extiende a través del software y hardware de la aplicación. Considerando las necesidades de compartición de información con diferentes componentes y usuarios.

La *Mineria de Datos* es el proceso de ordenación y recolección de información relevante dentro de grandes cantidades de datos, es comúnmente utilizada por la inteligencia de negocios en las organizaciones, análisis financieros, y también se esta incrementando su uso en las ciencias para extraer información de los enormes conjuntos de datos generados por métodos modernos de observación y métodos experimentales. Ha sido descrita como "la extracción no trivial de información implícita, previamente conocida y potencialmente útil existente en los datos" [11].

2.2.1. Clasificación

La *clasificación* puede ser formalizada como la tarea de aproximar una *función objetivo* desconocida $\Phi : I \times C \rightarrow \{T, F\}$ (que describe cómo instancias del problema deben ser clasificadas de acuerdo a un experto en el dominio) por medio de una función $\Theta : I \times C \rightarrow \{T, F\}$ llamada el *clasificador*, donde $C = \left\{C_1, ..., C_{|c|}\right\}$ es un conjunto de categorías predefinido, e I es un conjunto de instancias del problema. Comúnmente cada instancia $i_j \in I$ es representada como una lista $A = \left\{a_1, a_2, ..., a_{|A|}\right\}$ de valores característicos, conocidos como *atributos*, i.e. $i_j = \left\{a_{1j}, a_{2j}, ..., a_{|A|_j}\right\}$. Si $\Phi : i_j \times c_i \rightarrow T$, entonces i_j es llamado un *ejemplo positivo* de c_i, mientras si $\Phi : i_j \times c_i \rightarrow F$ éste es llamado un *ejemplo negativo* de c_i.

Para generar automáticamente el clasificador de c_i es necesario un proceso inductivo, llamado el *aprendiz*, el cual por observar los atributos de un conjunto de instancias preclasificadas bajo c_i o $\overline{c_i}$, adquiere los atributos que una instancia no vista debe tener para pertenecer a la categoría. Por tal motivo, en la construcción del clasificador se requiere la disponibilidad inicial de una colección Ω de ejemplos tales que el valor de $\Phi(i_j, c_i)$ es conocido para cada $(i_j, c_i) \in \Omega \times C$.A la colección usualmente se le llama *conjunto de entrenamiento (Tr)*. En resumen, al proceso anterior se le identifica como *aprendizaje supervisado* debido a la dependencia de Tr [31].

Recientemente se dio un cambio en el uso y el diseño de la tecnología Web el cual proporciona creatividad, compartición de información segura, colaboración y funcionalidad en el Web. Estos cambios han llevado a una revolución en la industria computacional causado por el hecho de moverse a la plataforma de internet, y el intento para entender las reglas para lograr el éxito en esta plataforma [27].

Esta revolución ha ayudado a lograr lo que se intenta en este proyecto de tesis, ofreciendo herramientas para la creación de excelentes soluciones web, las cuales, puedan satisfacer a los usuarios en el manejo del Web. *FLEX* es un framework de código libre altamente productivo para la construcción y el mantenimiento de aplicaciones web expresivas las cuales trabajan consistentemente en la mayoría de los navegadores, equipos de escritorio y sistemas operativos. Con FLEX se puede construir aplicaciones RIA (rich internet application *Web 2.0*) y dar a los usuarios una interface y performance como el que se da en una aplicación de escritorio.

Java es la plataforma perfecta para aplicaciones que necesitan estilos de interface tan poderosas y ricas como las de escritorio combinados con el crecimiento y las características de red de la Web.

Los *Servicios Web* son sistemas en la capa de software que están diseñados para la interoperabilidad maquina-a-maquina que existe dentro de una red [9]. Los servicios web son frecuentemente solo API's Web las cuales pueden ser accedidas a través de toda la red, así como el internet, y las solicitudes hechas son ejecutadas en un sistema remoto. Estos usan mensajes XML que siguen el estándar SOAP los cuales se han hecho

muy populares.

Los servicios web brindan la oportunidad de una fácil compartición del conocimiento con otras aplicaciones y usuarios, utilizando una forma estándar de representar datos de forma estructurada como XML. Estas aplicaciones pueden están escritas para otras plataformas o tecnologías y seguir teniendo acceso a los servicios.

2.3. Ingeniería de software

Como método para llegar a la construcción de un repositorio de calidad, siendo este, en si mismo software, seguiremos las reglas básicas de la ingeniería de software, señalando que, aunque no aplicando al 100 % cada uno de sus pasos y recomendaciones, si siguiendo los más importantes y que consideramos de provecho para nuestro proyecto o qué bien se adaptan a nuestra capacidad y tiempo de desarrollo.

Para esto hablaremos un poco de lo que es la ingeniería de software, los pasos o modelos que involucra y las recomendaciones generales del proceso. De esta manera podemos tomar la definición de que la Ingeniería de software es la disciplina o área de la informática que ofrece métodos y técnicas para desarrollar y mantener software de calidad.

Esta ingeniería trata con áreas muy diversas de la informática y de las ciencias de la computación, tales como construcción de compiladores, sistemas operativos, o desarrollos Intranet/Internet, abordando todas las fases del ciclo de vida del desarrollo de cualquier tipo de sistemas de información y aplicables a infinidad de áreas (negocios, investigación científica, medicina, producción, logística, banca, control de tráfico, meteorología, derecho, Internet, Intranet, etc.).

El desarrollo de software y la ingeniería de software no son lo mismo, ya que el segundo implica niveles de rigor y prueba de procesos que no son apropiados para todo tipo de desarrollo de software[25].

La metodología utilizada en la ingeniería de software cuenta básicamente de 6 etapas las cuales pudieran variar dependiendo la clase del proyecto, pero en si definen los pasos básicos que debe de cubrir un buen proceso de desarrollo, a continuación enumeramos estos pasos y posteriormente explicamos en qué consisten brevemente cada uno de ellos[33]:

- Identificación de necesidades
- Análisis de requerimientos
- Especificación
- Arquitectura
- Programación

- Pruebas
- Documentación
- Mantenimiento

2.3.1. Identificación de necesidades

En ocasiones existen clientes los cuales en su proceso empresarial jamás han involucrado un sistema de información y a lo largo del tiempo a administrado sus procesos de forma manual. Es aquí cuando a gran medida el cliente cuenta con determinadas necesidad de administración de información y es tarea del ingeniero de software el poder identificarlas y brindar ayuda al cliente con la experiencia obtenida a través del tiempo.

2.3.2. Análisis de requerimientos

Extraer los requisitos de un producto de software es la primera etapa para crearlo. Mientras que los clientes piensan que ellos saben lo que el software tiene que hacer, se requiere de habilidad y experiencia en la ingeniería de software para reconocer requisitos incompletos, ambiguos o contradictorios. El resultado del análisis de requisitos con el cliente se plasma en el documento *ERS*[2] (Especificación de Requerimientos del Sistema), cuya estructura puede venir definida por varios estándares, tales como *CMM-I*[3] (Modelo de Capacidad y Madurez). Asimismo, se define un *diagrama de Entidad/Relación*[4], en el que se plasman las principales entidades que participarán en el desarrollo del software.

La captura, análisis y especificación de requisitos (incluso pruebas de ellos), es una parte crucial; de esta etapa depende en gran medida el logro de los objetivos finales. Se han ideado modelos y diversos procesos de trabajo para estos fines. Aunque aún no está formalizada, ya se habla de la Ingeniería de Requisitos.

2.3.3. Especificación

La Especificación de Requerimientos describe el comportamiento esperado en el software una vez desarrollado. Gran parte del éxito de un proyecto de software

[2]En general en este documento se especifican las características o requerimientos que el sistema de cumplir para resolver las necesidades del cliente.

[3]Es un modelo para la mejora y evaluación de procesos para el desarrollo, mantenimiento y operación de sistemas de software. El cual cuenta con niveles, los cuales certifican a las organizaciones el grado de madurez de su proceso de desarrollo de software.

[4]Es una herramienta para el modelado de datos de un sistema de información. Estos modelos expresan entidades relevantes para un sistema de información así como sus interrelaciones y propiedades.

radicará en la identificación de las necesidades del negocio (definidas por la alta dirección), así como la interacción con los usuarios funcionales para la recolección, clasificación, identificación, priorización y especificación de los requerimientos del software.

Entre las técnicas utilizadas para la especificación de requerimientos se encuentran:

- Casos de Uso
- Historias de usuario

Siendo los primeros más rigurosos y formales, los segundas más ágiles e informales.

2.3.4. Arquitectura

La integración de infraestructura, desarrollo de aplicaciones, bases de datos y herramientas gerenciales, requieren de capacidad y liderazgo para poder ser conceptualizados y proyectados a futuro, solucionando los problemas de hoy. El rol en el cual se delegan todas estas actividades es el del Arquitecto. El Arquitecto de Software es la persona que añade valor a los procesos de negocios gracias a su valioso aporte de soluciones tecnológicas. La Arquitectura de Sistemas en general, es una actividad de planeación, ya sea a nivel de infraestructura de red y hardware, o de software. La Arquitectura de Software consiste en el diseño de componentes de una aplicación (entidades del negocio), generalmente utilizando patrones de arquitectura. El diseño arquitectónico debe permitir visualizar la interacción entre las entidades del negocio y además poder ser validado, por ejemplo por medio de diagramas de secuencia. Un diseño arquitectónico describe en general el cómo se construirá una aplicación de software. Para ello se documenta utilizando diagramas, por ejemplo:

- Diagramas de clases
- Diagramas de base de datos
- Diagramas de despliegue
- Diagramas de secuencia
- Diagramas de infraestructura física

Siendo los dos primeros los mínimos necesarios para describir la arquitectura de un proyecto que iniciará a ser codificado. Depende del alcance del proyecto, complejidad y necesidades, el arquitecto elige qué diagramas elaborar. Entre las herramientas para diseñar arquitecturas de software se encuentran:

- Enterprise Architect

- Microsoft Visio for Enterprise Architects

Para ilustrar un poco el proceso hasta este punto presentamos la figura 2.2 la cual muestra un diagrama de cómo se van dando las etapas en el proceso y algunos documentos o diagramas que se desarrollan en ellas.

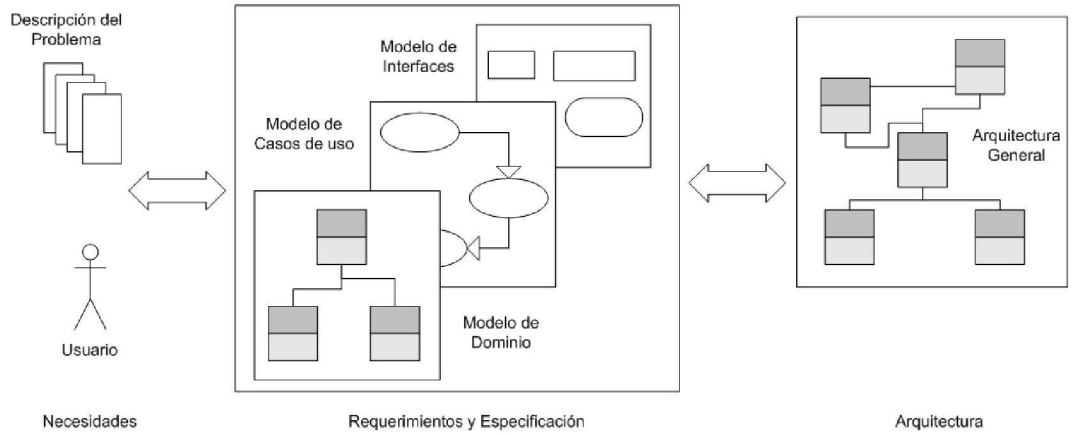

Figura 2.2: Parte del proceso de ingeniería de software.

Parte esencial de una buena arquitectura, confiable y escalable, es la que se fundamenta en patrones de diseño, de los cuales hablaremos a continuación.

Patrones de diseño de software

Los patrones de diseño (design patterns) son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software y otros ámbitos referentes al diseño de interacción o interfaces.

Un patrón de diseño es una solución a un problema de diseño. Para que una solución sea considerada un patrón debe poseer ciertas características. Una de ellas es que debe haber comprobado su *efectividad* resolviendo problemas similares en ocasiones anteriores[29]. Otra es que debe ser *reusable*, lo que significa que es aplicable a diferentes problemas de diseño en distintas circunstancias.

El objetivo que los patrones de diseño pretenden es:

- Proporcionar catálogos de elementos reusables en el diseño de sistemas software.
- Evitar la reiteración en la búsqueda de soluciones a problemas ya conocidos y solucionados anteriormente.
- Formalizar un vocabulario común entre diseñadores.
- Estandarizar el modo en que se realiza el diseño.

- Facilitar el aprendizaje de las nuevas generaciones de diseñadores condensando conocimiento ya existente.

Asimismo, no pretenden:

- Imponer ciertas alternativas de diseño frente a otras.
- Eliminar la creatividad inherente al proceso de diseño.

No es obligatorio utilizar los patrones, solo es aconsejable en el caso de tener el mismo problema o similar que soluciona el patrón, siempre teniendo en cuenta que en un caso particular puede no ser aplicable. Abusar o forzar el uso de los patrones puede ser un error.

A continuación mencionaremos algunos de los principales patrones de diseño:

- Abstract Factory (Fábrica abstracta)
- Factory Method (Método de fabricación)
- Singleton (Instancia única)
- Adapter (Adaptador)
- Composite (Objeto compuesto)
- Decorator (Envoltorio)
- Facade (Fachada)
- Proxy
- Observer (Observador)
- State (Estado)
- Strategy (Estrategia)
- Template Method (Método plantilla)

2.3.5. Programación

Reducir un diseño a código puede ser la parte más obvia del trabajo de ingeniería de software, pero no necesariamente es la que demanda mayor trabajo y ni la más complicada. La complejidad y la duración de esta etapa está íntimamente relacionada al o a los lenguajes de programación utilizados, así como al diseño previamente realizado.

2.3.6. Pruebas

Consiste en comprobar que el software realice correctamente las tareas indicadas en la especificación del problema. Una técnica de prueba es probar por separado cada módulo del software, y luego probarlo de forma integral, para así llegar al objetivo. Se considera una buena práctica el que las pruebas sean efectuadas por alguien distinto al desarrollador que la programó, idealmente un área de pruebas; no obstante el programador debe hacer sus propias pruebas. En general hay dos grandes formas de organizar un área de pruebas, la primera es que esté compuesta por personal inexperto y que desconozca el tema de pruebas, de esta forma se evalúa que la documentación entregada sea de calidad, que los procesos descritos son tan claros que cualquiera puede entenderlos y el software hace las cosas tal y como están descritas. El segundo enfoque es tener un área de pruebas conformada por programadores con experiencia, personas que saben sin mayores indicaciones en qué condiciones puede fallar una aplicación y que pueden poner atención en detalles que personal inexperto no consideraría.

2.3.7. Documentación

Todo lo concerniente a la documentación del propio desarrollo del software y de la gestión del proyecto, pasando por modelaciones (*UML*[5]), diagramas, pruebas, manuales de usuario, manuales técnicos, etc.; todo con el propósito de eventuales correcciones, usabilidad, mantenimiento futuro y ampliaciones al sistema.

2.3.8. Mantenimiento

Mantener y mejorar el software para enfrentar errores descubiertos y nuevos requisitos. Esto puede llevar más tiempo incluso que el desarrollo inicial del software. Alrededor de 2/3 de toda la ingeniería de software tiene que ver con dar mantenimiento. Una pequeña parte de este trabajo consiste en arreglar errores, o *bugs*. La mayor parte consiste en extender el sistema para hacer nuevas cosas. De manera similar, alrededor de 2/3 de toda la ingeniería civil, arquitectura y trabajo de construcción es dar mantenimiento.

[5]Lenguaje Unificado de Modelado, es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad. Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables.

2.4. Modelos de desarrollo de software

La ingeniería de software tiene varios modelos, paradigmas o filosofías de desarrollo en los cuales se puede apoyar para la realización de software, de los cuales podemos destacar a éstos por ser los más utilizados y los más completos:

- Modelo en cascada o Clásico (modelo tradicional)
- Modelo en espiral (modelo evolutivo)
- Modelo de prototipos
- Desarrollo por etapas
- Desarrollo iterativo y creciente o Iterativo e Incremental
- RAD (Rapid Application Development)

2.5. Consideraciones de elección de herramientas y diseño

En orden para alcanzar nuestro objetivo de construir nuestro repositorio inteligente es esencial que contemos con las herramientas necesarias. Esto es especialmente cierto cuando nuestra meta es alcanzar inteligencia de negocios. En general al querer construir un repositorio se cuenta con una cierta complejidad debido a que se debe de interactuar con distintos departamentos dentro de las organizaciones, por eso es fácil ver que una buena elección de software para inteligencia de negocio y personal capacitado es muy importante[17]. En esta sección hablaremos un poco de las distintas clases de herramientas y consideraciones que existen en este proceso, la cuales se agrupan de la siguiente manera:

- Consideraciones generales
- Base de datos / hardware
- Modelado de datos
- Herramientas ETL
- Herramientas OLAP
- Herramientas de reporteo
- Herramientas para metadatos

Categoria	Comprar	Construir
Costo		√
Tiempo de Implementación	√	
Documentación	√	
Funcionalidad / Características especiales	√	
Diseñado para necesidades especificas		√
Confianza para involucrar otras herramientas		√

Cuadro 2.1: Tabla de decisión para comprar o construir.

2.5.1. Consideraciones generales

Cuando evaluamos que herramienta para inteligencia de negocios usar, lo primero que debemos de determinar para nuestra decisión es *Comprar o Construir*. Podemos usar la tabla 2.5.1 para comparar las dos posibles decisiones.

Claramente podemos ver que cada una tiene sus ventajas y desventajas, por lo que a menudo es bueno considerar cada uno por separado. Por ejemplo, es claro que no es viable el escribir el código necesario para una base de datos relacional. En general, el decidir que opción tomar la basamos en los siguientes criterios:

- Habilidades técnicas del usuario
- Requerimientos
- Presupuesto disponible
- Tiempo

Debido a que cada herramienta del área de inteligencia de negocios tiene diferentes funciones, los criterios de *Comprar o Construir* se basaran en cada uno por separado. Entraremos en discusión detalladamente para cada una de las herramientas adelante.

Además de las funcionalidad de la herramienta, la cual discutiremos en las siguientes secciones. Existen algunas consideraciones que debemos de tomar en cuenta cuando deseamos comprar una herramienta en general:

La *Estabilidad del fabricante de la herramienta* es el factor más importante para decidir que herramienta comprar, aun más importante que las funcionalidades que la herramienta posee, debido a la simple razón que el fabricante debe seguir en el mercado, y dado eso hacer mejoras a la herramienta. De otra forma, si la compañía es candidata a salir del negocio en 6 meses, entonces no importa que esta tenga en ella el estado del arte en su funcionalidad, porque tarde o temprano estará desactualizada.

Algunas formas de saber cómo se encuentra la estabilidad de la compañía son:

- ¿Qué tipo de espacio u oficina ocupan?
- ¿Están gastando en renta por la mejor oficina solo para ser tomados en cuenta?
- ¿Están invirtiendo parte de sus ganancias en la mejora de sus productos?

Además, viendo la capacidad del gerente general, podemos ver que tan exitosa puede ser la compañía, esto es, que en este sector, por lo general las compañías son nuevas y basta ver la experiencia en otras empresas de estos dirigentes para ver el rumbo que puede tomar la nueva empresa.

El *soporte* es muy importante, estos es, ¿Qué tipos de soporte ofrecen? En general es un estándar que los fabricantes cobren un porcentaje del costo de la licencia de su producto por soporte.

Servicios profesionales entre los cuales se encuentran la consultoría y educación. ¿Qué tipo de propuesta en consultoría ofrece? ¿Nuestros requerimientos y el precio de la consultoría son razonables? ¿El fabricante nos irá a colocar a un recién egresado y querer cobrar demasiado caro por la consultoría? Una decisión sabia seria el hablar con personal del equipo de consultoría antes de firmar cualquier tipo de contrato. Consideraciones y preguntas como esta, son las que nos debemos hacer antes de tomar la decisión de comprar herramientas a cualquier vendedor.

2.5.2. Selección de hardware y base de datos

Aquí la única opción es ver que hardware y software comprar debido a que es casi imposible construir cada uno de estos desde cero. Para tomar nuestra decisión de que plataforma de hardware/software existen varios aspectos que deben de ser cuidadosamente considerados:

- *Escalabilidad:* ¿De que forma el sistema puede crecer en medida que nuestras necesidades de almacenamiento crezcan? ¿Que RDBMS y plataforma de hardware puede manejar grandes cantidades de datos de manera eficiente? Para tener una idea de esto, se debe de calcular la cantidad de datos de manera aproximada que existirá en el repositorio cuando este se encuentre de manera estable y madura funcionando y partir de esto basarse en pruebas y números disponibles.
- *Soporte para el procesamiento paralelo:* Los días de las computadoras con un solo procesador que valían miles de millones de dólares se han ido, y hoy en día las computadoras o servidores más poderosos usan múltiples procesadores, donde cada procesador puede realizar parte de la tarea, todos al mismo tiempo, por lo cual es importante considerar si el tipo de licencia que maneja el RDBMS es por procesador.

- *Combinación de RDBMS/Hardware:* Debido a que el RDBMS reside físicamente en la plataforma de hardware, habrá ciertas partes del código que serán dependientes al hardware, como resultado, errores y correcciones a errores serán dependientes al hardware.

Bases de datos relacionales populares

- Oracle
- Microsoft SQL Server
- IBM DB2
- Teradata
- Sybase
- MySQL

Sistemas operativos populares

- Linux
- FreeBSD
- Microsoft Windows

2.5.3. Modelado de datos

En orden para detallar y definir bien que es lo que se hace en esta fase, comenzaremos con explicar el modelado conceptual, lógico y físico de los datos. Los tres modelos tratan de explicar en si las entidades que existirán en el repositorio, y cada uno de ellos se diferencia del otro por el nivel de detalle con el que cuentan[30], la tabla 2.5.3 muestra que características incluye cada uno de estos modelos:

La figura 2.3 muestra gráficamente como serian cada uno de los modelos:

2.5.4. Herramientas ETL (Extracción, transformación y carga)

Cuando tenemos que seleccionar una herramienta de este tipo, no siempre es necesario comprar una herramienta. Esta decisión básicamente se puede tomar basada en los siguientes aspectos:

Característica	Conceptual	Lógico	Físico
Nombre de entidades	√	√	
Relaciones de entidades	√	√	
Atributos		√	
Llave primaria		√	√
Llaves foráneas		√	√
Nombre de tabla			√
Nombre de columna			√
Tipo de dato de columna			√

Cuadro 2.2: Características de modelos de datos.

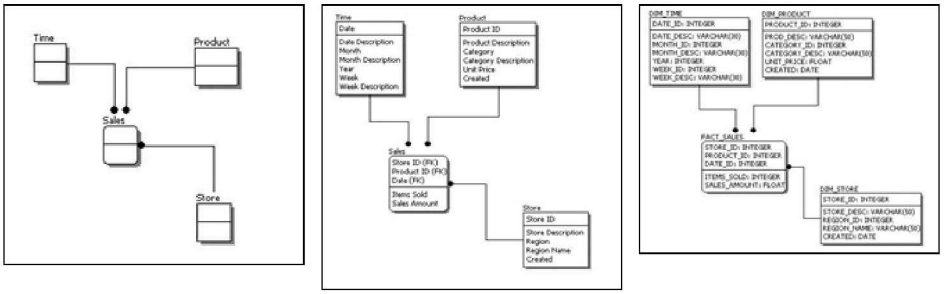

Figura 2.3: Muestra de izquierda a derecha los modelos conceptual, lógico y físico.

- *Complejidad en la transformación de los datos:* Entre más complicado sea la transformación de los datos, más viable será la adquisición de una herramienta ETL.

- *Necesidades de limpiado de datos:* Si los datos necesitan ir de limpiado en limpiado para poder ser almacenados en el repositorio, entonces, lo mejor es comprar una herramienta con funcionalidades muy fuertes en el limpiado de datos. De otra manera, será suficiente con construir una herramienta de limpiado desde cero.

- *Volumen de datos:* Existen herramientas comerciales con características para acelerar el movimiento de los datos. De esta manera, si el volumen de los datos lo requiere, es recomendable comprar una herramienta.

Mientras la selección de la plataforma de hardware y la base de datos es una obligación, la selección de una herramienta ETL es altamente recomendada, mas sin embargo no es una obligación. Cuando se evalúa una herramienta ETL, es recomendable ver las siguientes características:

- *Capacidad de la herramienta:* Esto incluye la transformación y limpiado de datos, en general las herramientas ETL se caracterizan por contar con una de las dos de

manera fuerte y solida, pero existen algunas que son bastante fuertes en ambas. Como resultado, si sabes que tus datos de entrada vendrán sucios, asegúrate que tu herramienta tenga buenas capacidad de limpiado de datos. Si sabes que tus datos necesitaran de varias transformaciones antes de quedar listos, pues es necesario que elijas una herramienta con fuertes capacidades de transformación.

- *Habilidad de poder leer directo de nuestra fuente de datos:* Para cada organización, existen diferentes tipos de fuentes de datos. Por lo que es importante que nos aseguremos de que la herramienta pueda leer directamente de la fuente de nuestros datos.

- *Soporte para metadatos:* Las herramientas ETL juegan un papel muy importante en los metadatos porque mapean la fuente hacia el destino, lo cual es una pieza importante en los metadatos. En algunos casos, las organizaciones han confiado en la documentación de la herramienta como su fuente de metadatos.

Herramientas populares

- IBM WebSphere Information Integration (Ascential DataStage)
- Ab Initio
- Informatica
- Talend

2.5.5. Herramientas OLAP

Las herramientas OLAP(Procesamiento analítico en línea) se enfocan en partir y cortar los datos. Para esto, ellos necesitan una fuerte capa de metadatos, así como flexibilidad en la aplicación final. Estas características son típicamente difíciles de encontrar en los sistemas hechos en casa. Así que si el análisis OLAP es parte importante en la integración del repositorio, la recomendación es comprar una herramienta de este tipo. Después de hablar de las recomendaciones en este tipo de herramientas, ahora hablaremos de los tipos de herramientas disponibles, MOLAP (OLAP Multidimensional) Y ROLAP (OLAP Relacional).

- *MOLAP:* Para este tipo, un cubo es agregado desde la fuente de datos relacional. Cuando los usuarios solicitan un reporte, la herramienta MOLAP puede generar la instrucción rápidamente porque todos los datos se encuentran pre-agregados en el cubo.

- *ROLAP:* Para este caso, en lugar de tener los datos pre-agregados en el cubo, el núcleo ROLAP actúa esencialmente como un pequeño generador de comandos SQL[6]. Estas herramientas cuentan generalmente con un diseñador grafico, donde el administrador del repositorio puede especificar las relaciones entre las tablas relacionales, así como sus dimensiones, atributos, y jerarquías entre las tablas.

En estos días, existe una convergencia entre los fabricantes tradicionales de herramientas MOLAP y ROLAP. Los fabricantes de ROLAP reconocen que los usuarios quieren sus reportes estén lo más rápido posible, así que están implementando funcionalidad MOLAP en sus herramientas; Por otro lado, los fabricantes MOLAP reconocen que muchas veces es necesario desplegar a mayor nivel de detalle la información, niveles a los cuales los cubos tradicionales no llegan debido a razones de tamaño y performance.

Algunos criterios para evaluar que herramienta OLAP seleccionar son los siguientes:

- *Habilidad para controlar el paralelismo proporcionado por el RDBMS y el hardware:* Esto puede incrementar de gran manera el performance de la herramienta y ayudar a la carga de los datos hacia los cubos los más rápido posible.

- *Performance:* Además de soportar el paralelismo, la herramienta debe de ser rápida en términos de la carga de los datos hacia los cubos y en leer los datos desde los cubos.

- *Esfuerzos por hacerse a la medida:* Mas y mas herramientas están siendo usadas como herramientas de reporteo avanzado. Estos es porque en muchos casos, específicamente en las implementaciones ROLAP. En muchos casos el poder cambiar a la medida el sistema final al usuario (front-end) se convierte en un factor para la selección de la herramienta.

- *Características de seguridad:* Debido a que estas herramientas podrían ser usadas por diferentes usuarios, asegurarse que la gente sólo pueda ver lo que está relacionado con su actividad. La mayoría de las herramientas cuentan con una capa de seguridad que puede interactuar con los protocolos de autenticación corporativos. Sin embargo existen casos que grandes empresas han desarrollado sus propios mecanismos de autenticación y cuentan con una simple y única firma para todo. Para esto casos, el integrar la herramienta con el mecanismo de seguridad local puede requerir de algo de trabajo. En estos casos es recomendable que el equipo del fabricante venga y haga una prueba en relación a la posible interacción de los mecanismos.

[6]Lenguaje de consulta estructurado, es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones en éstas.

- *Soporte para metadatos:* Debido a que las herramientas OLAP agregan los datos al cubo y en algunas ocasiones sirven como aplicación a los usuarios finales, es esencial que trabaje con la estrategia de metadatos que has elegido.

Herramientas populares

- Business Objects
- Cognos
- Hyperion
- Microsoft Analysis Services
- MicroStrategy
- Pentaho
- Palo OLAP Server

2.5.6. Herramientas de reporteo

Existe una gran variedad de requerimientos de reporteo, y construir o comprar una herramienta de reporteo para nuestras necesidades de inteligencia de negocios depende en gran parte de nuestros requerimientos. Normalmente, la decisión se basa en lo siguiente:

- *Numero de reportes:* Entre más grande sea el número de reportes que necesitemos, más viable es la opción de comprar una herramienta. Esto no es debido a que las herramientas de reporte nos ayuden a crear nuevos reportes fácilmente (ofreciendo componentes de re-uso), pero estos cuentan con herramientas de administración de reportes y funciones de fácil mantenimiento.
- *Forma de distribución del reporte deseada:* Si el reporte sólo será consumido desde una sola vía de acceso (por ejemplo, el correo o el navegador), entonces deberíamos de considerar fuertemente la opción de crear nuestro propio reporteador. Por otro lado, si los usuarios necesitan acceder a ellos por diferentes medios, entonces tiene sentido invertir en una herramienta de reporteo.
- *Creación de reportes Ad Hoc:* ¿Los usuarios son capaces de crear sus propios reportes a la medida? Si así es el caso, es buena idea comprar una herramienta de reporteo. Los fabricantes de este tipo de herramientas tiene una gran experiencia y conocimiento de las características que son importantes para los usuarios que

están creando los reportes. Una segunda razón es que para contar con la habilidad de poder crear reportes ad hoc es necesario contar con una fuerte capa de metadatos, y es algo difícil el contar con los metadatos cuando se está construyendo la herramienta desde cero.

Los datos no sirven de nada si lo único que hacen es residir en el repositorio. A causa de eso, la capa de presentación es de gran importancia. La mayoría de los fabricantes de herramientas OLAP cuentan con una capa de presentación que permiten a los usuarios las llamadas a reportes pre-definidos o la creación de reportes *ad hoc*. De cualquier forma, los siguientes puntos son importantes en la evaluación de la herramienta.

- *Capacidad de conexión hacia las fuentes de datos:* En general existen dos tipos de fuentes de datos, una son las bases de datos relacionales (RDBMS), y la otra son las fuentes multidimensionales OLAP. En la actualidad, las posibilidades que cuentes con ambas fuentes son grandes. Muchos fabricantes ofrecen en sus herramientas la posible conexión a ambas, pero revisando la herramienta a más detalle, es posible que solo sea buena para un tipo de fuente de datos, y el hacerlo para otra sea algo difícil a la hora de programar.

- *Capacidad de distribución y programación de corridas de reportes:* En los escenarios reales de repositorios usados por ejecutivos de alto rango, todos tienen como costumbre de inicio de semana, revisar los números más importantes de la semana pasada (digamos el numero de ventas), y así es como ellos satisfacen sus necesidades de inteligencia de negocios. Toda la deslumbrante funcionalidad de reportes con detalle y ad hoc no es importante para ellos, porque ellos no hacen mucho en realidad con estas características. Basados en ese escenario, la herramienta debe de contar con capacidad de distribución y programación o calendarización para la corrida de estos. Los reportes semanales se programan para que corran el lunes en la mañana, y los resultados de los reportes son distribuidos a los ejecutivos vía email o su publicación en la web. Existen fabricantes que promocionan que sus herramientas pueden distribuir los reportes mediante varias formas, pero realmente los que importan para los ejecutivos son el email y la publicación de estos vía internet.

- *Características de seguridad:* Debido a que las herramientas de reporteo son similares al caso de las OLAP, dado que llegan a un gran número de usuarios, el asegurarse que los usuarios sólo puedan ver lo que se supone deben ver es muy importante. La seguridad puede residir a nivel de reporte, carpeta, columna, registro y hasta celda. La mayoría de las herramientas establecidas cuentas con estas características. Además está el caso de las grandes empresas que cuentas con sus propios protocolos de autenticación mencionado en las herramientas OLAP.

- *Hechos a la medida:* Cada uno de nosotros que nos dedicamos a este trabajo, hemos tenido dolores de cabeza alguna vez al pasar bastante tiempo tratando de dar formato a un reporte para que luzca bonito. Esto es en definitiva pérdida de tiempo, pero desafortunadamente es un mal necesario. De hecho, muchas veces, los analistas desean tomar el reporte tal cual y agregarlo a sus presentaciones o reportes que entregan a sus jefes. Si el reporte les ofrece una fácil manera de pre-establecer de alguna forma la manera en que se quiere hacer el copiado y pegado para que se ajuste al estándar de la empresa, hiciera el trabajo de los analistas mucho más fácil y el ahorro de tiempo sería enorme.

- *Capacidad de exportación:* Las necesidades más comunes de exportación de datos son a archivos de Excel, planos y PDF, y una buena herramienta debe de exportar a estos tres tipos de formato. Para Excel, si la situación lo amerita, es bueno verificar que el formato también se exporte y no solo los datos, debido a que esto nos puede ahorrar una gran cantidad de tiempo.

- *Integración con Microsoft Office:* La mayoría de las personas está familiarizada o usan esto producto, especialmente Excel, para la manipulación de datos. Antes, las personas estaban acostumbradas a exportar los datos a Excel y a partir de ahí realizar labores de formateo y calculo de datos. Ahora muchas herramientas de reporteo ofrecen ambientes parecidos a office para la edición de los usuarios, así que todo el formateo puede realizarse con el reporteador en sí, sin la necesidad de exportar a Excel. Esto es agradable y conveniente para los usuarios.

Herramientas populares

- Business Objects (Crystal Reports)
- Cognos
- Actuate

2.5.7. Herramientas para metadatos

Solo en casos más raros tiene sentido construir una herramienta para metadatos desde cero. Esto es debido a que el hacer esto requiere de recursos que están íntimamente ligados a la operación técnica, y aspectos de negocio del repositorio, además que estos recursos son difíciles de conseguir. Aun cuando estos recursos están disponibles, muy a menudo existen otras tareas de mayor valor para la organización que construir una herramienta para los metadatos.

En efecto, la pregunta es cuando cualquier tipo de herramienta de metadatos es necesitada en su totalidad. Aunque los metadatos juegan un papel muy importante

en la implementación del repositorio, esto no quiere decir que siempre se necesita una herramienta para tener todos los *datos sobre los datos*. Es posible decir que esta información puede estar en un documento de texto, una presentación o hasta en un hoja de cálculo.

Al decir esto, algunos autores creen que el contar con una sólida base de metadatos es una de las claves para que nuestro repositorio tenga éxito. Aun cuando la herramienta de metadatos no sea seleccionada desde el inicio del proyecto, es esencial contar con una estrategia de metadatos, estos es, como los metadatos serán almacenados en el repositorio.

Sin duda esta es la herramienta más difícil de elegir, porque claramente no sigue un estándar. De hecho, esta sección debería ser llamada una estrategia para los metadatos. Tradicionalmente, la gente ha puesto la información del modelado de datos en herramientas como ERWin y Oracle Designer, pero es difícil extraer de estas herramientas la información una vez ingresada. Por ejemplo, una de las metas de nuestra selección de metadatos es el proveer información a los usuarios finales. Claramente esto es una tarea difícil con las herramientas de modelado de datos.

Así que, lo típico es que se hagan esfuerzos extras para crear una capa de metadatos que sea de uso para los usuarios finales. Mientas esto provee a los usuarios del conocimiento necesario para saber que significan los datos y los reportes, es claramente ineficiente debido a que esta información actualmente reside en las herramientas ETL, OLAP, reporteo o modelado de datos. Existen esfuerzos por parte de los fabricantes de repositorios para unificar el modelado de metadatos, en junio del 2000 la OMG lanzó un estándar de metadatos llamado CWM (Common Warehouse Metamodel), y algunos fabricantes como Oracle lo han implementado. Este estándar incluye lo último en tecnología como XML, UML y SOAP, y si se acepta de manera mundial, es claramente lo mejor que le podría pasar a la industria de los repositorios.

Dado esto, ¿qué significa esto para nuestro esfuerzo en la elaboración de nuestros metadatos? En ausencia de cualquier cosa, es recomendable que cualquier herramienta que elijamos soporte XML, y cualquier herramienta que vaya a usar los metadatos de igual manera lo soporte. Al mismo tiempo tenemos que no debemos de preocuparnos por criterios que son importantes para otras herramientas como el paralelismo o el performance dado que la cantidad de datos es relativamente pequeña comparada con lo que almacena o utiliza el repositorio en sí.

2.5.8. Consideraciones de diseño y performance de base de datos

Debido a que nuestro ambiente será un ambiente ROLAP donde las consultas corren sobre la base de datos relacional, el performance de las consultas puede ser un problema. Un estudio muestra que el usuario pierde interés en lo que está haciendo

si el reporte tarda más de 30 segundos en correr, lo cual en estos ambientes es muy común. Por lo cual también es importante que se tome algún tiempo ideando que se puede hacer para solucionar esto, mínimamente en los reportes de más demanda. Para lo cual seguiremos las siguientes reglas en los reportes de mayor demanda y duración.

Para cualquier base de datos en producción, el performance en las consultas SQL serán un problema tarde o temprano. El tener consultas de larga duración no solo consumen recursos del sistema que hacen que el servidor y la aplicación corren lentamente, también lleva a bloqueos de tablas y corrupción de los datos. Así que, el performance de las consultas es una tarea importante. Para esto ofrecemos algunos consejos para la optimización de las consultas[13]:

1. *Entender como la base de datos está ejecutando las consultas:* Hoy en día todas las bases de datos tienen su propio optimizador de consultas, y ofrecen una forma grafica a los usuarios para comprender como se está ejecutando la consulta. Por ejemplo, que índice de que tabla está siendo usado para ejecutar la consulta. El primer paso para optimizar las consultas es saber que está haciendo la base de datos. Existen diferentes comandos para poder ver esto dependiendo del servidor de base de datos. Por ejemplo en MySQL se puede utilizar "EXPLAIN [Consulta SQL]" para ver el plan de ejecución de la consulta.

2. *Obtener de la consulta la menor cantidad de datos posible:* Entre más cantidad de datos regrese la consulta, mayor la cantidad de recursos que necesita la base de datos para procesar y almacenar los datos, además de generar mayor cantidad de tráfico en la red. Por ejemplo, si solo necesitas obtener una columna de una tabla, no utilices "SELECT *"

3. *Almacenar resultados en tablas intermedias:* En ocasiones la lógica de las consultas puede ser algo complicado. A menudo, para poder obtener un resultado deseado es necesario utilizar sub consultas, vistas e instrucción de "UNION". Para estos casos, los resultados intermedios no son guardados en la base de datos, si no que son usados inmediatamente por la consulta en cuestión. Esto puede ocasionar resultados de performance, especialmente cuando estos resultados intermedios contienen una gran cantidad de registros.

   La forma de optimizar este tipo de consultas es el almacenar los datos intermedios en tablas temporales, y fragmentar la forma en que estaba escrita la consulta inicialmente en pequeñas sentencias con propósitos específicos y atacando partes especificas de la consulta. En muchos casos, se pueden construir índices en estas tablas temporales para incrementar el performance aun más. El hacer esto, agrega un poco de complejidad en el manejo de la consulta, pero el incremento en la velocidad de la consulta es algo que generalmente vale la pena.

A continuación se mencionan algunas estrategias de optimización:

- *Usar índices:* Usar un índice es la primer estrategia que se debe de usar para incrementar la velocidad en las consultas. De hecho, esta estrategia es tan importante que la optimización por medio de índices es algo discutida. Hay que tener mucho cuidado en no caer en el juego de crear índices en exceso en las tablas y más aun si estas son de gran tamaño.

- *Tablas de datos agregados:* Pre-popular las tablas con datos agrupados, es decir, a un más alto nivel de abstracción nos lleva a que menor cantidad de datos sea revisada.

- *Partir verticalmente:* Partir la tabla por columnas. Esta estrategia reduce la cantidad de datos que la consulta SQL necesita procesar.

- *Partir horizontalmente:* Partir la tabla por valores, es decir, registros en especifico. De igual forma que la estrategia anterior reduce la cantidad de datos que la consulta SQL necesita procesar.

- *Des normalizar:* El proceso de des normalizar consiste en combinar múltiples tablas en una sola. Esto incrementa la velocidad de la consulta debido a que menos conjunción de tablas es necesaria.

- *Configuración de servidor:* Cada servido tiene sus propios parámetros, y en ocasiones el afinar estos parámetros es necesaria para que el servidor pueda tomar y hacer uso de forma completa la capacidad de hardware. Lo cual nos lleva a incrementar la velocidad de las consultas.

2.6. Repositorios inteligentes

Los repositorios inteligentes están basados en los repositorios electrónicos (Data Warehouse), por eso es necesario tener claro el concepto, cual es su función y cómo están compuestos. Dado esto a continuación se da una pequeña introducción a los repositorios electrónicos.

2.6.1. Data warehouse

Es un repositorio de información de alguna empresa, institución u organización la cual se encuentra almacenada electrónicamente. Estos repositorios se encuentran diseñados para facilitar el reporteo y análisis de información[16].

La definición de repositorio electrónico se basa en el almacenamiento de datos. De cualquier forma, algunos aspectos como el análisis, extracción, transformación y carga

de los datos, así como también el manejo del diccionario de datos son componentes esenciales en un repositorio electrónico de información.

Una definición ampliada de los repositorios electrónicos incluye herramientas para la inteligencia de negocios, así como también de herramientas para la extracción, transformación y carga de los datos al repositorio, así como también como herramientas para el manejo y extracción de metadatos.

El concepto de repositorios electrónicos data desde los años 80's cuando los investigadores de IBM Barry Devlin y Paul Murphy desarrollaron el *business data warehouse*. En esencia el concepto intentaba proveer de una arquitectura computacional para el paso de la información de sistemas transaccionales a ambientes para la ayuda y soporte de decisiones. Este concepto intentaba atacar varios problemas asociados al paso de la información, principalmente el alto costo que representaba. A falta de una arquitectura para la creación de repositorios electrónicos, una gran cantidad de redundancia de información era requerida para poder dar soporte a los múltiples sistemas para la toma de decisiones. En grandes compañías era común que estos sistemas operaran de manera independiente, estos ambientes daban soporte a diferentes usuarios pero a menudo requerían de la misma información.

El proceso de obtener, limpiar e integrar la información hacia varios destinos, usualmente grandes sistemas, era comúnmente replicada a cada uno de ellos. Aun más complicado hacían este proceso, el que los sistemas transaccionales eran frecuentemente modificados de acuerdo a nuevos requerimientos de la organización, lo que provocaba el re trabajo.

Basados en analogías de cómo se encontraba la información almacenada en la vida real, los repositorios electrónicos pretendían ser a gran escala áreas de colección, almacenamiento y paso de información corporativa. Los datos podían ser obtenidos de un punto central o de varios puntos alternos como podrían ser puntos de venta de las organizaciones.

2.7. Estado del arte

A continuación se presentan una serie de herramientas existentes en el mercado, las cuales poseen características similares a este proyecto de tesis, o bien, comparten un enfoque similar en sus objetivos. No es la intención comparar este proyecto contra todas las herramientas existentes en el mercado, pero si, ubicar en qué contexto se encuentra este.

2.7.1. ChemSpider

Es un servicio gratuito que provee un acceso para la comunidad interesada en la estructura de los químicos. Brindando acceso a millones de estructuras químicas e

integración a una multitud de servicios en línea[6].

Cuenta con una rica variedad para la búsqueda de estructuras químicas, entre las cuales se destacan las búsquedas por estructura, por elementos químicos, por propiedades, por citas en la literatura, etc.

Una vez seleccionada la estructura deseada, la información que presenta con relación a ella, es la siguiente:

- Imagen con la estructura
- Propiedades moleculares
- Propiedades inherentes, identificadores y referencias
- Bases de datos asociadas y proveedores comerciales
- Patentes
- Artículos PubMed[7].
- Nombres y sinónimos
- Predicción de propiedades

2.7.2. PubChem

Posee un enfoque muy parecido a ChemSpider. PubChem provee información de las actividades biológicas de pequeñas moléculas. Es un componente del NIH[8]. Incluye información de sustancias, estructuras de compuestos, y bio-actividad de datos en 3 bases de datos (Pcsubstance, Pccompound, PCBioAssay)[5].

La base de datos de compuestos y sustancias, a medida de lo posible, ofrece enlaces a la descripción del bio-ensayo, la literatura, referencias, y el ensayo de puntos de datos. La información contenida en esta base de datos es de solo lectura para los visitantes.

La información obtenida al realizar la consulta es la siguiente:

- Información química y de drogas
- Literatura (referencias)
- Resultados de bio-ensayo

[7]Es un servicio de la librería nacional de medicina de los Estados Unidos que incluye alrededor de 18 millones de citas a revistas de publicaciones científicas que datan desde los años cincuentas. Incluye ligas y artículos completos, además de otros recursos.

[8]Instituto Nacional de la Salud del Departamento de Salud y Servicios Humanitarios de los Estados Unidos.

- Sinónimos
- Propiedades
- Información de compuestos
- Información de sustancias

2.7.3. Diccionario de productos naturales (DNP)

Es una base de datos estructurada la cual contiene información sobre las sustancias químicas. Incluye datos descriptivos y numéricos en química, fisca y en las propiedades biológicas de los compuestos; sistemática y nombres comunes de los compuestos, referencias bibliográficas, diagramas de estructura y de sus tablas de conexión asociadas[1].

Esta base de datos también cuenta con un enfoque parecido al de las bases de datos mencionadas con anterioridad, los datos presentados al realizar una consulta, son los siguientes:

- Nombre brindado por el diccionario de productos naturales
- Formula estructural
- Nombres alternativos
- Formula molecular
- Fuente de datos
- Referencias bibliográficas

2.7.4. Centro nacional para la medicina alternativa y complementaria (NCCAM)

Es un grupo promovido por NIH conformado por diversos sistemas médicos, de salud, prácticas y productos que no están considerados generalmente por parte de la medicina convencional[2]. Esta base de datos cuenta con una búsqueda limitada a un índice de registros. La información presentada al realizar una consulta es la siguiente:

- Introducción
- Nombres comunes y latinos
- ¿Para qué se usa?
- ¿Cómo se usa?

- Lo que la ciencia dice acerca de
- Efectos secundarios y precauciones
- Referencias o fuentes
- Ligas a sitios de internet con información acerca de

2.7.5. Servicio de conservación de recursos naturales (USDA)

Es parte de un programa del departamento de agricultura de los Estados Unidos, el cual tiene el propósito de conservar la información acerca de plantas en una base de datos. La búsqueda de plantas es en base al nombre científico, común y simbólico. Además, cuenta con una búsqueda avanzada por localización de la planta en el territorio estadounidense[3].

Es importante mencionar, que parte de la información de esta base de datos, puede ser actualiza por parte de los usuarios (ubicación e imágenes de las plantas). A continuación se presenta la información que presenta la base de datos con relación a una planta:

- Nombre científico y variedad
- Clasificación
- Fuente y documentación (referencias)
- Imágenes
- Sinónimos
- Distribución y localización territorial (EU)

2.7.6. SciFinder

Es una herramienta para la búsqueda de investigaciones que permite a alumnos y facultades el acceso a una gran variedad de investigaciones de varias disciplinas científicas, incluyendo ciencias biomédicas, química, ingeniería, ciencias de materiales, agricultura y más. La información de esta base de datos se basa en publicaciones científicas y patentes de alrededor del mundo[4].

Las búsquedas que proporciona son bastante completas y se basan en 3 grupos: Literatura, sustancias y reacciones. Los datos que proporciona la búsqueda son:

- Referencia bibliográfica

- Resumen
- Información de la patente
- Clasificación de la patente
- Liga hacia la publicación o la patente

Capítulo 3

Desarrollo experimental

A continuación se describe el proceso que se siguió para desarrollar este proyecto, las especificaciones técnicas y diagramas realizados, la experimentación y prueba del repositorio con información de investigaciones, publicaciones y patentes.

Como se mencionó en la sección donde se habla de ingeniería de software, no todos los proyectos o desarrollos cumplen con cada uno de los elementos con estricta rigurosidad, siendo este proyecto uno de los casos, pero si es importante mencionar en que etapa del desarrollo entrarían los pasos seguidos en nuestra metodología, la cual se da en seguida.

La identificación de necesidades recae desde la puesta en marcha de este proyecto y la creación del equipo de trabajo. El análisis de requerimientos está compuesto por la recopilación de requerimientos. En este proyecto no existió una etapa de especificación debido al corto tiempo y a la naturaleza del proyecto. La parte de arquitectura se encuentra plasmada en los componentes reutilizables y estratégicos, los diagramas de paquetes, clases y el modelado de datos. La programación se dividió en el proceso ETL, el desarrollo del Front-End (Aplicación final) y el desarrollo de reportes. La sección de pruebas está cubierta por el aseguramiento de calidad. La documentación por el manual de usuario y finalmente el mantenimiento por el mantenimiento, cambios y mejoras.

3.1. Definición de equipo de trabajo

Como punto de partida para la construcción del repositorio, se decidió crear una taxonomía de plantas que incluyera y diera cabida a todos los aspectos de investigación relevantes realizados en el instituto. Con esta finalidad se integró un grupo de doctores investigadores con conocimientos en áreas relacionadas como la biología, química, fitoquimica, etc. Este grupo estuvo conformado por el Dr. José Aldo Díaz Prado, Dr. Mario Moisés Álvarez, Dra. Elsa María Guajardo Touché y la MC. Elda Graciela Gómez López, a lo cuales en el proceso se aunaron el Dr. Sergio Serna Saldívar, el biólogo Gerónimo Cano y la Dra. Laila Pamela Partida Martínez, sin dar mayor o menor importancia al orden en que se nombran.

3.2. Elección de herramientas y software a utilizar

Una parte fundamental del proyecto es la plataforma a utilizar, así como los conocimientos técnicos requeridos para poder montar el repositorio en la plataforma elegida y que esta pueda brindar un puente para lograr los objetivos marcados en nuestro proyecto.

Para el desarrollo se cuentan con 2 ambientes, uno de desarrollo y otro de producción, el ambiente de desarrollo es donde se construye y se prueba el repositorio, siendo este una computadora portátil con un procesador 2.0 GHz de 64 bits, 4 MB en RAM y Windows 7 ultímate como sistema operativo. En el ambiente de producción se cuenta de un equipo de escritorio con 2.8 GHz, 3 MB de RAM y Windows 7 ultímate como sistema operativo.

La elección de servidor de *base de datos* es MySQL y está fundamentada de la siguiente manera. Como servidor de base de datos común, es una de las herramientas más populares del mercado por su performance, distintas estrategias de manejo de datos y el tipo de licenciamiento que en nuestro caso será gratuito.

Además MySQL cuenta con una estrategia de mercado para entrar en el mundo de los repositorios o data warehouses [26], la cual consta de:

- Brindar soporte a los casos más comunes de tipos de repositorios.
- Haciéndose socio de las compañías más importantes en el área de inteligencia de negocios.
- Ofreciendo un costo muy atractivo para las instalaciones que soportan los tipos más comunes de repositorios.

El punto de escalabilidad lo brindan los diferentes engines o cores [1] para el manejo de datos, los cuales podemos aplicar a conveniencia en la definición de tablas transaccionales y analíticas de información.

Finalmente MySQL está disponible para cada uno de los sistemas operativos existentes en el mercado, por lo cual hace la decisión fácil para la combinación de RDBMS/Hardware.

El software se construyó sobre el ambiente de desarrollo *eclipse*, el cual provee un gran soporte para el desarrollo de aplicaciones en distintos lenguajes y arquitecturas. Sobre eclipse se montó la estrategia cliente-servidor brindada por *Java* del lado del servidor y *FLEX* del lado del cliente. Comunicando estas 2 capas mediante servicios web embebidos en el marco de trabajo de FLEX.

[1] son los diferentes manejos o tartos a los datos que puede usar MySQL, algunos de estos son InnoDB o Archive, los cuales cuentan con características especificas como el manejo simultaneo de transacciones y la inserción rápida de registros respectivamente. Además estos engines pueden ser creados por los clientes mismos si es que necesitaran una característica especial en su manejo de datos.

El conjunto de herramientas y software elegido, provee las bases al proyecto para lograr una aplicación Web 2.0, la cual brinde al usuario una experiencia de fácil manejo y rapidez. Haciendo posible a través de él, el trabajo en equipo de los investigadores de manera remota, ingresando a la aplicación desde cualquier equipo para la consulta o modificación de los datos.

3.3. Recopilación de requerimientos y definición de taxonomía completa

Con base a los requerimientos de información y la experiencia profesional de cada uno de los involucrados en el equipo, especialmente del biólogo del equipo, se acordó manejar la taxonomía básica o clasificación de plantas que se muestra en la figura 3.1.

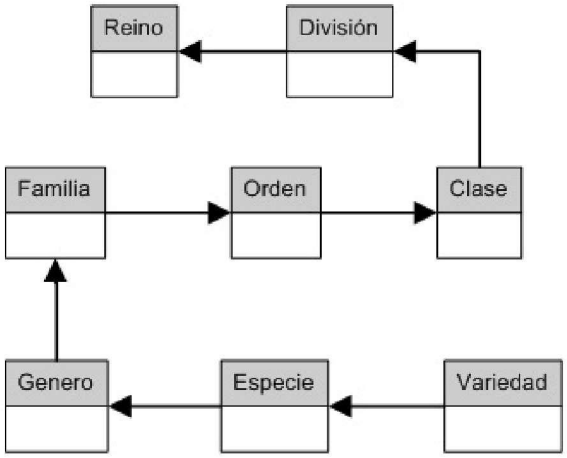

Figura 3.1: Clasificación botánica.

El diagrama que se presenta en la figura 3.1 se puede interpretar de la siguiente manera, una variedad pertenece a una especie en especifico y esa especie pertenece a un genero en particular, y en base a esa definición es como representamos las dependencias entre los elementos de la taxonomía creada, cabe resaltar que la *variedad* es el elemento principal y sobre el cual se construye y se modela este repositorio.

Como parte del modelo general, el diagrama de la figura 3.1 solo representa la clasificación taxonómica o botánica de la planta. El modelo completo comprende una clasificación enriquecida por los participantes del proyecto, la cual, contempla diversos elementos. Estos elementos son: formas de llamar a la misma planta en distintas partes del mundo, metabolitos asociados a la planta, micro organismos, usos y alivios a padecimientos, ubicación de las plantas en sus respectivos estados, hábitats y tipos de tierra

en las que se encuentran, así como también las referencias bibliográficas que avalan la existencia o relación entre ellos.

La figura 3.2 muestra un diagrama de cómo están relacionados los elementos mencionados anteriormente y en conjunto con el diagrama de la figura 3.1 representan el modelado conceptual de nuestro repositorio.

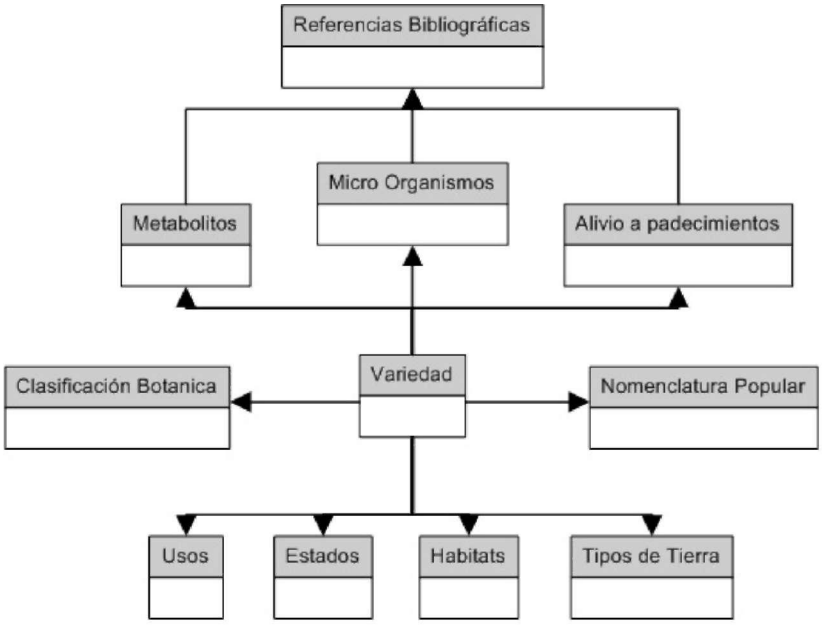

Figura 3.2: Taxonomía completa.

3.4. Comparativo entre repositorios existentes

En esta sección se hace un comparativo de los repositorios (bases de datos) existentes que tienen relación con este proyecto, en base a las funcionalidades de éstos y la funcionalidad que se pretende dar a este proyecto con la recopilación de requerimientos realizada. Es importante resaltar que este proyecto no intenta desbancar o superar a los mencionados en esta comparación, la intención es ubicar al lector de que es lo que contiene el repositorio, lo que éste puede hacer y sus limitantes.

A continuación se presenta la tabla 3.4, donde se hace un comparativo de las funcionalidades y enfoques que presentan algunos repositorios, de los cuales ya se hizo mención en el desarrollo teórico.

Como se puede ver en la tabla 3.4, los repositorios cumplen con características de acuerdo a su enfoque, aclarando que algunas características son más complejas o detalladas que otras y que no son todas las características que un repositorio de este enfoque puede tener. En este caso particular, se cumplen con un buen número de características y se definen los puntos débiles o necesidades de crecimiento del proyecto.

Repositorio / Característica	ChemSpider	PubChem	DNP	NCCAM	USDA	SciFinder	Badiano S21
Nombre científico, variedad					√		√
Clasificación botánica					√		√
Resumen o introducción		√	√	√		√	√
Imagen de estructura	√	√	√			√	
Propiedades moleculares	√		√				
Propiedades inherentes	√	√					
Patentes	√					√	√
Nombres y sinónimos	√	√	√	√	√		√
Identificadores en otras bases de datos	√		√				
Predicción de propiedades	√						
Información química y de drogas		√					
Literatura (referencias bibliográficas)		√	√	√	√	√	√
Resultados de bio-ensayo		√					
Información de compuestos	√	√					
Información de sustancias		√				√	√
Usos				√			√
Efectos secundarios y precauciones				√			
Ligas a sitios con información acerca de	√			√		√	
Imágenes					√		√
Distribución y localización territorial					√		√
Condiciones en la que se desarrolla							√
Posicionamiento global							√

Cuadro 3.1: Comparativo entre repositorios.

3.5. Construcción de componentes reutilizables y estratégicos

Entre los componentes reutilizables y estratégicos se cuenta con componentes del lado del cliente y del servidor, debido a la naturaleza de la aplicación web. Para explicar de manera más clara estos componentes, se inicia con la explicación de los componentes del lado del servidor, para lograr esto se hace uso de diagramas de clases (clases JAVA).

El diagrama que se muestra en la figura 3.3, es el diagrama de paquetes de componentes, en la cual se aprecia como el paquete base o principal se nombra *arquitectura*, brindando la posibilidad de en caso de emprender un nuevo proyecto, importar este paquete para tener la base arquitectónica para empezar a desarrollar.

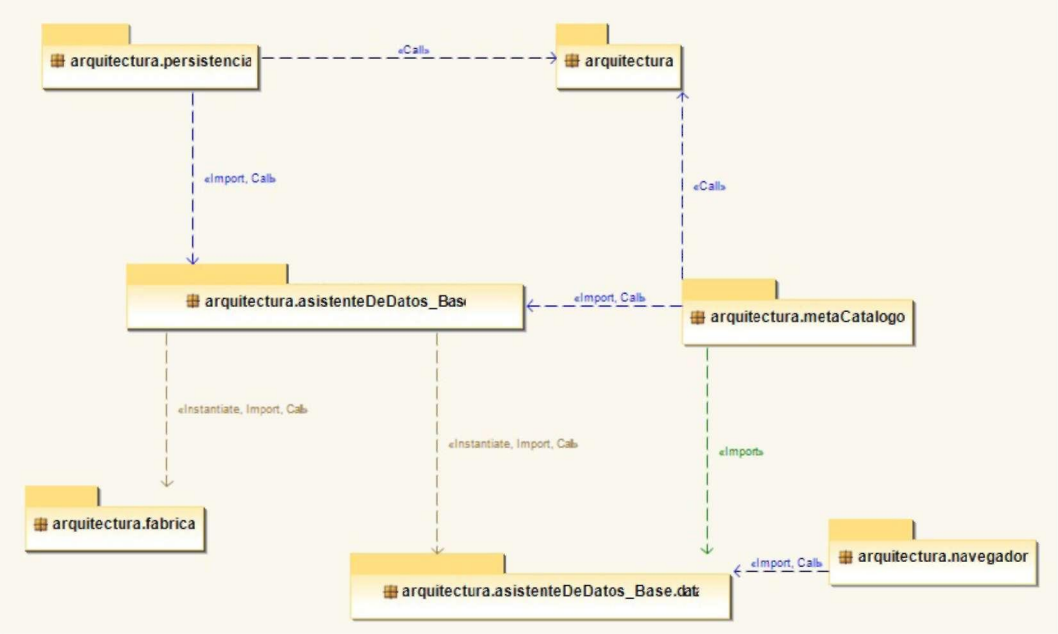

Figura 3.3: Diagrama de paquetes de arquitectura.

En primer lugar se tiene el componente de acceso a datos llamado *asistenteDeDatos_Base*, el cual contiene las clases para acceder a los datos, brindando una interfaz estándar, organizada y ágil para acceder a los datos sin importar la fuente u origen de estos.

El diagrama representado en la figura 3.4 muestra la estructura de las clases que contiene este paquete, es importante mencionar que la clase *AsistenteDeDatos_Base* implementa el patrón de diseño *Template Method* definiendo en una operación el esqueleto del acceso a datos estándar, delegando en las subclases algunos de sus pasos, esto permite que las subclases redefinan ciertos pasos del algoritmo sin cambiar su estructura. En el caso de esta implementación, lo que se intenta es ser flexible en cuanto a posibles nuevas fuentes de datos.

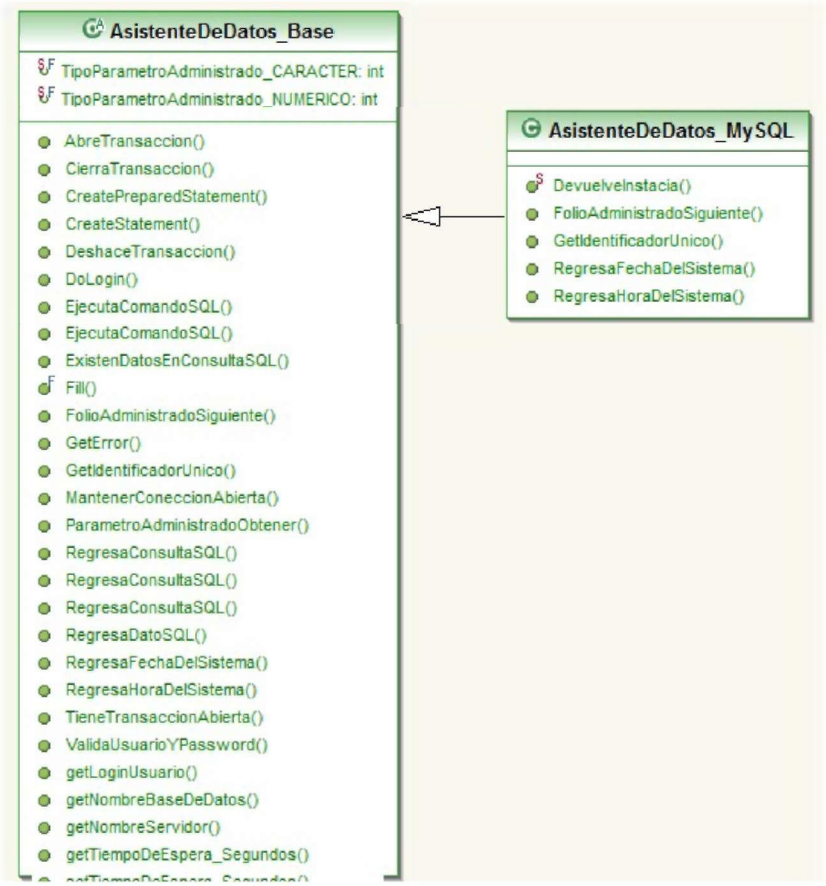

Figura 3.4: Diagrama de clases que incluye el paquete asistenteDeDatos_Base.

El paquete anterior se apoya del paquete *data* el cual tiene como propósito brindar una especie de base de datos en memoria, la cual cuenta con tablas, registros y columnas las cuales asemejan la estructura de una base de datos. Esta base de datos en memoria puede ser poblada por distintas fuentes o proveedores. La figura 3.5 muestra a detalle las clases y relaciones entre clases pertenecientes al paquete.

En seguida se tiene el paquete de *persistencia* el cual contiene las clases base, de las cuales heredaran las clases de nuestro sistema para proporcionar persistencia en la base de datos. El paquete *fábrica* cuenta con las clases que proporcionan la creación de instancias y el puente entre el lado del cliente y el servidor publicando los métodos necesarios de acceso de información.

Los paquetes *navegador* y *metaCatalogo* proporcionan las clases necesarias para la construcción del menú del sistema, validando los derechos y permisos sobre opciones para el primer caso. Para el segundo proporciona una estructura para definir los catálogos, que como su nombre lo dice, son datos acerca de los catálogos. Estos datos y la forma en que se utilizan serán explicados a mayor detalle más adelante.

De lado del cliente, se cuenta con el equivalente al paquete *data* en la estructura

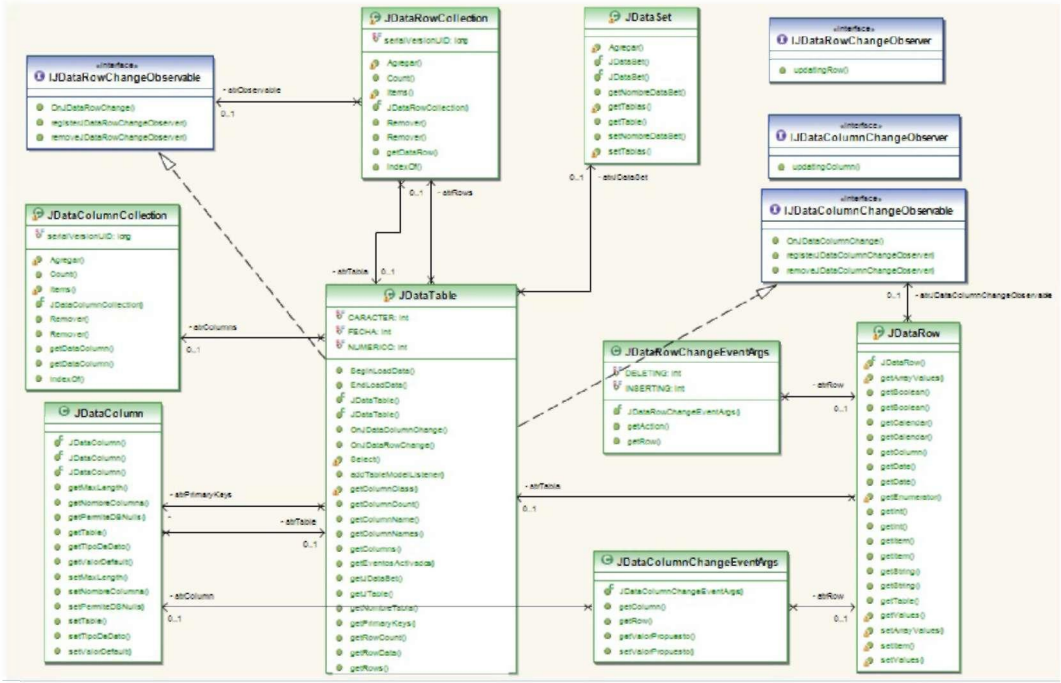

Figura 3.5: Diagrama de clases del paquete asistenteDeDatos_Base.data.

de clases de FLEX, esto para evitar el mapeo excesivo de clases, el cual nos obliga a realizar el paradigma de programación que implementa la comunicación del cliente con el servidor en FLEX. De esta forma, cuando es necesario hacer un acceso a datos del lado del cliente solo de consulta, no es necesario crear una clase del lado del cliente y otra equivalente del lado del servidor para esta tarea. Simplemente se obtienen los datos a través de objetos de tipo *JDataSet* o *JDataTable*.

Ahora es tiempo de hablar de los *MetaCatalogos*, que son, para que sirven y cuando los utilizamos. De inicio como el nombre lo dice, los MetaCatalogos son datos acerca de los catálogos, esto es decir, indican la tabla correspondiente al catálogo, una vista o estructura que define un acceso rápido y entendible para el usuario final, los campos primarios y de búsqueda para cada una de ellas, además de criterios de acceso y muestra al usuario de datos. Nos sirven para contar con una presentación y búsqueda rápida de datos para el usuario en un catálogo determinado, además de proporcionar el rehusó de estos en diferentes opciones del sistema con un comportamiento estándar. Finalmente los utilizamos en cada una de las opciones del sistema donde es necesario proporcionar un valor de un catálogo determinado, agilizando de gran manera el proceso de programación.

La figura 3.6 muestra la opción de búsqueda de un elemento en un catalogo cualquiera, como se puede ver en la parte superior izquierda, se encuentra la elección de la columna por la cual se realizará la búsqueda, en seguida de este se encuentra el criterio de búsqueda. Es importante mencionar que de acuerdo al tipo de dato de

la columna es el criterio que se introduce, ya sea tipo texto, numérico o un rango de fechas. Mas a la derecha se tiene el botón de búsqueda el cual al dar un clic sobre él se ejecutará la búsqueda con el criterio proporcionado, existen MetaCatalogos los cuales de inicio muestran todos los elementos del catalogo, esto para el caso de los catálogos que se sabe de antemano tendrán pocos elementos. Finalmente una vez hecha la búsqueda, los elementos que cumplen el criterio son mostrados para la elección del correspondiente por parte del usuario.

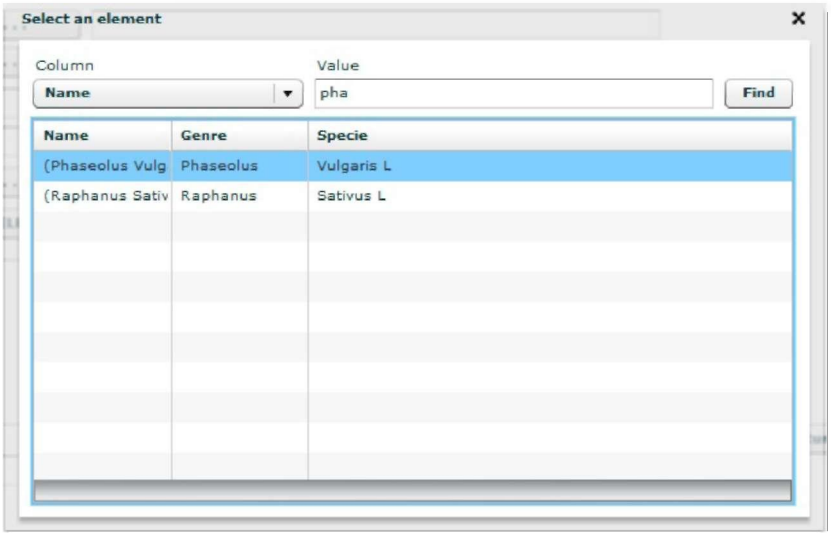

Figura 3.6: Interfaz para la selección de un elemento de catalogo.

3.5.1. Diagrama de paquetes y clases

A continuación se presenta el diagrama de paquetes del proyecto, estos paquetes comprenden las clases escritas en JAVA que se encuentran del lado del servidor, la figura 3.7 muestra el diagrama paquetes. Para este diagrama solo falta explicar los paquetes que pertenecen al paquete *badianoS21* los cuales corresponden al diseño y lógica del proyecto en sí. Los paquetes *fabricas* y *escribanos* contienen clases para la consulta y persistencia de los objetos del proyecto respectivamente. El paquete *referencias* contiene las clases correspondientes a lo relacionado con las referencias bibliográficas que maneja la aplicación. El paquete *catalogos* contiene las clases relacionadas a los catálogos como su nombre lo dice. El paquete *común* contiene clases que son utilizadas por los demás paquetes y por lo general contienen funciones publicas que son utilizadas en todo el proyecto. Finalmente los paquetes *mapping* y *persistencia* contienen la definición de un patrón de diseño *Template Method* para mapear y almacenar una clase con su correspondiente tabla en el sistema y la definición de las clases que implementan ese patrón respectivamente.

No se presenta un diagrama de clases general, porque la explicación redundaría con la dada más adelante en el modelo de datos.

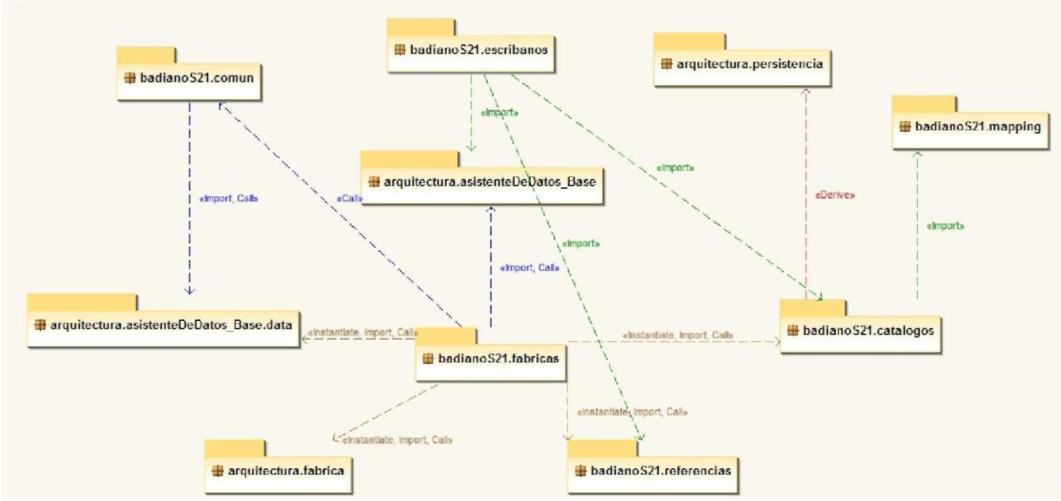

Figura 3.7: Diagrama de paquetes del proyecto BadianoS1.

3.6. Modelado de datos

A continuación se describe el modelo entidad-relación o modelo físico, pero antes de iniciar con la explicación del modelo, es importante señalar que para la nomenclatura de nombres de campos, se usa una especial, la cual brinda una fácil comprensión del tipo de dato que maneja el campo, y además establece un estándar para la declaración de nombres de campos.

La nomenclatura que se usa agrega al inicio del nombre del campo un prefijo que indica el tipo de dato de este, como ejemplo, si se desea nombrar a un campo "nombreCompleto" el cual será de tipo carácter variable o fijo, le añade el carácter "c" al principio, quedando el nombre del campo de la siguiente manera "cNombreCompleto".

La tabla 3.6 define los distintos tipos de prefijos que se utilizan para la nomenclatura de atributos.

Una vez definida la nomenclatura anterior, se da pie a la descripción del modelo completo, el cual es algo extenso, debido a esto, se separa en diagramas de acuerdo al contexto, los cuales son:

- Clasificación botánica.
- Referencias bibliográficas.
- Taxonomía completa.
- De sistema (seguridad e internas).

Prefijo	Descripción
c	Tipos de datos carácter variable o fijo en cualquiera de sus variantes.
n	Tipo de datos numéricos con precisión o sin ella.
b	Tipos de datos booleanos (si o no).
i	Tipos de datos que representan una imagen.
o	Tipos de datos que pueden almacenar distintos tipos o con propósitos variables.

Cuadro 3.2: Prefijos para la nomenclatura de atributos.

3.6.1. Clasificación botánica

La figura 3.8 muestra el diagrama de la clasificación botánica, y se dará una explicación de ella, porque es redundante con la explicación que se dio en la figura 3.1.

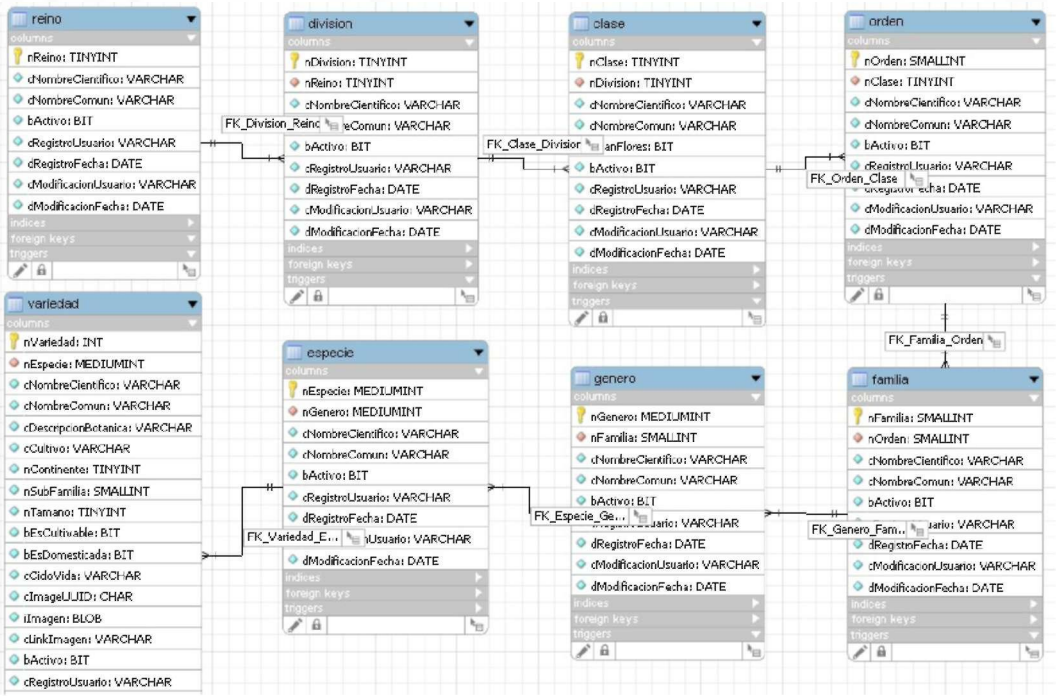

Figura 3.8: Modelo físico de clasificación de plantas.

3.6.2. Referencias bibliográficas

A continuación se presenta la figura 3.9, la cual muestra el diagrama para el control de las referencias bibliográficas.

Lo que se plasma en el diagrama de la figura 3.9 es la estructura que almacenara los datos relacionados con las referencias bibliográficas, la cual, básicamente se

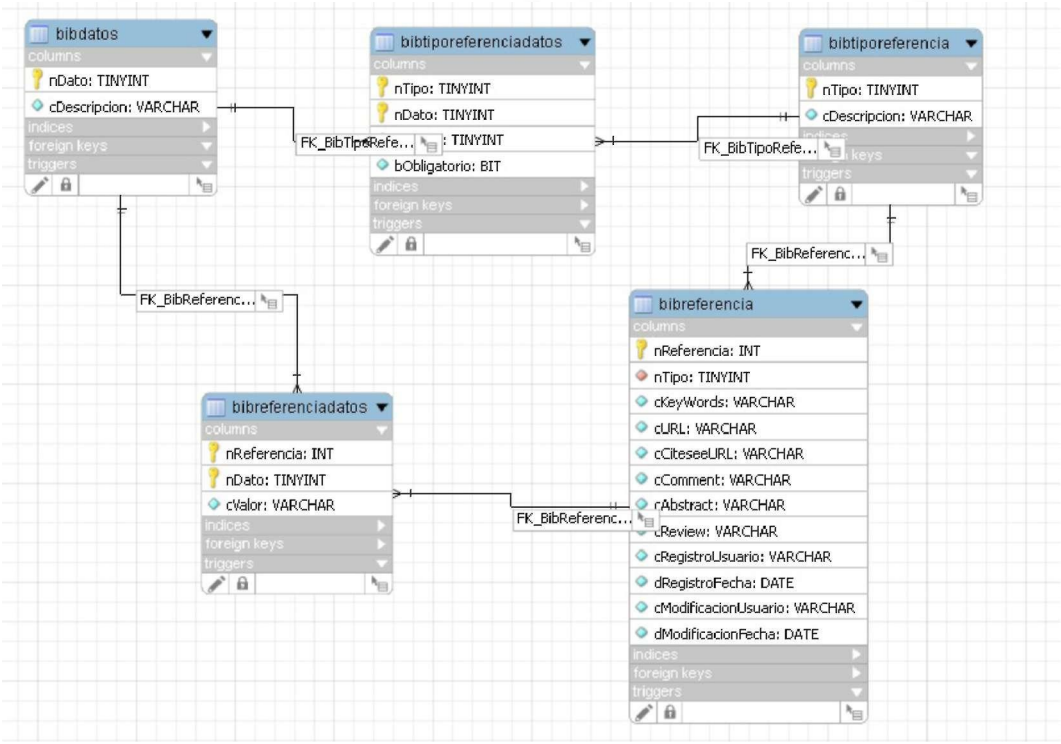

Figura 3.9: Modelo físico de referencias bibliográficas.

tomo del estándar de archivos .bib. Y se organiza de la siguiente manera, la tabla *BibTipoReferencia* define los diferentes tipos de referencias bibliográficas que existen, la cuales podrían ser, publicaciones científicas, tesis de maestría o doctorado y patentes por mencionar algunas. La tabla *BibDatos* define los metadatos posibles de captura, un ejemplo seria el autor, la editorial, el año, etc. La tabla *BibTipoReferenciaDatos* define los datos a capturar por cada tipo de referencia.

Las tablas *BibReferencia* y *BibReferenciaDatos* son en las que se almacena las definiciones de fuentes bibliográficas, basándose en las tablas anteriormente mencionadas y respetando las reglas definidas de captura en ellas.

Como se puede observar, la definición de tablas cuenta con un diseño normalizado [2]aunado a un diseño escalable, donde no es necesario el cambiar la estructura de las tablas en caso de contar con un nuevo tipo de captura o definición de esta. Esto se logra, representando cada valor de la referencia, en un registro mapeado a un metadato que nos indica que representa el valor.

[2]El proceso de normalización de bases de datos, consiste en aplicar una serie de reglas a las relaciones obtenidas tras el paso del modelo entidad-relación al modelo relacional. Esto con el propósito de evitar la redundancia de los datos, evitar problemas de actualización de los datos en las tablas y proteger la integridad de los datos [20].

3.6.3. Taxonomía completa

A continuación se muestra un diagrama completo, simplificando en las partes relacionadas con la clasificación botánica y las referencias bibliográficas vistas anteriormente. El diagrama se presenta en la figura 3.10.

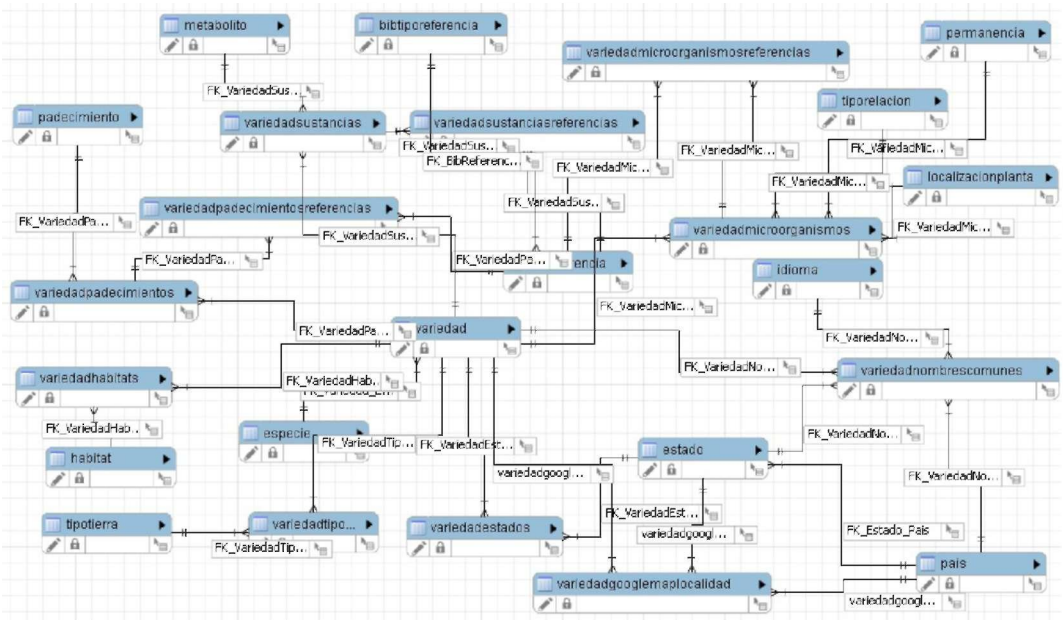

Figura 3.10: Modelo físico de la taxonomía completa.

El diagrama mostrado en la figura 3.10 es mucho más complicado que los diagramas mostrados con anterioridad, pero para analizarlo, se puede tomar como entidad principal a *Variedad*. A partir de ahí se puede ver como se representan las relaciones de acuerdo al diagrama mostrado en la figura 3.2, por ejemplo la tabla *variedadpadecimientos* representa la relación de la planta con la cura de padecimientos, en una relación de uno a muchos [3], es decir, que una planta puede o no estar asociada a la cura de uno o muchos padecimientos. Por otro lado la tabla *variedadmicroorganismos* representa la relación de la planta con ciertos Micro Organismos, igual con una relación de uno a muchos.

3.6.4. De sistema (seguridad e internas)

A continuación se presenta un diagrama que muestra las tablas y relaciones utilizadas de manera interna por el repositorio, para brindar una funcionalidad y escala-

[3]En el diagrama, la relación que llega a una entidad con una pata de gallo, representa que la entidad del otro lado de la relación puede aparecer muchas veces en ella. Éste es el tipo de relación más común, pero también existen otros como de cero a muchos, o de uno a uno.

bilidad de acuerdo a las necesidades de los usuarios y administradores. La figura 3.11 muestra el diagrama de las tablas de sistema.

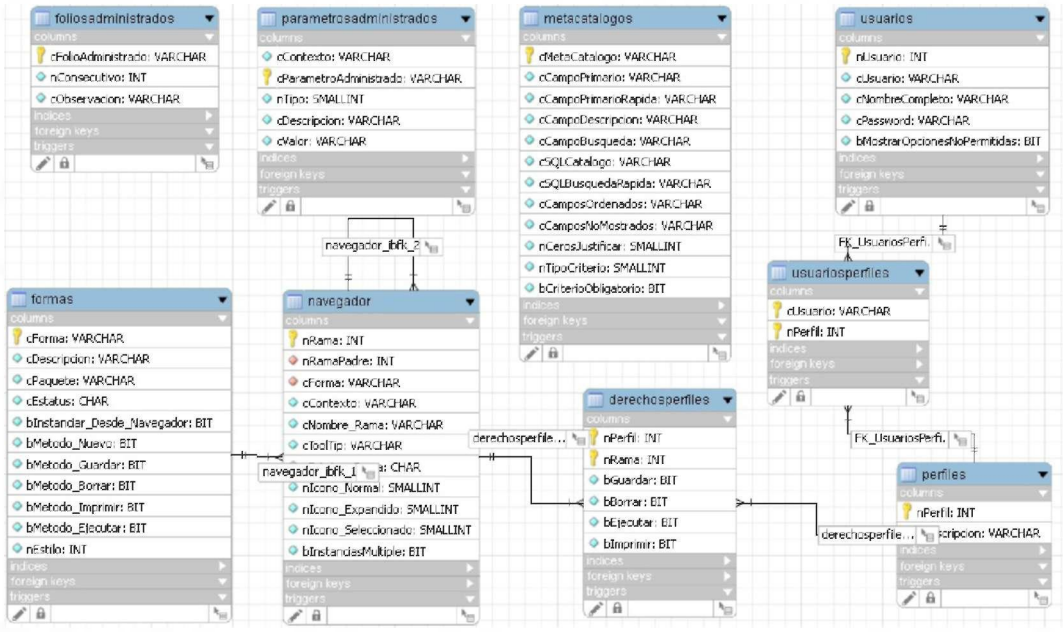

Figura 3.11: Modelo físico de la seguridad e internas.

A continuación se da inicio a explicar cada una de las tablas del diagrama, y lo que representan en la arquitectura del repositorio. La tabla *foliosadministrados* lleva el consecutivo a asignar en las tablas del sistema, es decir, al querer insertar un registro con un identificador o consecutivo en cualquier tabla del repositorio, el sistema solicita el siguiente folio a esta tabla.

La tabla *parametrosadministrados* tiene como propósito almacenar valores de parámetros del sistema, con la intención de poder cambiar cierta funcionalidad del sistema sin necesidad de recompilar y volver a generar versión con el cambio de la aplicación.

En la tabla *metacatalogos* se define la estructura de un rápido, fácil, reusable y entendible acceso a los datos para el usuario de ciertas entidades del sistema, esto es decir, existen columnas de las tablas las cuales contienen información que el usuario no necesita o no debe de ver, dado esto, en esta definimos todas esas reglas o necesidades de acceso a datos.

La tabla *usuarios* almacena todos los usuarios del sistema (la contraseña del usuario se encuentra almacenada encriptada en la tabla). La tabla *perfiles* actúa como agrupador de roles de usuarios del sistema, como ejemplo, en ella damos de alta el perfil administrador el cual se desea tenga privilegios sobre todas las opciones del sistema. La tabla *usuariosperfiles* relaciona a los usuarios con los perfiles. Por último, con relación a estas tablas de derechos se cuenta con la tabla *derechosperfiles*, la cual mapea un

perfil con una rama en el árbol del menú (y sus correspondientes hijos) a la cual tiene derecho el perfil.

Finalmente las tablas *formas* y *navegador* definen las posibles opciones del sistema y definen un orden jerárquico de visualización en el menú respectivamente.

3.7. Extracción, transformación y carga (ETL)

Debido a que desde el inicio del proyecto se tenía contemplado el análisis de información masivo, este proceso se resumió y quedó verdaderamente sencillo. Como primer punto para lograr esto, nuestra base de datos transaccional será la misma que sirva como repositorio, siendo este un repositorio transaccional ROLAP.

Como segundo y último punto en este proceso, se realizó una tarea de transformación de datos, donde el diseño normalizado transaccional pasa a ser una estructura con diseño no normalizado, integrando todas las entidades de la taxonomía creada. La figura 3.12 muestra un diagrama de cómo se hace la transformación de manera simplificada e ilustrativa.

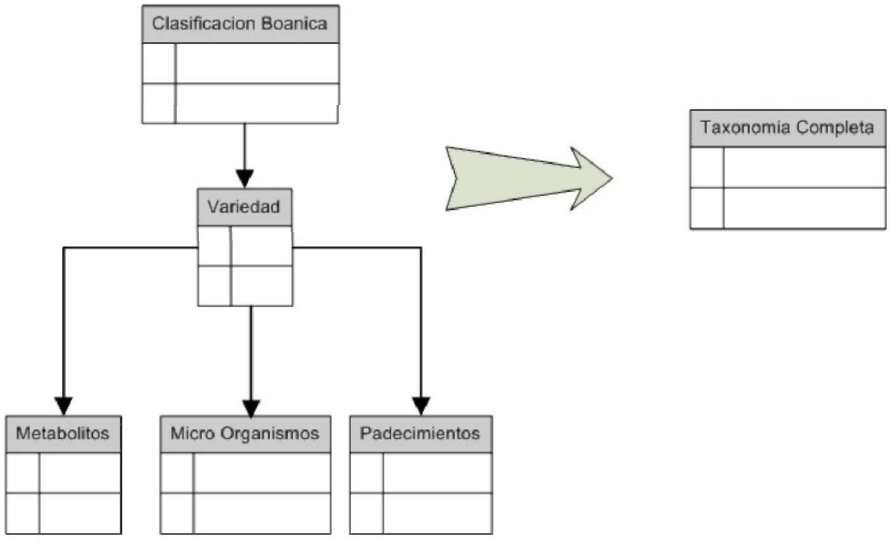

Figura 3.12: Proceso de ETL.

3.8. Desarrollo de aplicación final (front-end)

El desarrollo de la interfaz de usuario para el uso y explotación del repositorio está basado en el ambiente Web 2.0. Esto con la finalidad de brindar un acceso y explotación masiva de este, además de brindar al usuario una experiencia agradable,

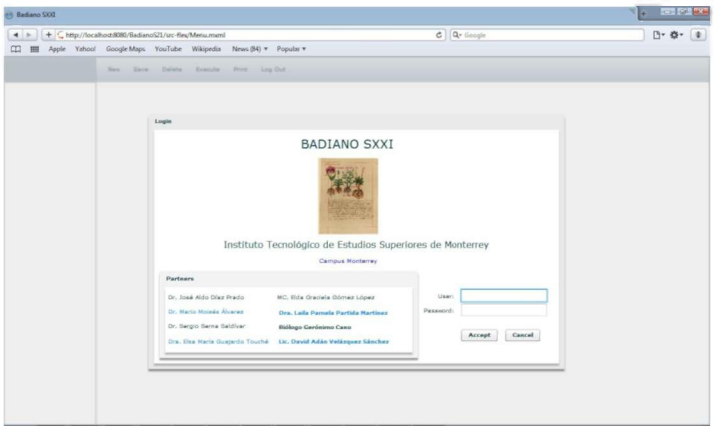

Figura 3.13: Interfaz de inicio al sistema.

confiable y rápida. El acceso al sistema se da a través de una autentificación de usuario y contraseña y la figura siguiente 3.13 ilustra esto.

Como muestra la figura 3.13 se cuenta con enlaces para mostrar perfiles de los colaboradores y participantes del proyecto, con una estructura como lo muestra la figura 3.14.

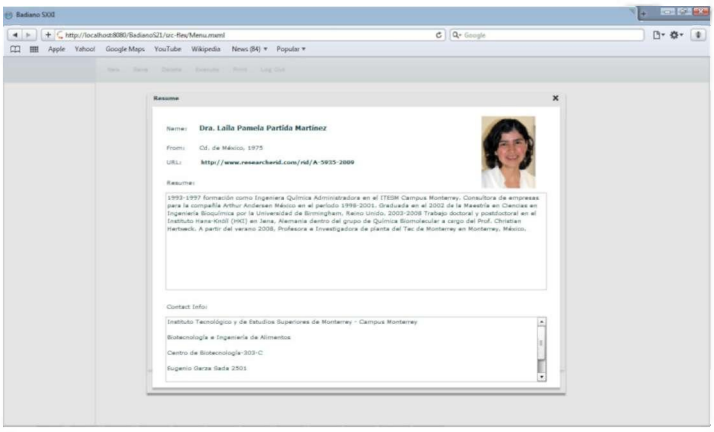

Figura 3.14: Muestra de colaborador currículo y datos personales.

Es importante mencionar que el idioma que se eligió para el Front-End de la aplicación es el idioma ingles, debido a que se intenta que este repositorio sea público y con un idioma estándar para la comunidad científica.

El menú principal es el que se muestra en la figura 3.15 el cual nos brinda la opción de poder trabajar con distintas opciones del sistema a la vez sin necesidad de hacer recargas a la pagina [4]. Mostrando las opciones en forma jerárquica o de árbol, además de permitir una interfaz de trabajo común para todas las opciones del sistema.

[4]Siendo este uno de los principales beneficios del Web 2.0

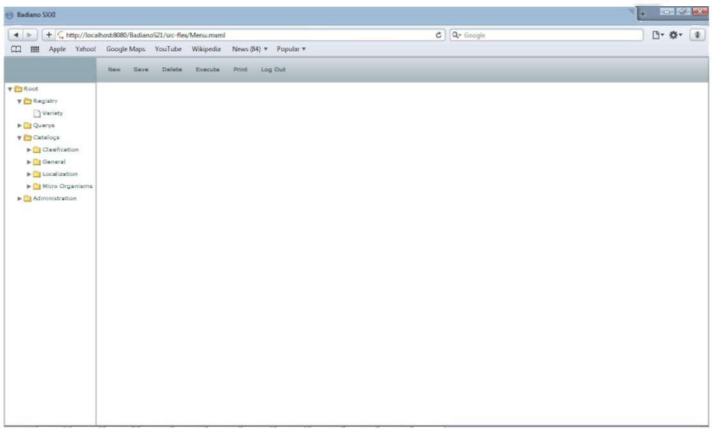

Figura 3.15: Menu del sistema.

La interfaz cuenta con las siguientes secciones para agrupar las opciones de esta:

- *Registros*: En esta sección se encuentran las opciones que definen a la variedad con todas las características definidas en nuestra taxonomía.
- *Consultas*: A partir de aquí se consulta y hacer uso a la información almacenada en el repositorio.
- *Catálogos*: Define los elementos por los cuales está compuesta la variedad y da la normalización que se necesita para la no redundancia de la información en el repositorio.
- *Administración*: Define usuarios y privilegios de uso del repositorio.

El diseño del registro de variedad, respeta la taxonomía creada y cumple con las especificaciones de esta, agrupando cada uno de los aspectos de la variedad en su respectiva sección, la siguiente es una figura 3.16 que muestra la organización de la misma.

3.8.1. Integración con Google Maps

El API de Google Maps para Flash proporciona una nueva forma de añadir mapas de Google interactivos a un sitio web mediante el complemento Flash de Adobe para la visualización de contenido dinámico. Esta API es una alternativa completamente independiente al API de Google Maps para JavaScript existente. Además de ofrecer muchas de las funciones de dicha API, permite combinar contenido Flash con Google Maps. En nuestro caso la localización de las plantas en una zona geográfica se vuelve de gran utilidad y valía, debido a algunos factores como escasez o peligro de extinción de una variedad, y el crecimiento urbano de las ciudades.

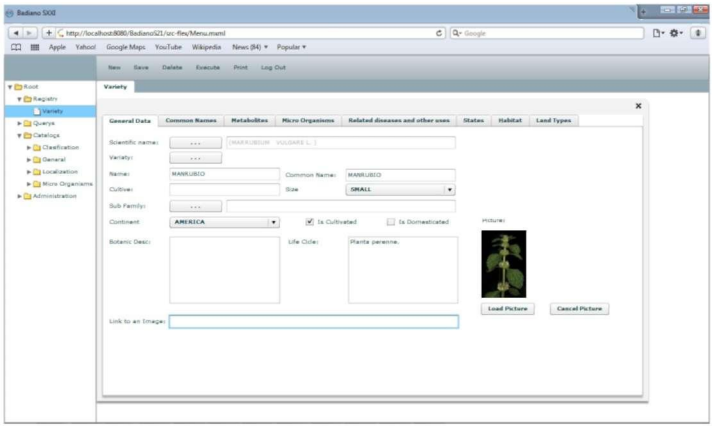

Figura 3.16: Opción de registro de variedad.

La búsqueda y localización de objetos y direcciones mediante los mapas que brinda Google se han vuelto muy populares y de gran utilidad en la actualidad, siendo una de las formas más eficaces de encontrar una dirección o tener una referencia de la topografía de una zona determinada.

Existen diversas variedades las cuales se dan sólo de manera no cultivada y encontrar yacimientos o brotes de estas se vuelve complicado, en ocasiones estos brotes que antes se encontraban en cierta zona han desaparecido por la construcción de un nuevo fraccionamiento sobre esta.

Siendo estos los principales factores se tomó la decisión de integrar la ubicación de las variedades sobre un mapa, con el objetivo de contar con una ubicación precisa, la cual brinde una rápida y segura fuente de donde se puede encontrar la variedad que se busca. Aunado a esto se tienen los datos o información que se puede obtener de la ubicación especifica, como características de suelo, clima, vegetación, etc. Enriqueciendo el conocimiento existente sobre la variedad y la región.

La localización de las variedades se da en 2 pasos en la aplicación y son los siguientes:

- Determinar la existencia de la variedad a nivel de estado.
- Proporcionar la ubicación de esta en el estado a nivel de detalle, ubicándola incluso hasta en una calle o dirección especifica.

La determinación de existencia a nivel de estado se da en el registro de variedades en la sección de estados. La ubicación a detalle y localización en el mapa de una variedad se da en la opción de localización como lo muestra la figura 3.17. Teniendo la opción de determinar a detalle la ubicación en un estado, con esto se puede localizar de manera exacta de la variedad a nivel de dirección, incluso proporcionando la latitud y longitud del punto deseado en caso de necesitar comparar la ubicación contra un GPS.

En el caso de que el mapa proporcionado por Google no defina la ubicación de alguna población, se cuenta con la opción de localización manual en la opción de registro de variedad. En la cual se proporciona el estado en la que se ubica y los datos acerca de la localización para que ésta pueda ser ubicada en el mapa.

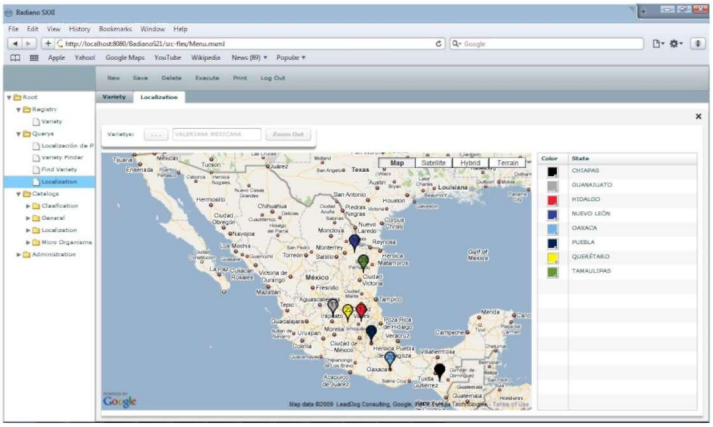

Figura 3.17: Localización en el mapa de una variedad.

3.9. Desarrollo de reportes

Para el desarrollo de reportes, se desarrolló una herramienta de reporteo, debido al poco tiempo que tiene la plataforma FLEX en el mercado y al costo que tienen las existentes. Existen algunas opciones entre los fabricantes con opción a prueba pero a final de cuentas todas ellas tienen un costo, por lo cual se optó por la decisión de construir un reporteador básico, del cual hablaremos un poco a continuación.

3.9.1. Herramienta de reporteo

Debido a la necesidad de reporteo y la falta de una herramienta gratuita para este proyecto, se construyó un herramienta que crea reportes de manera ágil y fácil en un archivo con extensión PDF, el cual puede ser mostrado en nuestra aplicación.

Los detalles que incluye el reporteo son:

- La estandarización en el formato de los reportes, como títulos, tipos de letra, imágenes o logos.
- Configuración y diseño.
- Trabajo con datos de diversas fuentes sin importar la fuente al trabajar directamente con las clases de manejo de datos definidas, estas clases semejan las

existentes en el espacio de nombres System.Data del framework .NET[5].

- La posible reutilización de esta herramienta en alguna otra aplicación.

Los reportes que incluirá el repositorio serán los siguientes y son descritos a continuación:

- Clasificación dinámica.
- Taxonomía de variedad.
- Búsqueda general ad-hoc.
- Referencias asociadas.

3.9.2. Clasificación dinámica

Se creó un árbol de clasificación en base a un objetivo a clasificar de tipo booleano o en el caso de ser nominal se define un valor nominal el cual será la muestra verdadera y para los valores nominales restantes la muestra es falsa. Además del atributo objetivo a clasificar, se definen un conjunto de atributos, los cuales serán parte de la clasificación.

El algoritmo utilizado para la implementación de la clasificación es el algoritmo ID3 y esta implementado parcialmente en java y en consultas SQL a la base de datos, esto con la intención de agilizar el proceso de cálculo, el algoritmo utilizado es el siguiente:

El cálculo del mejor elemento se hace en base a la *entropía* y esta entropía se basa en la entropía de Shannon, la cual se define a continuación:

$$H(x) = \sum_i p(x_i) \times log_2(1 - p(x_i)) \tag{3.1}$$

Donde x es la variable aleatoria discreta y los subíndices de x son los posibles valores de la variable.

3.9.3. Taxonomía de variedad

En este reporte se despliegan todas las propiedades relacionadas con la variedad, dejando a decisión del usuario la inclusión de los elementos que integran dichas propiedades.

[5]Es un componente de software que puede ser o es incluido en los sistemas operativos Microsoft Windows. Provee soluciones pre-codificadas para requerimientos comunes de los programas y gestiona la ejecución de programas escritos específicamente para este framework.

Algoritmo 1 Algoritmo de clasificación ID3 recursivo

```
1: if Todos los Ejemplos son positivos then
2:    Devolver un nodo positivo
3: else if Todos los Ejemplos son negativos then
4:    Devolver un nodo negativo
5: else if Atributos está vacío then
6:    Devolver el voto mayoritario del valor del atributo objetivo en Ejemplos  //
      Ejemplos: Conjunto de instancias a clasificar. Atributo-objetivo: Atributo objetivo
      a clasificar.
7: else
8:    Sea A Atributo el MEJOR de atributos  // Atributos: Conjunto de atributos a
      tomar en cuenta para la clasificación.
9:    for Para cada valor v del atributo hacer do
10:      Sea Ejemplos(v) el subconjunto de ejemplos cuyo valor de atributo A es v
11:      if Ejemplos(v) esta vacío then
12:         Devolver un nodo con el voto mayoritario del Atributo objetivo de Ejemplos
13:      else
14:         Devolver ID3(Ejemplos(v), Atributo-objetivo, Atributos/A)
15:      end if
16:   end for
17: end if
```

3.9.4. Búsqueda general ad-hoc

Reporte en el cual, el usuario elige los datos que quiere ver (propiedades de las variedades) con los filtros que él desea, por ejemplo, el usuario desea ver todos las variedades de plantas que se dan en el estado de Sinaloa. Además, cuenta con la opción de grabar una especie de formato de reporte, en el cual se definen los datos de este, para posteriormente solo definir el filtrado conveniente.

3.9.5. Referencias asociadas

Su objetivo es describir los datos de la referencia, así como también las variedades, padecimientos, metabolitos y microorganismos asociados a la referencia. Con la posibilidad de buscar referencias por cualquier atributo relacionado a esta.

3.10. Aseguramiento de calidad

Este se dará por parte del equipo de trabajo definido, tomando como ejemplo de funcionamiento la introducción de diversas variedades del frijol. Estos experimentos se encuentran descritos a continuación.

3.10.1. Experimentación y prueba de funcionamiento

La muestra experimental consta de las diversas variedades del frijol, debido a la variedad existente para esta planta, el conocimiento acerca de cura a padecimientos que posee y los estudios realizados en el instituto.

Otro punto importante para elegir a esta especie como base experimental, es que existen estudios en los cuales relacionan a esta con la cura del cáncer de colon (en ratas), ya que el cáncer en general es una de las enfermedades que causan el mayor número de muertes en nuestro país.

Entre las referencias tomadas para la carga de información se tienen las siguientes referencias de especies y variedades que curan algún tipo de cáncer:

- Composición y efectos quimiopreventivos de los polisacáridos del frijol común (*Phaseolus Vulgaris L.*) en el cáncer de colon inducido por azoximetano[10].
- Inhibición del crecimiento de las células cancerígenas por el extracto del frijol negro (*Phaseolus Vulgaris L.*)[15].
- Método de preparación de concentrados de la parte superior del rábano (*Raphanus Sativus L.*) con una mayor absorción de hierro, alto contenido de fibra dietética y una excelente actividad antitumoral en cáncer de hígado humano[18].

- Extractos de rábano japonés (*Raphanus Sativus L.*), salud alimenticia y medicinas para prevenir el cáncer[22].

Con relación a la cura de diversos padecimientos, se hizo uso de las siguientes referencias:

- (Diabetes, obesidad, hiperlipidemia y las enfermedades cardiovasculares) Método para la preparación de papillas para la prevención y el tratamiento dietético de la diabetes, obesidad, hiperlipidemia y las enfermedades cardiovasculares[32].
- (Diabetes y enfermedades cardiacas) El consumo del frijol pinto (*Phaseolus Vulgaris L.*) reduce los bio-marcadores en riesgos de enfermedades cardiacas y diabetes[34].
- (Hígado fibroso) Extracto de frijol negro (*Phaseolus Vulgaris L.*) aminora la fibrosis del hígado en ratas con lesiones inducidas por CCl4[23].
- (Uso cosmético) Composición cosmética que contienen péptidos que son preparados por la hidrólisis enzimática de frijol negro (*Phaseolus vulgaris L.*) germinado y o el arroz negro[8].
- (Antioxidante) Efecto de los procesos biológicos de la actividad antioxidante de las semillas de leguminosas seleccionadas, entre ellas el frijol rojo (*Phaseolus Vulgaris L.*)[14].
- (Antioxidante) Propiedades antioxidantes de los nuevos alimentos concentrados que contengan frijol rojo (*Phaseolus vulgaris L.*)[21].
- (Presión alta) Propiedades funcionales y de digestibilidad in vitro de la tripsina del aislado de proteína del frijol rojo (*Phaseolus Vulgaris L.*) Tiene efecto en el tratamiento de la presión alta[35].

Las variedades ingresadas al sistema son las que se muestran en la tabla 3.10.1, las cuales en conjunto con todas las propiedades relacionadas a estas, forman una base de 2960 instancias. Sobre este conjunto de instancias, a las cuales se le nombran como de entrenamiento, se corrió la clasificación dinámica, teniendo como resultado el árbol de clasificación que se muestra en la figura 3.18.

Si bien es cierto el número de variedades capturadas no es muy extenso, para la experimentación, se busca verificar la funcionalidad del sistema con ejemplos reales, a los cuales los avalen investigaciones científicas, el potencial que este tiene y podría tener en base al conocimiento que se le valla agregando con el paso del tiempo.

En la figura 3.18 se puede ver como en base a la información existente, se clasifican de manera automática los atributos que contribuyen para que una variedad cure o no el cáncer.

Especie	Variedad
(*Phaseolus Vulgaris L.*)	Black Bean
(*Phaseolus Vulgaris L.*)	Red Kidney Bean
(*Phaseolus Vulgaris L.*)	Northern Beans
(*Phaseolus Vulgaris L.*)	White Bean
(*Phaseolus Vulgaris L.*)	Yellow Bean
(*Phaseolus Vulgaris L.*)	Pinto Bean
(*Raphanus Sativus L.*)	Radish

Cuadro 3.3: Variedades de experimentación.

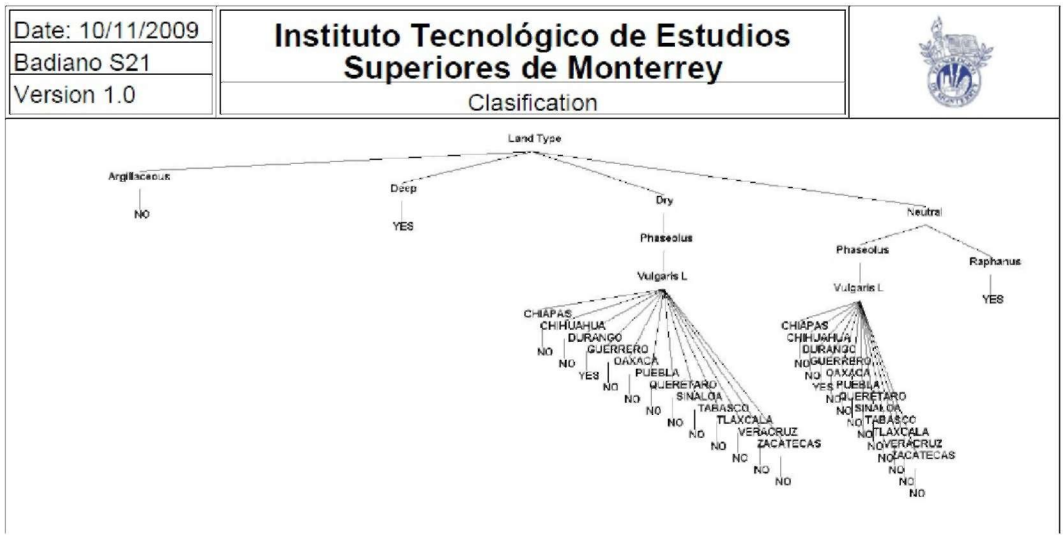

Figura 3.18: Árbol de clasificación con la cura del cáncer como objetivo.

Aunado a la experimentación con datos reales, se hicieron pruebas unitarias y de componentes durante el desarrollo de la aplicación, así como también se trabajó con un conjunto de 69 variedades las cuales tienen relacionadas curas a padecimientos, los cuales en su gran mayoría son remedios caseros o regionales. Estas variedades conforman un conjunto de 298561 instancias de prueba, las cuales son un número considerable para la pruebas de estrés y performance de la aplicación.

3.11. Puesta en producción, mantenimiento, cambios y mejoras.

Se dará una vez completado el proceso de aseguramiento de calidad y se hará sobre el ambiente de producción mencionado con anterioridad. Se dará un soporte a usuarios

mediante correo electrónico. Sobre cambios y mejoras se propone el siguiente esquema:

- *Cambios a la información del sistema*: Estos cambios pueden darse en cualquier momento por los usuarios con derechos a la opción. Por lo que se propone que exista un grupo de usuarios con ese derecho, los cuales analicen el cambio o agregado de información al sistema (Este grupo lo puede definir el administrador en el sistema).

- *Cambios y mejoras al software*: Lo cambios y mejoras serán analizados por parte del equipo que integra el proyecto para su aceptación. Una vez aceptada la modificación por parte del equipo, se escribirá un memo con la solicitud de cambio, para que esta pueda ser evaluada en cuanto a factibilidad, impacto y tiempo de desarrollo. Una vez revisado la solicitud de cambio con su resultado por el equipo de trabajo, se decidirá si se hace la modificación y quien será el responsable a realizarla (quedo en la mejor disposición de ayudar a guiar a realizar el cambio e inclusive, si se llega a un acuerdo, a realizarlo).

- *Continuación del proyecto por parte de terceros*: quedo en la mejor disposición a realizar una explicación conceptual y arquitectónica de la aplicación a la persona que vaya a continuar con el proyecto.

3.12. Manual de usuario

Consta de un formato ilustrativo de los diferentes tipos de opciones del sistema y que se puede hacer en cada uno de ellas. Se encuentra definido, detallado y disponible para su descarga en la barra de menú del repositorio.

Capítulo 4

Conclusiones

En este último capítulo se exponen las conclusiones a las cuales llega esta investigación, las contribuciones y el trabajo futuro.

4.1. Conclusiones

Se concluye que la taxonomía creada involucra las propiedades más importantes relacionadas a las plantas, las cuales en conjunto y con una gran cantidad de información fidedigna en el sistema, integran un conocimiento permite la aplicación de algoritmos y técnicas computacionales para la asociación de propiedades y atributos para su análisis. La arquitectura computacional seleccionada y creada para atacar este proyecto cumple con el objetivo de publicación, disponibilidad y preservación del conocimiento, aunado a la ubicación geográfica para una rápida ubicación de ejemplares.

4.2. Contribuciones

Se generó una herramienta para la administración del conocimiento de plantas medicinales mexicanas, la cual apoya al proceso de documentación y preservación de investigaciones, así como a la integración y publicación de información por parte de diversos grupos investigadores.

En el área de ingeniería de software, se creó una arquitectura computacional abierta para atacar diversos problemas de necesidad de información, en los cuales, las necesidades sean el acceso y disposición del sistema de manera mundial a través de internet, así como también, una experiencia agradable para el usuario en relación a rapidez y confiabilidad para el manejo de sus datos.

4.3. Trabajo futuro

La integración de una mayor cantidad de información al sistema (las plantas descritas por el Códice Badiano), para hacer experimentos y análisis de información que

guíen a seguir una línea de investigación para encontrar nuevos usos medicinales o fortalecer los ya conocidos sobre las plantas mexicanas.

Bibliografía

[1] Dictionary of natural products. Pagina web. http://dnp.chemnetbase.com/.

[2] National center for complementary and alternative medicine. Pagina web, 1998. http://nccam.nih.gov/.

[3] United states department of agriculture. Pagina web, 1998. http://plants.usda.gov/index.html.

[4] Scifinder. Pagina web, 2000. https://scifinder.cas.org/.

[5] Database of information on the biological activities of small molecules. Pagina web, 2005. http://pubchem.ncbi.nlm.nih.gov/.

[6] Database of chemical structures and property predictions. Pagina web, 2007. http://www.chemspider.com/.

[7] Don juan badiano y don martín de la cruz. Pagina web, 2009. http://es.wikipedia.org/wiki/Don_Juan_Badiano_y_don_Mart%C3%ADn_de_la_Cruz.

[8] J. T. Bae, T. B. Choi, J. H. Kim, B. C. Lee, D. H. Lee, U. I. Lee, H. B. Pyo, and E. J. Yoon. Cosmetic composition containing peptides which are prepared by enzymatic hydrolyzing germinated black bean and/or black rice, 2003:2004108226.

[9] A. Brown. Web service glossary. documento disponible en http://www.w3.org/TR/ws-gloss/, Junio 2002.

[10] A. A. Feregrino-Perez, L. C. Berumen, G. Garcia-Alcocer, R. G. Guevara-Gonzalez, M. Ramos-Gomez, R. Reynoso-Camacho, J. A. Acosta-Gallegos, and G. Loarca-Pina. Composition and chemopreventive effect of polysaccharides from common beans (*Phaseolus vulgaris L.*) on azoxymethane-induced colon cancer. *Journal of Agricultural and Food Chemistry*, 56(18):8737–8744, 2008.

[11] W. Frawley, G. Piatetsky-Shapiro, and C. Matheus. Knowledge discovery in databases: An overview. *AI Magazine*, pages 213–228, 1992.

[12] A. Garritz-Ruiz and J. A. Chamizo. *Del tequesquite al ADN. Algunas facetas de la química en México.* desconocido, 1997.

[13] P. Gulutzan and T. Pelzer. *SQL Performance Tuning.* Pearson Education, 2006.

[14] M. Gumienna, M. Czarnecka, and Z. Czarnecki. Effect of biological processes on the antioxidant activity of selected legume seeds. *Polish Pharmaceutical Society*, 41(3):553–557, 2008.

[15] J. A. Gutierrez-Uribe, S. R. O. Serna-Saldivar, J. E. Moreno-Cuevas, C. Hernandez-Brenes, and E. M. Guajardo-Touche. Cancer cell growth inhibition by black bean (*Phaseolus Vulgaris L.*) extracts, 2005:424757000.

[16] W. H. Inmon. *Tech Topic: What is a Data Warehouse?* Prism Solutions, 1995.

[17] W. H. Inmon. *Building the Data Warehouse.* WILEY, 2005.

[18] Y. J. Kim and K. S. Seong. Method of preparing radish tops concentrates having enhanced iron absorption, high contents of dietary fiber and excellent antitumor activity in human liver cancer, 2004:2006015997.

[19] R. Kimball, M. Ross, W. Thornthwaite, J. Mundy, and B. Becker. *The Data Warehouse Lifecycle Toolkit.* Wiley, 2008.

[20] D. Kroenke. *Procesamiento de bases de datos: fundamentos, diseño e implementación.* Pearson Education, 2003.

[21] M. Kulczak, K. Przygonski, M. Jezewska, I. Blasinska, and H. Luczak. Antioxidant properties of new food concentrates containing prepared red kidney beans. *Polish Pharmaceutical Society*, 41(3):293–297, 2008.

[22] H. Kumagaya. Japanese radish (*Raphanus sativus*) extracts as health foods and medicines for preventing cancer, 2004:2005206495.

[23] A. G. Lopez-Reyes, N. Arroyo-Curras, B. G. Cano, V. J. Lara-Diaz, G. E. Guajardo-Salinas, J. F. Islas, V. Morales-Oyarvide, L. A. Morales-Garza, F. J. Galvez-Gastelum, G. Grijalva, and J. E. Moreno-Cuevas. Black bean extract ameliorates liver fibrosis in rats with ccl4-induced injury. *Annals of hepatology : official journal of the Mexican Association of Hepatology*, 7(2):130–135, 2008. http://www.medigraphic.com/pdfs/hepato/ah-2008/ah082f.pdf.

[24] R. Martinez-Ortiz. Contribución al estudio fitoquímico de brickellia laciniata, traxinus greggi y cephalantus salicifolia. Master's thesis, ITESM campus Monterrey, 1986.

[25] P. McBreen. *Software craftsmanship: the new imperative.* Addison-Wesley, 2002.

[26] MySQL. Enterprise data warehousing with mysql. Business White Paper publicado MySQL, Diciembre 2007.

[27] T. O'Reilly. Web 2.0 compact definition: Trying again. documento disponible en http://radar.oreilly.com/archives/2006/12/web-20-compact-definition-tryi.html, Enero 2007.

[28] M. Rojo-Alba. Identificación de algunas plantas del códice badiano y sus nombres científicos y populares. documento disponible en http://www.tlahui.com/educa/comunidad/tesinas/aidenti.htm, 2002. Medicina Tradicional de México y sus Plantas Medicinales.

[29] J. Stelting. *Patrones de diseño aplicados a Java.* Pearson Educación, 2004.

[30] T. J. Teorey, S. S. Lightstone, and T.Ñadeau. *Database Modeling and Design: Logical Design.* Morgan Kaufmann, 2005.

[31] A. Téllez-Valero. Extracción de información con algoritmos de clasificación. Master's thesis, INADE, 2005.

[32] A. Wang and J. Zhao. Method for preparing composite porridge for prevention and dietary therapy of diabetes, obesity, hyperlipidemia, and cardiovascular diseases, 2007.

[33] A. Weitzenfeld. *Ingeniería de software orientada a objetos con UML, Java e Internet.* Cengage Learning Editores, 2005.

[34] D. M. Winham, A. M. Hutchins, and C. S. Johnston. Pinto bean consumption reduces biomarkers for heart disease risk. *American College of Nutrition*, 26(3):243–249, 2007.

[35] S. W. Yin, C. H. Tang, Q. B. Wen, X. Q. Yang, and L. Li. Functional properties and in vitro trypsin digestibility of red kidney bean (*Phaseolus vulgaris L.*) protein isolate: Effect of high-pressure treatment. *Food Chemistry*, 110(4):938–945, 2008.

[36] Y. Zhong, S. Jung, S. Pramanik, and J. H. Beaman. Data model and comparison and query methods for interacting classifications in a taxonomic database. *International Association for Plant Taxonomy (IAPT)*, 45:223–241, 1996.