Benemérita Universidad Autónoma de Puebla

Facultad de Ciencias de la Computación

Descubrimiento de patrones de comportamiento en datos urológicos utilizando Aprendizaje Automático

T E S I S

PARA OBTENER EL TÍTULO DE:

Ingeniero en Ciencias de la Computación

PRESENTA:

Josué Abraham López Sánchez

DIRECTORES DE TESIS:

Dra. María Josefa Somodevilla García

Dr. Ivo Humberto Pineda Somodevilla

Heróica Puebla de Zaragoza, Puebla, Septiembre 2020

La vida nos brinda oportunidades infinitas para reconocer a las personas que de una u otra manera influyen en nuestros proyectos, pero ninguna como la fortuna que hoy me embarga para agradecer a ustedes su contribución en mi desarrollo profesional.

A mis padres quiero expresarles mi gratitud, por los momentos que peldaño a peldaño contribuyeron a que hoy concluya una de las metas más importantes en mi vida, mi reconocimiento a su paciencia y dedicación para hacer la persona perseverante que han hecho de mí.

A mi asesora la Dra. María Josefa Somodevilla García, quien ha sido una persona clave en todo este proceso de formación profesional, quiero agradecerle las enseñanzas que me ha brindado, su conocimiento y su dirección, así mismo reconocerle sinceramente la acertada orientación que me permitió culminar exitosamente este proyecto.

A mi asesor el Dr. Ivo Pineda Somodevilla un agradecimiento especial por haberme confiado su trabajo, por su disposición y paciencia, virtudes que fueron imprescindibles para la realización de la presente tesis.

Resumen

El aprendizaje automático está siendo de gran utilidad en el entendimiento de fenómenos en diversas áreas del conocimiento, ya que, mediante técnicas matemáticas, una computadora puede procesar grandes cantidades de información para obtener conocimiento que resulte útil en la toma de decisiones.

El objetivo de la presente investigación es determinar y aplicar las técnicas de aprendizaje automático que resultan más útiles para obtener conocimiento en expedientes clínicos de la especialidad de Urología. En este contexto, el estudio aborda desde el diseño de la base de datos hasta el proceso de análisis con la metodología *KDD*.

Las respuestas obtenidas muestran que resulta de gran utilidad desarrollar un flujo de trabajo que incluya algoritmos de agrupamiento y que los resultados obtenidos de estos sirvan para entrenar a un clasificador basado en árboles de decisión. Estos resultados indican que un modelo híbrido entre aprendizaje supervisado y no supervisado resulta muy útil para tratar con un dominio tan complejo la Medicina.

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

Introducción

La calidad de vida de una persona es un concepto que cambia en función de la percepción del contexto social y cultural, sin embargo, puede considerarse en términos generales, como la suma del bienestar físico, emocional, sexual, psicológico y social.

En medicina, la calidad de vida está directamente relacionada con la salud y la enfermedad. La Organización Mundial de la Salud (OMS) declara a la salud como "la opinión del individuo de su posición de la vida, en el contexto cultural y sus valores en lo referente a las expectativas, a los objetivos, a los estándares y a las preocupaciones" (OMS, 2005)

Los tratamientos médicos actuales no solo tienen como objetivo atender el proceso de la enfermedad enfocándolo a la sobrevida, además se centran en la mejora del aspecto físico, emocional y social de los pacientes. La medicina tiene subramas que se dedican específicamente a estudiar y atender problemas que mejoren la calidad de vida.

La Urología es una rama de la medicina que atiende las disfunciones de las vías urinarias, el aparato genital y los elementos adyacentes a él, el Urólogo conoce métodos de diagnóstico, así como procedimiento médicos y quirúrgicos de los órganos de las vías urinarias. Actualmente el Urólogo está preparado para abordar problemas del aparato urinario masculino y femenino (Caunet, s.f.).

Existen diversas enfermedades que son casos de estudio para la Urología, como, por ejemplo, las enfermedades de transmisión sexual y las neoplasias malignas en la próstata. En la ciudad de Puebla, la tasa de mortalidad por VIHI/SIDA es de 3.06 % ocupando el 24º lugar a nivel nacional (SALUD, 2013), así mismo la tasa de mortalidad por cáncer de próstata es del 42.67 % posicionándose en el 9º lugar a nivel nacional (SALUD, 2013).

El proceso clínico que se sigue en la atención de enfermedades genera datos, los datos en conjunto constituyen a la información y el conocimiento que se encuentra implícito en esa información puede ser útil para la mejora en el diagnóstico y tratamiento de las enfermedades, sin embargo, extraer el conocimiento de la información no siempre es una tarea fácil y cuando las cantidades de información son grandes, se vuelve una tarea imposible para un humano.

La Minería de Datos es un conjunto de técnicas que permiten al analista de datos generar y estudiar modelos para extraer el conocimiento de grandes cantidades de información. El aprendizaje automático aporta el enfoque teórico para generar dichos modelos, pues éste estudia los algoritmos que realizan esta tarea. La unión de la Urología con el Aprendizaje Automático puede ser benéfica para describir mejor los padecimientos y mejorar los tratamientos.

1.1. Planteamiento de la investigación

En esta sección se expone el origen del problema que es objeto de la investigación, así como los objetivos particulares y generales de ésta. También se presenta la justificación con las aportaciones que la presente tesis proporciona al campo de la bioinformática para el beneficio de la población que se estudia.

1.1.1. Problema a resolver

El desarrollo de la presente investigación tiene origen en el repositorio de archivos médicos proporcionados por el experto del área, el Dr. Ivo Humberto Pineda Somodevilla, en dicho repositorio hay datos Urológicos que pueden ser analizados mediante diversos algoritmos de aprendizaje automático, para obtener conocimiento útil que ayude a mejorar la atención al paciente, la toma de decisiones, el descubrimiento de nuevas tendencias en la Urología y la creación de campañas de promoción de la salud Urológica.

Los archivos médicos proporcionados corresponden a pacientes de la ciudad de Puebla y municipios cercanos, comenzaron a ser recolectados por el experto desde el 03 de junio del 2015 y continúan en recolección por lo que el volumen de datos continúa en crecimiento, esto es una ventaja debido a que los algoritmos de Aprendizaje Automático trabajan mejor con volúmenes grandes de datos, sin embargo, como actualmente no se tiene un sistema de bases de datos, el incremento del volumen de los datos complica cada vez más el proceso de extracción y transformación de los mismos para su análisis.

La recolección de los datos ha sido en las consultas a los pacientes a través de un archivo de texto plano, la gestión de estos se realiza entrando a modificar el contenido de cada archivo, las imágenes y resultados de los estudios que se han solicitado se encuentran, en algunos casos, en otros archivos. Este escenario da lugar a la modificación errónea de los datos, la recolección inconsistente, la posibilidad de redundancia de información, la ambigüedad, así como la falta de persistencia de los datos a lo largo del tiempo.

La falta de una base de datos que almacene la información del Urólogo dificulta la posibilidad conocer los estadísticos básicos de toda la muestra recolectada, implica carecer de conocimiento del estado general de la salud Urológica de Puebla, además, limita el poder hacer afirmaciones respecto a las tendencias de salud sexual de la población y

el no tener un punto de partida para resolver problemas que se presenten en el futuro.

1.1.2. Objetivos de la investigación

El objetivo general es utilizar algoritmos de Aprendizaje Automático para descubrir patrones de comportamiento en un repositorio de datos urológicos que faciliten la toma de decisiones.

Objetivos específicos:

- Crear un repositorio de datos urológicos utilizando tecnología SQL.
- Desarrollar un *script* en Python que procese recetas médicas en formato Microsoft Word e ingrese los datos en el repositorio SQL.
- Desarrollar una aplicación móvil basada en *android* y un sistema web que permitan la captura y administración de datos urológicos.
- Aplicar técnicas de aprendizaje automático descriptivas y/o predictivas para el descubrimiento de patrones, las cuales permitirán segmentar pacientes, asociar grupos de pacientes con sus antecedentes y clasificar los nuevos pacientes que asistan a consulta en el futuro.
- Evaluar el modelo con el experto del área de conocimiento.

1.1.3. Justificación de la investigación

La principal motivación de la investigación es el aporte que se puede hacer al campo de la Urología. Actualmente, en México no existe un sistema comercial personalizado para administrar pacientes de la especialidad de Urología y que a su vez realice análisis utilizando el Aprendizaje Automático, por lo tanto, el desarrollo de este proyecto se constituye en una aplicación pionera que, con base en el Análisis de Datos, ayude en la toma de decisiones en el campo de la Urología.

1.1.4. Preguntas de investigación

- ¿Cómo construir un repositorio *SQL* para representar los datos Urológicos?
- ¿Cómo desarrollar una aplicación móvil basada en *Android* y un sistema web que permitan la captura y administración de datos Urológicos?
- ¿Qué técnicas de aprendizaje automático son necesarias para generar patrones Urológicos?

1.2. Aportaciones de la investigación

Con la presente investigación se pretende generar motivación para mejorar los procesos de atención y tratamiento a pacientes en el ámbito médico, mediante el uso de tecnologías de la información. En el campo de la Bioinformática se busca proveer a los expertos del área de Urología con una herramienta de toma de decisiones, esta solución podría incrementar las posibilidades de éxito al tratar patologías difíciles de diagnosticar y encontrar conocimiento útil, válido, inteligible y novedoso implícito en los archivos clínicos de los pacientes a los que atienden.

1.3. Estructura de la tesis

Capítulo 1: Introducción: Se describe la problemática que motiva la realización de la investigación con su justificación, así mismo, se plantean los objetivos generales y específicos del proyecto, por último, se explican las aportaciones de la presente investigación.

Capítulo 2: Fundamentos teóricos: Se presentan todos los conceptos teóricos que sustentan los procedimientos realizados para alcanzar los objetivos. Primero se explican las bases de datos relacionales junto con su proceso de diseño, después se habla de inteligencia artificial, sus orígenes y la manera en que el área ha tenido auge en los

últimos años, finalmente, se presenta una introducción al aprendizaje automático, su relación con la minería de datos y la metodología *KDD*.

Capítulo 3: Estado del arte: Se presenta un contexto del área de la salud en México, se hace referencia a los trabajos más recientes de la salud en unión con la minería de datos y se estudian las aplicaciones de las técnicas utilizadas en dichos trabajos.

Capítulo 4: Marco de trabajo: En este capítulo se muestra la aplicación de la teoría enunciada en el capítulo 2 y en la problemática planteada en el capítulo 1. Se explica cómo se diseñó la base de datos, cómo se implementó el algoritmo que procesa las recetas médicas y finalmente cómo se generaron los modelos de aprendizaje automático.

Capítulo 5: Evaluación de la propuesta y resultados: En este capítulo se presentan los resultados obtenidos de toda la investigación y se realiza una interpretación aprobada por el experto.

Capítulo 6: Conclusiones y trabajo a futuro: En este capítulo se exponen las conclusiones obtenidas, enfatizando las consideraciones iniciales de esta investigación, también se describen las limitaciones de los resultados obtenidos, así como el trabajo futuro que puede dar continuación a la investigación.

Capítulo 2

Fundamentos teóricos

A continuación se explica el fundamento teórico en relación con las bases de datos y los algoritmos de aprendizaje automático en el contexto de la investigación. En principio, se presentan definiciones que sustentan el correcto diseño de una base de datos relacional, el uso comercial que estas tienen y las tecnologías bajo las cuales son utilizadas.

En esta sección se aborda el aprendizaje automático y las dos grandes vertientes de algoritmos que estudia, también se habla de las técnicas de minería de datos que tienen base en algoritmos de aprendizaje automático y por último para contextualizar la teoría revisada, se presenta el proceso *KDD*, sus etapas y su relación con la Minería de datos.

2.1. El auge de la capacidad de cómputo

Desde la formalización de la teoría de la computación en la década de 1950, Alan M. Turing introdujo en su artículo *Computing Machinery and Intelligence*, la famosa pregunta *"can machines think?"*, con la que estableció la base de lo conocido actualmente como Inteligencia Artificial, desafortunadamente en la época de Turing no existían computadoras que tuvieran la capacidad de ejecutar en un tiempo razonable algún algoritmo de aprendizaje computacional, lo cual limitó el crecimiento del área.

En noviembre de 1988 DEC (Digital Equipment Corporation) presentó la VAX 11/780,

la primera computadora comercial del mundo con arquitectura de 32 bits (Digital Equipment Corporation, 1977). Para su tiempo fue una computadora con prestaciones muy extendidas, sin embargo, en los siguientes años diferentes compañías crearon computadoras que fueron superando a la VAX 11/780 hasta dejarla completamente obsoleta. En la figura 2.1 puede apreciarse de manera gráfica la comparación de la VAX 11/780 y varias computadoras con diferentes procesadores presentadas a lo largo del tiempo.

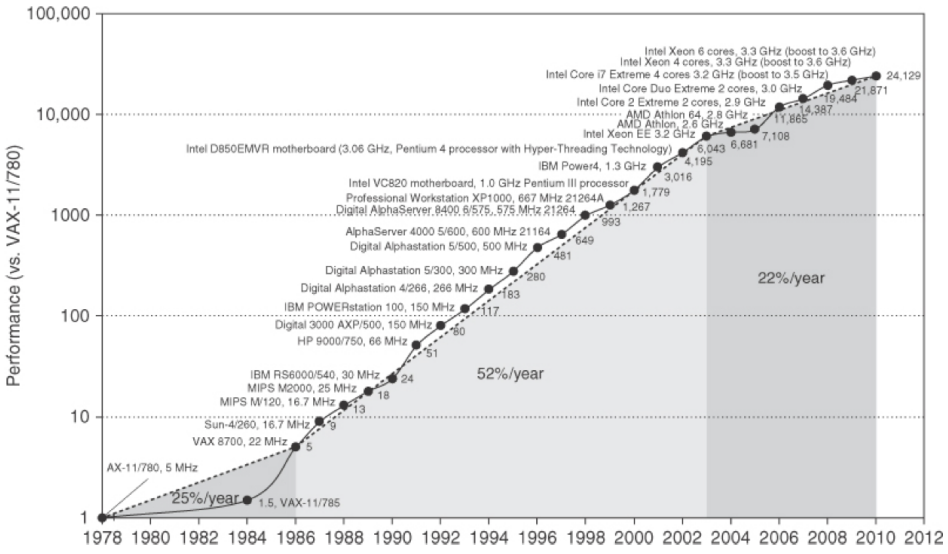


Figura 2.1. Evolución de los procesadores

La tendencia del rendimiento computacional (Figura 2.1) va en incremento conforme transcurre el tiempo (L. Hennessy & A. Patterson, 1989), además, si se considera que un ejemplar de la VAX 11/780 costaba 350,000 USD en 1989 y que una computadora con el procesador Intel Xeon E5-1620, cuesta actualmente 1,900 USD, se puede concluir que conforme pasa el tiempo, el rendimiento de una computadora incrementa y la proporción rendimiento-precio es mejor.

La evolución de los procesadores ha permitido que cada vez las organizaciones tengan

acceso a una computadora con prestaciones que permiten realizar tareas más complejas. Es entonces plausible argumentar, que el abaratamiento del hardware y la mejora del rendimiento de este es un factor que ha permitido que la ciencia de la computación se haya desarrollado tanto en los últimos años.

2.2. El problema de la explosión de información

Gracias a que cada vez es más fácil para las organizaciones tener acceso a computadoras, éstas comienzan a basar sus procesos en soluciones informáticas, lo que provoca que se tengan grandes cantidades de información que van en aumento y que cada vez son más difíciles de analizar por un humano. La limitante más grande de una organización en la actualidad no es la capacidad de recopilar datos sino la capacidad de gestionar, analizar, sintetizar y descubrir conocimiento relevante en ellos (Bernus & Noran, 2017).

Tener almacenados grandes volúmenes de información sin analizarlos nos hace ricos en información, pero pobres en conocimiento (Bernus & Noran, 2017), es por ello, que se han tenido que desarrollar áreas que aporten teorías para poder resolver el problema del análisis de grandes cantidades de información.

2.3. Bases de datos

Los sistemas de bases de datos están presentes como un componente cotidiano en la mayoría de las actividades de la sociedad moderna, desde las interacciones que hay en redes sociales y el comercio por Internet hasta las transacciones bancarias que se realizan diariamente. Debido al avance de la tecnología a lo largo de los años, las aplicaciones de las bases de datos se extienden, y el papel que éstas fungen se vuelve cada vez más crítico (Silberschatz, F. Korth, & Sudarshan, 2002).

Antes de la existencia de los sistemas de bases de datos existieron los sistemas de archivos, los cuales eran conjuntos de ficheros de datos y programas de aplicación que

permitían a los usuarios finales trabajar con los mismos (Marqués, 2009). Los sistemas de archivos estaban establecidos bajo sistemas descentralizados, situación que provocaba que cada departamento o sección de una organización gestionara de manera independiente sus datos con su propio sistema de archivos, sin embargo, apostar por esta estructura tuvo inconvenientes que provocaron la emigración a un sistema centralizado.

Algunos problemas que presentan los sistemas de archivos son los enlistados a continuación (Marqués, 2009):

- Aislamiento de los datos: Debido a la descentralización de los archivos, algunos pueden estar escritos en diferentes formatos y eso implica tener que desarrollar programas de aplicación capaces de leer múltiples tipos de formatos.
- Problemas de integridad: Los valores que pueden ser almacenados en los sistemas de archivos muchas veces tienen restricciones dadas por las reglas del negocio, por ejemplo: "En una cuenta bancaria el saldo no debe ser inferior a 2000.00 MXN", esto implica programar directamente la condición, sin embargo, cuando el número de restricciones es muy grande, resulta ser más complicado modificar los programas para hacer que las reglas de negocio se cumplan.
- Problemas de atomicidad: Debido a que cualquier sistema computacional está sujeto a fallos mecánicos o eléctricos, puede ocurrir que una operación con los datos se vea interrumpida por uno de estos fallos. Los sistemas de archivos no garantizan que una transacción se realice completamente, en otras palabras, en un sistema de archivos no se puede garantizar que las operaciones con los datos sean atómicas.
- Redundancia de datos: Cuando se tiene un sistema de archivos es común encontrarse con problemas de redundancias en los datos y esto es porque en varios departamentos de una organización puede estar almacenado el mismo dato. La redundancia provoca inconsistencia porque si el dato que varios departamentos

comparten sufre una modificación tendría que modificarse en todos los departamentos que este almacenado.

- Dificultad en el acceso a los datos: Como en un sistema de archivos los programas de aplicación son diseñados para cada departamento y aparte los programas de aplicación son estáticos, no es posible consultar una lista de datos desde una perspectiva no prevista.

- Anomalías en el acceso concurrente: Los sistemas de archivos complican el acceso multiusuario para realizar transacciones.

Estos inconvenientes motivaron el cambio de los sistemas de archivos por los sistemas de bases de datos, en las siguientes secciones se explicará porque en un sistema de bases de datos los inconvenientes de los sistemas de archivos no existen.

Una base de datos es una colección de datos persistentes relacionados que están almacenados en un soporte informático de acceso directo y que además contienen información relevante para una organización (Silberschatz, F. Korth, & Sudarshan, 2002) (J. Date, 2001) (Llanos Ferraris, 2007), por consiguiente, una base de datos cumple las siguientes características:

- Representa un aspecto del mundo real.

- Los datos que almacena son coherentes y están lógicamente relacionados.

- Tiene un grupo de usuarios previsto para un objetivo específico.

2.3.1. Introducción a las bases de datos

Un modelo de datos es una colección de conceptos que describen las relaciones, la semántica y las restricciones de consistencia en los datos (Silberschatz, F. Korth, & Sudarshan, 2002), así mismo un modelo de datos proporciona el nivel de detalle, la organización, el almacenamiento y las características que permiten comprender los datos (Elmasri & Navathe, 2015), dicho en otras palabras, el modelo de datos enuncia el nivel

de abstracción de los datos.

Debajo de la estructura de una base de datos se encuentra el fundamento teórico del funcionamiento, este es llamado modelo de datos. A lo largo de los años se han propuesto diferentes modelos de datos para satisfacer diferentes necesidades de la industria, sin embargo, el modelo de bases de datos más utilizado es el relacional.

El modelo relacional fue propuesto por Edgar Frank Codd en su artículo titulado *A relational model of data for large shared data banks* (Llanos Ferraris, 2007). Codd planteó que su modelo opera con relaciones (término matemático para referirse a una tabla) y que sus operadores generan nuevas relaciones. El modelo relacional tiene como base la teoría de conjuntos, esto es una gran ventaja, pues las matemáticas le proporcionan una base teórica sólida que permite demostrar los enunciados del modelo, por ello, es uno de los modelos más estudiados y utilizados en la actualidad.

2.3.2. Diseño de bases de datos relacionales

Diseñar una base de datos consiste en estudiar las reglas de operación de una organización para definir una estructura de datos que se adapte a las necesidades de almacenamiento de información requeridas. La estructura de datos resultante será un esquema compuesto por relaciones con interrelaciones y atributos con sus dominios.

Dividir en etapas el proceso de diseño facilita en gran medida la obtención del resultado esperado, porque habitualmente la complejidad de los requisitos o la cantidad de información impiden ver con claridad un modelo final desde el principio, por ello, el diseño de bases de datos relacionales se divide en 3 etapas tal y como se muestra en la figura 2.2.

El diseño conceptual es una etapa que tiene como base las concepciones de los hechos de la vida real y no las representaciones de los datos, por lo que en esta fase no importa si el modelo de datos será relacional, orientado a objetos, etc. El modelo ER

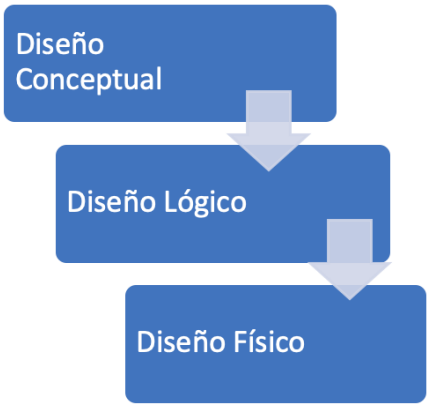


Figura 2.2. Etapas de diseño de bases de datos.

(Entidad Relación) o EER (Entidad Relación Extendido) es utilizado para diseñar el modelo conceptual y tiene una notación gráfica propuesta por Peter Chen que da como resultado el Diagrama ER o Diagrama EER (Chen, 1976).

Los modelos ER y EER se utilizan en la fase de más alto nivel del proceso de diseño de bases de datos porque permiten modelar un problema de la vida real de manera natural, además, los elementos que se utilizan para construirlo son de fácil comprensión. El diagrama ER o EER es útil para comunicarse con el usuario precisamente por el nivel de abstracción con el que representa un problema.

Los elementos que componen a un diagrama ER son los siguiente:

- Atributos: Los atributos son las propiedades relevantes de las entidades o de las relaciones.
    - Atributo Llave: Es el atributo que identifica a toda la entidad.
    - Atributo Univaluado: Es el atributo que tiene un único valor para cada ocurrencia de la entidad.
    - Atributo Multivaluado: Es el atributo que tiene más de un valor para cada

ocurrencia de la entidad.

- Atributo Derivado: Es el atributo que obtiene su valor de valores de otros atributos.
- Atributo Compuesto: Es el atributo que se puede dividir en otros atributos.

- Entidades: Son objetos del mundo real de los cuales nos interesan una serie de características.
  - Entidad: Es una entidad independiente de la existencia de otra entidad.
  - Entidad Débil: Es una entidad que solamente existe si otra entidad existe.
- Relaciones Binarias: Son asociaciones entre las entidades. A continuación, se clasifican por su tipo de conectividad a excepción del último ejemplo que hace referencia a una abstracción que se hace para el Modelo Entidad Relación Extendido:
  - Razón de Cardinalidad.
    - Uno a Uno: En una relación binaria la conectividad uno a uno (1:1) se utiliza para modelar que para cada ocurrencia de la entidad A le corresponde una ocurrencia de la entidad B.
    - Uno a Muchos: En el caso de uno a muchos (1: N) se utiliza para modelar que para una ocurrencia de la entidad A le corresponde un número indefinido de ocurrencias de la entidad B.
    - Muchos a Muchos: Para la conectividad Muchos a Muchos (N:M) ocurre algo similar que con la 1: N solo que ahora es en ambos sentidos, para cada ocurrencia de la entidad A le corresponden muchas ocurrencias de la entidad B y para cada ocurrencia de la entidad B le corresponden muchas ocurrencias de la entidad A.
  - Participación
    - Participación Total: Define que una entidad siempre va a participar en una relación.

- Participación Parcial: Define que no siempre una entidad va a tener participación en una relación.

- Restricción estructural en la participación.
  - Expresada en términos de máximos y mínimos, la restricción de la participación de una entidad en una relación determina cuál es el rango de ocurrencias de una entidad.

- Abstracciones del Modelo Entidad Relación Extendido.
  - Generalización/Especificación: Es utilizada para definir una entidad general llamada entidad superclase que puede derivarse en entidades específicas llamadas entidades subclase.

En la etapa del Diseño Lógico es imprescindible contar con el resultado del diseño conceptual, puesto que este será transformado al modelo de datos de la tecnología que se empleará en la etapa del diseño físico; más concretamente, para las bases de datos relacionales se utiliza el Modelo Relacional que posteriormente se traducirá en el modelo físico en instrucciones para un SGDB relacional como MySQL.

Antes de llegar directamente al proceso de conversión al modelo relacional es oportuno realizar un cambio de terminología respecto al Modelo Entidad Relación, dado que en modelo relacional una relación es un término matemático para referirse a una tabla y que una relación entre tablas es llamada interrelación vamos, a referirnos a las relaciones del modelo Entidad Relación como interrelaciones y las entidades seguirán denominándose con el mismo término.

A continuación, se enlistan en los siguientes puntos el proceso de conversión del diseño conceptual al diseño lógico:

1. Transformación de entidades a relaciones (Elmasri & Navathe, 2015):

   a) Las entidades fuertes se convierten en una relación.

- Los atributos univaluados pasan a ser atributos de la relación.
- La llave primaria pasa a ser llave primaria de la relación.
- Los atributos derivados se eliminan en la relación.
- Los atributos compuestos pasan a ser atributos independientes en la relación.
- Los atributos multivaluados se convierten en una nueva relación, la clave primaria de la entidad es una combinación de la clave primaria de la misma junto con el atributo multivaluado.

b) Las entidades débiles se convierten en una relación.

- La clave primaria de la nueva relación es al mismo tiempo una clave foránea a la entidad fuerte de la que depende.
- Todos los atributos son tratados tal y como se hace en una entidad fuerte.

c) Transformación de interrelaciones binarias:

- Relaciones de uno a uno.
  - Si una de las entidades tiene participación total en la interrelación:
    - La clave primaria de la entidad con participación total, pasa a ser clave foránea en la entidad con participación parcial.
  - Si ambas entidades tienen participación total o participación parcial en la interrelación:
    - Se elige arbitrariamente una de las dos entidades para que de ella se tome la clave primaria y pase a ser clave foránea de la otra entidad.
  - Si ambas entidades tienen participación total y una de ellas no participa en ninguna otra interrelación:
    - Se unen ambas entidades para formar una sola relación.
  - Si hay atributos en la interrelación se añaden a la relación en la que se decidió agregar la clave foránea.

- Relaciones de uno a muchos:
  - Se toma la clave primaria de la entidad con razón de cardinalidad 1 y este pasa a ser una llave foránea de la entidad que tiene la razón de cardinalidad N.
  - Si hay atributos en la interrelación se agregan a la relación en donde la razón de cardinalidad es N.
- Relaciones de muchos a muchos:
  - Se crea una nueva relación que tendrá como clave primaria una composición de las claves primarias de las entidades participantes en la interrelación.
  - Si hay atributos en la interrelación se agregan en la nueva relación.
- Interrelaciones de Generalización/Especificación:
  - Se crea una relación por cada entidad.
  - Las interrelaciones hijo (las que son subentidades en el Diagrama EER) heredan la clave primaria de la entidad padre (La entidad genérica en el Diagrama EER), clave que es foránea y primaria a la vez en la nueva interrelación.

El resultado del diseño lógico es un esquema que expresa los datos en términos de Relaciones (Tablas). En la tabla 2.1 se muestra un resumen de la equivalencia de los elementos del modelo Entidad Relación Extendido y el Modelo Relacional. (Llanos Ferraris, 2007)

Tabla 2.1

Mapeo Diseño conceptual - Lógico.

Modelo Entidad Relación	Modelo Relacional
Entidad	Relación
Entidad débil	Relación
Interrelación 1:1	Clave foránea
Interrelación 1:N	Clave foránea
Interrelación N:M	Relación
Generalización/Especialización	Relación por cada entidad
Atributo multivaluado	Relación

Antes de continuar con la última etapa del proceso de diseño de una base de datos relacional, es conveniente tratar con el concepto de normalización en el Modelo Relacional. Para contextualizar la normalización en un esquema del modelo relacional se considera lo siguiente:

Sea R un esquema del modelo Entidad Relación.

1. R puede haber sido creada a partir del diseño general de una base de datos, es decir, a partir de un modelo Entidad Relación (Tal y como se ha abordado en esta sección).

2. R pudo haber sido una sola relación en un principio que posteriormente se descompuso en diversas relaciones.

3. R pudo haber sido generado a partir de otro tipo de diseño adecuado para el modelo relacional.

La normalización de un esquema del Modelo Relacional define la calidad del mismo, verificar en qué forma normal se encuentra un esquema permite definir formalmente por qué un esquema es mejor que otro. Un esquema relacional normalizado carece de redundancias de datos, lo que implica que esté libre de anomalías de actualización y borrado. La normalización permite entender por completo al diseñador todos los atri-

butos por los que está compuesta su base de datos.

El proceso de normalización se realiza por etapas llamadas formas normales y se llega a estas de manera consecutiva, las etapas siguen un orden estricto. A continuación, se explica de manera resumida cómo verificar las formas normales más usuales en un esquema.

Un concepto crucial para explicar la normalización es el de Dependencia Funcional. Considere lo siguiente: Sea R una relación y sean X, Y subconjuntos del conjunto de atributos de R, entonces se dice que Y depende funcionalmente de $X (X \implies Y)$ si y solo si cada valor de X en R está asociado con un valor de Y en R (J. Date, 2001).

Una relación R se encuentra en Primera Forma Normal si todos los atributos de R son atómicos (Silberschatz, F. Korth, & Sudarshan, 2002), un atributo es atómico si este es indivisible, en otras palabras, el atributo no puede descomponerse en subatributos.

Una relación R se encuentra en Segunda Forma Normal si está en Primera Forma Normal y si cada atributo no principal de R depende completamente de la clave primaria de R (Elmasri & Navathe, 2015). Un atributo no principal de una relación es todo atributo que no es llave primaria.

La segunda forma normal utiliza el concepto de dependencia funcional total. Una dependencia funcional total es una dependencia funcional en la que si se elimina un atributo del conjunto de claves primarias de X la dependencia no es mantenida, es decir, en una dependencia funcional total, todos los atributos no clave deben depender totalmente de la clave primaria de la relación.

Una relación R está en Tercera Forma Normal si y solo si se encuentra en segunda

forma normal y si los atributos no clave son dependientes en forma no transitiva de la clave primaria (Elmasri & Navathe, 2015). Una dependencia funcional transitiva, es una dependencia funcional en la que existe un conjunto de atributos Z que no es subconjunto de la clave primaria donde se mantiene $X \implies Z$ y $Z \implies Y$.

Para asegurarse que un esquema cumple con alguna de las formas normales es necesario verificar todas las relaciones. En caso de que alguna relación no cumpla con alguna forma normal es necesario normalizarla, este proceso consiste en descomponer la relación tomando los atributos que no cumplen con alguna forma normal y agregándolos a una nueva relación.

Cuando ya se tiene un Modelo Relacional robusto es posible pasar a la última fase, el **diseño físico**, diseñarlo consiste en implementar el Modelo Lógico en un SGBD específico. El lenguaje de consulta SQL es un estándar para las bases de datos relacionales y es considerado como una de las razones del éxito comercial de las bases de datos.

2.4. Aprendizaje Automático y Minería de Datos

Los diagnósticos médicos, los reconocimientos visuales, la ingeniería de diseño y la detección de patrones en grandes conjuntos de datos, son un ejemplo de problemas que carecen de soluciones algorítmicas o que están definidos informalmente. En dichos tipos de problemas la solución clásica, mediante un programa basado solo en un algoritmo y en datos, puede no ser la mejor opción al abarcar problemas tan complicados como los ya mencionados. El aprendizaje automático es una solución (Garcia Serrano, 2016) para resolver este tipo de problemas dado que suma el concepto del conocimiento del dominio que permite desarrollar soluciones más elaboradas y precisas.

El aprendizaje automático o aprendizaje de máquina es una rama de la Inteligencia Artificial que busca hacer que las máquinas sean capaces de hacer generalizaciones a

partir de información suministrada en forma de ejemplos. Esto aporta a un ente con inteligencia artificial la capacidad de aprender cosas para adaptarse al medio tal y como lo hace un ente inteligente naturalmente.

Para poder hablar de aprendizaje automático o aprendizaje de máquina es necesario comprender cual es la definición de aprendizaje. La Real academia española lo define como: "Adquirir conocimiento de algo a través, del estudio o la experiencia".

Un niño aprende desde muy temprana edad que si se cae puede lastimarse, pero para llegar a esta conclusión primero tuvo que caerse varias veces, es decir, llegó a una generalización a partir de ejemplos. El aprendizaje automático utiliza en esencia el mismo modelo cognitivo para alcanzar sus objetivos.

Las técnicas que estudia el aprendizaje automático son utilizadas para descubrir patrones que se encuentran implícitos en un conjunto de datos, esta aplicación del aprendizaje automático es estudiado en la Minería de Datos (Witten, Frank, & Hall, 2011). El enfoque que la minería de datos le da a las técnicas de Aprendizaje Automático es práctico, no teórico.

La minería de datos da la posibilidad de extraer patrones, descubrir tendencias, predecir comportamientos y, en general, sacar provecho a un conjunto grande de datos, sin embargo, la minería de datos es sólo una etapa de un proceso más completo llamado *KDD* (Descubrimiento de conocimiento a partir de datos). Este proceso está compuesto de varias etapas en las que hay diversas técnicas del Aprendizaje Automático y otras áreas de la informática y las matemáticas.

2.4.1. Introducción al Aprendizaje Automático

En la sección 2.4 se explicó en líneas generales cómo trabaja el aprendizaje automático, haciendo énfasis en la idea de que, a partir de un conjunto de ejemplos, en

los que se encuentra el conocimiento de un dominio específico, se obtenga una generalización. En la figura 2.3 se aprecia ese proceso de inducción y deducción que utiliza el aprendizaje automático.

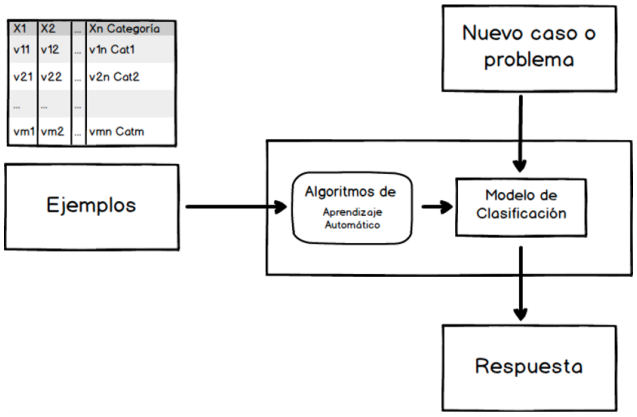


Figura 2.3. Funcionamiento del aprendizaje supervisado.

El aprendizaje automático es un área extensa que para poder ser estudiada de mejor manera se divide en dos principales ramas: el aprendizaje supervisado y el aprendizaje no supervisado.

El aprendizaje supervisado estudia métodos que intentan descubrir una relación entre los atributos de entrada y un atributo objetivo (Maimon & Rockach, 2010), la relación descubierta representa un modelo que describe los fenómenos ocultos en el conjunto de datos y estos fenómenos pueden ser utilizados para predecir el valor objetivo de un nuevo ejemplo solamente conociendo los atributos de entrada.

El aprendizaje supervisado es utilizado en diversos problemas como: el diagnóstico de enfermedades, la concesión de créditos bancarios, la predicción de quiebra de una empresa, el reconocimiento de caracteres escritos a mano, anomalías en cromosomas, etc. (Sierra Araujo, 2006). Para que un algoritmo de aprendizaje supervisado cree un modelo que cumpla con los resultados esperados se le debe proporcionar un conjunto

de ejemplos con las etiquetas correctamente asignadas.

Las tareas del aprendizaje supervisado pueden dividirse en dos: La clasificación y la regresión. En la clasificación el algoritmo determina el valor de la variable en un dominio nominal, por ejemplo, si se quiere saber si un correo es *spam* o no, o determinar si el clima de mañana será nublado, soleado o lluvioso. Cuando se quiere determinar el valor de una variable numérica se realiza una tarea de regresión, por ejemplo, si se desea predecir cuánto caerá el dólar el año siguiente o si se desea predecir el alcance de una campaña publicitaria en redes sociales.

2.4.2. Clasificación

Suponga que tiene un conjunto de variables predictivas $\{X_1, X_2, X_3, ..., X_n\}$ y una variable C que representa a la clase real de un ejemplo, también suponga que en una base de datos D se encuentran N ejemplos diferentes, en los cuales el valor de la clase es de la forma $\{(x_1, o_1), (x_2, o_2), (x_3, o_3), \ldots, (x_n, o_n)\}$, en un problema con M clases distintas o_i e, $\{c_1, c_2, c_3, \ldots, c_m\} c_i = 1, 2, 3, \ldots, N$ (Sierra Araujo, 2006). En la tabla 2.2 se observan los datos descritos con los términos anteriores.

Tabla 2.2

Conjunto de datos de entrenamiento.

Caso	X_1	X_2	...	X_n	C
1	X_{12}	X_{21}	...	X_{n2}	C_2
2	X_{11}	X_{23}	...	X_{n1}	C_M
...	...	...	...	...	
N	X_{11}	X_{22}	...	X_{n2}	C_2

Al clasificador se le proporciona un conjunto de datos D llamado conjunto de entrenamiento, su objetivo es aprender qué clase del conjunto C le corresponde a cada conjunto $\{x_1, x_2, x_3, \ldots, x_n\}$ (Tabla 2.2). Entonces, el algoritmo de clasificación construye un modelo que puede usarse para predecir ejemplos nunca vistos en el conjunto

D (Maimon & Rockach, 2010).

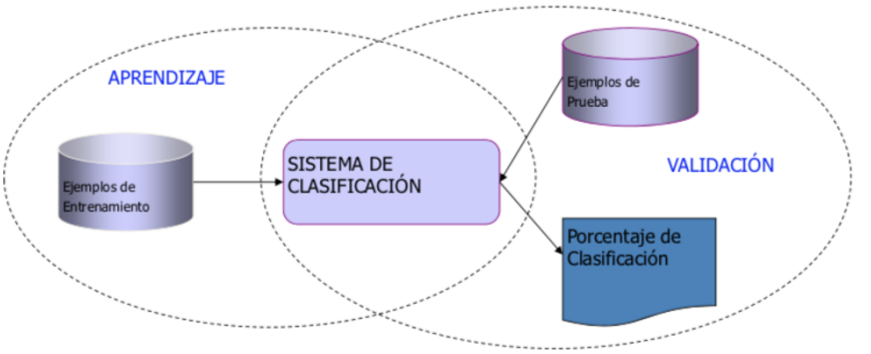


Figura 2.4. Evaluación de un clasificador.

La evaluación de un clasificador es una tarea primordial en el aprendizaje automático, pues con ella se puede comprender la calidad de un modelo generado, existen diferentes criterios para evaluar los modelos de clasificación y a pesar de que los modelos con una alta precisión se consideran mejores también, puede haber criterios que definan la calidad de un modelo como: la complejidad computacional, la interpretabilidad y la escalabilidad (Maimon & Rockach, 2010).

A continuación, se revisarán las definiciones de las medidas de evaluación de un clasificador (Han, Kamber, & Pei, 2012) y para complementarlas se presentarán las expresiones para calcularlas (tabla 2.3).

- *Accuracy* de un clasificador es el porcentaje de tuplas del conjunto de entrenamiento que fueron clasificadas correctamente.
- *Error rate* es: 1−*accuracy*.
- La sensibilidad y la especificidad son tasas positiva verdadera y negativa verdadera respectivamente, estas medidas son utilizadas para evaluar modelos en los que hay un desequilibrio considerable en las clases a predecir.

- *Precision and Recall:* precision es una medida de exactitud que indica qué porcentaje de ejemplos etiquetados como positivos son realmente positivos, mientras que el *recall* es una medida de integridad, pues indica qué porcentaje de ejemplos positivos están etiquetados como tales. *Recall* se corrresponde con *sesitivity*.

Tabla 2.3

Medidas de evaluación del clasificador.

Medida	Expresión
Accuracy	$\left(\frac{VP+VN}{P+N}\right)$
Error rate	$\left(\frac{FP+FN}{P+N}\right)$
Sensitivity/recall	$\left(\frac{VP}{P}\right)$
Specififity	$\left(\frac{VN}{N}\right)$
Precision	$\left(\frac{VP}{VP+FN}\right)$

- Verdaderos Positivos (VP): Son los ejemplos positivos clasificados correctamente.
- Verdaderos Negativos (VN): Son los ejemplos negativos clasificados correctamente.
- Falsos Positivos (FP): Son los ejemplos que negativos que fueron clasificados incorrectamente como positivos.
- Falsos Negativos (FN): Son los ejemplos positivos que fueron clasificados incorrectamente como negativos.

Estos parámetros son concentrados en la matriz de confusión de la figura 2.5. La matriz de confusión es una herramienta de gran utilidad para analizar qué tan bueno resultó el clasificador al determinar la clase correcta de cada uno de los ejemplos con los que este fue entrenado.

		Clase Predicha		
		Si	No	Total
Clase Actual	Si	VP	FN	P
	No	FP	VN	N
	Total	P	N	P + N

Figura 2.5. Matriz de confusión.

La determinación de que un modelo sea factible o no, está dada por los resultados de las métricas enunciadas en la tabla 2.3, sin embargo, pueden existir otros criterios en los que el modelo no sea viable; por ejemplo, el caso en el que la multinacional Netflix no llevó a producción la versión mejorada de su sistema de recomendaciones por el alto costo computacional que esto implicaba (Masnick, 2012).

2.4.3. Aprendizaje no supervisado

El aprendizaje no supervisado tiene la característica de tratar con problemas en los que los ejemplos proporcionados al modelo no cuentan con una etiqueta o variable objetivo, es decir, a diferencia del aprendizaje supervisado este no requiere conocimiento a priori para funcionar. El aprendizaje no supervisado está muy relacionado con el problema de la estimación de la densidad en la estadística (Hinton & Sejnowski, 1999).

Las aplicaciones del aprendizaje no supervisado son esencialmente en problemas de agrupamiento en los que se pueden descubrir clases dentro de los datos; por ejemplo, se pueden tener como datos de entrada un conjunto de imágenes con dígitos del 0 al 9 escritos a mano, entonces, un algoritmo de aprendizaje no supervisado puede encontrar 10 grupos en dicha información de entrada (Han, Kamber, & Pei, 2012).

Debido a que el objetivo del aprendizaje no supervisado es descubrir nuevos conjuntos de categorías, la evaluación de estos es intrínseca, a diferencia de los métodos de

aprendizaje supervisado en los que la evaluación es extrínseca, puesto que los grupos reflejan una clase de referencia. Por lo que los métodos no supervisados tienen objetivos descriptivos y no predictivos (Maimon & Rockach, 2010).

El agrupamiento o *clustering* es una técnica que agrupa instancias de datos en subconjuntos que compartan características similares, de esta forma los ejemplos son organizados en una representación eficiente que puede conducir al descubrimiento de grupos previamente desconocidos dentro de los datos (Maimon & Rockach, 2010), (Han, Kamber, & Pei, 2012), en este contexto, diferentes métodos de agrupamiento pueden generar diferentes combinaciones de clusters en un conjunto de datos.

El agrupamiento abarca disciplinas desde las matemáticas y la estadística hasta la genética y la informática, por lo que es una herramienta ampliamente usada para diversos estudios. Formalmente, la estructura del agrupamiento se representa como un conjunto de subconjuntos $C = C_1, C_2, C_3, \ldots, C_k$ de S en donde $S = \cup_{(i=1)}^{k} C_i$ y $C_i \cap C_j = \emptyset$ con $i \neq j$ (Maimon & Rockach, 2010), por ello, cualquier ejemplo en S pertenece únicamente a un solo subconjunto tal y como se muestra en la figura 2.6

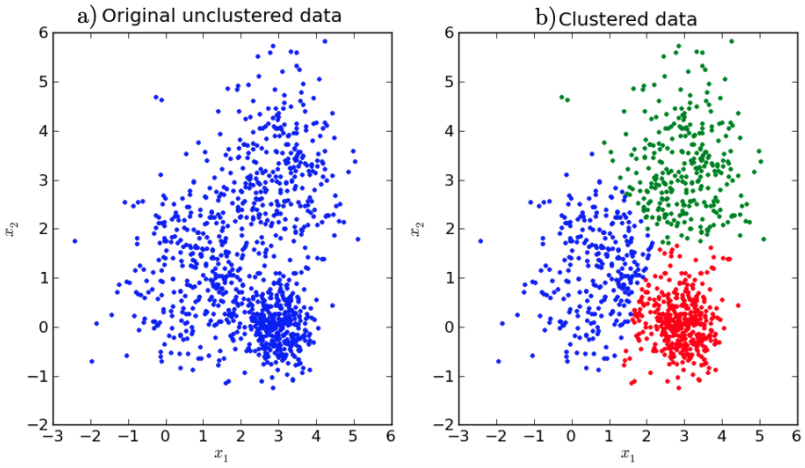


Figura 2.6. Proceso de agrupamiento: (a) conjunto original (b) datos agrupados.

Las reglas de asociación tratan de buscar las posibles relaciones existentes entre la ocurrencia de un hecho en determinados conjuntos de transacciones (Vila Miranda, Sanchez Fernandez, & Cerda Leiva, 2004), la definición formal de una regla de asociación es una implicación de la forma $X \implies Y$, en donde $X, Y \subset I$ y $X \cap Y = \emptyset$ denotando de esta forma que todo evento que satisface a X también satisface a Y (Molina Lopez & Garcia Herrero, 2006).

Un ejemplo didáctico para comprender las reglas de asociación puede referirse a las transacciones que ocurren en el ámbito comercial, por ejemplo, el 95 % de las personas que compran jabón, compran shampoo, en términos coloquiales se puede expresar como: "La mayoría de las personas que compran jabón compran shampoo", pudiendo expresar la regla de asociación como jabón $\implies$ shampoo en donde el jabón es el antecedente y shampoo es el consecuente de la regla.

2.4.4. Minería de datos y metodología *KDD*

Knowledge Discovery in Databases (KDD) es el proceso de identificar patrones válidos, novedosos, útiles e intelegibles a partir de grandes conjuntos de datos (Maimon & Rockach, 2010). La minería de datos es el núcleo de la metodología *KDD* que involucra algoritmos estudiados por el Aprendizaje automático, mismos algoritmos que permiten explorar datos, desarrollar modelos matemáticos y descubrir patrones significativos que se traduzcan en conocimiento útil.

El proceso de descubrimiento de conocimiento de la figura 2.7 es iterativo y consta de 5 etapas principales. Comienza con el entendimiento del dominio y termina con la aplicación del conocimiento adquirido en todo el proceso, así mismo, el proceso *KDD* está diseñado para regresar a cualquier etapa anterior desde cualquier etapa de la metodología, lo que permite reajustar el proceso en todo momento.

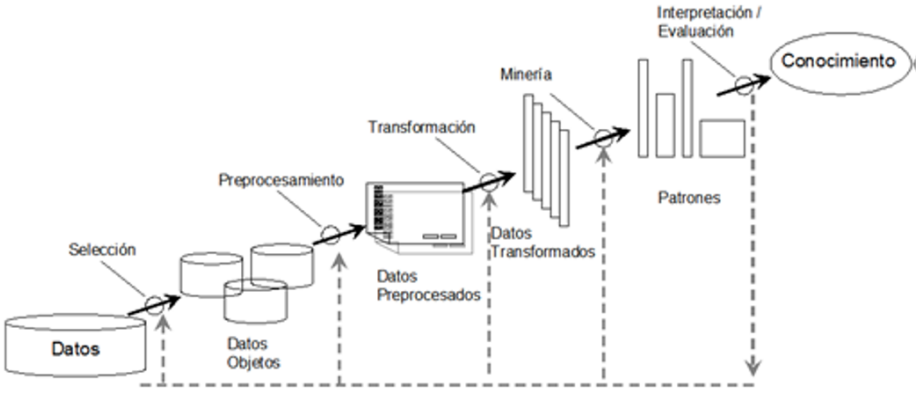


Figura 2.7. Proceso KDD (Maimon & Rockach, 2010)

1. **Comprensión del dominio:** La primera etapa del proceso *KDD* tiene el objetivo de llegar a la comprensión del dominio de los datos que se van a analizar, esto con el objetivo de definir las metas que se quieren conseguir con el análisis de datos.

2. **Selección de datos:** Como etapa siguiente del proceso es necesario crear un conjunto de datos que será el objeto de estudio. La relación con la etapa de comprensión del dominio es muy estrecha puesto que los objetivos definidos dictan qué datos se requieren, por tanto, en la selección se datos se tendrá que averiguar qué datos están disponibles, obtener datos adicionales e integrarlos en un solo conjunto de datos.

3. **Preprocesamiento y limpieza:** En esta etapa se busca mejorar la calidad de los datos, esto es, eliminar datos ruidosos, tratar valores atípicos y el borrado de datos. Se realiza el proceso de limpieza de datos porque la calidad del modelo que se busca crear depende en gran parte de la calidad de los datos que se le proporcionen, si se proporcionan datos con errores de captura o con atributos poco confiables el modelo final puede resultar con un sesgo importante.

4. **Transformación de datos:** En esta etapa se reduce la dimensión de los datos mediante la selección de características y el muestreo, también se discretizan

atributos numéricos, todo con el objetivo de tener un conjunto de datos óptimo para algoritmos específicos de aprendizaje supervisado y no supervisado.

5. **Minería de datos:** La etapa de minería de datos está orientada a cumplir de manera directa los objetivos establecidos al iniciar el proceso *KDD*, por eso se comienza con la elección de la tarea adecuada para ello. Con la minería de datos se pueden hacer dos tipos de tareas: las descriptivas y las predictivas, mismas que corresponden a los métodos estudiados en el aprendizaje no supervisado y el aprendizaje supervisado, respectivamente.

   Una vez teniendo la estrategia para alcanzar los objetivos, es necesario decidir qué tácticas utilizar, es decir, elegir qué método específico se utilizará, por ejemplo, suponga que se está resolviendo un problema de clasificación y se tiene la opción de utilizar una red neuronal artificial o un árbol de decisión, la primera opción aporta más precisión al modelo, sin embargo, la segunda opción proporciona más comprensibilidad al modelo.

   Con la tarea definida y el método elegido con base en la naturaleza del problema, finalmente, se llega a la implementación del algoritmo y en este paso solo queda hacer múltiples experimentos con ajustes diferentes en los diversos parámetros que el algoritmo necesita, con el fin de obtener el modelo más preciso posible.

6. **Evaluación:** En esta etapa se evalúan e interpretan los patrones minados con respecto a los objetivos iniciales del proceso. Aquí es fundamental la valoración del experto del dominio de los datos que se están minando, él es quien puede determinar con más precisión la comprensibilidad y la utilidad del modelo obtenido. En caso de no haber alcanzado los resultados esperados, es oportuno repetir el proceso desde cualquier etapa con alguna modificación.

7. **Aplicar el conocimiento descubierto:** La etapa final del proceso *KDD* tiene como principal tarea la utilización del conocimiento en la toma de decisiones

futuras y es hasta este punto que se puede determinar la efectividad de todo el proceso.

2.4.5. Técnicas de Minería de datos basadas en Aprendizaje Automático

En la sección 2.4.4 se enunció la metodología *KDD* y las etapas que la componen, ahora se hace énfasis en la etapa de minería de datos porque es una de las más importantes, por ello, es conviene enunciar las técnicas que la componen. En la figura 2.8 se muestra un diagrama jerárquico de la minería de datos.

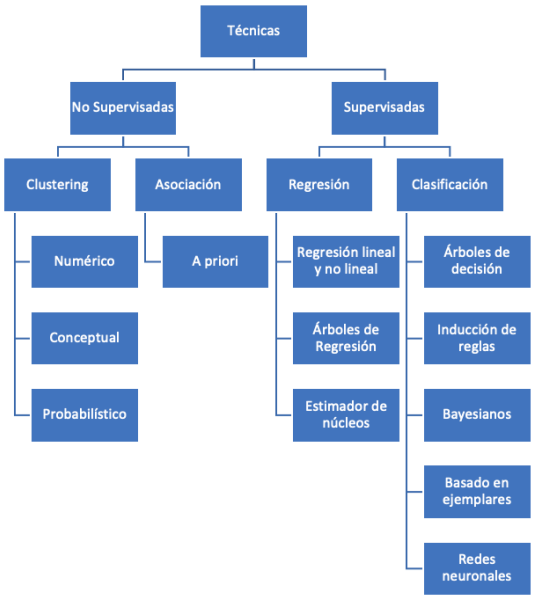


Figura 2.8. Técnicas de minería de datos. (Molina López & García Herrero, 2006)

Las técnicas de minería de datos enunciadas en la figura 2.8 están clasificadas dentro de la rama en la que son más utilizadas, pero esto no significa que los algoritmos o técnicas no puedan utilizarse para distintos propósitos, por ejemplo, las redes neuronales pueden ser utilizadas como clasificadores o como regresores y hasta ser utilizadas en el aprendizaje no supervisado, lo mismo ocurre con los árboles de decisión o de regresión.

2.4.6. Weka

Weka (figura 2.9) es un software de aprendizaje automático y de código abierto (Eibe Frank, 2016), es ampliamente utilizado para la investigación y aplicaciones industriales, se puede acceder a él mediante una interfaz gráfica, desde una terminal o con una *API*. Weka se puede integrar con las herramientas de ciencia de datos más populares como R y Python. Una de las grandes ventajas de Weka es que facilita la experimentación con diversos algoritmos de aprendizaje automático, permite entrenar, ejecutar y evaluar clasificadores sin escribir una sola línea de código.

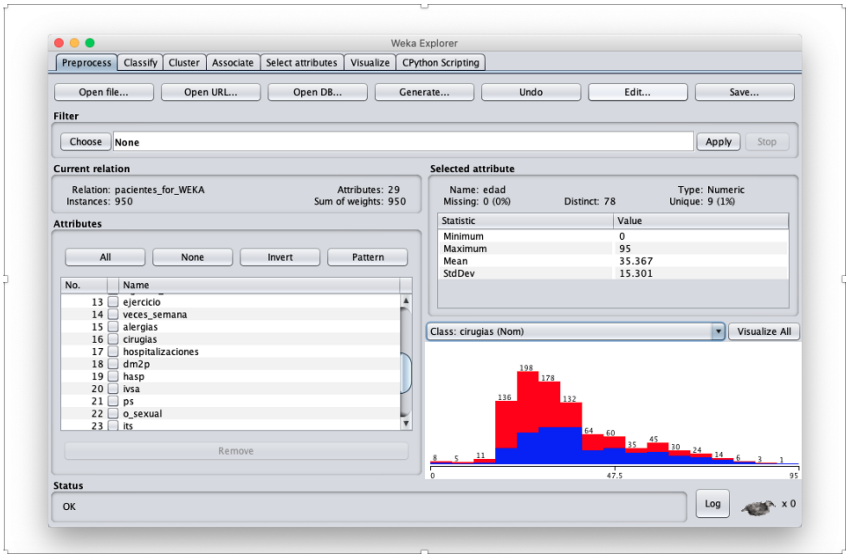


Figura 2.9. Interfaz del software WEKA

2.4.7. *Anaconda navigator*

Anaconda (figura 2.10) es una *suite* multiplataforma de código abierto que incluye un conjunto de bibliotecas y herramientas diseñadas para el desarrollo de proyectos de ciencia de datos, tiene soporte para diversos entornos de desarrollo y para los lenguajes de programación más importantes en el ámbito de la ciencia de datos, Python y R. (Anaconda, Inc., 2019). Anaconda incluye entornos de desarrollo como: JupyterNotebook,

JupyterLab, spyder, Rstudio junto con las bibliotecas analíticas más importantes como NumPy, ScPy, Numba, Pandas y Dask. Esta además incluye por defecto bibliotecas de visualización como MatPlotLib, Bokeh, HoloViews, Seaborn, por último, también incluye bibliotecas de Aprendizaje Automático como TensorFlow, Sklearn, H2o y Theano (Rodríguez, 2018).

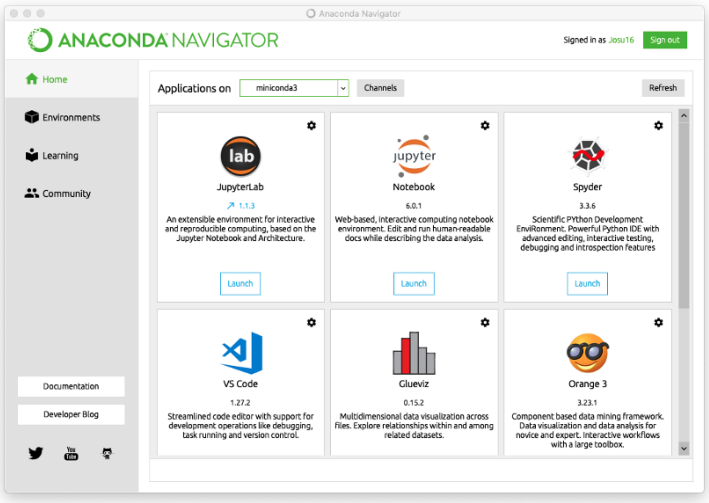


Figura 2.10. Menú principal de Anaconda Navigator.

Las aplicaciones que tiene la suite Anaconda son ilimitadas, puesto que, está respaldada por los dos lenguajes de programación más utilizados en la ciencia de datos, teniendo la posibilidad de desarrollar: herramientas *ETL* en Python, hacer análisis estadísticos con R y hasta desarrollar un completo sistema de toma de decisiones con ambos lenguajes de programación utilizando la biblioteca rpy2 (Gautier, 2016).

Capítulo 3

Estado del arte

En los últimos años la Minería de Datos se ha consolidado como una disciplina ampliamente reconocida por distintas instituciones y organizaciones alrededor del mundo (Oscar Nigro, Xodo, Corti, & Terren, 2004), por ello, se está convirtiendo en una herramienta de gran utilidad para la investigación en diversas áreas de la ciencia, la medicina no es una excepción ,pues la minería de datos puede apoyar a los expertos del sector salud y a los pacientes a mejorar en las siguientes dimensiones: (FMed, 2019)

- **Pronóstico:** Identificar patrones de comportamiento de enfermedades en determinados pacientes con base en múltiples historiales clínicos, estimar el tiempo de recuperación de los pacientes ante determinados padecimientos, entender mejor los progresos de las enfermedades y tratamientos.
- **Diagnóstico:** Ayudar a los expertos a identificar posibles diagnósticos en las visitas clínicas, descubrir posibles padecimientos futuros con base en los historiales clínicos de los pacientes, los modelos de aprendizaje automático también pueden ser de utilidad para los expertos como un recurso de respaldo para diagnosticar enfermedades con sintomatologías confusas.
- **Tratamiento:** Entrenar un modelo de aprendizaje automático puede ayudar a los expertos a automatizar sugerencias de tratamientos basadas en grandes bases de datos de pacientes con diagnósticos similares.
- **Flujo de trabajo clínico:** Mejorar y simplificar los registros médicos electróni-

cos, esto con el objetivo de quitarles esa carga de trabajo a los médicos para que puedan pasar más tiempo en contacto directo con los pacientes.

- **Ampliar el acceso a los conocimientos especializados:** Facilitar el acceso a un sistema que pueda sugerir o alertar atención médica especializada a pacientes de regiones geográficas remotas con escasez de especialistas médicos.

3.1. Minería de datos y Medicina

En el año 2014, Henry Jesús Hernández publicó una investigación en la que utiliza técnicas de minería de datos para obtener patrones de comportamiento en los expedientes clínicos de pacientes prediabéticos (Hernández Gómez, 2014), el autor utilizó la base de datos de la Encuesta Nacional de Salud y Nutrición (ENSANUT) del año 2012.

El problema fue abordado desde la perspectiva descriptiva y predictiva, en su etapa descriptiva utiliza aprendizaje no supervisado con la técnica de agrupamiento *K-Means* (Hernández Orallo, Ramírez Quintana, & Ferri Ramírez, 2004); en la etapa predictiva, utiliza el algoritmo de aprendizaje supervisado J48 (Hernández Orallo, Ramírez Quintana, & Ferri Ramírez, 2004).

El resultado que obtuvo al aplicar aprendizaje supervisado con el algoritmo J48 fue: "Se descubrió que las pacientes a las que algún médico les ha dicho que, Si tienen diabetes o azúcar alta en la sangre, presentan presión alta en un tiempo después del diagnóstico. Aunque no es causa para el desarrollo de la patología, debido a que existen pacientes con diabetes y no presentan presión alta. Pero si puede considerarse como factor de riesgo para quienes presentan diabetes" (Hernández Gómez, 2014), además, obtuvo la siguiente lista de reglas de clasificación del conjunto de datos estudiado:

- Existen pacientes con diabetes desde hace 10 años, pero desconocen si tienen presión alta.

- Hay pacientes que tienen presión alta desde hace 6 años y llevan 10 años padeciendo diabetes.
- Quienes llevan 2 años con presión alta, tienen diabetes desde hace 10 años.
- Se descubrió que quienes asumen que tienen 10 años con presión alta, tiene 5 años con diabetes.
- Las pacientes que no tienen presión alta, presentan diabetes desde hace 10 años.

Respecto a los resultados de aplicar la técnica de clustering con el algoritmo K-Means obtuvo 4 grupos de pacientes descritos a continuación:

- Grupo 0.- Es prevalecido por Mujeres en edad de 38 años, ningún médico le ha dicho que tiene presión alta, no tienen diabetes y por consecuente no presentado ningún padecimiento relacionado con la patología.
- Grupo 1.- Conformado en su mayoría por Mujeres en edad de 36 años, no presentan diabetes, tampoco presentan presión alta o colesterol.
- Grupo 2.- Al igual que el grupo 0 predominan las mujeres en edad de 38 años, estas han presentado colesterol alto, así como triglicéridos en límite establecido para considerarlo como normal.
- Grupo 3.- Se constituye por hombres en edad de 20 años, no presentan diabetes, tampoco padecen colesterol o triglicéridos.

La investigación de Hernández Gómez es una de las pocas aportaciones del Aprendizaje Automático a la medicina en México, sin embargo, en el campo de la Urología no existe una investigación similar publicada a nivel nacional, por ello, la presente tesis tiene una oportunidad de innovar en este campo.

En México aún no existe un expediente clínico digital universal que sea utilizado por todas las instituciones de salud pública y privada (Medina, 2017), por esta razón, la

información de historias clínicas de los pacientes no se encuentra concentrada en un repositorio universal, esto implica que cada institución cuente con su propio repositorio de información hecho a su especificación y con sus consideraciones, sin embargo, existe una norma que describe como debe estar estructurada una historia clínica (NOM-004-SSA3-2012, 2010).

Capítulo 4

Marco de trabajo

En este capítulo se planteará todo el desarrollo del trabajo de investigación para la presente tesis, desde el proceso de abstracción de flujo de trabajo de un Urólogo para el modelado de la base de datos hasta llegar a los resultados del algoritmo de aprendizaje automático. Los conceptos abordados en el capítulo 2 se volverán a revisar en el presente capítulo, pero con un enfoque práctico.

4.1. Diseño de la base de datos

Para poder alcanzar los objetivos del presente proyecto es necesario comenzar todo el proceso del desarrollo con el diseño de una base de datos, esta permitirá alojar toda la información necesaria a los sistemas de administración de pacientes y posteriormente facilitará el análisis de los datos urológicos. Se ha elegido modelar una base de datos relacional con tecnología SQL.

4.1.1. Análisis de requerimientos

Para el proyecto se cuenta con un repositorio con 2610 prescripciones médicas con información de pacientes, sus historias clínicas y las consultas a las que asistieron. El primer paso en la recolección de requerimientos es el análisis por inspección de las recetas médicas, esto con el objetivo de tener un primer contacto con el dominio de los datos que deben almacenar en el repositorio.

La revisión de la información de las recetas ayuda a profundizar en la terminología utilizada en el área de Urología y a detectar las necesidades que facilitan la comprensión de las reglas del negocio.

La entrevista al experto del área es de gran utilidad para resolver las dudas que han surgido con el análisis que se menciona en el párrafo anterior. De la entrevista con el experto se obtiene el siguiente planteamiento que describe las características detalladas que debe abarcar la base de datos a diseñar.

Se necesita modelar una base de datos para un sistema que debe mantener información de pacientes y todos los detalles necesarios de las consultas a las que asisten, la descripción del proceso que el médico lleva a cabo cuando atiende pacientes se resume en los siguientes puntos.

- Cuando el paciente acude por primera vez al médico, se da de alta su información personal (nombre, dirección, género, edad, RFC, email, ocupación, estado civil, el responsable del paciente y por quien fue referido) además de guardar su información de identificación es necesario dar de alta su historia clínica. En la historia clínica debe haber antecedentes heredofamiliares, antecedentes personales no patológicos, antecedentes personales patológicos y antecedentes Urológicos.
- Cuando un paciente acude al médico por un padecimiento se inicia una línea de seguimiento del caso, es decir, todas las consultas posteriores a ella son consideradas subsecuentes de la inicial y solo están dedicadas a tratar el padecimiento inicial. Si el paciente presenta otro padecimiento se le inicia otra rama de tratamiento, por lo tanto, las consultas subsecuentes solamente tienen que registrar datos básicos de la evolución del caso junto con la fecha.
- En el sistema hay dos roles de acceso a la información, el administrador, quien puede realizar cualquier cambio en la información y el auxiliar, quien solamente puede dar de alta fichas de identificación de los pacientes y ver su información

básica de contacto.

- Para todo tipo de consultas debe ser posible registrar una impresión diagnóstica, un plan de tratamiento y una solicitud de estudios, pero solo las consultas primarias pueden tener el padecimiento o padecimientos actuales del paciente y el motivo o motivos de la consulta.
- Una consulta (subsecuente o inicial) puede solicitar estudios de laboratorio o de imagenología, por ello, es necesario que sea posible almacenar imágenes y/o resultados de los estudios en texto.

Las conclusiones obtenidas del análisis por simple inspección de las recetas y el enunciado anterior, respecto a las reglas del negocio que deben regir al sistema, dan paso al diseño conceptual de la base de datos.

4.1.2. Diseño conceptual

El diagrama Entidad Relación Extendido que se muestra en la figura 4.1 el resultado del diseño conceptual de la base de datos, este propone que para modelar los datos de los expedientes clínicos hacen falta 14 entidades con sus respectivas relaciones. A continuación, se enlistan las entidades expuestas en la figura 4.1 junto con una explicación detallada:

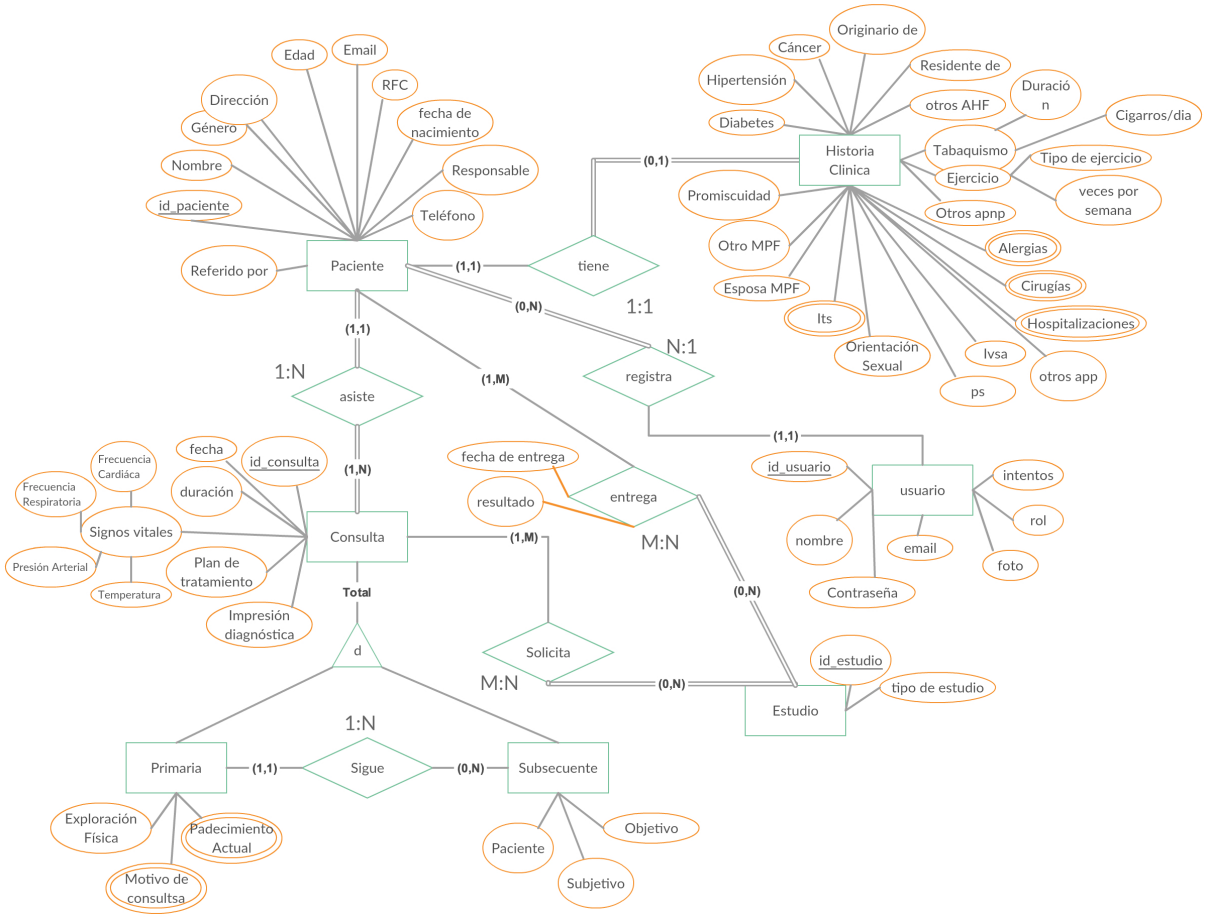


Figura 4.1. Diagrama Entidad Relación Extendido de la base de datos clínicos.

- Paciente: La entidad paciente almacena solamente la información de identificación de los pacientes, está involucrada en la relación "tiene" con la entidad Historia Clínica, la relación es de uno a uno puesto que un paciente solo puede tener una historia clínica y una historia clínica solo puede ser de un paciente, la participación del paciente en la relación "tiene" es parcial porque un paciente puede no tener historia clínica, también participa en la relación "asiste" con la entidad Consulta, dicha relación es de uno a muchos puesto que un paciente puede asistir a muchas consultas, pero a una consulta solo puede asistir un paciente, la participación del

paciente en la relación es total porque no puede haber consulta sin un paciente.

- Historia Clínica: Es una entidad que tiene como atributos los antecedentes clínicos del paciente, en esta entidad son destacables los atributos multievaluados: tipos de cáncer, tipos de alergias, tipos de cirugías, motivos de hospitalizaciones y tipos de infecciones de transmisión sexual.

- Consulta: Esta entidad tiene una relación de especificación con la entidad primaria y subsecuente, esto se modela así por la jerarquía que está descrita en las reglas del negocio, además, se ha establecido como una especificación disjunta porque un miembro de la entidad consulta solo puede ser mapeada para ser consulta primaria o subsecuente no ambas a la vez. La restricción de cobertura de la especificación se ha elegido como total debido a que no existe el caso en que una consulta sea algo diferente a primaria o subsecuente. La entidad Consulta está involucrada en las relaciones "solicita" y "entrega" ambas de N a M las cuales a su vez están relacionadas con la Entidad estudio, se ha modelado así puesto que en una consulta se pueden solicitar cero o muchos estudios de laboratorio e imagenología y en una consulta pueden entregarse resultados de cero o muchos estudios de laboratorio e imagenología.

  - Primaria: Una consulta primaria es la única que debe registrar la exploración física, los padecimientos actuales y los motivos de la consulta por ello se han puesto los atributos mencionados únicamente en la entidad Primaria.
  - Subsecuente: La entidad subsecuente almacena los atributos que solo una consulta subsecuente puede tener, además la consulta subsecuente participa en la relación "sigue" con cardinalidad 1 a N junto con la entidad Primaria, en dicha relación se establece que una consulta primaria es seguida por cero o muchas consultas subsecuentes, pero una consulta subsecuente solo puede seguir a una primaria.

- Estudio: La entidad estudio participa en las relaciones "solicita" y "entrega", relaciones en las que también la entidad consulta está involucrada; la participación

de “Estudio” en ambas relaciones es total porque no puede haber una solicitud o entrega de resultados de estudio sin una consulta.

- Usuario: La entidad usuario representa a la persona que interactúa con el sistema, en dicha entidad se define el atributo rol y tipo de acceso que tendrá, además, Usuario participa en la relación “registra” con la entidad Paciente, con esto se modela que un usuario puede registrar muchos paciente pero un paciente solo puede ser registrado por un usuario.

Las entidades: cáncer_, hospitalización_, cirugía_, alergia_, e its_ están relacionadas con la entidad Historia Clínica, mientras que las entidades motivo_consulta y padecimiento_actual están relacionadas con la entidad Primaria, dichas entidades representan características que puede o no haber en una historia clínica o una consulta primaria respectivamente, además estas características pueden ser cero o varias, por ejemplo, la entidad hospitalización_ representa los motivos por los cuales el paciente ha sido hospitalizado, puede darse el caso en el que un paciente haya sido hospitalizado por múltiples razones o que no haya sido hospitalizado nunca.

4.1.3. Diseño lógico

A continuación, se explicará el resultado del mapeo del diagrama Entidad Relación Extendido de la sección 2.1.2 al diseño lógico para la base de datos relacional que se necesita para la presente tesis. En la figura 4.2 se puede observar el modelo lógico de la base de datos.

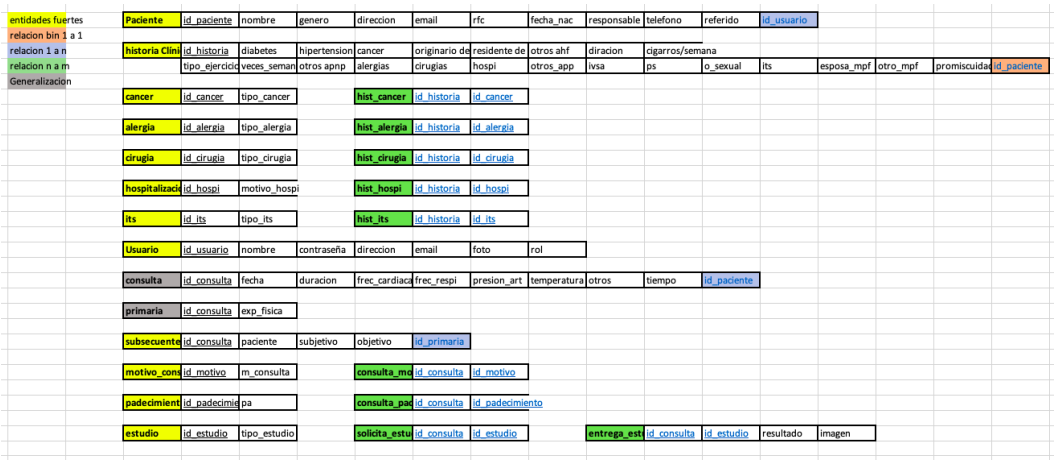


Figura 4.2. Modelo relacional de datos.

Tal y como se explicó en la sección 2.3.2 el procedimiento del mapeo de un diagrama EER al modelo lógico convierte cada tipo de relación, entidad y atributo a su equivalencia del modelo lógico, por lo que la figura 1 tiene acotaciones con código de color para identificar el motivo por el cual se generaron los elementos del modelo lógico, además todas las claves primarias están subrayadas y las claves foráneas están subrayadas y de color azul.

El modelo expuesto en la figura 4.2 se encuentra en Segunda Forma Normal debido a que en todos los atributos de todas las tablas la dependencia es total. Las tablas que corresponden a la acotación en color verde son las únicas en las que hay una llave primaria compuesta y todas a excepción de la tabla entrega_est carecen de atributos a parte de la llave primaria compuesta. Los atributos de la tabla entrega_est tienen una dependencia total de la llave primaria porque el resultado de un estudio solicitado depende de la consulta en que se solicitó y del tipo de estudio, este razonamiento aplica para la imagen del resultado de un estudio.

Para comprobar la Tercera Forma Normal en el modelo lógico se ha verificado que no existan dependencias funcionales transitivas y como no las hay se puede concluir

que se tiene una base de datos en modelo relacional en 3FN. La razón por la que no hizo falta aplicar ningún proceso de normalización al modelo creado es porque este se obtuvo a partir de un diagrama entidad relación.

4.1.4. Diseño físico

En esta última sección del diseño de base de datos se mostrará el resultado de todo el proceso de diseño de la base de datos, en la figura 4.3 está el diagrama obtenido de la conversión del modelo lógico realizado en la sección 3.1.3 al modelo físico. El *DBMS* elegido para la base de datos es MySql con la distribución MariaDB versión 10.1.25 para Linux, la herramienta utilizada para administrar la base de datos es MySql Workbench.

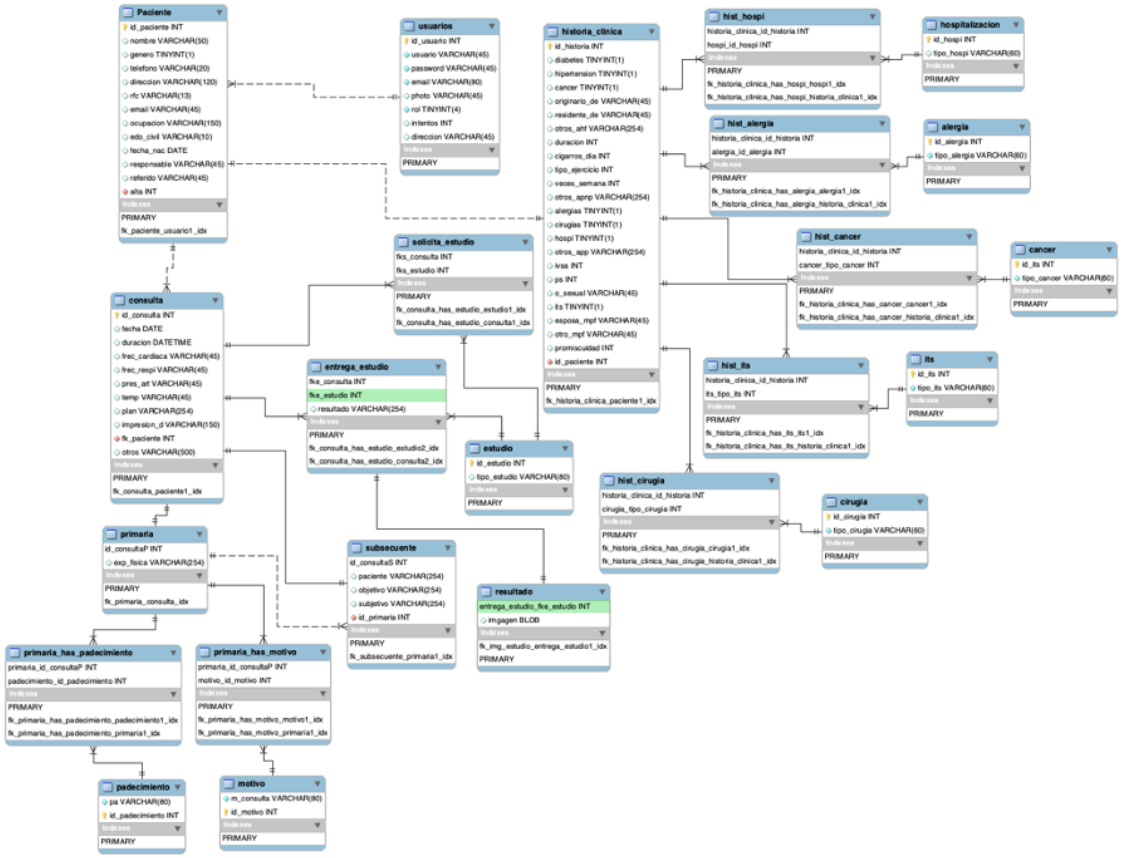


Figura 4.3. Base de datos en MariaDB.

En el diagrama de la figura 4.3 se aprecia el diseño físico de la base de datos a la cual nos referiremos de ahora en adelante como datos clínicos, la diferencia más notoria del diagrama físico y lógico es la incorporación de los tipos de datos de cada atributo.

Las reglas de comportamiento para las claves foráneas elegidas para datos clínicos son las enunciadas a continuación:

- Si un usuario es eliminado la clave foránea de paciente es asignada como nula, la modificación de un usuario propaga la modificación en todas las tuplas de la tabla pacientes.
- Si un paciente es eliminado también es eliminada toda información relacionada con él, como la historia clínica y sus consultas, la modificación de un paciente es propagada a la tabla historia clínica y a la tabla consulta.
- La eliminación de una consulta se propaga a la tabla de especificación, y de la tabla de especificación y las modificaciones serán propagadas a todas las tablas que hagan referencia a ella.
    - Si la consulta es primaria solo puede eliminarse si no hay una referencia directa a ella por una tupla de la tabla consulta subsecuente, en otras palabras, solo puede eliminarse una consulta primaria si no tiene consultas subsecuentes. La eliminación será propagada a todas las tablas que hagan referencia a la primera consulta.
    - Si la consulta es subsecuente se puede eliminar sin ninguna restricción.
- La eliminación de una tupla de historia clínica se propaga a todas las tablas que hacen referencia a ella, de igual manera sucede con la modificación.
- Las tablas, cáncer, alergia, cirugía, hospitalización, its, motivo y padecimiento están restringidas para eliminarse siempre y cuando no haya una referencia hacia ellas desde otra tabla, la modificación en dichas tablas se propaga.

4.2. Algoritmo de mapeo de prescripciones médicas

En esta sección se explicará a detalle cómo funciona el algoritmo que procesa las recetas médicas otorgadas para el desarrollo de la presente tesis. El algoritmo de mapeo es un *Script* de Python 3.7.3 programado en el IDE PyCharm 2019.1.1 en macOS Mojave.

Los objetivos generales del algoritmo son:

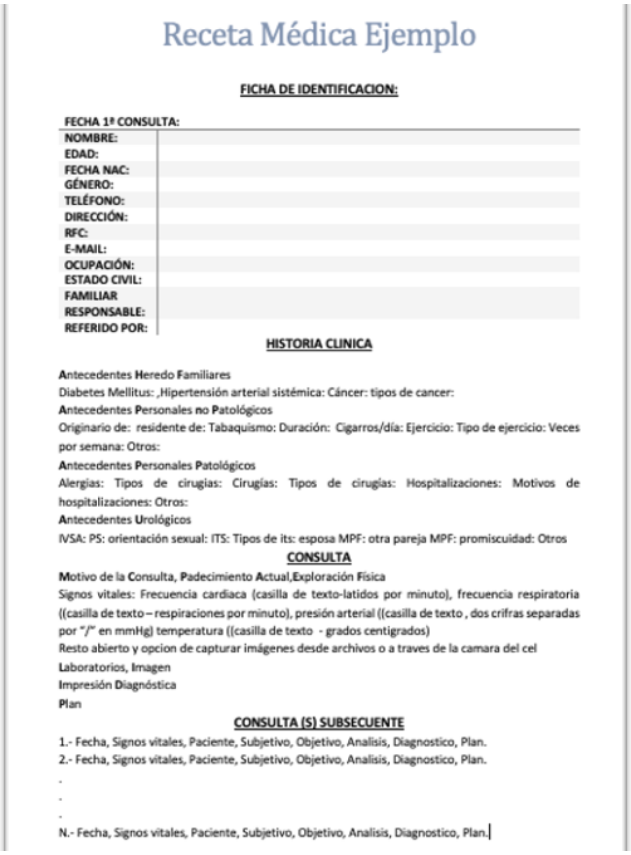
1. Hacer una búsqueda de los datos requeridos.
2. Estandarizar el formato de los datos.
3. Almacenar los datos en un diccionario.
4. Ingresar la información en la base de datos.

Los archivos del repositorio son documentos de Microsoft Word 2007-2019 (.doc/.docx) en los que está presente información de identificación del paciente, su historia clínica y la información de sus consultas, La figura 4.4 muestra la generalización de la estructura de todos los archivos del repositorio, misma que está establecida con base en la revisión de las recetas y la entrevista con el experto que las generó.

En la estructura de las recetas médicas del repositorio están presentes los datos en forma (clave, valor), solamente con el caso particular de que en las partes de historia clínica, consulta y consultas subsecuentes pueden aparecer varios valores para una sola clave (clave, valor1, valor2, ... , valorN), por lo tanto, el algoritmo debe estar preparado para tratar con ambos casos. Las bibliotecas que hacen posible el trabajo del script son:

- Scandir: Facilita la navegación y apertura de archivos en directorios.
- Docx2txt: Permite hacer una lectura de un documento de Microsoft Word y convertirlo en una cadena de Python.
- Time: Hace posible el conteo del tiempo que tarda la ejecución.
- Nltk: Utilizada para tokenizar oraciones.

- Re: Valida expresiones regulares.



Receta Médica Ejemplo

FICHA DE IDENTIFICACION:

FECHA 1ª CONSULTA:	
NOMBRE:	
EDAD:	
FECHA NAC:	
GÉNERO:	
TELÉFONO:	
DIRECCIÓN:	
RFC:	
E-MAIL:	
OCUPACIÓN:	
ESTADO CIVIL:	
FAMILIAR RESPONSABLE:	
REFERIDO POR:	

HISTORIA CLINICA

Antecedentes **H**eredo **F**amiliares
Diabetes Mellitus: ,Hipertensión arterial sistémica: Cáncer: tipos de cancer:
Antecedentes **P**ersonales **n**o **P**atológicos
Originario de: residente de: Tabaquismo: Duración: Cigarros/día: Ejercicio: Tipo de ejercicio: Veces por semana: Otros:
Antecedentes **P**ersonales **P**atológicos
Alergias: Tipos de cirugias: Cirugías: Tipos de cirugías: Hospitalizaciones: Motivos de hospitalizaciones: Otros:
Antecedentes **U**rológicos
IVSA: PS: orientación sexual: ITS: Tipos de its: esposa MPF: otra pareja MPF: promiscuidad: Otros

CONSULTA

Motivo de la **C**onsulta, **P**adecimiento **A**ctual,**E**xploración **F**isica
Signos vitales: Frecuencia cardiaca (casilla de texto-latidos por minuto), frecuencia respiratoria ((casilla de texto – respiraciones por minuto), presión arterial ((casilla de texto , dos crifras separadas por "/" en mmHg) temperatura ((casilla de texto - grados centigrados)
Resto abierto y opcion de capturar imágenes desde archivos o a traves de la camara del cel
Laboratorios, **I**magen
Impresión **D**iagnóstica
Plan

CONSULTA (S) SUBSECUENTE

1.- Fecha, Signos vitales, Paciente, Subjetivo, Objetivo, Analisis, Diagnostico, Plan.
2.- Fecha, Signos vitales, Paciente, Subjetivo, Objetivo, Analisis, Diagnostico, Plan.
.
.
.
N.- Fecha, Signos vitales, Paciente, Subjetivo, Objetivo, Analisis, Diagnostico, Plan.

Figura 4.4. Receta médica generalizada

A continuación se presenta el fragmento más importante del algoritmo en el que se procesa una receta. Considere las siguientes afirmaciones:

1. Candidatos: Arreglo bidimensional nx4, es la estructura de datos encargada de almacenar cada palabra del resultado de la tokenización de un fragmento de texto a procesar, también almacena características que ayudan al algoritmo a determinar qué tipo de palabra es.

   a) Columna 1: almacena una palabra.

b) Columna 2: define qué tipo de subcadena es, 0 es complementaria, 1 válida y 2 inválida.

c) Columna 3: almacena la etiqueta de la palabra con el objetivo de poderla identificar en todo momento.

d) Columna 4: contiene la cadena encontrada en el proceso de búsqueda.

2. deseadas: Arreglo bidimensional nx2 que contiene en su primera columna la expresión regular que ayuda a encontrar una palabra clave, en la segunda columna está almacenada la etiqueta de la coincidencia que se desea encontrar con la expresión regular.

3. noDeseadas: Arreglo unidimensonal que contiene expresiones regulares de cadenas que ayudan al algoritmo a encontrar con una mayor precisión solo la información que se necesita.

4. limpiadoras: arreglo unidimensional que contiene expresiones regulares de cadenas de texto que solamente limpiarán el resultado final (limpiadoras es opcional).

```
%1 reconocimiento de claves utiles y claves no utiles
para i hasta candidatos.tam hacer:
        band := falso
        j := 1
        mientras band = falso y j < deseadas.tam hacer
           :
                patron := compilarExpReg(deseadas[j
                   ][1])
                resultado := patron.buscar(candidatos[
                   i][1])
                si resultado = verdadero
                        candidatos[i][2] := 1
```

```
                    candidatos[i][3] := deseadas[j
                        ][2]
                    candidatos[i][4] := resultado.
                        encontrada()
                    band := verdadero
            fin si
            j := j + 1
    fin mientras
    %reconocimiento de claves no utiles
    band := falso
    j := 1
    mientras band = falso y j < noDeseadas.tam
        hacer:
            patron := compilarExpReg(noDeseadas[j
                ])
            resultado := patron.buscar(candidatos[
                i][1])
            si resultado = verdadero hacer:
                    candidatos[i][2] := 2
                    k := i + 1
                    mientras k < candidatos.tam y
                        candidatos[k][2] = 0 hacer:
                            candidatos[k][2] := 2
                            k := k + 1
                    fin mientras
                    band := verdadero
            fin si
            j := j +1
    fin mientras
```

```
fin para
%2 eliminacion de entradas no necesarias.
conta := 0
para i hasta candidatos.tam hacer:
        si candidatos[i-conta][2] = 2 hacer:
                candidatos.eliminar(i-conta)
                conta := conta +1
        fin si
fin para
%3 analisis
Para i hasta candidatos.tam hacer:
        si candidatos[i][2] = 1 hacer:
                patron := compilarExpReg('([Nn]eg([o]|
                   ad[ao]s {0,1} ). {0,1} )|([Nn]iega)
                   . {0,1}')
                resultado := patron.buscar(candidatos[
                   i][1])
                si resultado = verdadero hacer:
                        candidatos[i][1] := Nulo
                sino
                        aux := candidatos[i][1]
                        aux := aux.reemplaza(
                           candidatos[i][4],'')
                        %buscamos las que limpiaran
                           la cadena
                        para j hasta limpiadoras.tam
                           hacer:
                                patron :=
                                   compilarExpReg(
```

```
                                    limpiadoras[j])
                                resultado := patron.
                                    buscar(aux)
                                si resultado =
                                    verdadero hacer:
                                        aux := aux.
                                            reemplazar(
                                            resultado.
                                            encontrada
                                            (),'')
                                fin si
                            fin para
                            %se buscaran los completos
                            k := i +1
                            mientras k<candidatos.tam y
                               candidatos[k][2] = 0 hacer:
                                    aux2 := candidatos[k
                                        ][1]
                                    %buscamos las
                                        limpiadoras y las
                                        eliminamos
                                    aux := aux+','+aux2
                                    k := k +1
                            fin mientras
                            candidatos[i][0] := aux
                    fin si
        fin si
        Fin para
```

La idea fundamental del algoritmo enunciado consiste en que a partir de una cadena de texto que contenga la información que se desea extraer, se construya una estructura de datos capaz de almacenar todas las palabras de la cadena de texto junto con información que se necesita para reconstruir, reordenar y depurar las subcadenas de la forma $(clave, valor1, valor2, \ldots, valorN)$ que contienen la información deseada. A continuación, se presenta un ejemplo que explicará de manera más concreta qué recibe, cómo lo procesa y qué retorna el algoritmo.

Considere el siguiente enunciado que hace referencia a un ejemplo de la descripción que puede encontrarse en una de las prescripciones médicas, con referencia a los antecedentes personales patológicos de un paciente:

"alergias negadas, cirugía de apéndice, vesícula, quiste tirogloso, crónicas: artritis, epilepsia y esclerosis múltiple, medicamento suministrado permanentemente prednisona 5mg. tratamiento actual contra el acné vulgar con doxiciclina 50mg. resto negado"

En el esquema datos_clinicos de la sección 3.1.4 las tablas encargadas de almacenar la historia clínica de un paciente, requieren solo 3 datos de los antecedentes personales patológicos de un paciente, las alergias, las cirugías y las hospitalizaciones; por lo tanto, se mostrará la configuración del algoritmo para extraer dicha información.

```
deseadas := [('[Aa]lergi[ao]s{0,1}', 'alergias'),
                ('([Qq]uir[u]rgico(s){0,1})|([Cc]irug[i]a(s)
                    {0,1})', 'cirugias'),
              ('[Hh]ospitalizaci[o](n|es)', 'hospi')]

noDeseadas := ['resto', '[Cc]r[o]nic[ao]s{0,1}', '[Tt]rata((
    mientos{0,1})|(ndo))', 'medicamentos{0,1}']

limpiadoras := ['[Pp][Oo][Ss][Ii][Tt][Ii][Vv][OoAa][Ss
```

```
]{0,1}.{0,1}']
```

Tabla 4.1

Estado de la estructura de datos después de reconocer claves útiles y no útiles

Número	Frase	Tipo	Etiqueta	Encontrada
1	Alergias negadas	1	Alergias	Alergia
2	Cirguría de apéndice	1	Cirugías	Cirugía
3	Vesícula	0	-1	”
4	Quiste tirogloso	0	-1	”
5	Crónicas: artritis	1	-1	”
6	Epilepsia y esclerosis múltiple	1	-1	”
7	Medicamento suministrado permanentemente prednisona 5mg	1	-1	”
8	Tratamiento actual contra el acné vulgar con doxicilina 50mg	1	-1	”
9	Resto negado	1	-1	”

Cuando la ejecución completa el paso 1 la estructura de datos tiene el estado que se muestra en la tabla 4.1, la columna “Tipo” ya tiene etiquetadas las entradas de la estructura de datos que no son necesarias, indicadas con el número 2, que para este caso particular son la mayoría.

Tabla 4.2

Estado de la estructura de datos después de la eliminación de entradas.

Número	Frase	Tipo	Etiqueta	Encontrada
1	Alergias negadas	1	Alergias	Alergia
2	Cirguría de apéndice	1	Cirugías	Cirugía
3	Vesícula	0	-1	”
4	Quiste tirogloso	0	-1	”

Después de la eliminación de las entradas innecesarias la estructura de datos se tiene el estado descrito en la tabla 4.2 y ahora está preparada para pasar a la última etapa.

Tabla 4.3

Estado de la estructura de datos al finalizar el algoritmo. .

Número	Frase	Tipo	Etiqueta	Encontrada
1	NULO	1	Alergias	Alergia
2	de apéndice, vecícula, quiste tirogloso	1	Cirugías	Cirugía
3	Vesícula	0	-1	”
4	Quiste tirogloso	0	-1	”

La tercera etapa del proceso concatena con una coma todas las entradas válidas de la estructura de datos con las entradas posteriores inmediatas establecidas como indefinidas, es por esta razón que a la entrada número dos se le concatenaron las entradas 3 y 4 que estaban marcadas por la columna “Tipo” (Tabla 4.3) como indefinidas, a parte, la entrada 1 fue asignada como NULO porque el enunciado original indicaba negación.

La estructura de datos de la tabla 4.3 es la que retorna el algoritmo para que en un

módulo externo obtenga la columna “frase” mediante la cadena de la columna etiqueta esto con el objetivo de que pueda ser generalizado para funcionar en diversos casos y con múltiples formatos de archivos clínicos.

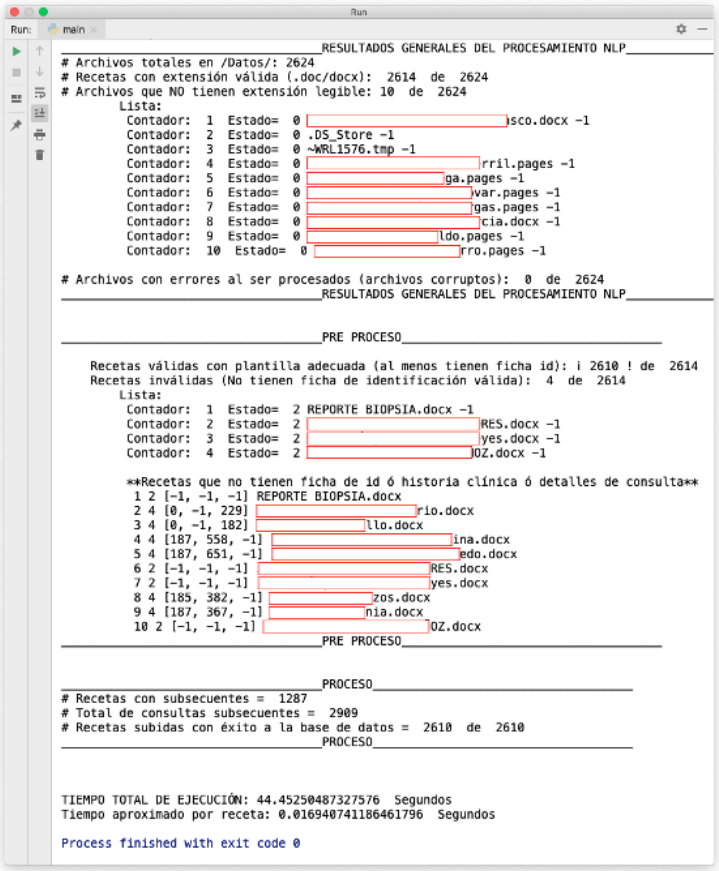


Figura 4.5. Resultados del algoritmo.

En la figura 4.5 se tiene el LOG de la ejecución del algoritmo en la consola de PyCharm. En el repositorio de archivos se tienen 2624 archivos de los cuales 10 no están en el formato adecuado de Microsoft Word, y otros 4 no tienen una estructura mostrada en la figura 4.4, por tanto, otros 14 archivos fueron descartados lo que deja un total de 2610 archivos procesados.

4.3. Sistema web y aplicación móvil de gestión de pacientes

El ingreso de la información de los expedientes clínicos a la base de datos a partir del algoritmo propuesto en la sección 4.2, es útil para los expedientes que ya se encuentran en ficheros, sin embargo, tiene dos limitantes; la imposibilidad de separar campos muy específicos que no fueron considerados al momento de su diseño y la falta de practicidad para estarlo ejecutando cada vez que se tengan datos actualizados de uno o más pacientes.

Para resolver el problema de administración de expedientes clínicos a la base de datos se propone un sistema web junto con una aplicación móvil que deben cumplir con las siguientes características:

- Permitir la inserción, actualización y borrado de cualquier expediente clínico.
- Mostrar Estadísticas básicas de la base de datos.
- Ser accesible desde cualquier parte del mundo.
- Garantizar la seguridad de acceso a la información.
- Ser más práctico que la edición directa de expedientes en un fichero en Microsoft Word.

La figura 4.6 muestra un servidor web y un servidor de bases de datos al cual se puede acceder desde internet a través de un navegador web de una computadora personal o desde un dispositivo móvil con una aplicación nativa. La arquitectura del sistema web es de tipo cliente–servidor en donde el cliente puede ser una computadora personal y/o un dispositivo móvil.

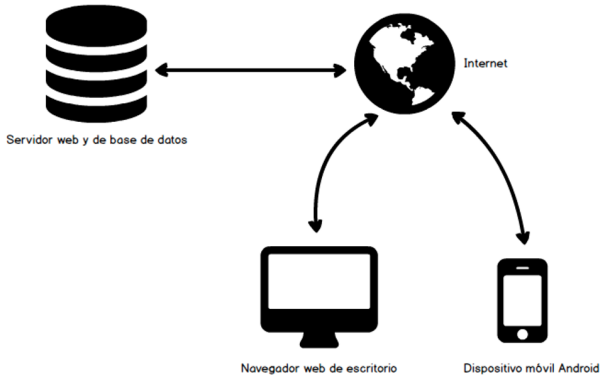


Figura 4.6. Arquitectura del sistema web y aplicación móvil.

A continuación, se describen las características más importantes que constituyen al sistema de gestión de pacientes en términos de los dos tipos de cliente:

- Aplicación web para escritorio (figura 4.7): Desarrollada con el MVC (Modelo Vista Controlador) consta de dos partes, *Frontend*, es la parte de la aplicación que interactúa con el usuario, se encuentra desarrollada en la biblioteca Bootstrap 4 (Bootstrap, 2020), esta parte de la aplicación web se ejecuta en el navegador web del usuario; *Backend*, es la parte de la aplicación web que se ejecuta en el servidor y el usuario no interactúa de manera directa, está desarrollada en lenguaje PHP en su versión 7.4 (PHP Hypertext Preprocessor, 2019) y es a este nivel en done se hace la conexión con la base de datos creada en la sección 4.1.

- Aplicación Móvil (figura 4.8): Desarrollada para dispositivos móviles *Android* utilizando el MVP (Modelo Vista Presentador) con el soporte de la *Api* 29 de Google para lenguaje JAVA (Android Developers, 2019), para consumir servicios web se utiliza la biblioteca Retrofit 2.

Figura 4.7. Panel de control del sistema web: pantalla de alta de nuevo paciente y pantalla de alta de nueva consulta.

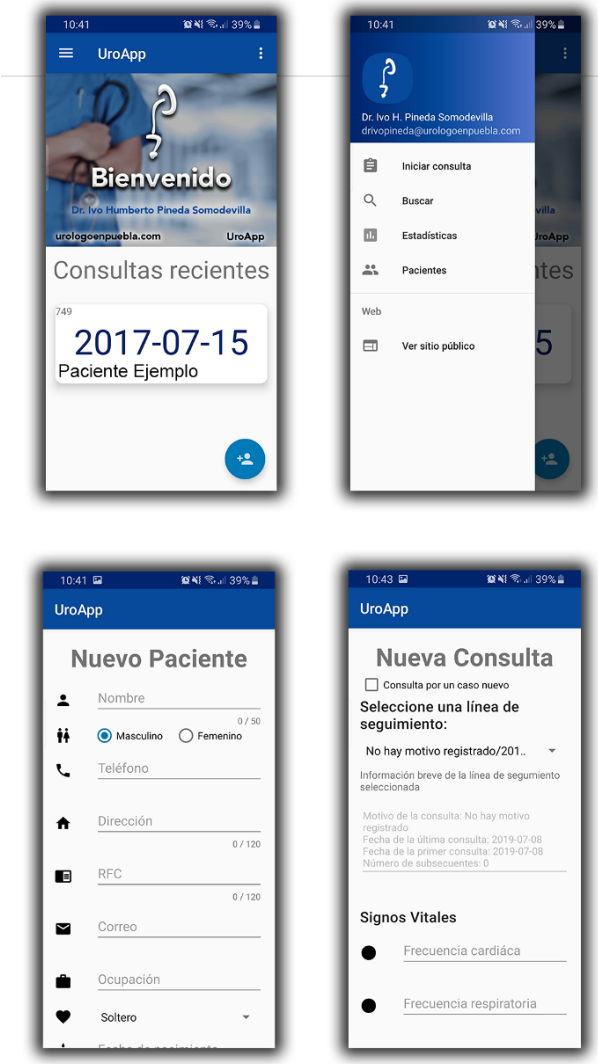


Figura 4.8. Panel de control del sistema web: pantalla de alta de nuevo paciente y pantalla de alta de nueva consulta.

Ambas aplicaciones interactúan con el mismo servidor y base de datos, esto permite que la información que se agregue desde cualquiera de las dos aplicaciones se actualice en tiempo real y que se pueda tener acceso a ellas desde cualquier parte del mundo en donde haya conexión a internet.

4.4. Análisis de datos

A lo largo de las secciones 3.1 a la 3.4 se han explicado los procesos con los que se han tratado los datos originales proporcionados para la presente tesis, esto con el fin de poder realizar los objetivos establecidos en la sección 1.1.2, por tanto, ahora es oportuno comenzar con la etapa de análisis de la información siguiendo los pasos de la metodología *KDD* junto con las tareas específicas descritas a continuación:

1. Selección
    - *a*) Elaborar una consulta SQL para extraer los atributos que a priori parezcan útiles para conseguir los objetivos.
    - *b*) Exportar la consulta a un archivo legible para las herramientas de aprendizje automático utilizadas en las siguientes etapas.
2. Preprocesamiento
    - *a*) Homogeneizar los valores que toman los atributos.
    - *b*) Establecer estrategias para el tratamiento de datos faltantes.
    - *c*) Eliminar valores atípicos que degradan la precisión de los algoritmos.
    - *d*) Decidir si se descartan atributos.
3. Transformación
    - *a*) Seleccionar atributos relevantes
    - *b*) Reducción de dimensionalidad.
4. Minería de Datos
    - *a*) Experimentar con algoritmos de agrupamiento, reglas de asociación y correlaciones.
    - *b*) Experimentar con algoritmos de clasificación y regresión.

5. Evaluación

   a) Ponderar los resultados obtenidos en todo el proceso.

   b) Dar una conclusión de los resultados.

4.4.1. Selección de datos

Para obtener el conjunto de datos e iniciar el proceso de análisis es necesario ejecutar el *script* SQL enunciado en la figura 4.9, éste obtiene datos de las tablas: Paciente, historia_clinica e Impresión_diagnostica. Los únicos atributos que han sido ignorados son datos de contacto de los pacientes tales como los nombres, números de teléfono y correos electrónicos.

En las líneas 6 a 9 de la figura 4.9 se observa un conjunto de instrucciones que efectúa un proceso de exportación de los resultados obtenidos a un archivo con extensión .csv, este será legible para las herramientas de aprendizaje automático que se utilizan en las posteriores etapas de esta sección.

```
1  SELECT Paciente.nombre, Paciente.genero,Paciente.nombre, Paciente.direccion, Paciente.fecha_nac, Paciente.ocupacion,
2      Paciente.referido_por, historia_clinica.*, consulta.fecha_consulta, TIMESTAMPDIFF(YEAR,  Paciente.fecha_nac, consulta.fecha_consulta) as edad
3  FROM Paciente
4  LEFT JOIN historia_clinica on Paciente.id_paciente = historia_clinica.id_paciente
5  LEFT JOIN consulta on Paciente.id_paciente = consulta.id_paciente
6  INTO OUTFILE  './pacientes.csv'
7  FIELDS TERMINATED BY ','
8  OPTIONALLY ENCLOSED BY ''
9   LINES TERMINATED BY '\n';
```

Figura 4.9. Script SQL con instrucciones de exportación a archivo separado por comas.

Otro conjunto de datos necesario para enriquecer el análisis de los expedientes clínicos es el de la zona territorial del estado de Puebla (INEGI, 2016) , en él se encuentra el estado dividido por Municipios, Localidades y AGEB, con dicho conjunto de datos es posible hacer una geolocalización de los pacientes para asignarles un AGEB y mejorar la interpretación de los resultados finales.

La descripción general del conjunto de datos principal es presentada a continuación (Tabla 4.4) junto a una breve descripción de cada uno de los atributos.

Tabla 4.4

Descripción de los atributos del conjunto de datos.

Nombre del campo	Tipo de atributo	Frecuencia	Descripción
genero	Categórico	2	Sexo biológico del paciente.
ocupacion	Categórico	268	Actividad laboral del paciente.
diabetes_mellitus	Categórico	2	¿Ha tenido familiares con diabetes tipo 2?
hipertensión_arteria	Categórico	2	¿Ha tenido familiares con hipertensión arterial sistémica?
cancer	Categórico	2	¿Ha tenido familiares con cáncer?
originario_de	Categórico	298	Municipio y entidad federativa de nacimiento
residente_de	Categórico	176	Municipio y entidad federativa en donde actualmente erradica
drogas	Categórico	2	¿Consume algún tipo de drogas ilegales?
alcoholismo	Categórico	2	¿Consume alcohol?
duracionAlcohol	Entero	–	Tiempo en años que lleva consumiendo alcohol
alcoholSemana	Entero	–	Número de días a la semana que consume alcohol.

Continúa en la siguiente página

Tabla 4.4 – Continuación de la página anterior

Nombre del campo	Tipo de atributo	Frecuencia	Descripción
tabaquismo	Categórico	2	¿Es fumador activo?
duracion_t	Entero	–	Tiempo en años como fumador activo
cigarros_dia	Entero	–	Número de cigarros que consume al día
ejercicio	Categórico	2	¿Realiza alguna actividad física?
veces_semana	Entero	–	Número de días que realiza actividad física
alergias	Categórico	2	¿Tiene algún tipo de alergia?
cirugias	Categórico	2	¿Ha sido intervenido quirúrgicamente?
hospitalizaciones	Categórico	2	¿Ha sido hospitalizado?
dm2p	Categórico	2	¿Padece de diabetes tipo 2?
hasp	Categórico	2	¿Padece de hipertensión arterial sistémica?
ivsa	Entero	–	Edad de inicio de vida sexual activa
ps	Entero	2	Número de parejas sexuales
o_sexual	Categórico	3	Orientación sexual
its	Categórico	2	¿Ha tenido alguna infección de transmisión sexual?
esposa_mpf	Categórico	82	Método anticonceptivo que utiliza con su pareja estable.

Continúa en la siguiente página

Tabla 4.4 – Continuación de la página anterior

Nombre del campo	Tipo de atributo	Frecuencia	Descripción
otra_pareja_mpf	Categórico	26	Método anticonceptivo que utiliza con otra pareja sexual.
promiscuidad	Categórico	2	¿Cambia frecuentemente de pareja sexual?
edad	Entero	–	Edad del paciente al recibir la consulta.
direccion	Cadena	–	Dirección del domicilio de residencia del paciente.

4.4.2. Preprocesamiento

La importancia de la limpieza de los datos se ve reflejada en las etapas finales del proceso *KDD*, puesto que entrenar un modelo de aprendizaje automático con un conjunto de datos pre procesado incorrectamente podría incurrir en conclusiones incorrectas y eso podría influir negativamente en la toma de decisiones. A continuación, se presenta la estrategia utilizada para mejorar la calidad de los datos:

- Atributos categóricos no binarios:
  1. Se aplicaron filtros que reducen la cantidad de valores posibles (Figura 4.10).
  2. Se decidió junto con el experto una equivalencia de valores para unificar los repetidos.
- Atributos numéricos:
  1. Se eliminaron valores atípicos con la ayuda del experto utilizando la inspección simple.

2. Se visualizaron histogramas y se convirtieron todas las distribuciones sesgadas a una distribución normal.
3. Se eliminaron valores atípicos con las funciones matemáticas Z-Score e IQR.

- Direcciones de pacientes:

1. Se utilizó el fragmento de *script* de la figura 4.11 para convertir las direcciones al sistema de coordenadas geográficas ESPG:4326.

```
In [10]: # 5
def limpiar_valores(cadena):
    regex = r"([^n\u0300-\u036f]|n(?!\u0303(?![\u0300-\u036f])))[\u0300-\u036f]+"
    aux = cadena
    if type(aux) == float:
        retorno = cadena
    else:
        # primero se separan las cadenas y se toma la primera
        arr = aux.split(" ")
        aux = arr[0]
        # ahora se quitan espacios finales e iniciales, puntos y comas
        aux = aux.strip(" .,")
        # ahora se toma la cadena y se obtiene el grafema base, (si recibe la Á se convierte en A)
        aux = re.sub(regex, r"\1", normalize("NFD", aux), 0, re.I)
        # Ahora se convierte todo a minúsculas
        retorno = aux.lower()

    return retorno

In [11]: # 5
ds["ocupacion"] = ds["ocupacion"].apply(limpiar_valores)
```

Figura 4.10. Fragmento de código en Python que hace una limpieza de un atributo en un conjunto de datos.

```
In [47]: def geocoding(dir):
    geocode_result = gmaps.geocode(dir)
    if len(geocode_result) > 0:
        lat = geocode_result[0]['geometry']['location'] ['lat']
        long = geocode_result[0]['geometry']['location']['lng']
    else:
        lat = np.nan
        long = np.nan
    return [lat, long]

In [49]: pacientes_geo[['lat', 'long']] = pacientes_geo.apply(lambda row: pd.Series(geocoding(row['dirección'])), axis=1)
```

Figura 4.11. Fragmento de código en Python que obtiene las coordenadas geográficas dada una dirección.

Debido a que se encontraron atributos que contienen altos porcentajes faltantes, se tomó la decisión junto con el experto de desechar los atributos: otra_pareja_mpf y esposa_mpf. En la figura 5.1 se indican los porcentajes de valores faltantes por atributo.

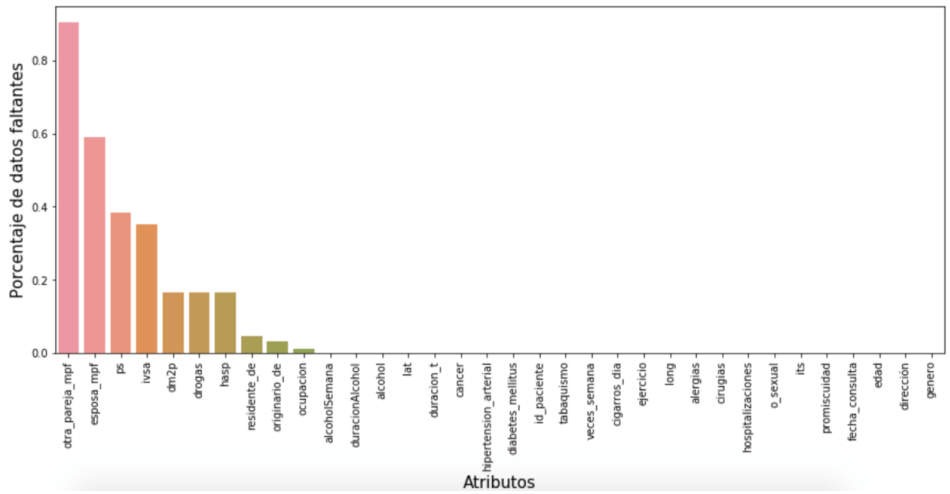


Figura 4.12. Histograma de datos faltantes por atributo.

Capítulo 5

Evaluación de la propuesta y resultados

Como continuación del proceso *KDD* iniciado en el capítulo 4, en este capítulo se mostrarán las dos etapas siguientes: la transformación del conjunto de datos y la aplicación de los algoritmos de aprendizaje automático, en esta última se dará la explicación del flujo de proceso de los datos y los parámetros de los algoritmos aplicados, así mismo se indicarán cuáles fueron las razones de la elección de dichos parámetros.

La propuesta para abordar el problema fue la de la creación de un modelo híbrido entre aprendizaje no supervisado y aprendizaje supervisado. El modelo de aprendizaje no supervisado encontró nuevos grupos de pacientes y posteriormente estos fueron utilizados para entrenar un modelo de aprendizaje supervisado para clasificar pacientes nunca vistos por ambos modelos.

5.1. Transformación

La transformación necesaria para la construcción del modelo inicial de agrupamiento estuvo constituida de la selección de atributos, de la cual, se obtuvo la siguiente lista:

Tabla 5.1

Atributos seleccionados para la generación de grupos de pacientes.

Número	Nombre
1	ocupacion
2	alcohol
3	alcoholSemana
4	tabaquismo
5	duracion_t
6	cigarros_dia
7	ejercicio
8	veces_semana
9	cirugias
10	hasp
11	ivsa
12	ps
13	o_sexual
14	its
15	promiscuidad
16	edad
17	long
18	lat
19	gmu

Se planeó y realizó un segundo experimento de agrupamiento de los pacientes con un algoritmo diferente, por ello, se seleccionó un conjunto de datos diferente, solo para las características Urológicas numéricas de los pacientes. En la figura 5.1 se tiene un ejemplo de los primeros 5 pacientes con los atributos seleccionados para este segundo experimento: edad, ivsa y ps.

```
In [242]: 1 ds_agrupado_jerarquia.head()
Out[242]:
```

	id_paciente	edad	ivsa	ps
2	906.0	28.0	19.0	4.0
3	727.0	22.0	18.0	5.0
5	1320.0	39.0	20.0	30.0
6	160.0	27.0	18.0	3.0
7	98.0	27.0	18.0	9.0

Figura 5.1. Conjunto de datos para el segundo experimento de agrupamiento.

5.2. Minería de datos e interpretación

Después de tener el conjunto de datos preprocesado adecuadamente se procedió a la etapa en la que los algoritmos de aprendizaje automático se aplican, se validan los resultados y se da una interpretación a éstos.

5.2.1. Agrupamiento

En la aplicación del algoritmo de agrupamiento *KMeans* (figura 5.2)para el primer experimento se obtuvieron 3 grupos que son pertenecientes al grado de marginación del AGEB en el que se encuentra el paciente.

En el primer grupo es correspondiente a los pacientes que residen en un AGEB con un grado de marginalidad bajo, el promedio de edad de los pacientes es de 39.5 años. Las características más sobresalientes son que los pacientes tienen como principal actividad laboral el comercio, suelen fumar en promedio medio cigarro al día, en general han sido intervenidos quirúrjicamente. Su número promedio de parejas sexuales es de 17.5 e iniciaron su vida sexual activa a los 18 años y el centroide geográfico del grupo corresponde a la colonia El Carmen.

El segundo grupo de pacientes corresponde al grado de marginalidad muy bajo y en él se encuentran pacientes con un promedio de edad de 32 años, son estudiantes, suelen consumir alcohol al menos una vez por semana, no son fumadores activos, tienen actividad física muy esporádicamente, no han tenido intervenciones quirúrgicas y el centroide geográfico del grupo corresponde a la colonia Rivera de Santiago.

El último grupo de pacientes correspondiente al grado de marginación bajo corresponde a los pacientes que tienen un promedio de edad de 35 años, este grupo es empleados de alguna institución pública o privada, no consumen alcohol ni son fumadores, no han sido intervenidos quirúrgicamente, iniciaron su vida sexual activa a los 18 años y en

promedio han tenido 12 parejas sexuales, el centroide geográfico del tercer grupo se encuentra en el barrio de Analco.

```
kMeans
======

Number of iterations: 12
Within cluster sum of squared errors: 2426.6512636133107

Initial starting points (random):

Cluster 0: comerciante,No,0,Si,0,0,0,0,Si,No,17.858044,14.457096,Hetero,No,No,45,-97.8
Cluster 1: empleado,Si,0,No,0,0,0,0,No,No,18,10,Hetero,No,No,33,-98.224752,19.059552,'
Cluster 2: empleado,No,0,No,0,0,1,5,No,No,17,40,Hetero,No,Si,43,-98.25138,19.106909,Me

Missing values globally replaced with mean/mode

Final cluster centroids:
                                                Cluster#
Attribute                          Full Data          0          1          2
                                     (950.0)    (311.0)    (414.0)    (225.0)
=============================================================================
ocupacion                       estudiante comerciante  estudiante    empleado
alcohol                                 Si         No         Si         No
alcoholSemana                       0.0526     0.0193     0.1014     0.0089
tabaquismo                              No         Si         No         No
duracion_t                          0.5558     0.8585     0.4807     0.2756
cigarros_dia                        0.3179     0.4952      0.244     0.2089
ejercicio                           0.1853     0.0707     0.1787     0.3556
veces_semana                        0.1105     0.0482      0.058     0.2933
cirugias                                No         Si         No         No
hasp                                    No         No         No         No
ivsa                                17.858    17.8253    17.6486    18.2886
ps                                 14.4571    17.5608    13.0078    12.8338
o_sexual                            Hetero     Hetero     Hetero     Hetero
its                                     No         No         No         No
promiscuidad                            No         No         No         No
edad                                35.3674    39.5016    32.2874      35.32
long                               -98.2045   -98.2091    -98.208   -98.1919
lat                                 19.0389    19.0358    19.0426    19.0366
_GM_Base_marginacion_AGEB_00-10_GMU    Bajo       Bajo   Muy bajo       Bajo
```

Figura 5.2. Resultados del algoritmo K-means.

Ahora se presenta el segundo experimento de agrupamiento con la utilización del algoritmo *Cobweb* para agrupamiento jerárquico. Antes de ejecutar el algoritmo se normalizaron los atributos puesto que se compararon las edades con número de parejas sexuales. El dendrograma de la figura 5.3 indica el resultado del algoritmo, y mediante la técnica del codo se determinó que el número óptimo para este experimento fue de 4 grupos. En la figura 5.4 se aprecia en un diagrama de dispersión la separación de los 4 grupos encontrados.

```
In [474]:
1 dendrogram_tune(Z, truncate_mode='lastp',p=12, leaf_rotation=90.,
2                 leaf_font_size=12., show_contracted=True,
3                 annotate_above=10, max_d=1.9)
4 plt.show()
```

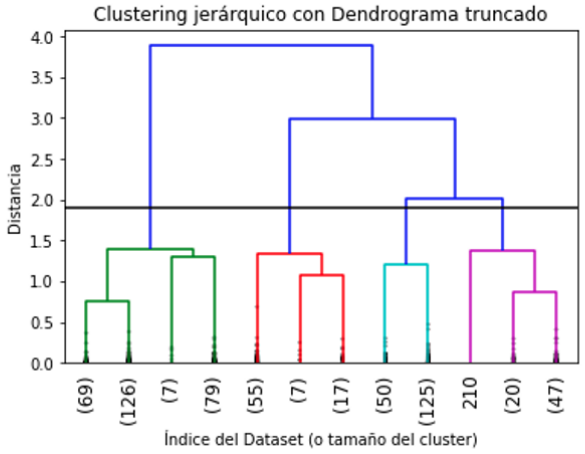


Figura 5.3. Dendrograma truncado de los grupos encontrados por el algoritmo de agrupamiento jerárquico aglomerativo.

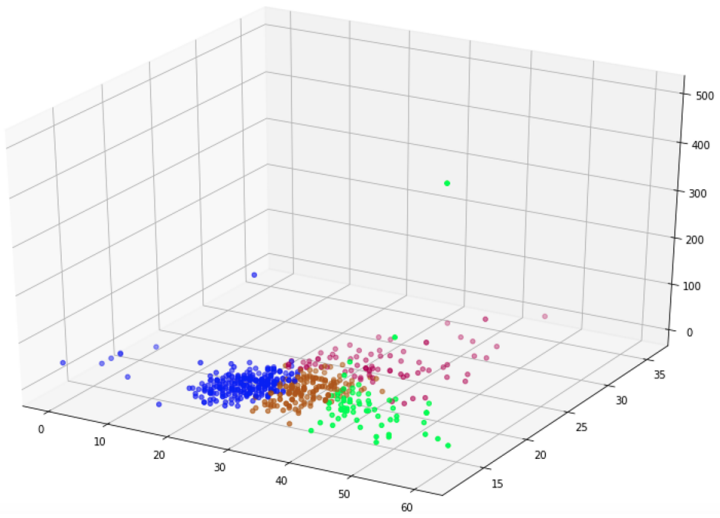


Figura 5.4. Gráfica de dispersión de los 4 grupos encontrados en el segundo experimento.

El primer grupo de pacientes corresponde a los pacientes menores de edad que iniciaron su vida sexual activa a los 16.7 años y tienen un promedio de 11.8 parejas sexuales. El segundo grupo corresponde a los pacientes con un promedio de 35 años, con inicio de vida sexual activa a los 23 años y un promedio de 10 parejas sexuales. En el tercer grupo entran los pacientes con 32 años que iniciaron su vida sexual activa a los 17 años y que tienen un promedio de 14.5 parejas sexuales. El último grupo es de los pacientes con un promedio de 43.8 años con inicio de vida sexual activa a los 16 años y que han tenido en promedio 31 parejas sexuales (figura 5.5).

```
In [432]: 1 ds_agrupado_jerarquia.groupby("cluster").mean()
```

Out[432]:

cluster	id_paciente	edad	ivsa	ps
1	947.804270	22.946619	16.768683	11.807829
2	905.481013	35.303797	23.607595	9.873418
3	900.388571	32.405714	17.480000	14.525714
4	955.367647	43.882353	15.985294	31.000000

Figura 5.5. Resultados de los grupos de pacientes del segundo experimento.

5.2.2. Clasificación

Ahora se presenta la segunda fase correspondiente a la etapa de minería de datos, en la sección 5.2.1 se obtuvo un conjunto con una nueva clasificación, ahora se mostrarán los resultados obtenidos de haberle proporcionado el conjunto de datos resultante al algoritmo de aprendizaje supervisado para que entrenar el modelo que permitió clasificar pacientes no vistos por el primer modelo. La creación del modelo de clasificación se consiguió gracias al algoritmo J48 el cual creo un árbol de decisión, los resultados del clasificador son los presentados en la figura 5.6

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         860               90.5263 %
Incorrectly Classified Instances        90                9.4737 %
Kappa statistic                          0.8493
Mean absolute error                      0.1037
Root mean squared error                  0.2305
Relative absolute error                 24.9756 %
Root relative squared error             50.5928 %
Total Number of Instances              950

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.982    0.048    0.896      0.982    0.937      0.911    0.980     0.942     cluster1
                 0.913    0.025    0.973      0.913    0.942      0.890    0.969     0.963     cluster2
                 0.776    0.061    0.768      0.776    0.772      0.712    0.899     0.636     cluster3
Weighted Avg.    0.905    0.039    0.908      0.905    0.906      0.860    0.958     0.889

=== Confusion Matrix ===

   a   b   c   <-- classified as
 277   0   5 |   a = cluster1
   0 431  41 |   b = cluster2
  32  12 152 |   c = cluster3
```

Figura 5.6. Resultados del algoritmo de clasificación.

El porcentaje de instancias clasificadas correctamente por el algoritmo es de 90.5 %, y gracias a que los grupos que el algoritmo de agrupamiento encontró estaban aceptablemente balanceados, el porcentaje de clasificación no se vio sesgado por esta condición, por esto, ahora se tiene un modelo de clasificación para los pacientes que se atiendan de ahora en adelante.

Capítulo 6

Conclusiones y trabajo a futuro

En esta tesis se presentó el diseño e implementación de todo un sistema de información para la toma de decisiones con base en aprendizaje automático del área de Urología. Las conclusiones del proyecto de investigación son:

- Para construir un repositorio SQL para representar datos urológicos es necesario, entender los procesos implicados en las consultas urológicas y seguir una metodología de diseño de bases de datos relacionales.
- Desarrollar una aplicación móvil y un sistema web es posible si se hace un levantamiento de requerimientos adecuado y que este corresponda con el esquema de una base de datos con el mismo objetivo, todo esto para garantizar que los sistemas cubran por completo las necesidades del usuario.
- Las técnicas de aprendizaje automático que son necesarias para generar patrones urológicos son los algoritmos de agrupamiento y los clasificadores con base en árboles de decisión, con ellos, es posible describir con mayor claridad grandes conjuntos de datos. Los algoritmos de aprendizaje no supervisado como K-Means y Cobweb son útiles para encontrar una forma de clasificar a los pacientes en función a múltiples características y esto permite una segmentación más compleja, escalable y matemáticamente respaldada. Los algoritmos de aprendizaje supervisado basados en árboles como J48 generan un modelo que ya no necesita ser entrenado después de generarlo para determinar el grupo al que corresponden los

pacientes nuevos.

Los modelos híbridos de aprendizaje supervisado y no supervisado trabajan en conjunto para obtener una segmentación de pacientes y tener un propio clasificador que permita ir etiquetando pacientes nuevos conforme se vayan dando de alta en la base de datos, sin la necesidad de pasar por todo el flujo que se revisó en el presente trabajo.

6.1. Trabajo a futuro

El repositorio de datos con el que se trabajó en la presente tesis tiene un crecimiento promedio de 30 pacientes por mes esto abre la oportunidad de que en el futuro se repitan las técnicas aplicadas para ir mejorando los modelos, así mismo llegará el momento en que los datos tendrán un volumen lo suficientemente grande para aplicar técnicas Deep learning que mejoren los resultados de los modelos.

En la base de datos se tienen más relaciones que contienen información de consultas y tratamientos, sin embargo, estas se encuentran escritas en lenguaje natural, para poder saber de qué padecimientos y diagnósticos se trata, será necesario utilizar las técnicas que el PLN (Procesamiento del Lenguaje Natural) estudia. El CIE-10 tienen un conjunto de datos de diagnósticos que están aceptados por la OMS, una continuación del presente proyecto puede ser la conversión del lenguaje de los expedientes médicos al estándar del CIE-10 para poder generalizar las categorías de los mismos y probar la clasificación de diagnósticos.

Bibliografía

Anaconda, Inc. (2019). The Anaconda Difference. Recuperado el 15 de Noviembre de 2019, de Anaconda.

Android Developers. (3 de Septiembre de 2019). Android Developers. Recuperado el 22 de Noviembre de 2019, de https://developer.android.com/about/versions/10/behavior-changes-10

Bernus, P., & Noran, O. (2017). Data Rich – but Information Poor.

Bootstrap. (20 de 1 de 2020). getbootstrap. Obtenido de https://getbootstrap.com

Caunet. (s.f.). Caunet.org. Recuperado el 22 de Noviembre de 2019, de ¿Qué es la urología?: https://caunet.org/que-es-la-urologia/

Chen, P. P.-S. (1976). The entity-relationship model—toward a unified view of data. ACM Transactions on Database Systems (TODS), 9-36.

Digital Equipment Corporation. (1977). Archive ECE. Recuperado el 3 de Mayo de 2019, de Electrical & Computer Engineering: http://www.archive.ece.cmu.edu/ẽce447/s13/lib/exe/fetch.php?media=vax_archhbkvol1_1977.pdf

Eibe Frank, M. A. (2016). The WEKA Workbench. Online Appendix. Obtenido de Data Mining: Practical Machine Learning Tools and Techniques: https://www.cs.waikato.ac.nz/ml/weka/

Elmasri, R., & Navathe, S. B. (2015). Fundamentals of Database Systems. Pearson.

FMed. (Abril de 2019). El médico y la máquina. Recuperado el 5 de Octubre de 2019, de Medium: https://medium.com/@infofmed/el-médico-y-la-máquina-b933771a22e7

Garcia Serrano, A. (2016). Inteligencia Artificial. Fundamentos Practica y Aplicaciones. Alfaomega.

Gautier, L. (2016). rpy2.readthedocs. Obtenido de https://rpy2.readthedocs.io/en/version_2.8.x/overview.html

Han, J., Kamber, M., & Pei, J. (2012). Data Mining Concepts and Techniques. Morgan Kaufmann.

Hernández Gómez, H. J. (2014). Aplicación de minería de datos a información de pacientes prediabéticos. Revista Iberoamericana de Producción Académica y Gestión Educativa.

Hernández Orallo, J., Ramírez Quintana, M. J., & Ferri Ramírez, C. (2004). Introducción a la minería de datos. Madrid: Pearson.

Hinton, G., & Sejnowski, T. J. (1999). Unsupervised Learning: foundations of neural computation. Cambridge.

INEGI. (2016). Biblioteca digital de mapas. Recuperado el 18 de Octubre de 2019,

de inegi.org: https://www.inegi.org.mx/app/biblioteca/ficha.html?upc=702825218881

J. Date, C. (2001). Introducción a los sistemas de bases de datos. México: Pearson.

L. Hennessy, J., & A. Patterson, D. (1989). Arquitectura de computadoras: Un enfoque cuantitativo. Madrid: McGraw-Hill.

Llanos Ferraris, D. R. (2007). Fundamentos de informática y programación en C. Madrid: Paraninfo.
Maimon, O., & Rockach, L. (2010). Data Mining and Knowledge Discovery Handbook. Israel: Springer.

Marqués, M. (2009). Bases de datos. Castellón de la Plana.

Masnick, M. (13 de Abril de 2012). Why Netflix Never Implemented The Algorithm That Won The Netflix $1 Million Challenge. Recuperado el 12 de Junio de 2019, de TechDirt: https://www.techdirt.com/articles/20120409/03412518422/why-netflix-never-implemented-algorithm-that-won-netflix-1-million-challenge.shtml

Medina, A. (13 de Septiembre de 2017). A México le urge el Expediente Clínico Electrónico Universal. Forbes México.

Molina Lopez, J. M., & Garcia Herrero, J. (2006). TÉCNICAS DE ANÁLISIS DE DATOS. Madrid, España.

NOM-004-SSA3-2012. (5 de Octubre de 2010). NORMA Oficial Mexicana NOM-004-SSA3-2012, Del expediente clínico.

OMS. (2005). The World Health Organization Quality oThe World Health Organi-

zation Quality of Life assessment (WHOQOL): position paper from the World Health Organization.f Life assessment (WHOQOL): position paper from the World Health Organization. Soc. Sci. Med, 41: 1403.

Oscar Nigro, H., Xodo, D., Corti, G., & Terren, D. (2004). KDD (Knowledge Discovery in Databases): Un proceso centrado en el usuario. VI Workshop de Investigadores en Ciencias de la Computación. Buenos Aires.

PHP Hypertext Preprocessor. (24 de Octubre de 2019). Recuperado el 22 de Noviembre de 2019, de PHP.net: https://www.php.net

Rodríguez, D. (12 de Noviembre de 2018). Analytics Lane. Obtenido de Cuatro librerías para ciencia de datos en Python: https://www.analyticslane.com/2018/11/12/cuatro-librerias-para-ciencia-de-datos-en-python/

SALUD. (2013). datos.gob. Recuperado el 15 de Enero de 2019, de https://datos.gob.mx/busca/dataset/indicadores-ods-de-salud/resource/2b3ba7ee-1800-410b-8906-c592616fcbcb

SALUD. (2013). datos.gob. Recuperado el 15 de Enero de 2019, de https://datos.gob.mx/busca/dataset/indicadores-ods-de-salud/resource/56eae72f-688c-4f1a-8d9b-48ff1c21e3e9

Sierra Araujo, B. (2006). Aprendizaje Automático: Conceptos básicos y avanzados. PRENTICE HALL/PEARSON.

Silberschatz, A., F. Korth, H., & Sudarshan, S. (2002). Fundamentos de bases de datos. Madrid: McGRAW-HILL.

Vila Miranda, M. A., Sanchez Fernandez, D., & Cerda Leiva, L. (2004). REGLAS DE ASOCIACIÓN APLICADAS A LA DETECCIÓN DE FRAUDE CON TARJETAS DE CRÉDITOS. XII CONGRESO ESPAÑOL SOBRE TECNOLOGÍAS Y LÓGICA FUZZY, 491-496.

Witten, I., Frank, E., & Hall, M. (2011). Data Mining practical machine learning tools and techniques. Hamilton: Morgan Kaufmann.