RESOLUCIÓN DE INCOSISTENCIAS EN BASE DE DATOS RELACIONAL PARA EL CONTROL DE DOCUMENTOS DEL ÁREA DE OFICIALÍA MEDIADORA, CONCILIADORA Y CALIFICADORA (CODOMCyC) PARA EL MUNICIPIO DE TEPETLAOXTOC, ESTADO DE MÉXICO.

TESIS

PARA OBTENER EL GRADO DE:

LICENCIADO EN INFORMÁTICA ADMINISTRATIVA

PRESENTA:

MIGUEL ÁNGEL ROMERO PÉREZ

DIRECTOR DE TESIS:

ALMA DELIA CUEVAS RASGADO

Índice.

Introducción

La presente investigación está orientada a la resolución de inconsistencias en la base de datos relacional debido a que los registros del departamento de la oficialía mediadora, conciliadora y calificadora del municipio de Tepetlaoxtoc, Texcoco, Estado de México se habían llevado de forma manual. Como resultado del control manual los datos de las personas involucradas en un caso se escribían en libretas o en hojas de cálculo habiendo inconsistencias en dichos registros de datos, el objetivo de desarrollar un sistema de información, para el control de actas y citas, ha resuelto mayor parte de este problema, así como la reducción del tiempo para generar los reportes y la administración de las citas.

La Oficialía Calificadora conoce, califica e impone sanciones administrativas a las personas que infringen el Bando Municipal en apoyo a la Autoridad Municipal, la conservación del orden público y verificación de daños a los bienes de propiedad Municipal. (Ortega, 2017)

Como se mencionó anteriormente los procesos realizados en el área, el manejo de su documentación es en archiveros, por lo cual al generar un reporte de las diferentes actas o por persona, el tiempo de respuesta es largo.
El manejo de citas para las personas involucradas en dichas actas es por medio de hojas de cálculo (Excel).

Este documento está dividido en cuatro capítulos, en el primero se plantea el problema, se da a conocer la justificación, así como los objetivos, tanto el general, como los específicos y por último se da a conocer la hipótesis. En el capítulo dos se presenta el marco teórico; las definiciones, antecedentes de la investigación, tipos de sistemas, componentes, necesidades y vistas de UML, patrones de diseño MVC, bases de datos relacionales, su estructura básica y su teoría, así como ventajas y desventajas de las mismas, el SQL, las tareas de un lenguaje de consulta y su estructura básica. En el capítulo tres se explica el método: como el diseño de la investigación, el programa y su procedimiento y en el cuatro se presentan las conclusiones, los trabajos futuros y el cronograma.

Capítulo 1. Planteamiento del problema

Los niveles jerárquicos y la relación que guardan entre si cada una de las áreas que integran la estructura de la Oficialía Mediadora Conciliadora y la Oficialía Calificadora, es autorizada en la Ley Orgánica Municipal. Constituyéndose por dos áreas sustantivas y un área de apoyo para ejercer una mejor función de dirección y control por parte de las unidades y conforme al Reglamento Interno de Trabajo es una dependencia Auxiliar que depende directamente de la Presidencia Municipal.

La misión de esta área es coadyuvar en la solución de conflictos -no constitutivos de delito- para lograr un ambiente de Seguridad, Paz y Tranquilidad en el Municipio y, en caso, de incumplimiento al Bando Municipal calificar y sancionar adecuadamente las faltas o infracciones, mediante vigilancia e investigación, actuando en todo momento con Imparcialidad y con total apego a la ley. (Ortega, 2017)

La visión de esta área es mediar, conciliar y ser árbitro en los conflictos de intereses entre los particulares y sancionar las faltas administrativas, todo esto de acuerdo a las atribuciones contempladas en el Bando Municipal y con total transparencia. (Ortega, 2017)

Al conocer la importante función del área en estudio y ver que su tiempo de respuesta es muy larga y considerando la pérdida de organización en las diferentes actas a expedir, se requiere agilizar los tiempos de respuesta y también de llevar el control de las actas expedidas.

Debido a la anterior situación se plantea lo siguiente:

¿Desarrollando un sistema de información, se logrará llevar el control de las diferentes actas y citas para reducir el tiempo de generar reportes y mejorar el control de citas?

1.2 Justificación

El principal objetivo de la Oficialía Mediadora, Conciliadora y Calificadora es: Ejercer la justicia en materia de faltas administrativas al Bando Municipal y demás disposiciones municipales, así como la mediación y conciliación a fin de brindar un mejor servicio a los habitantes del Municipio, mediante el conocimiento, atención, seguimiento, calificación, imposición de sanción y/o finalización de las solicitudes y presentados. Se pretende que este objetivo se cumpla con la función de las áreas administrativas que son las siguientes: (Ortega, 2017)

- Determinar las directrices técnicas y formalidades de los procedimientos de mediación y/o Conciliación de conflictos y redactar los convenios conciliatorios y acuerdos de reparación de daños.
- Atender las solicitudes de información y llenado de formatos administrativos, así como entregar los informes y/o reportes solicitados.
- Elaboración de actas Informativas y/o certificaciones de hechos.
- Determinar las políticas para la atención de hechos de tránsito suscitados en el territorio Municipal y de las faltas administrativas conforme a los lineamientos de garantía de audiencia y respeto de Derechos humanos,
- Elaborar citatorios y atender exhortos de otras dependencias.

Por lo cual se ve la necesidad de trabajar con un sistema de información, que apoye al mejoramiento de los procesos de documentación, ya que cada actividad que se realice deberá estar registrada en el sistema para un mejor control. Esta base de datos conecta el registro, así mismo el sistema le permite imprimir el acta en un archivo de texto (Word) para limitar el tiempo de captura, se maneja un registro de citas para un mejor control. Proporciona los reportes correspondientes por cada personal o por rango de fechas.

El beneficio se da a corto plazo y tiene un alcance a nivel municipal sin embargo otros municipios del Estado de México y otros Estados de la República Mexicana podrán usarlo con los mismos éxitos que en Tepetlaoxtoc, por lo tanto, el alcance sería a nivel estatal y Nacional debido a la estandarización de funciones.

1.3 Objetivos de la investigación

A continuación, se describe los objetivos de esta tesis.

1.3.1 Objetivo general

- Desarrollar un sistema de información donde se registran las actas y citas, con el fin de reducir el tiempo, al generar reportes y llevar un control de citas.

1.3.2 Objetivo específicos

- Analizar y diseñar un diagrama de flujo con el fin de detallar los procesos.
- Analizar y diseñar la base de datos, para el control de las actas a través de un gestor de base de datos.
- Desarrollar algoritmos para la gestión de datos a través de vistas.
- Programar el sistema con el lenguaje de programación orientado a objetos para satisfacer las necesidades de los reportes y citas.
- Elaboración de las actas en archivos de textos (Word).
- Guardar los reportes en hojas de cálculo (Excel).

1.4 Hipótesis

Desarrollando e implementando un sistema de información ayudará a mejorar el control de citas, así como también reducirá el tiempo de la generación de reportes en el área de Oficialía Mediadora y Conciliadora.

1.5 Alcances y limitaciones

- Alcances. El sistema generará las actas correspondientes y llevará el control de citas para el área de oficialía mediadora y conciliadora.
- Limitaciones. Solo se instalará en una máquina por el tamaño de área.

Capítulo 2. Marco Teórico

2.1 Base de datos relacionales

Actualmente las empresas de diferentes sectores tanto públicas como privadas utilizan en sus diversas áreas o departamentos sistemas de información (SI) para realizar diversos procesos de negocio; en los cuales las bases de datos (BD) son la columna vertebral de éstos, ya que es donde se resguardan y recuperan los datos para su procesamiento y generación de información; siendo las bases de datos relacionales (BDR) las más utilizadas.

Cuando se manipula una sola BDR es sencillo la recuperación de datos y el procesamiento de los mismos; pero cuando en las empresas existe la necesidad de compartir información tiene lugar el procesamiento utilizando bases de datos heterogéneas (BDH); es decir, con esquemas diferentes que podrían incluir un campo de tipo imagen como una fotografía tomada en diferentes momentos, ya que fueron desarrolladas en momentos diferentes e implementados con diversos sistemas gestores de bases de datos (DBMS). Es por ello, que se debe contar con algún mecanismo que permita, por un lado, identificar que el dominio de un atributo corresponde y es común al dominio de otro atributo, y por el otro resolver las inconsistencias entre esquemas que permita poder realizar el procesamiento de datos, incluye las fotografías, como si se tratara de una única BD.

Con base a lo anterior surge la idea de este proyecto, ya que la interoperabilidad entre BDRH donde se requiere la integración de los datos, sigue siendo un reto dentro de las empresas. Existen diversos esfuerzos para dar solución a esta problemática, pero en ninguno de ellos fusionan las ontologías, el reconocimiento de patrones y la arquitectura de tres capas; algunos trabajos de investigación son citados a continuación:

(Coetzer, Moodley, & Gerber, 2013) describen un caso de estudio en esquemas de bases de datos relacionales y aplicaciones estandarizadas, orientadas a la biodiversidad de museos en Sudáfrica, con la finalidad de extraer resúmenes significativos con semántica heterogénea., ya que a pesar de que se tenían datos de calidad y esquemas consistentes, la heterogeneidad semántica seguía siendo un reto. Para el caso de estudio en cuestión, se consideraron tres bases de datos relacionales de diferentes organizaciones, limitándose el estudio a angioespermas. Los almacenes de datos contenían datos relacionados con colecciones de insectos de visita de flores, y para obtener representaciones semánticas de las tres BD se utilizó una ontología expresada en OWL (Ontology Web Language) teniendo como resultado que el uso de la ontología presenta observaciones semánticamente y que por lo tanto se puede utilizar como un indicador para determinar las ocurrencias y los visitantes de flores (insectos).

En el enfoque presentado por (Karasneh et. Al., 2009) se realiza la integración de esquemas en BDRH mediante la coincidencia de esquemas, más no así con el uso de ontologías, considerando la heterogeneidad sintáctica y semántica, para la integración de datos sin ayuda del usuario. Para llevar a cabo la integración, se realiza un proceso binario, es decir se consideran como entrada dos esquemas de BDR que generan un esquema que será la entrada para procesarse con otro esquema, así se genera un esquema global que satisface las propiedades de los esquemas entrantes. Con base a los resultados expuestos en este trabajo existen un mayor porcentaje de similitudes y de atributos coincidentes que en otros esfuerzos realizados, ya que se explotan los diversos elementos de las BDR en el proceso de coincidencia.

En otro enfoque se propone una ontología automática basada en una base de datos relacional, donde la BDR es recibida como entrada extrayendo información y tablas con las cuales se crea un modelo conceptual para generar una ontología en OWL mediante reglas de transformación. La transformación de la BDR en una ontología conlleva tres pasos: 1) la entrada de la BDR teniendo como salida una ontología con estructura OWL a partir del conocimiento extraído de los componentes de la BDR; 2) Se utiliza la información obtenida en el paso anterior para generar los componentes del grafo que se utiliza como un modelo conceptual intermedio; y 3) se recibe el grafo creando la ontología final que es almacenada en la estructura OWL y que además considera disparadores para convertir datos SQL en su ontología. El resultado de este trabajo es una ontología que mantiene las relaciones y semántica de los datos (Dadjo & Kheirkhah, 2015).

(Bakkas & Bahaj, 2013), proporcionan un método que permite la extracción automática de una BDR y su estructuración en forma de grafos RDF utilizando la API de Jena para que estén disponibles en la Web semántica. Para extraer el esquema de la base de datos relacional se utiliza un catálogo, con el cual se crean clases que componen el modelo de ontología correspondiente al esquema. Posteriormente el modelo creado se modifica mediante consultas en SPARQL. Para los registros se crean instancias de clases utilizando los datos de las bases de datos relacional. Las pruebas para obtener los resultados fueron realizadas mediante un prototipo que demuestra la precisión y rendimiento del enfoque propuesto.

Un trabajo similar al que se está proponiendo en este documento, con respecto al uso de reconocimiento de patrones y ontología, es el propuesto por (Sooksawaddee & Sasiporn, (2013) que proponen un sistema denominado OMNN- IMAGEN con información relacionada a la biblioteca para la categorización de objetos clasificando imágenes de origen, incluye rostros, que representan los mismos conceptos de ontología según el grado de valores de similitud, así se utiliza un reconocimiento de patrones basado en el mapeo de ontologías. No obstante, utilizan redes neuronales., que es un tema que no se aborda en el

proyecto que estamos proponiendo. En este trabajo se propone la idea de la cartografía de ontología basado en el contenido de la imagen para apoyar a futuras aplicaciones multimedia; sin embargo, los autores comentan que debido a la similitud entre imágenes existe una igualación inadecuada en el sistema, por lo que es necesario hacer un trabajo futuro que permita clasificar las imágenes ambiguas.

Sobre reconocimiento de patrones faciales, se propone un sistema para recuperar imagen facial a base de semántica (SFIR) utilizando APS0 y distancia euclidiana al cuadrado. Para ello se utilizan tres etapas: 1) extracción de características (color, boca, ojos y textura por mencionar algunos); 2) optimización, donde se utiliza APS0 para dotar de semántica y 3) recuperación de imágenes, donde se utiliza la distancia euclidiana al cuadrado, para recuperar imágenes de rostros que tienen menos distancia con la imagen de búsqueda. Los resultados muestran un 95% de recuperación precisa de la imagen, pero no es aplicable a imágenes en tiempo real (Kalimuthu & Krishnamurthi, 2015).

Arguello, F. (2011), hace un compendio de las principales investigaciones para el reconocimiento facial, siendo las técnicas más utilizadas: PCA, análisis de discriminantes lineales de Fisher (Fisherfaces), conservación de proyecciones locales (LPP), proyecciones aleatorias (Randomefaces) , redes neuronales artificiales (RN), Análisis de componentes independientes (ICA), análisis de subregiones, correlación (CORR), DCT; sin embargo los investigadores, prefieren utilizar una combinación de las técnicas antes mencionadas. Asimismo, se indica que la extracción de características y su procesamiento es altamente costoso a nivel computacional, ya que el procesamiento debe pasar por 6 fases: captura de la imagen, preprocesamiento, localización, escalamiento y ajuste, extracción de características y, clasificación y toma de decisión. No obstante, se concluye que después de realizar un análisis de los resultados de las investigaciones que la técnica más utilizada es PCA, redes neuronales y la combinación de técnicas. Los resultados muestran que utilizar PCA con LDA y RN presenta una eficiencia del 95.5%.

2.1.2 Interoperabilidad en base de datos

La interoperabilidad en bases de datos se refiere a poder compartir información entre bases de datos heterogéneas independientemente del sistema gestor de bases de datos (SGBD) en que se haya implementado.

Lo anterior tiene lugar en las empresas u organizaciones debido a que el desarrollo de sistemas en las diversas áreas se realiza en diferentes momentos, con diferentes entornos de desarrollo, diferentes plataformas y diferentes SGBD, existiendo la problemática de poder

compartir información y recursos. Para efectos de este trabajo, la interoperabilidad está orientada a las bases de datos relacionales.

La heterogeneidad que existe en las bases de datos se da a nivel estructural, se presenta cuando se tienen esquemas diferentes; nombrado de atributos, nombres de atributos diferentes; semántico, se tiene el mismo nombre con significados diferentes; y sintáctico, que tiene lugar cuando se trabaja con dominios de datos diferentes. Los tres primeros son los más relevantes para las bases de datos relacionales.

Las bases de datos relacionales tienen un esquema de bases de datos que describe su estructura, es decir las tablas, atributos y relaciones a partir de las llaves primarias y foráneas. La heterogeneidad estructural y de nombrado de atributos, presentan un problema al querer recuperar información de diversas bases de datos relacionales en el mismo dominio o contexto de operación.

Para ejemplificar lo anterior, considere dos esquemas de bases de datos, cada uno con sus respectivas tablas y atributos nombrados de manera diferente, los cuales podrían ser considerados como sinónimos. La base de datos Calificaciones tiene la tabla Alumno es sinónimo de la tabla Estudiante, en la base de datos Becas; lo mismo sucede con los atributos matricula que es sinónimo de num. cuenta. En ambos casos, el nombre de la tabla y del atributo son necesarios para poder realizar consultas en una base de datos. Sin embargo, si se desea realizar alguna consulta recuperando datos de ambas bases de datos, se deben identificar los sinónimos y los datos para poder acceder a éstos, recuperarlos y procesarlos, para obtener información mediante consultas siendo transparente para el usuario, a esto se le denomina integración de datos.

Para la integración de datos no basta con tener conexiones a las bases de datos como: conectividad de bases de datos abierta (ODBC), conectividad a bases de datos Java (JCBD), conectividad de bases de datos abierta independiente (IODBC) y objetos de enlaces y bases de datos embebidas (OLEDB), para poder acceder a los datos; debido a que con ellos no se resuelve la sinonimia en los atributos y tablas de diversas bases de datos como si se tratará de una sola.

Es por lo anterior que se requiere de una estructura que permita representar un conjunto de conceptos y relaciones en un dominio dado, como lo hace una ontología. Los conceptos permiten formalizar el conocimiento, en este caso los nombres de las tablas y los atributos; las relaciones representan la conexión entre ellos.

2.1.3 Ontología

Hay varias definiciones de ontología, la clásica es la de Gruber que la define como una representación o especificación formal (una estructura sintáctica) de una conceptualización (conjunto de conceptos) compartida (Cuevas, 2006) (común a varias personas u otros programas).

OM de las siglas Ontology Merging es un lenguaje desarrollado con la finalidad de diseñar ontologías con conceptos y relaciones que proporcionan más semántica a las operaciones de búsqueda de conocimiento. (Cuevas, 2006),

La estructura semántica de las ontologías definidas en este lenguaje, es a través de un conjunto de etiquetas (como en XML) que identifican el concepto y sus relaciones, por ejemplo, <concept> que indica el nombre del concepto. Esta etiqueta permite el anidamiento de conceptos, <language> que representa el lenguaje del concepto, <word> donde se encuentran las palabras que definen al concepto y <relation>que representa el tipo de relación que conecta al concepto.

2.1.2.1 Las relaciones en OM pueden ser implícitas y explícitas

Implícitas: son las relaciones expresadas por el anidamiento, el concepto externo se reconoce como antecesor mientras que el interno como sucesor. Estas relaciones son: member, part, part* y subset.

Explícitas: son las que se encuentran definidas entre las etiquetas <relation></relation>, puede ser una actividad (eats), propiedad (color) o atributo del concepto. No se permiten relaciones anidadas.

Existe otro tipo de relación llamada Partición, esta se diferenciará con la palabra “Partition” y contiene una estructura diferente a las relaciones explícitas. Una partición es un conjunto cuyos subconjutos están bien definidos por un rango o intervalo. Cada instancia de un subconjunto no puede ser instancia de otro dentro de la misma partición, es decir cada subconjunto de la partición son mutuamente exclusivos y colectivamente exhaustivos.

2.1.2.2 Ventajas

El lenguaje permite que un concepto pueda tener relaciones n-arias (múltiples).
Una relación puede ser un concepto.
Se permiten las relaciones de tipo partición.
Proporciona más semántica a la interpretación de los conceptos de la ontología al usar sinónimos en la definición de los conceptos.

2.1.2.3 Desventajas

Pese a que cuenta con su propio lenguaje de definición de ontologías y su propia estructura de datos hace falta más pruebas para demostrar la riqueza de su representación.

2.1.2.4 Aplicaciones de las Ontologías

Web Semántica: La necesidad en esta área es que las máquinas (programas, robots, "bots", "crawlers", "arañas", agentes) y no solamente los humanos, puedan entender la gran cantidad de información en la red (Web). Por lo tanto, aquí se centra la importancia de deducir información para añadirla al conocimiento de una ontología. La unión automática de ontologías (que se presenta en este trabajo) parte de la ontología A que puede estar formada de un documento en la Web (actualmente este paso se hace a mano) y tomar otro documento en la Web que será la ontología B uniéndolas con cuidado, devolviendo una ontología C con información de A y B muy útil para responder preguntas de un usuario.

Actualmente, ya se cuenta con Google una especie de recolector automático de información sobre algún tema t, pero la interpretación o fusión de la información contenida en los documentos recolectados queda a cargo del lector o usuario.

Comercio electrónico: Un agente A puede enriquecer el conocimiento de su ontología A (conocer sinónimos de sus productos, tomar nuevas características o propiedades de sus productos) uniendo su ontología A con la de otros agentes B con la ontología B, C con C, etc. que vendan el mismo producto, entonces logra hacerse "entender mejor" con otro agente y su ontología X que compre los productos que vende el agente A.

Recuperación de información: Una ontología A puede enriquecer sus nodos con otras ontologías B, C, etc. y responder preguntas complejas que se deriven de A y de las

ontologías que haya copiado. Por ejemplo, si en A dice que el delincuente es hombre, B dice que es de raza negra y C dice que es joven entonces la extracción del conocimiento resulta ser una información interesante.

Educación virtual: Un libro virtual A puede unir su ontología A (donde los nodos son temas discutidos en el libro) con la ontología de otro libro virtual B que trata del mismo tema o similares. Esta ontología enriquecida será mejor aprovechada por los estudiantes que aprenden de A. También A puede unir su ontología con una ontología "predecesora", por ejemplo, el libro A que trata de Finanzas Internacionales puede encontrar útil fusionar su ontología con B que trata de Economía o el libro A que trata de la Administración en los negocios encontrará útil fusionar su conocimiento con la ontología Gerencia. Esto ayudará a los estudiantes a recapitular conceptos previos.

Bases de datos heterogéneas: Se puede obtener una ontología A de una base de datos, definiendo cada entidad como un nodo y sus atributos como relaciones y unirla con otras ontologías B, C, etc. resultando bases de datos similares. Tal fusión debe hacerse cuidadosamente, verificando los dominios y tipos de datos de los atributos, la sinonimia, evitando información redundante y contradictoria. El resultado de la fusión será una base de datos más útil a cualquier corporación que desea administrar y explotar mejor su información.
Por ejemplo, la ontología de WalMart se puede fusionar con ontologías de otras firmas, para generalizar y complementar la descripción de sus productos y para encontrar los productos similares que Walmart no vende pero lo hacen otras firmas.

2.1.2.4 Ejemplo de Ontología

A continuación, se presenta un ejemplo de ontología en la cual los conceptos están relacionados entre si y describen las características del concepto Oaxaca. Las relaciones tienen una etiqueta que enlaza a un nodo de otro. Por ejemplo, Oaxaca y Puebla están relacionados por el enlace "Colinda al norte con", y así sucesivamente. Figura 1.

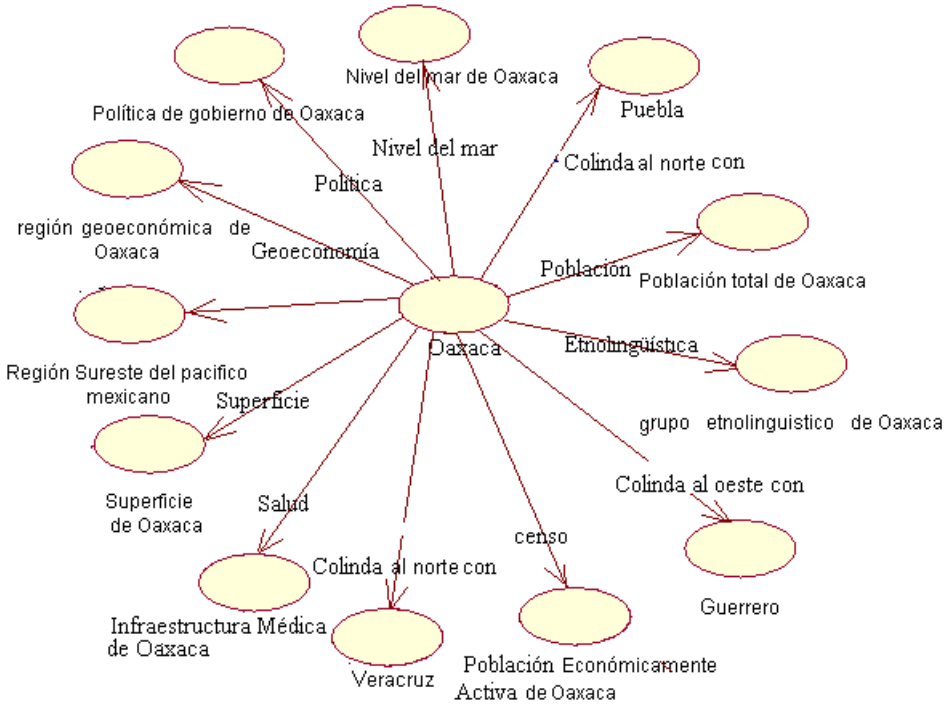


Figura 1 Ejemplo de concepto llamado Oaxaca.

Una ontología puede ser el intérprete entre las relaciones de una base de datos relacional heterogénea en la cual, la relación ALUMNO puede ser un concepto en la ontología con los sinónimos ESTUDIANTE, EGRESADO y USUARIO, mismos que pueden ser relaciones de otras bases de datos.

Cuando una consulta de una base de datos quiera solicitar un dato de otra base de datos con diferentes campos: por ejemplo

SELECT nombre, Apellido FROM ALUMNO

Una ontología tendrá definido el sinónimo de ALUMNO por EGRESADO y podrá interpretar que la consulta en la otra base de datos se refiere a:

SELECT nombre, apellido FROM EGRESADO

Devolviendo los datos solicitados.

Los sinónimos se representan en los nodos de la ontología.

2.2. Antecedentes

La Oficialía Mediadora Conciliadora y Calificador conoce, califica e impone sanciones administrativas a las personas que infringen el Bando Municipal en apoya a la Autoridad Municipal en la conservación del orden público y verificación de daños a los bienes de propiedad Municipal.

Canalizar las solicitudes que llegan a la Oficialía Mediadora Conciliadora y Calificadora a fin de indicarle al solicitante la acción o acciones a seguir con respecto al problema planteado.

La Oficialía Calificadora y la Oficialía Calificadora son las responsables de realizar todas las funciones encomendadas en el ámbito de su competencia por la Ley Orgánica Municipal del Estado de México, así como de elaborar y actualizar los manuales de procedimientos.

Los titulares de la Oficialía calificadora y/o Oficialía Mediadora deberán canalizar e instruir al personal del área a fin de que cualquiera de ellos pueda recibir al ciudadano conforme los principios generales de ética en el servicio público.

Atender y asesorar a los solicitantes con conocimientos jurídicos sólidos a fin de ofrecerle alternativas económicas, expeditas y suficientes para solucionar la problemática planteada. Y de ser necesario canalizarle a otras áreas e Instancias de las que se requiera intervención para solucionar ampliamente la problemática.

La función de las áreas administrativas que son las siguientes: (Ortega, 2017)

- Determinar las directrices técnicas y formalidades de los procedimientos de mediación y/o Conciliación de conflictos y redactar los convenios conciliatorios y acuerdos de reparación de daños.
- Atender las solicitudes de información y llenado de formatos administrativos, así como entregar los informes y/o reportes solicitados.
- Elaboración de actas Informativas y/o certificaciones de hechos.
- Determinar las políticas para la atención de hechos de tránsito suscitados en el territorio Municipal y de las faltas administrativas conforme a los lineamientos de garantía de audiencia y respeto de Derechos humanos,
- Elaborar citatorios y atender exhortos de otras dependencias.

Al conocer la función de esta área y ver que su tiempo de respuesta es muy larga y a la vez la perdida de organización en las diferentes actas a expedir. Se ve la necesidad de agilizar los tiempos de respuesta y también de llevar el control de dichas actas.

2.3 Tipo de Investigación

2.3.1 Por el propósito o finalidades perseguidas

APLICADA, PRÁCTICA O EMPÍRICA. Es la utilización de los conocimientos en la práctica, para aplicarlos, en la mayoría de los casos, en provecho de la sociedad.
Este tipo de investigación también recibe el nombre de investigación práctica o empírica. Se caracteriza porque busca la aplicación o utilización de los conocimientos que se adquieren. La investigación aplicada se encuentra estrechamente vinculada con la investigación básica, ya que depende de sus descubrimientos y aportes teóricos. Busca confrontar la teoría con la realidad.

2.4 Sistema de Información

Desde hace mucho tiempo, las organizaciones han reconocido la importancia de administrar recursos claves como la mano de obra y la materia prima. En la actualidad, la información se ha ganado el legítimo derecho de ser considerada como un recurso clave. Los encargados de la toma de decisiones por fin han comprendido que la información no es tan solo un producto derivado de la conducción de los negocios, sino un impulso de los mismos y que puede constituir un factor crucial en el éxito o fracaso de la empresa.

2.4.1 Antecedentes de sistemas

Un sistema de información es un conjunto de elementos interrelacionados con el propósito de presentar atención a las demandas de información de una organización, para elevar el nivel de conocimiento que permite un mejor apoyo a la toma de decisiones y desarrollo de acciones. (Botella & Olivella, 2011)

Un sistema de información es un conjunto de elementos o componentes interrelacionados para recolectar (entradas), manipulación (proceso) y diseminar (salida) datos e información y para proveer un mecanismo de retroalimentación en pro del cumplimiento de un objetivo. (R & G, 2000)Si estaban en la biografía del libro

2.4.2 Operaciones del Sistema

Ahora se cita el concepto de sistema de información: Un sistema de información realiza cuatro actividades básicas: entrada, procesamiento, salida y retroalimentación. (R & G, 2000) el autor desglosa este proceso de la siguiente manera:

- **Entrada**: Es la actividad que consiste en recopilar y capturar datos primarios.
- **Procesamiento**: Supone la conversión o transformación de datos en salidas útiles. Esto puede implicar ejecutar cálculos, realizar comparaciones y adoptar acciones alternas, y el almacenamiento de datos para su uso posterior.
- **Salida**: Implica producir información útil, por lo general en forma de documentos y/o reportes, podría ser la entrada de otro.
- **Retroalimentación**: Es la salida que se utiliza para efectuar cambios en actividades de entrada o procesamiento. (R & G, 2000)

2.4.3 Tipos de Sistemas

Los sistemas de información se desarrollaron con diversos propósitos, según las necesidades de la empresa, en la Figura 2 se puede observar los tipos de sistemas.

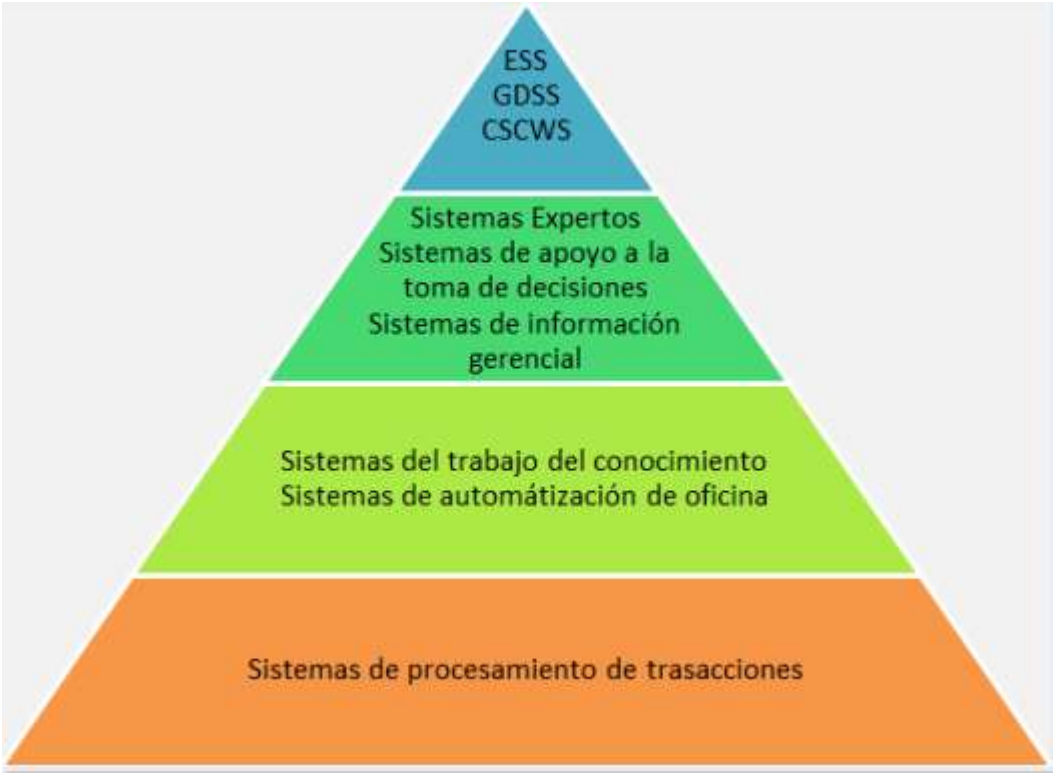


Figura 2 Pirámide de Tipo de Sistemas. (E. KENDALL & E. KENDALL, 2005) Así vienen en la bibliografía

1)ESS (Executive Support System) Sistemas de apoyo a ejecutivos, cuando los ejecutivos recurren a la computadora, por lo general lo hacen en busca de métodos que los auxilien en la toma de decisiones de nivel estratégico, ayuda a organizar sus actividades relacionadas con el externo mediante herramientas gráficas y de comunicaciones, por lo general se encuentran en salas de juntas o en oficinas corporativas personales**.**

GDSS (Group Decision Support System) Cuando los grupos requieren trabajar en conjunto para tomar decisiones semiestructuradas o no estructuradas, un sistema de apoyo a la toma de decisiones en grupo.

CSCWS (Computer-Support Collaborative Work System) Sistemas de trabajo colaborativo a poyados por computadoras.

2) Sistemas expertos (AI, Artificial Intelligence) Desarrollar máquinas que tengan un comportamiento inteligente, son la comprensión del lenguaje natural y el análisis de la capacidad para razonar un problema hasta una conclusión lógica**.**

Sistema de apoyo a la toma de decisiones (DSS, Decision Support System) Coincide con los sistemas de información general en que ambos dependen de una base de datos para abastecerse. Las diferencias en el DSS ponen énfasis en el apoyo a la toma de decisiones en todas las fases, aunque la decisión definitiva es responsabilidad exclusiva del encargado de tomarla.

Sistema de información gerencial (MIS, Managemen Information System) Son sistemas de información computarizados cuyo propósito es contribuir a la correcta interacción entre los usuarios y las computadoras. Debido a que los usuarios, software y hardware, funcionen de manera coordinada.

3) Sistema de trabajo de conocimiento (KWS, Knowledge Work System) Sirve de apoyo a los trabajadores profesionales, como los científicos, ingenieros y médicos, en sus esfuerzos de creación de nuevos conocimientos y dan a estos la posibilidad de compartirlo con sus organizaciones o con la sociedad.

Sistema de automatización de la oficina (OAS, Office Automation System). Apoyan a los trabajadores de datos, quienes por lo general no generan conocimiento nuevo, sino más bien analizan la información con el propósito de transformar los datos o manipularlos de alguna manera antes de compartirlo, distribuirlos formalmente con el resto de la organización y en ocasiones más allá de estas.

4) Sistema de procesamiento de transacciones (TPS)

Son sistemas de información computarizada creados para procesar grandes cantidades de datos relacionadas con las transacciones rutinarias del negocio. Expanden los límites de la

organización dado que les permite interactuar con entornos externos. Es importante para las operaciones cotidianas de un negocio, que estos sistemas funcionen sin ningún tipo de interrupción, puesto que los administradores recurren a los datos producidos por las TPS con el propósito de obtener información actualizada sobre el funcionamiento de su empresa. (E. KENDALL & E. KENDALL, 2005

2.5 UML

UML quiere decir (Unified Modeling Language) lenguaje de modelo unificado, es un lenguaje estándar para visualizar, especificar, consultar y documentar los artefactos de un sistema de software, puede usarse en las diferentes etapas del ciclo de vida del desarrollo y en diferentes tecnologías de implementación, es independiente del proceso de desarrollo de software. (Fawiler, Scott, & Scott, 2000)

El lenguaje UML proporciona una notación con la que podemos capturar e ilustrar diseños de programas. Es independiente del lenguaje de programación y utiliza términos genéricos. (Lewis & Chase, 2006)

2.5.1 Antecedentes

UML es una consolidación de muchas de las notaciones y conceptos más usados orientados a objetos, Empezó como una consolidación del trabajo de Grande Booch, James Rumbaugh, e Ivar Jacobson. (Fawiler, Scott, & Scott, 2000)

Estos autores fueron contratados por la Empresa Rational Software Co. Para crear una notación unificada en la que se basa para las construcciones de sus herramientas CASE (ingeniería de Software Asistida por computadoras). En el proceso de creación de UML han participado, no obstante, otras empresas de gran peso en la industria como Microsoft, Hewlett-Packard, Oracle o IBM, así como grupo de análisis y desarrolladores.

2.5.1.1 Componentes

Un diagrama de clases UML describe las clases de sistema, las relaciones estáticas entre ellas, los atributos y operaciones asociadas con cada clase y las fricciones aplicables a las conexiones entre objetos.

Un atributo es cualquier dato de nivel de clase, incluyendo las variables y las constantes.

Una operación es esencialmente equivalente a un método.

Las clases se representan en UML mediante rectángulos, usualmente divididos en tres secciones que contienen el nombre de la clase, sus atributos y sus operaciones. (Lewis & Chase, 2006) , la Figura 3 presenta de manera gráfica una clase en UML.

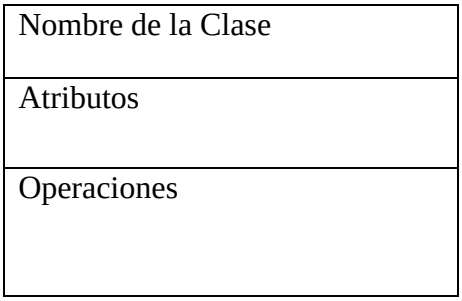


Figura 3 Definición de Clase en UML.

2.5.2 Necesidad del UML

Los tiempos cambian y las necesidades también. El trabajo colaborativo, el apego a los estándares, las grandes masas de información, y los infinitos niveles de jerarquía en cualquier organización empezaron a dar fin a viejo Diagrama de Flujo el cual al fin y al cabo sólo eran rombos, líneas y rectángulos.

Satisfacer estas necesidades condujo a la creación de un estándar internacional para el año 2005, el Lenguaje de Modelado Unificado (Unified Modeling Language), o mejor conocido como UML.

No es muy difícil de comprender a que viene el UML, técnicamente son los mismos "diagramas de flujo" pero llevados a un lenguaje que represente en forma de código lo que solíamos ver en forma de imágenes. Evidentemente que el UML va mucho más allá de las dos dimensiones a la que estábamos acostumbrados con los viejos diagramas de flujo, pero el objetivo es el mismo: organizar y representar. (Garzon, 2010)

2.5.3 Vistas UML

En la construcción del software utilizamos UML, existen cinco vistas para visualizar, especificar y documentar la arquitectura de software. UML permite presentar cada vista mediante un conjunto de diagramas. (Hinojosa, 2012), véase la Figura 4.

Nombre de la vista	Descripción
Vista de Diseño	Muestra la función del sistema desde el punto de vista de un actor externo que interactúa con el. Esta vista es útil a clientes, diseñadores y desarrolladores.
Vista de Diseño	Muestra la función del diseño dentro del sistema en término de la estructura estética y comportamiento dinámico del sistema. Esta vista es útil a diseñadores y desarrolladores. Se define propiedades tales como: persistencia, concurrencia, interfaces y estructura internas a las clases
Vistas de Procesos	Muestra la concurrencia de los sistemas y sincronización. Útil a desarrolladores e integradores
Vista de implementación	Muestra la organización de los componentes de código. Útil a desarrolladores.
Vista de implementación o despliegue	Muestra la implantación del sistema en la arquitectura física, Útil a desarrolladores, integradores y verificadores.

Figura 4 Tabla de Vistas UML. Fuente: Elaboración propia.

2.5.3.1 Importancia de los casos de Uso

Un caso de uso es una descripción del comportamiento de un sistema. Esa descripción está escrita desde el punto de vista de un usuario que acaba de oír que el sistema hace algo en particular. Un caso de uso captura la secuencia visible de eventos por lo que pasa un sistema como respuesta a un estímulo de un único usuario. Rober C., M. (2004).

2.5.4 Representación de modelo de caso un de Uso

El gran rectángulo es el contexto del sistema. Cualquier caso que esté en el interior es una parte del sistema que se desarrolla. En el exterior del rectángulo se muestra a los actores que actúan sobre el sistema. Los actores son entidades externas al sistema que proporcionan estímulos al mismo. Normalmente son seres humanos.

Dentro del rectángulo se presentan los casos de uso. Son los óvalos con nombres en su interior. Las líneas que conectan a los actores con los casos de uso que ellos estimulan. (Rober C., M. 2004), véase la Figura 5.

Figura 5 Ejemplo de Caso de Uso.

2.5.5 Patrón de diseño MVC (modelo de vista controlador)

A partir de las necesidades y comportamientos esperados del sistema, es posible elegir, adaptar o diseñar una arquitectura pertinente, que resuelva y facilite la estabilidad y adaptabilidad a las nuevas tendencias arquitectónicas y garantice en alto grado la evolución del sistema software antes las necesidades cambiantes del contexto. Es en este punto donde los patrones de arquitectura cobran importancia como soluciones recurrentes que pueden ser aplicadas en diferentes dominios de aplicación y entornos tecnológicos, ganando el cumplimiento de atributos de calidad esperados para todo sistema de software (Hinojosa, 2012), este es un patrón mayormente usado en el área de Diseño de Sistemas de Información.

El patrón MVC divide la responsabilidad del sistema en tres partes;

- MODELO; que mantiene la lógica y los datos del programa.
- VISTA; que proporciona una presentación Visual del modelo.
- EL CONTROLADOR; que procesa la entrada del usuario y hace modificaciones al modelo.

2.6 Base de datos

Como el lugar principal para el almacenamiento de datos, la base de datos es importante en cualquier sistema, por lo que a continuación se detallan definiciones y estructuras.

2.6.1 Definición de base de datos

Es una estructura de computadoras integradas, compartidas que alojan un conjunto de;

- Datos para el usuario final, es decir, hechos en bruto interesante para el usuario final.
- Metadatos o datos sobre datos mediante los cuales se integran datos. (Rob & Coronel, 2003)

Los metadatos describen las características de los datos y las relaciones que vinculan aquellos que están incluidos en la base de datos.

Sistema de administración de base de datos (DBMS), es un conjunto de programas que maneja la estructura de la base de datos y controla el acceso a los datos guardados en esta (Rob & Coronel, 2003).

2.6.2 Estructura básica

Se refiere a la estructura básica de una Base de Datos en la cual está estructurado de tres maneras: 1) desde el punto de vista conceptual o alto nivel, 2) punto de vista de Diseño o nivel medio y 3) y codificación o bajo nivel (nivel que solo entienden los programadores y Administradores de Base de Datos), la estructura básica se describe a continuación:

- Un archivo tiene muchos componentes.
- Un componente puede estar integrado por muchos ensambles.
- Un ensamble puede contener muchas piezas (Rob & Coronel, 2003)

2.6.3 Teoría de los modelos de datos relaciona

La teoría del modelo de datos relacional es obra del investigador de IBM Edgar Codd en 1970. Goza de una fuerte base matemática. El modelo relacional se caracteriza a muy grandes rasgos por disponer que toda la información está representada en tablas, y las relaciones entre datos están representadas explícitamente en esos mismos datos[1], véase la Figura 6.

Término relacional formal	Equivalente informal
Relación	Tabla
Tupla	Fila o registro
Cardinalidad	Número de filas o registros
Atributo	Columna o campo
Grado	Número de columnas o campos
Clave primaria	Identificador único
Dominio	Fondos de valores legales

Figura 6 Términos relacionales y equivalentes informales[2].

2.6.4 Ventajas

A continuación, se presentan las ventajas del modelo relacional:

1) **Simplicidad conceptual**: Es fácil ver la base de datos conceptual, lo que simplifica su proceso de diseño.
2) **Seguridad de la base de datos**: La seguridad de la base de datos es provista y ejecutada por el DBMS.
3) **Independencia de datos**: El DBMS crea un ambiente en el que las independencias de los datos pueden mantenerse, con lo que disminuye sustancialmente el esfuerzo de programación y el mantenimiento del programa.

[1] http://www.cs.us.es/cursos/bd-2001/temas/modelo_relacional.html
[2] http://elies.rediris.es/elies9/4-2-3.htm

4) **Eficiencia**: Cuando una base de datos contiene un gran volumen de datos relacionales y cuando los usuarios requieren muchas transacciones en las que utilizan datos cuyas relaciones se mantienen fijas con el tiempo. (Rob & Coronel, 2003)

2.7 SQL

Es un lenguaje por excelencia en la codificación de base de datos relacionales, sus siglas provienen de SQL (Structured Query Language, Lenguaje de Consulta Estructurado), siendo su máxima la funcionalidad de la base de datos a través de consultas.

2.7.1 Definición

SQL es un lenguaje estándar e interactivo que alimenta, consulta y actualiza una base de datos en la cual se relacionan las tablas usando un sistema gestor.

2.7.2 Tarea de un lenguaje de consulta

Un lenguaje de base de datos permitirá crear bases de datos y estructurar tablas para realizar tareas de administración de datos básicas (agregar, eliminar y modificar) y realizar consultas complejas diseñadas para transformar los datos sin procesar en información útil. (Rob & Coronel, 2003)

2.7.3 Estructura Básicas

El lenguaje SQL está compuesto por comandos, cláusulas, operadores y funciones de agregado. Estos elementos se combinan en las instrucciones para crear, actualizar y manipular las bases de datos.

Para exponer más claramente los conceptos se realizarán ejemplo sobre relaciones que se crearán aquí para entender mejor como funciona SQL.

Cuando se refiere a relación se habla más concretamente de la tabla de datos en sí, y sus atributos serán los campos de la tabla. Como ejemplo la siguiente relación (tabla) la llamaremos persona y sus atributos (campos) son nombre, apellido e Identificador (Id), véase la Figura 7.

PERSONA		
ID	NOMBRE	APELLIDO
34568	JOSE	ALVAREZ
54856	PABLO	LÓPEZ
56640	ROBERTO	SÁNCHEZ
71280	JUAN	BALDERAS
42389	RUBEN	LÓPEZ
54836	SANDRA	BULLOCK

Figura 7 Tabla de registro.

Existen dos tipos de comandos SQL:

- los DLL que permiten crear y definir nuevas bases de datos, campos e índices.
- los DML que permiten generar consultas para ordenar, filtrar y extraer datos de la base de datos. [3]

Capítulo 3. Metodología

Para la realización del sistema de información se ha usado el ciclo de vida Lineal, que consta de las siguientes etapas:

Análisis

Primera fase en la cual se obtienen los datos para identificar los requerimientos del sistema CODOMCyC:

3.1 Secuencia Metodológica

Paso 1: Entrevistas para conocer el control de las actas.

Paso 2: Identificación de datos y reportes.

Paso 3: Análisis y diseño del sistema de información con UML

Paso 4: Diseño de modelo relacional de la base de datos.

Paso 5: Codificar o programar los algoritmos previamente diseñados.

[3] http://www.ithinkweb.com.mx/capacita/sql_intr.html

Paso 6: Implementación de los algoritmos en lenguaje de programación Java, empleando el patrón MVC

3.2 Desarrollo de pasos metodológicos

3.2.1 Análisis de requerimientos de Oficial Mediadora y/o Calificador

Se realizaron entrevistas con la encargada del área, para conocer sus necesidades, así como, los conflictos a los que se enfrenta y al momento de realizar un reporte por las diferentes actas, se detectó la problemática que se encuentran debido a que la documentación la archivan por lo cual genera tiempo al realizar el reporte, el control de citas lo manejan por medio de un archivo de Excel. Para reducir el tiempo de generar los reportes y un mejor control de las citas, se propone la implementación de un sistema autodidáctico, que permite registrar las actas dando como resultado reporte en tiempo real y el control de citas.

3.2.2 Identificación de datos

Para desarrollar e implementar la base de datos fue necesario identificar los datos que se utilizan para realizar los reportes correspondientes en tiempo real, de igual manera mejorando el control de citas.

3.2.3 Identificación de datos para poder realizar los diferentes reportes y control de citas

Se ha ubicado los datos necesarios para dar de alta las actas:

Oficialía Calificadora:

- Fecha.
- Hora de detención.
- Hora de liberación.
- Nombre y domicilio de la persona acusada.
- Sexo.
- Estado Civil.
- Grado Estudios.
- Edad.
- Nombre de una persona responsable.
- Lugar de suceso.

- Calle, municipio, estado, colonia.
- Nombre de la persona afectada.
- Hora, monto, observaciones.
- Sanción Impuesta.
- Observación.
- Motivo.
- Tipo de Falta.
- Fracción.
- Artículo.

Oficialía Mediadora

- Tipo de Acta.
- Fecha de Suceso.
- Hora de Suceso
- Nombre y dirección de las personas involucradas.
- Sexo.
- Estado Civil.
- Grado Estudios.
- Edad.

Citas

- Nombre de la Persona.
- Fecha de la Cita.
- Estatus.
- Hora.

3.2.4 Identificación de reportes

- Reporte por acta.
- Reporte por persona.
- Reporte por rango de fechas.
- Reporte de citas por persona.
- Reporte de citas por fecha.

3.3 Diseño

Etapa en la cual se concretan los requerimientos a través de algoritmos, diagramas, etc con la finalidad de detallar los aspectos de la tecnología de software.

3.3.1 Diseño de UML

La realización del sistema de información ha sido utilizando el lenguaje UML, el cual permite modelar y realizar los diagramas de caso de uso para cada uno de los procesos del área Oficial Mediador y/o Calificador.

3.3.2 Caso de Uso

Dentro del sistema se puede obtener un menú que permita realizar diferentes tareas como "Guardar", que permite almacenar el registro en la base de datos, véase la Figura 8.

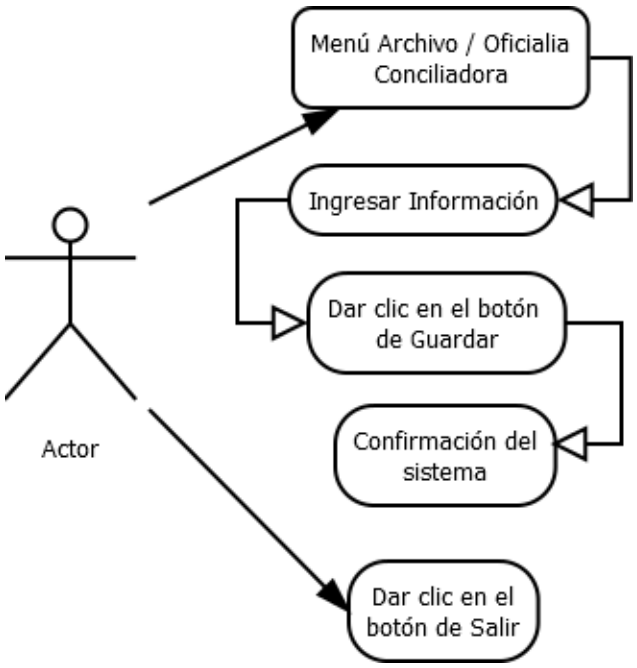


Figura 8 Diagrama de caso de uso "Guardar". Fuente: Elaboración propia 2017.

Buscar

Este menú permite de manera específica, mediante la utilización de una clave de identificación como la que cuenta cada registro, la información será mostrada dentro del área de búsqueda, véase en la Figura 9.

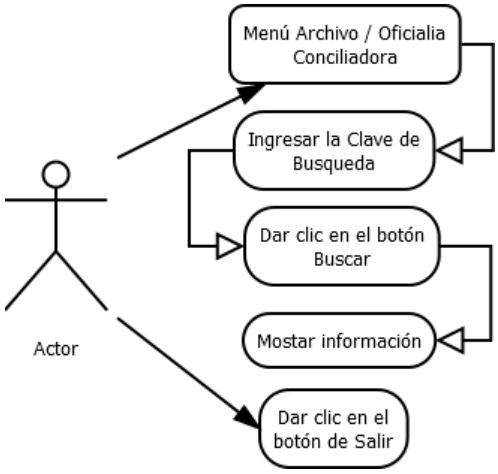


Figura 9 Diagrama de caso de uso "Búsqueda". Fuente: Elaboración propia 2017.

Modificar.

Menú mostrado en la Figura 10, que permite buscar y modificar un registro especifico, se utiliza de la siguiente forma; una vez que se ha realizado la búsqueda del registro que se desea modificar se realizan la modificación deseada en los campos de datos y enseguida se presiona el botón <modificar>.

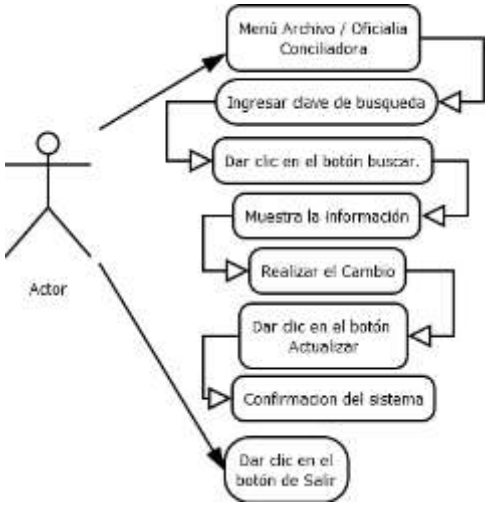


Figura 10 Diagrama de caso de uso <Actualizar>. Fuente: Elaboración propia 2017.

3.3.3 Diseño del modelo relacional

Diseño de la base de datos con el modelo relacional, la cual fue desarrollada utilizando el programa MySQL 8.1 OS, lo que permitió el desarrollo de la base de datos de una manera gráfica, véase la Figura 11.

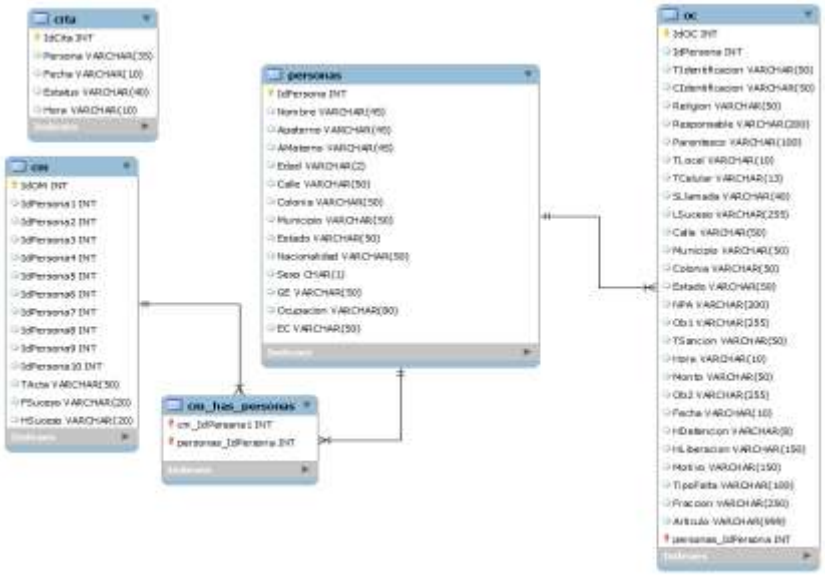


Figura 11 Modelo relacional de la base de datos de CODOMCyC. Fuente: Elaboración propia 2017.

3.3.4 Diseño de vistas

Para poder llevar a cabo el diseño de las vistas, se han tomado en cuenta los datos necesarios de las actas y citas. Generando diseños de prueba con la finalidad de facilitar a los usuarios el uso del sistema

3.4 Codificación

Esta es la fase que corresponde a la generación de código fuente del sistema, derivado de los diseños de la etapa anterior. A continuación, se presenta la forma en que se ha implementado la Base de Datos del sistema.

3.4.1 Implementación de la base de datos en MYSQL

Se ha normalizado la base de datos y desarrollado el modelo relacional, se ha implementado la base de datos utilizando MySQL Server 8.1, en un equipo de computadora con las siguientes características: (laptop emachines) procesador intel celeron 900, 1 GB de memoria RAM ejecutando un sistema operativo Windows 7.

Se siguen los siguientes pasos para la implementación de la base de datos en MySQL. Véase la Figura 12.

1. Se acceso a MySQL Server 8.1

2. Se crea una base de datos con el nombre "seguridad"

3. Se hace uso de la base de datos creada. **Anexo1 Script.**

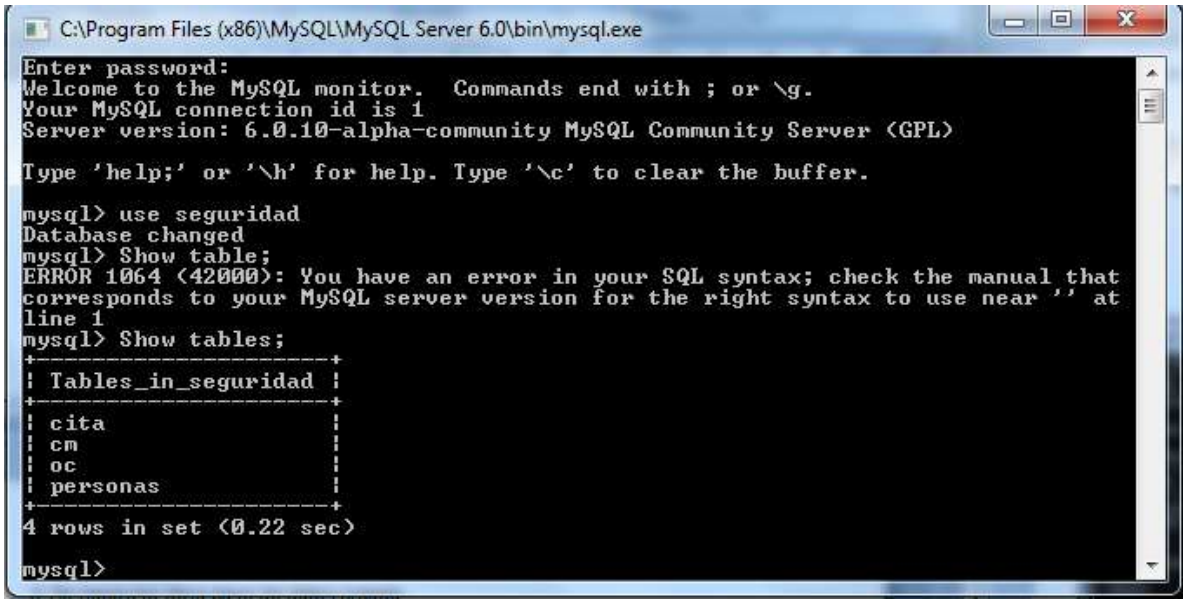


Figura 12 Creando y usando la base de datos. Fuente: Elaboración propia 2017.

Se han cargado cada una de las tablas que conforman el modelo relacional con sus respectivos campos, definiendo los tipos de datos que cada una de estas almacenan al igual que sus respectivas longitudes. Véase la Figura 13.

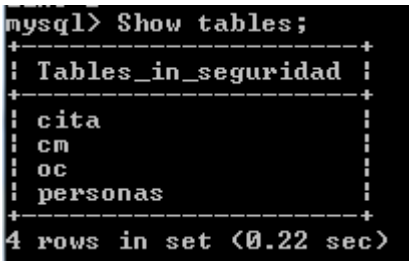


Figura 13 Tablas que conforman la base de datos cargadas en MySQL. Fuente: Elaboración propia 2017.

3.4.2 Implementación de algoritmo para realizar los procesos

Para realizar el planteamiento de los algoritmos fue necesario conocer las variables utilizadas dar de alta las actas y citas. Planteando el desarrollo de la base de datos en un modelo relacional que sirvió para definir las tablas y su contenido, utilizadas para programar los algoritmos mediante programación orientada a objetos utilizando Java en base al patrón MVC

3.4.3 Programación de los algoritmos

Para la codificación de algoritmos y que estamos cubriendo las necesidades de tienda se utiliza un lenguaje de programación Java mediante la aplicación del patrón MVC bajo la plataforma NetBeans IDE 8.1

NetBeans proporciona un entorno de desarrollo para el lenguaje de programación Java, se trata de un software libre y gratis sin restricciones de uso, que permite a las aplicaciones ser desarrolladas por modulos. Un módulo es un archivo Java que con tiene clases de Java escritas para interactuar con las APIs[4] de NetBeans.

3.4.4 Implementación de los algoritmos en el lenguaje java utilizando el patrón MVC

Para aplicar el patrón MVC en el desarrollo del sistema del control de inventario fue necesario crear paquetes que permiten mantener la estructura de dicho patrón, véase la Figura 14.

[4]**Interfaz de programación de aplicaciones** (**IPA**) o API (del inglés *Application Programming Interface*) es el conjunto de funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software

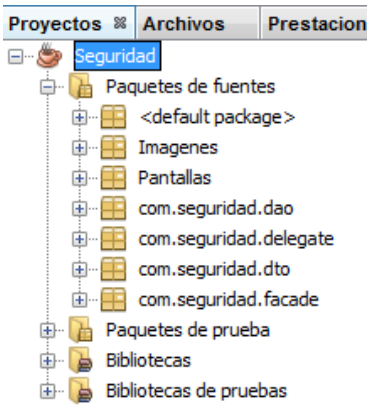


Figura 14 Paquetes de patrón MVC. Fuente: Elaboración propia 2017.

- **Paquete de Imagen:** Este paquete almacena las imágenes que utiliza el sistema tales como iconos, logotipos e imágenes ilustrativas que describen el uso del sistema.
- **Paquete com.seguridad.dto:** "Data Transfer Object" es un JavaBean que representa una entidad en la aplicación. Es un objeto que almacena información y que será usado para transferir información por la aplicación.
- **Paquete com.seguridad.dao:** Los "Data Access Objects" son los objetos que proporcionan acceso a los DTO. A través de ellos facilitaremos el acceso y manipulación de los datos en la aplicación.
- **Paquete com.seguridad.facade:** Simplifica el acceso a un conjunto de clases proporcionando una única clase que utiliza para comunicar un conjunto de clases que se encuentran en el paquete DAO.
- **Paquete com.seguridad.delegate:** Es implementado dentro del desarrollo del sistema porque se requiere extender y reutilizar la funcionalidad de una clase sin dependencia totalmente de la herencia, este patrón permite compartir código que no se puede heredar.
- **Paquete seguridad:** Mantiene las interfaces con que se relaciona el usuario.
- **Paquete Pantallas:** Contienen las pantallas del sistema. **Anexo 2 Pantallas.**

EL patrón Modelo de Vista Controlador centra la información con que el sistema opera en los distintos paquetes desarrollado a excepción del paquete seguridad, el paquete seguridad agrupa las interfaces de usuario, mientras que el controlador responde a eventos, usualmente acciones del usuario, e invoca peticiones al modelo y a la vista

3.5 Implementación

Esta fase se refiere a la puesta en marcha del sistema en el hardware del usuario

3.5.1 Tipo de implementación

Implementación directa: Es decir se proceder a trabajar directamente con la aplicación ya sea porque no existe aplicación alguna o bien se sustituye directa mente la anterior por el nuevo sistema y en la práctica se evaluará para mejoraras y ajustes a la aplicación. Este es el tipo de implementación que se ha elegido debido a que no existe un sistema anterior.

Implementación en paralelo: En esta se a trabajar con las dos aplicaciones, la nueva y la anterior comprobando resultados para ir evaluando la nueva y revisar que esté correcta o bien requiere de mejoras[5].

3.5.2 Capacitación

Se dará capacitación al personal para el buen funcionamiento del sistema.

3.6 Pruebas

Esta etapa es la que se encarga de la realización de las comprobaciones de cada módulo del sistema y la interacción de estos entre sí.

3.6.1 Tipo de Prueba

Existen diversos métodos para realizar las pruebas de software, entre las más importantes se encuentran la prueba de Caja Blanca y prueba de Caja Negra

El uso de la prueba de Caja Blanca es mejor para verificar que se recorran todos los caminos y detectar un mayor número de errores. Esta prueba es la que más se adapta a la situación debido a que se cuenta con el código fuente.

La Caja Negra brinda la posibilidad de cubrir la mayor parte de las combinaciones de entradas y lograr así un juego de pruebas más eficaz.

Las pruebas mencionadas permiten probar cada una de las condiciones existentes en el programa, identificar claramente las entradas, salidas y estudiar las relaciones que existen

[5] http://william-programasweb.blogspot.mx/2009/07/implementacion.html

entre ellas, permitiendo así maximizar la calidad de las pruebas y en dependencia del resultado se constará con un sistema más estable y confiable. (Martínez, 2015)

3.6.2 Prueba de Caja Negra

En ellas se pretende examinar el programa en busca de que cuente con las funcionalidades que debe tener y como lleva a cabo las mismas, analizando siempre los resultados que devuelve y probando todas las entradas en sus valores válidos e inválidos.

Se desarrollan casos de prueba reales para cada condición o combinación de condiciones y se analizan los resultados que arroja el sistema para cada uno de los casos. En esta estrategia se verifica el programa considerándolo una caja negra. Las pruebas no se hacen en base al código, sino a la interfaz. No importa que se cubran todas las rutas dentro del programa, lo importante es probar todas las entradas en sus valores válidos e inválidos y lograr que el sistema tenga una interfaz amigable. (Martínez, 2015)

3.6.3 Prueba de Caja Blanca

También suelen ser llamadas estructurales o de cobertura lógica. En ellas se pretende investigar sobre la estructura interna del código, exceptuando detalles referidos a datos de entrada o salida, para probar la lógica del programa desde el punto de vista algorítmico. Realizan un seguimiento del código fuente según se va ejecutando los casos de prueba, determinándose de manera concreta las instrucciones, bloques, etc. que han sido ejecutados por los casos de prueba.

Se desarrollan casos de prueba que produzcan la ejecución de cada posible ruta del programa o módulo, considerándose una ruta como una combinación específica de condiciones manejadas por un programa.

Hay que señalar que no todos los errores de software se pueden descubrir verificando todas las rutas de un programa, hay errores que se descubren al integrar unidades del sistema y pueden existir errores que no tengan relación con el código específicamente. (Martínez, 2015)

3.6.3.1 Características de las pruebas de Caja Blanca

Este tipo de pruebas se ha desarrollado en el sistema ya que se indaga sobre la estructura interna del código, omitiendo detalles referidos a datos de entrada o salida. Su objetivo principal es probar la lógica del programa desde el punto de vista algorítmico.

Estas se basan en el diseño de Casos de Prueba que usa la estructura de control del diseño procedimental para derivarlos. Mediante las pruebas de Caja Blanca el ingeniero de software puede obtener Casos de Prueba que:

- Garanticen que se ejerciten por lo menos una vez todos los caminos independientes de cada módulo, programa o método.
- Ejerciten todas las decisiones lógicas en las vertientes verdadera y falsa.
- Ejecuten todos los bucles en sus límites operacionales.
- Ejerciten las estructuras internas de datos para asegurar su validez.

Las pruebas de Caja Blanca son consideradas entre las más importantes que se aplican a los sistemas, con la que se obtienen como resultados la disminución en un gran porciento el número de errores existentes en el software y por ende una mayor calidad y confiabilidad en la codificación. (Martínez, 2015).

3.6.3.2 Técnicas de Caja Blanca

Este tipo de técnicas se le conoce también como Técnicas de Caja Transparente o de Cristal.

• Este método se centra en cómo diseñar los casos de prueba atendiendo al comportamiento interno y la estructura del programa.
• Se examina así la lógica interna del programa sin considerar los aspectos de rendimiento.
• El objetivo de la técnica es diseñar casos de prueba para que se ejecuten, al menos una vez, todas las sentencias del programa, y/o todas las condiciones tanto en su vertiente verdadera como falsa.
Puede ser impracticable realizar una prueba exhaustiva de todos los caminos o sentencias de un programa => se han definido distintos criterios de cobertura lógica, que permiten decidir qué sentencias o caminos se deben examinar con los casos de prueba.

3.6.3.2.1 Técnicas de cobertura de caminos

Camino: Secuencia de sentencias encadenadas desde la entrada del programa hasta su salida.
Se escriben casos de prueba suficientes para que se ejecuten todos o algunos de los caminos de un programa.
Criterios:

- **Todos los caminos.**
  - No es muy práctico por la cantidad de rutas a probar.
  - Un programa puede tener un gran número de rutas a probar o un número infinito.
  - Este criterio es considerable siempre que detecte los fallos, sin embargo, se presenta mucha dificultad para llevar a la práctica.

- **Cobertura de Sentencias.** (Figura 15)
  - Una cobertura de sentencias se ha logrado si todas las declaraciones han sido ejecutadas al menos una vez.
  - La cobertura de la sentencia completa es el criterio más débil de la cobertura en las pruebas
  - El problema básico es seleccionar unos pocos caminos que recorran todos los nodos de un control de flujo gráfico (CFG) para lograr la cobertura declaración completa.

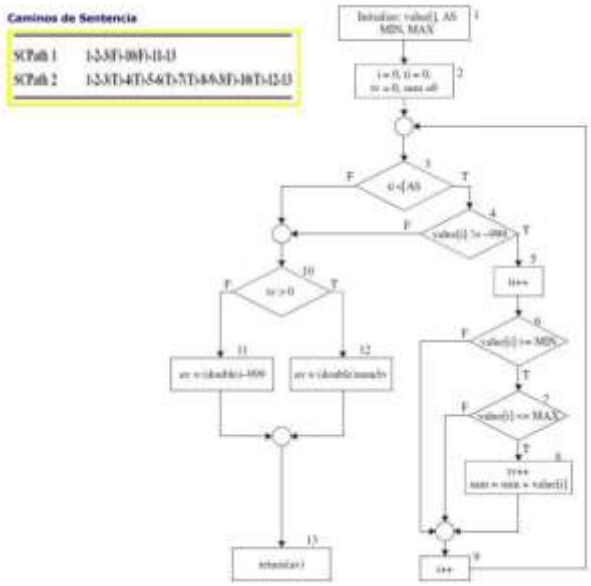


Figura 15 Cobertura de Sentencia.

- **Ramas:** Es una arista saliente de un nodo. (Figura 16)
  - Todos los nodos rectángulos deben tener como máximo una rama de salida.
  - Todos los nodos de diamante tienen dos ramas de salida.

- Cobertura de ramas completa significa la selección de un número de caminos de manera que cada rama se incluye en al menos una ruta.

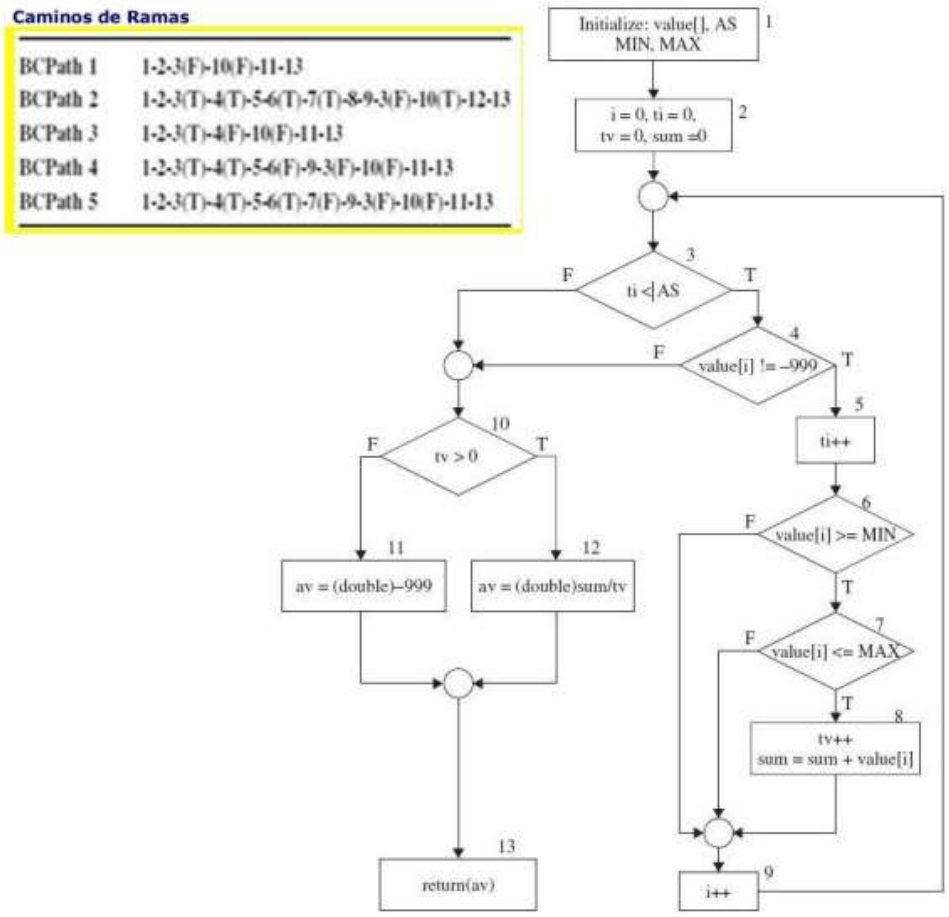


Figura 16 Ramas.

- **Predicados.** (Figura 17)

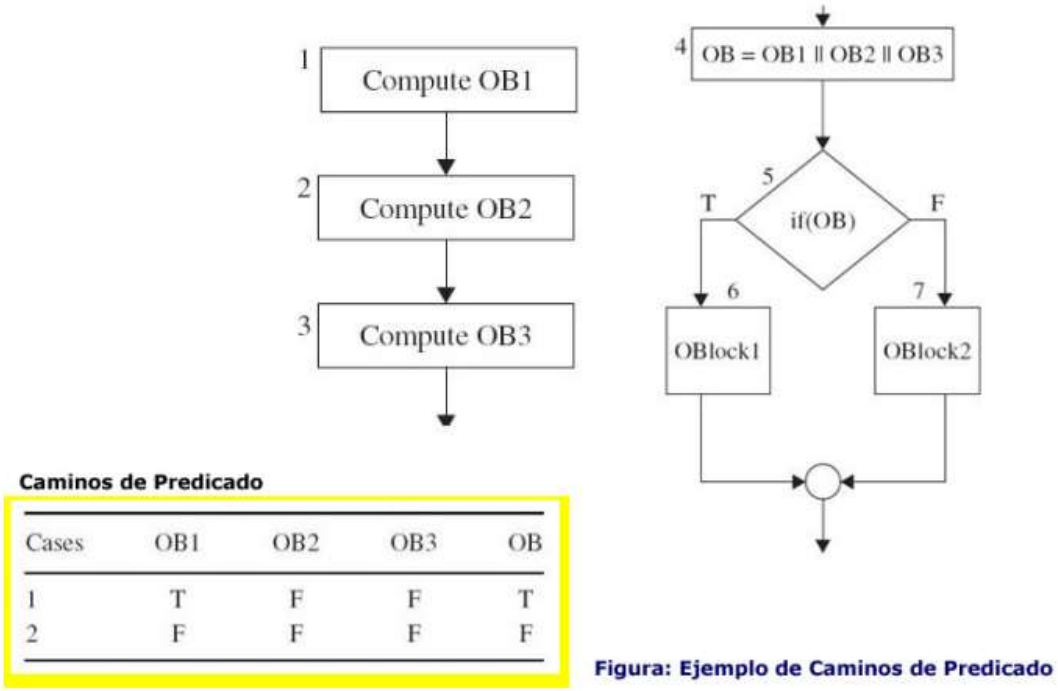


Figura 17 Camino de Predicado.

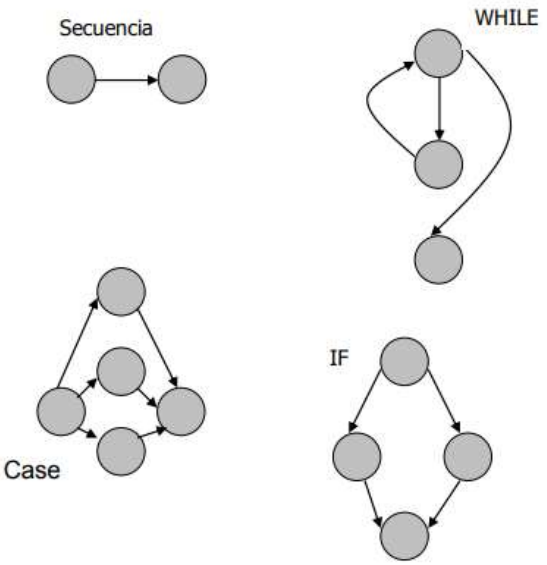
- **Ruta básica. (**Figura 18**)**

Procedimiento criterio de ruta(Camino) Básica(Co)

Se escriben casos de prueba suficientes para que se ejecuten todos los caminos de un programa. Los pasos a aplicar esta técnica son:

- Representar el programa en un grafo de flujo **(**Figura 19**)**
- Calcular la complejidad ciclomática
- Determinar el conjunto básico de caminos independientes
- Derivar los casos de prueba.

3.6.3.2.2 Técnicas de control de flujo



- **Nodos:** Representan cero, una o varias sentencias.
- **Aristas:** líneas que unen dos nodos.
- **Regiones:** áreas delimitadas por aristas y nodos. Cuando se contabilizan las regiones de un programa debe incluirse el área externa como una región más
- **Nodos Predicado:** Cuando en una condición aparecen uno o más operadores lógicos (AND, OR, XOR, ...) se crea un nodo distinto por cada una de las condiciones simples. Cada nodo generado de esta forma se denomina nodo predicado.

Figura 18 Control de Flujo.

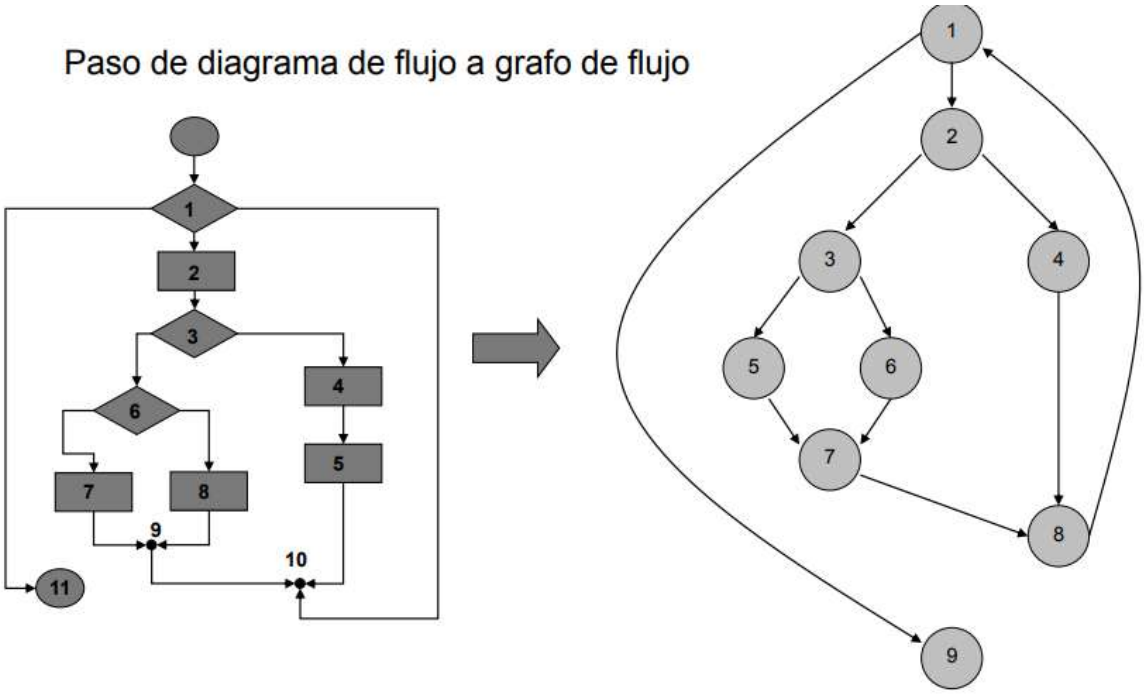


Figura 19 Diagrama de flujo a grafo de flujo.

3.6.3.2.3 Técnicas de cobertura de ciclos

Es una técnica de Caja Blanca, en donde el objeto es verificar los ciclos de un programa software. Estas técnicas se caracterizan por usar grafos para describir su funcionamiento. Estos grafos siempre se componen de: Aristas, nodos y regiones. (Figura 20)

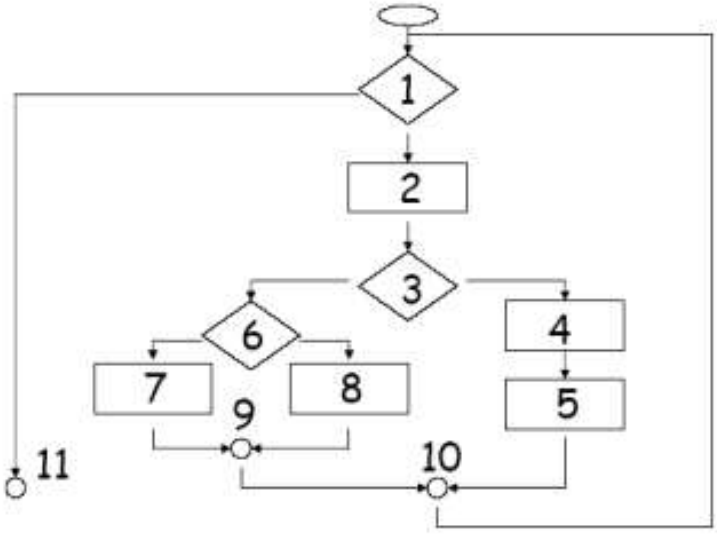


Figura 20 Cobertura de Ciclos.

Como existen diferentes tipos de ciclos hay que tenerlos en cuenta a la hora de analizarlos. Los tipos son:

• Simple
• Anidado
• Concatenado
• No estructurado Además también hay que tener las diferentes sentencias que hay para representar un ciclo:
• While
• Repeat
• For

Para usar esta técnica es importante tener el código del programa que estamos evaluando, buscando los algoritmos que contengan los ciclos. Una vez seleccionados los segmentos de código que contienen ciclos se procede a dibujar el grafo, esto se hace para poder identificar el recorrido lógico del código. Con el grafo y el código se identifica que criterio usar para aplicar pruebas.

3.6.4. Prueba de Caja blanca en CODOMCYC

Para realizar la implementación de las pruebas de Caja Blanca en el Sistema CODOMCYC, se tomó como referencia la técnica de cobertura de sentencias, se ha logrado si todas las declaraciones han sido ejecutadas al menos una vez.

En la ventana de Bienvenida, en esta interfaz solo se valida el acceso al sistema con un usuario y contraseña.

Botón de aceptar. (Figura 21)

Figura 21 Código de botón aceptar en la ventana de Bienvenida.

Diagrama de flujo. (Figura 22)

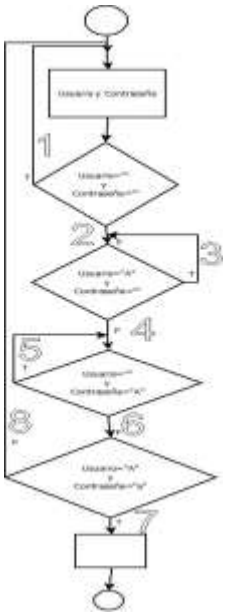


Figura 22 Diagrama de flujo del botón aceptar en la ventana de Bienvenida.

Ventana Conciliadora, en esta interfaz de manejan barios botones, en el botón de guardar se valida que los campos no estén vacíos, si es así se ejecuta el método crearPersonas y crearOC en donde se guarda la información en la base de datos.

Botón Guardar. (Figura 23)

```
if(jtFInterno.getText().equals("") || jLabel42.getText().equals("") || jTOcupacion.getText().equals("") || jTIdentificacio
    || jTRelacion.getText().equals("") ||jTGEstudios.getText().equals("")|| jTResponsable.getText().equals("") || jTParent
    || jTLocal.getText().equals("") ||jTCelular.getText().equals("")|| jTSLlamada.getText().equals("") || jLSuceso.getText
    || jTPMunicipio.getText().equals("") ||jTPColonia.getText().equals("")|| jTPEstado.getText().equals("") || jTNPA.getTe
    || jCHora.getSelectedItem().equals("")|| jTMonto.getText().equals("") || jTOb2.getText().equals("")|| jTFecha.getText(
    || jLabel42.getText().equals("") || jTFNombre.getText().equals("") || jTFAPaterno.getText().equals("") ||jTFAMaterno.g
    || jTCalle.getText().equals("") ||jTColonia.getText().equals("")|| jTMunicipio.getText().equals("") || jTEstado.getTex
{
    JOptionPane.showMessageDialog(this,"Por favor ingresa todos los datos","Aviso del Sistema",JOptionPane.ERROR_MESSAGE);
    return;
}
else
{   del.crearPersonas(dto);
    ddel.crearOC(ddto);
    JOptionPane.showConfirmDialog(this,"Registro dado de alta","Aviso del Sistema ",JOptionPane.DEFAULT_OPTION,JOptionPane
    Limpiar();
    jCHDetencion.removeAllItems();
    jCHLiberacion.removeAllItems();
    jCEdad.removeAllItems();
    jCCivil.removeAllItems();
    jCHora.removeAllItems();
    jTBSancion.removeAllItems();
    jCTFalta.removeAllItems();
    verCon(); comboEdad();comboCivil();comboFalta();comboHora();comboSancion();
}
```

1

2

Figura 23 Botón de guardar en la ventana de conciliadora.

Diagrama de flujo. (Figura 24)

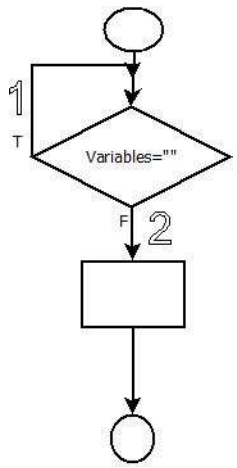


Figura 24 Diagrama de flujo del botón de guardar en la ventana de conciliadora.

En el botón de buscar, consulta en la base de datos los parámetros correspondientes y los muestra en la interfaz, en caso contrario que no existe mandara un mensaje el sistema.

Botón buscar. (Figura 25)

Figura 25 Botón de buscar de la ventana conciliadora.

Diagrama de Flujo. (Figura 26)

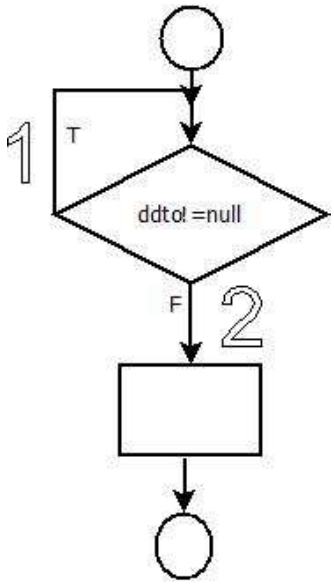


Figura 26 Diagrama de flojo del botón buscar de la ventana de conciliadora.

En el botón de actualizar, valida que todos los campos no estén vacíos, manda un mensaje de confirmación que, si requiere actualizar el registro, si es así, manda un mensaje que se actualizo correctamente, en caso contrario manda el mensaje registro no actualizado.

Botón Actualizar. (Figura 27)

```
if(jtfInterno.getText().equals("") || jLabel42.getText().equals("") || jTOcupacion.getText().equals("") || jTIdentifica
 || jTRelacion.getText().equals("") ||jTGEstudios.getText().equals("")|| jTResponsable.getText().equals("") || jTParent
 || jTLocal.getText().equals("") ||jTCelular.getText().equals("")|| jTSLlamada.getText().equals("") || jLSuceso.getText
 || jTFMunicipio.getText().equals("") ||jTPColonia.getText().equals("")|| jTPEstado.getText().equals("") || jTNPA.getTe
 || jCHora.getSelectedItem().equals("")|| jTMonto.getText().equals("") || jTOb2.getText().equals("")|| jTFecha.getText(
 || jLabel42.getText().equals("") || jTFNombre.getText().equals("") || jTFAPaterno.getText().equals("") ||jTFAMaterno.g
 || jTCalle.getText().equals("") ||jTColonia.getText().equals("")|| jTMunicipio.getText().equals("") || jTEstado.getTex
{
    JOptionPane.showMessageDialog(this,"Por favor ingresa todos los datos","Aviso del Sistema ",JOptionPane.ERROR_MESSAGE)
    return;
}
Component unComponentePadre = null;
int seleccion = JOptionPane.showOptionDialog(unComponentePadre,"¿Seguro que quieres actualizar\n el Registro?","Aviso del
if (seleccion == 0)
{
    del.actualizaPersonas(dto);
    ddel.actualizaOC(ddto);
    JOptionPane.showConfirmDialog(this,"Registro actualizado","Aviso del Sistema",JOptionPane.DEFAULT_OPTION,JOptionPane.I
    jCHDetencion.removeAllItems();jCHLiberacion.removeAllItems();jCEdad.removeAllItems();jCCivil.removeAllItems();
    jCHora.removeAllItems();jTBSancion.removeAllItems();jCTFalta.removeAllItems();
}else
{
    JOptionPane.showConfirmDialog(this,"Registro no actualizado","Aviso del Sistema",JOptionPane.DEFAULT_OPTION,JOptionPan
}
```

1
2
3

Figura 27 Botón de actualizar de la ventana conciliadora.

Diagrama de Flujo. (Figura 28)

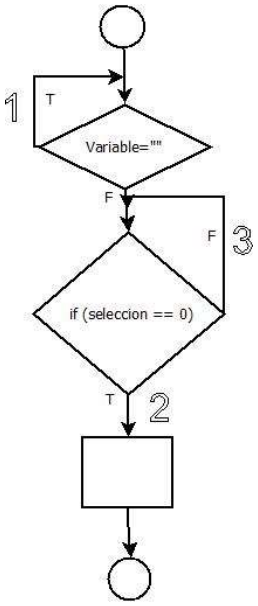


Figura 28 Diagrama de flujo del botón de actualizar de la ventana conciliadora.

Ventana de Mediadora, en esta interfaz de manejan barios botones, en el botón de guardar se valida que los campos no estén vacíos, si es así se ejecuta el método crearMediadora, en donde se guarda la información en la base de datos.

Botón de Guardar. (Figura 29)

Figura 29 Botón de guardar de la ventana de mediadora.

Diagrama de Flujo. (Figura 30)

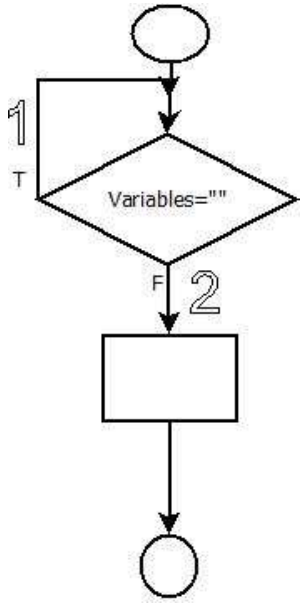


Figura 30 Diagrama de flujo del botón guardar de la ventana mediadora.

En el botón de buscar, consulta en la base de datos los parámetros correspondientes y las muestras en la interfaz, en caso contrario que no existe mandara un mensaje el sistema. Botón buscar. (Figura 31)

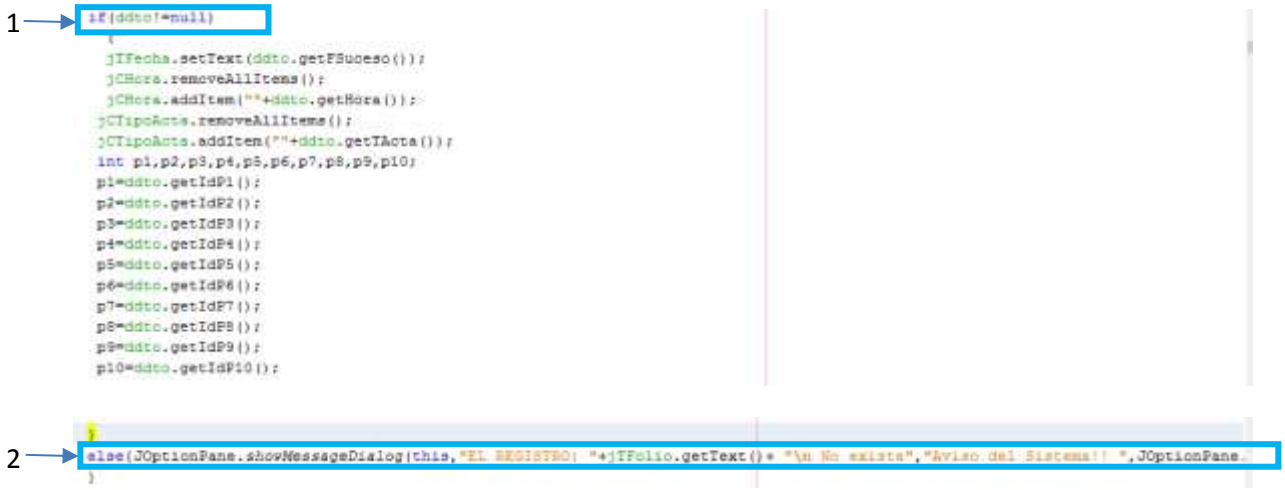


Figura 31 Botón de buscar de la ventana de mediadora.

Diagrama de Flujo. (Figura 32)

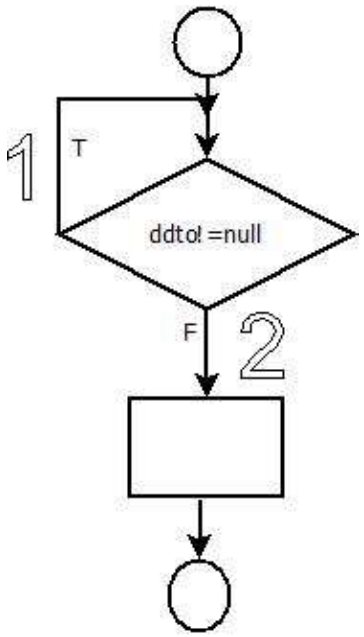


Figura 32 Diagrama de Flujo del botón de buscar de la ventana de mediadora.

En el botón de actualizar, valida que todos los campos no estén vacíos, manda un mensaje de confirmación que, si requiere actualizar el registro, si es así, manda un mensaje que se actualizo correctamente, en caso contrario manda el mensaje registro no actualizado.

Botón de Actualizar. (Figura 33)

```
1  if(jTFolio.getText().equals("")||jTFecha.getText().equals("")||jCHora.getSelectedItem().equals("")||L1.getText().equals("
     ||L4.getText().equals("")||L5.getText().equals("")||L6.getText().equals("")||L7.getText().equals("")||L8.getText().equa
   {
       JOptionPane.showMessageDialog(this,"Por favor ingresa todos los datos","Aviso del Sistema ",JOptionPane.ERROR_MESSAGE)
       return;
   }
   Component unComponentePadre = null;
   int seleccion = JOptionPane.showOptionDialog(unComponentePadre,"¿Seguro que quieres actualizar\n el Registro?","Aviso del
2  if (seleccion == 0)
   {   del.actualizarMediadora(ddto);
       JOptionPane.showConfirmDialog(this,"Registro actualizado","Aviso del Sistema",JOptionPane.DEFAULT_OPTION,JOptionPane.I
     jTFolio.setText("");
     jCHora.setSelectedIndex(0);
     jTFecha.setText("");
     Limpiar();
     jCTipoActa.removeAllItems(); jCEdad.removeAllItems(); jCCivil.removeAllItems();
     LimpiarTable();
     jTFecha.setText("");
   }
3  else
   {
       JOptionPane.showConfirmDialog(this,"Registro no actualizado","Aviso del Sistema",JOptionPane.DEFAULT_OPTION,JOptionPan
   }
```

Figura 33 Botón de actualizar de la ventana de mediadora.

Diagrama de Flujo. (Figura 34)

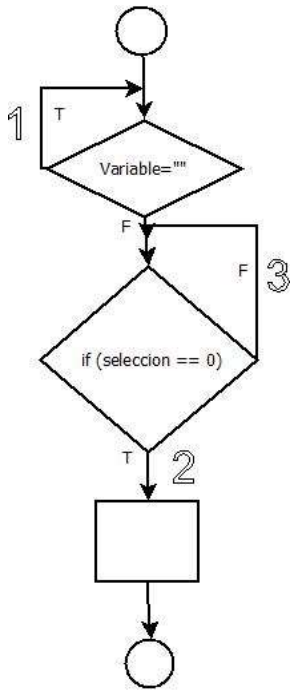


Figura 34 Diagrama de flujo del botón de actualizar de la ventana de mediadora.

Botón Agregar en la tabla, valida que todos los campos no estén vacíos, se guarda la información en la base de datos y se muestra en la tabla de la interfaz. (Figura 35)

Figura 35 Botón de agregar en la tabla de la ventana de mediadora.

Diagrama de Flujo. (Figura 36)

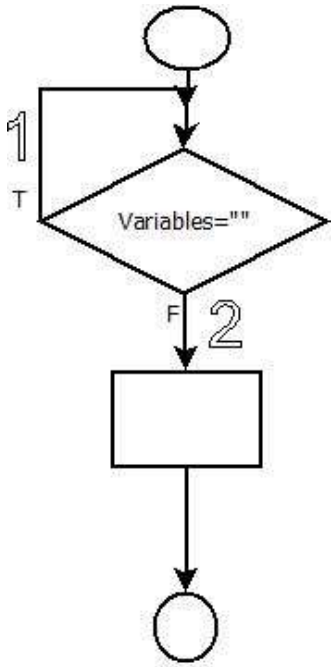


Figura 36 Diagrama de flujo del botón de agregar en la tabla de la ventana de mediadora.

Botón Actualizar para la tabla, en este botón toma los datos que se encuentra en cada campo, se me coloca en un for para actualizar la información en la tabla. (Figura 37)

```
String[] datos = new String[14];
datos[0] = con1.getText();
datos[1] = jTFNombre.getText();
datos[2] = jTFAPaterno.getText();
datos[3] = jTFAMaterno.getText();
datos[4] = String.valueOf(jCEdad.getSelectedItem());
datos[5] = jTCalle.getText();
datos[6] = jTColonia.getText();
datos[7] = jTMunicipio.getText();
datos[8] = jTEstado.getText();
datos[9] = jTNacionalidad.getText();
datos[10] = rsexo.getText();
datos[11] = Estudios.getText();
datos[12] = Ocupacion.getText();
datos[13] = String.valueOf(jCCivil.getSelectedItem());
1 → for (int i = 0; i < jTPersona.getColumnCount(); i++) {
        model.setValueAt(datos[i], filas, i);
}
```

Figura 37 Botón de actualizar tabla de la ventana de mediadora.

Diagrama de Flujo. (Figura 38)

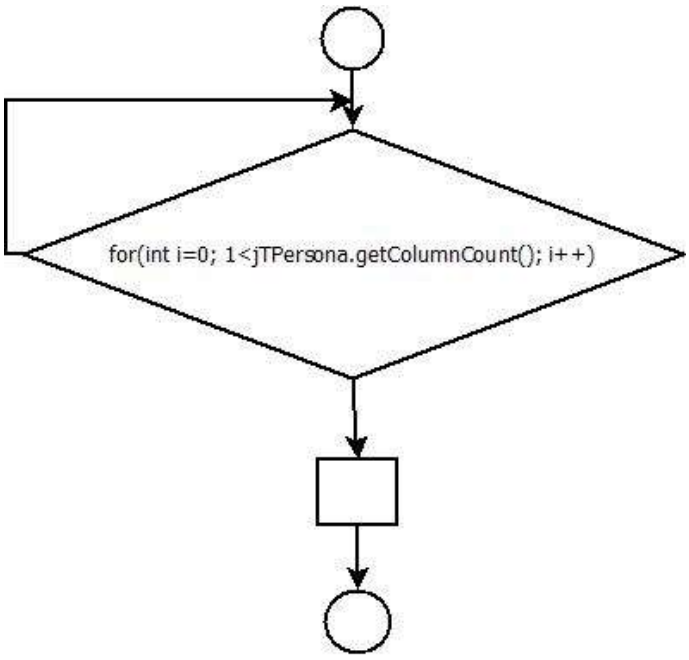


Figura 38 Diagrama de flujo del botón de actualizar de la ventana mediadora.

Ventana de citas

Guardar. Se crea el registro en la base de datos, validando que todos los campos estén llenos. (Figura 39)

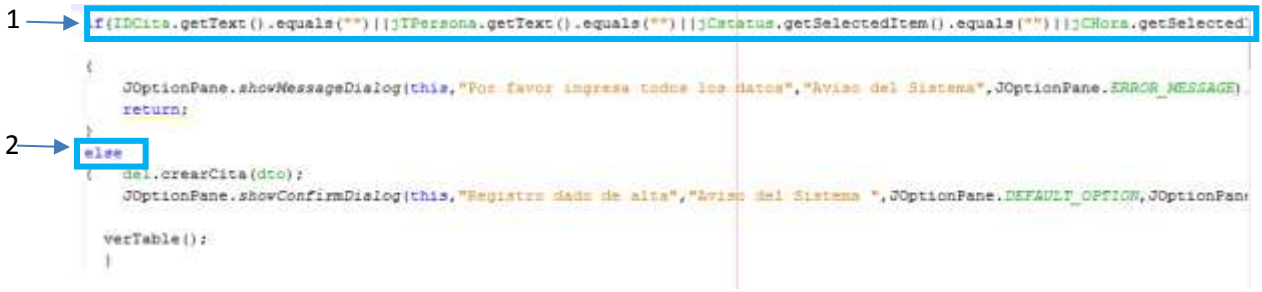


Figura 39 Botón de guardar de la ventana de citas.

Diagrama de Flujo. (Figura 40)

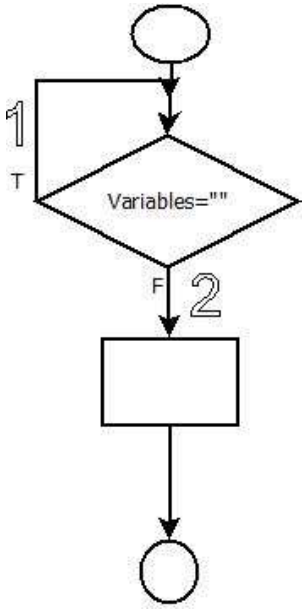


Figura 40 Diagrama de flujo del botón de guardar de la ventana de citas.

Buscar. Se realiza la búsqueda del registro, donde se muestra en la interfaz los datos, en caso contrario mostrara un mensaje que no existe el registro. (Figura 41)

```
1  if(dto!=null)
    {
    jTPersona.setText(dto.getPersona());
    jTFecha.setText(dto.getFecha());
   jCstatus.removeAllItems();
   jCHora.removeAllItems();
   jCstatus.addItem(""+dto.getEstatus());
   jCHora.addItem(""+dto.getHora());

2  }
   else{JOptionPane.showMessageDialog(this,"EL REGISTRO: "+IDCita.getText()+ "\n No existe","Aviso del Sistema!! ",JOptionPane.
   }
```

Figura 41 Botón de buscar de la ventana de citas.

Diagrama de Flujo. (Figura 42)

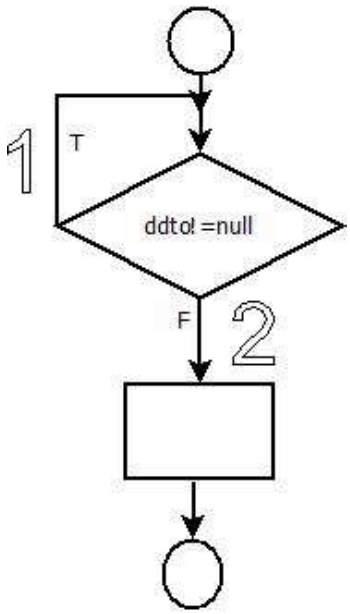


Figura 42 Diagrama de Flujo del botón de buscar de la ventana de citas.

Actualizar. Por medio de este botón de puede actualizar la información ya antes guardada en el registro, validando que los campos no estén vacíos. Mandando mensaje cuando se actualiza la información y otro cuando no se actualiza el registro. (Figura 43)

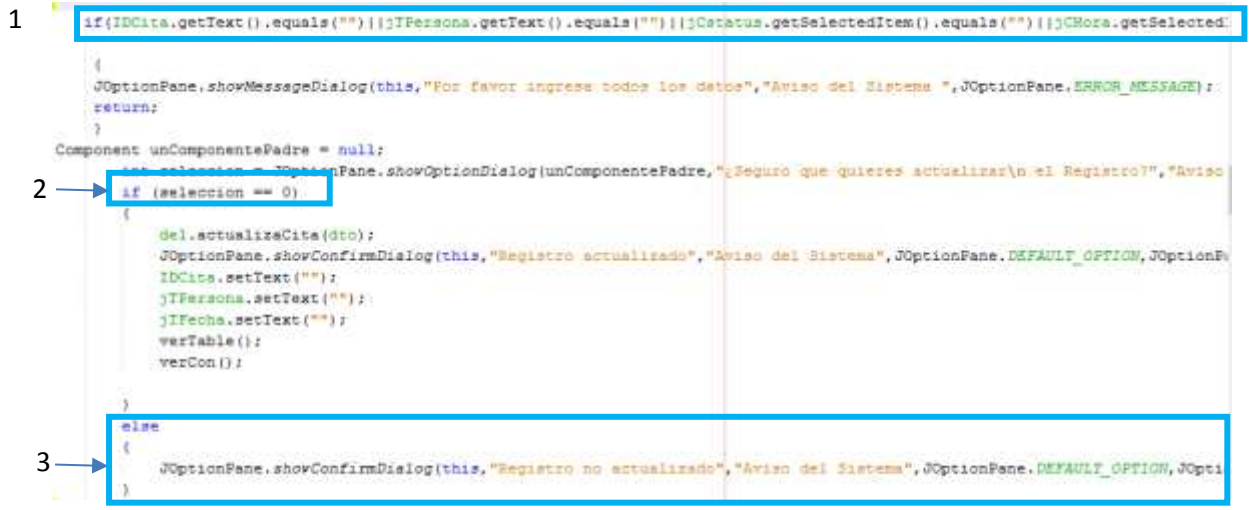


Figura 43 Botón de actualizar de la ventana de citas.

Diagrama de Flujo. (Figura 44)

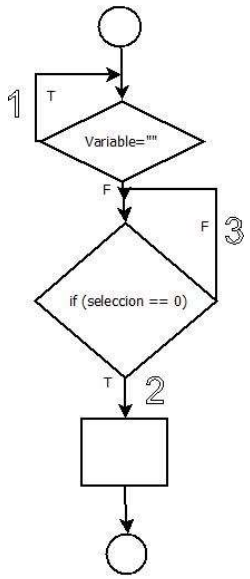


Figura 44 Diagrama de flujo del botón de actualizar de la ventana de citas.

Reporte de citas. En esta interfaz se realizan los reportes correspondientes, por ejemplo por un rango de fecha, por el nombre de la persona, por el estatus de la cita.

Botón de buscar. (Figura 45)

Figura 45 Botón buscar de la ventana de reportes de citas.

Diagrama de Flujo. (Figura 46)

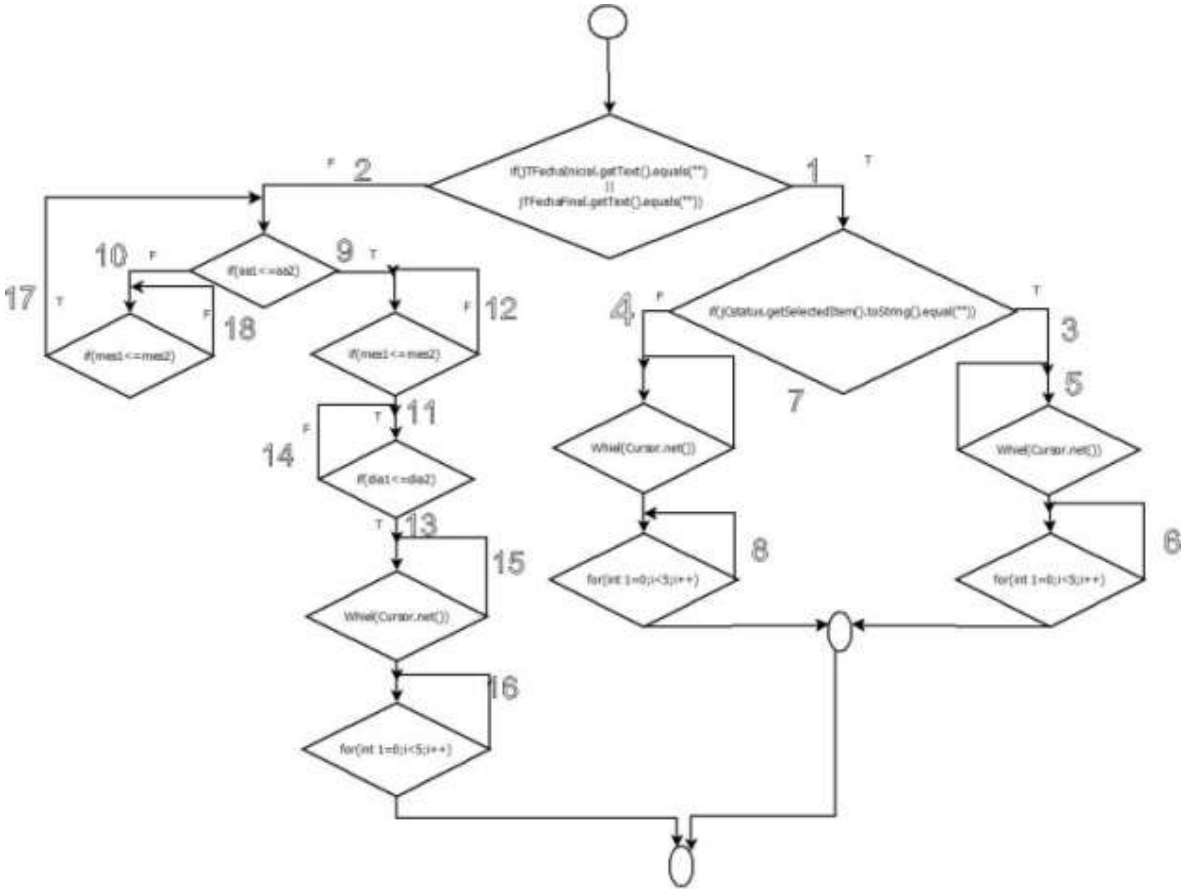


Figura 46 Diagrama de flujo del botón buscar de la ventana de reportes de citas.

Botón de Todos. En esta interfaz se realizan los reportes correspondientes, dando como resultado todos los registros dados de alta. (Figura 47)

```
Connection conexion;
        conexion = Conexion_BD.obtenerConexion();
        try (String sql_command = "select * from cita;";
        Statement st = (Statement) conexion.createStatement();
        ResultSet cursor = st.executeQuery(sql_command);
        DefaultTableModel modelo = (DefaultTableModel) jTBCitas.getModel();
1 ->    while (cursor.next()) {
        Object[] fila = new Object[5];
2 ->    for (int i = 0; i < 5; i++) {
        fila[i] = cursor.getObject(i + 1);
        }
         modelo.addRow(fila);
         }
        ) catch (SQLException e) {
          JOptionPane.showConfirmDialog(this, e.getMessage(), "Error de SQL (Al hacer SELECT)", JOptionPane.DEFAULT_OPTION, JOpt
        }
```

Figura 47 Botón todos de la ventana de reportes de citas.

Diagrama de Flujo. (Figura 48)

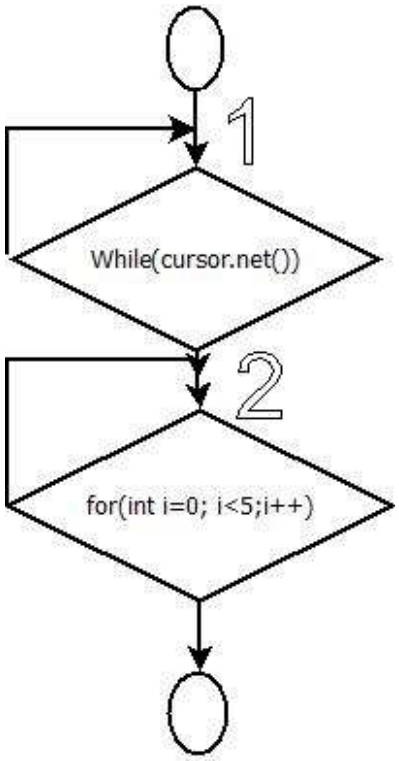


Figura 48 Diagrama de flujo del botón todos de la ventana de reportes de citas.

Botón exportar de la ventana de reportes de citas. (Figura 49)

```
1  if(selecArchivo.showDialog(null,"Exportar")==JFileChooser.APPROVE_OPTION)
   {archivo=selecArchivo.getSelectedFile();
3    if(archivo.getName().endsWith("xls")||archivo.getName().endsWith("xlsx"))
       {JOptionPane.showMessageDialog(null, modeloE.Exportar(archivo,jTRCitas));}
4    else{JOptionPane.showMessageDialog(null, "Seleccione un formato valido.");}
   }
```

Figura 49 Botón de exportar de la ventana de reportes de citas.

Diagrama de Flujo. (Figura 50)

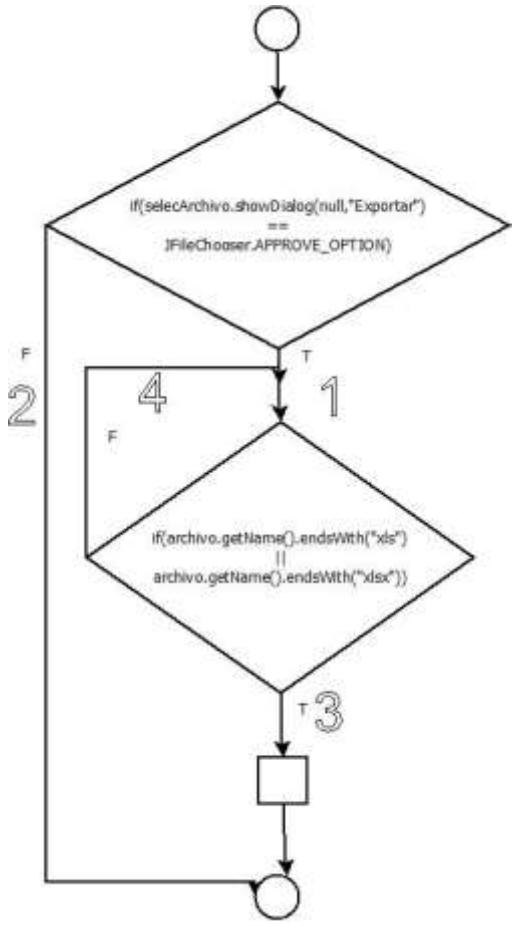


Figura 50 Diagrama de flujo del botón de exportar de la ventana de reportes de citas.

Venta de reporte de Conciliadora

Botón exportar. En esta interfaz se realizan los reportes correspondientes, por ejemplo, por un rango de fecha, por el nombre de la persona, por el estatus de la cita. (Figura 51)

```
2 →  else
     {Connection conexion;
         conexion = Conexion_BD.obtenerConexion();
         try {
             String sql_command = "Select a.IdOC,concat(b.Nombre,' ',b.APaterno,' ',b.AMaterno)as Nombre,a.TIdentificacion,a.CIdentificacion
             Statement st = (Statement) conexion.createStatement();
             ResultSet cursor = st.executeQuery(sql_command);
             DefaultTableModel modelo = (DefaultTableModel)jTConciliadora.getModel();
16 →         while (cursor.next()) {
                 Object[] fila = new Object[28];
17 →            for (int i = 0; i <28; i++) {
                     fila[i] = cursor.getObject(i + 1);
                 }
                 modelo.addRow(fila);
             }
         }
         catch (SQLException e) {
             JOptionPane.showConfirmDialog(this, e.getMessage(), "Error de SQL (Al hacer SELECT)", JOptionPane.DEFAULT_OPTION, JOptionPane.E
         }
     }
```

Figura 51 Botón de buscar de la ventana de reporte de Conciliadora.

Diagrama de Flujo. (Figura 52)

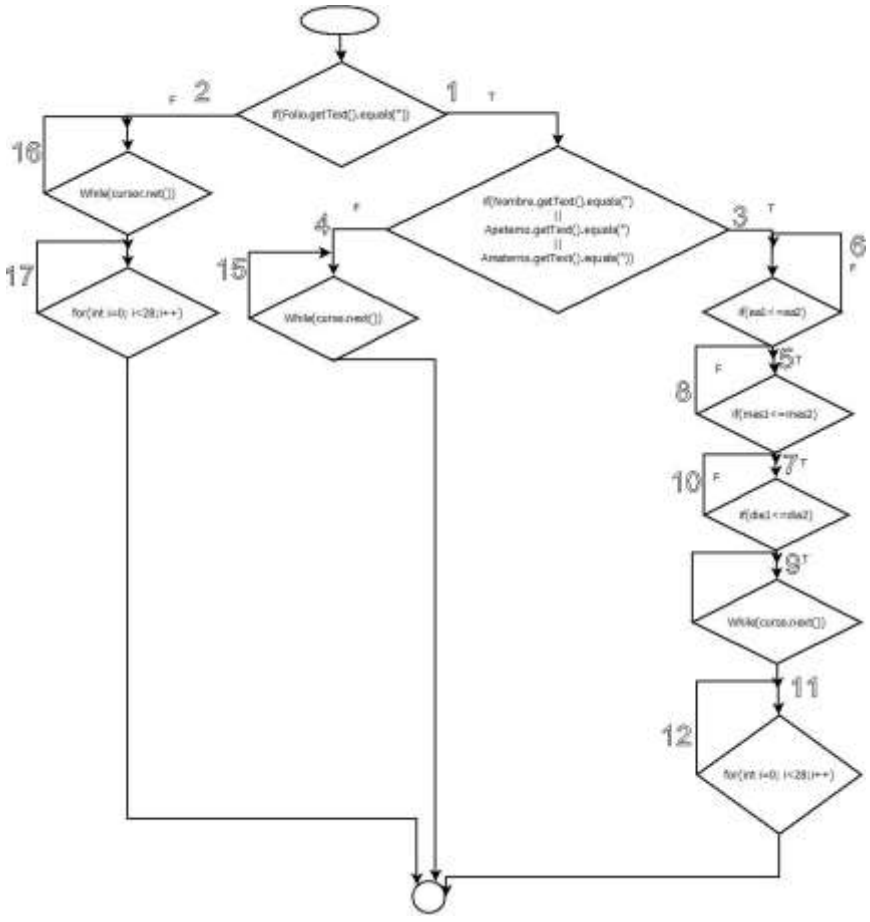


Figura 52 Diagrama de Flujo Botón del buscar de la ventana de reporte de Conciliadora.

Botón de Todos. En esta interfaz se realizan los reportes correspondientes, por ejemplo, por un rango de fecha, por el nombre de la persona, por el estatus de la cita. (Figura 53)

```
Connection conexion;
conexion = Conexion_BD.obtenerConexion();
try {
    String sql_command = "Select a.IdCC,concat(b.Nombre,' ',b.APaterno,' ',b.AMaterno)as Nombre,a.TIdentificacion,a.CIdentificacion
    Statement st = (Statement) conexion.createStatement();
    ResultSet cursor = st.executeQuery(sql_command);
    DefaultTableModel modelo = (DefaultTableModel)jTConciliadora.getModel();
1   while (cursor.next()) {
        Object[] fila = new Object[28];
2       for (int i = 0; i <28; i++) {
            fila[i] = cursor.getObject(i + 1);
        }
        modelo.addRow(fila);
    }
}
catch (SQLException e) {
    JOptionPane.showConfirmDialog(this, e.getMessage(), "Error de SQL (Al hacer SELECT)", JOptionPane.DEFAULT_OPTION, JOptionPane.E
}
```

Figura 53 Botón de todos de la ventana de reporte de Conciliadora.

Diagrama de Flujo. (Figura 54)

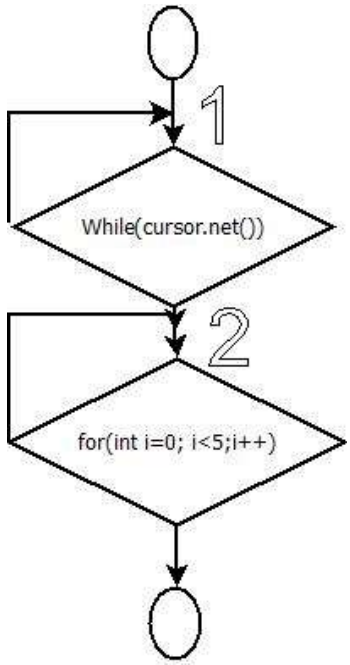


Figura 54 Diagrama de flujo del botón de todos de la ventana de reporte de Conciliadora.

Botón de exportar en la ventana de reportes mediadora. La operación de este botón es tomar la información consultado y exportarlo a un archivo de Excel, en una ruta que usuario seleccione. (Figura 55)

Figura 55 Botón de exportar en la ventana de mediadora.

Diagrama de Flujo. (Figura 56)

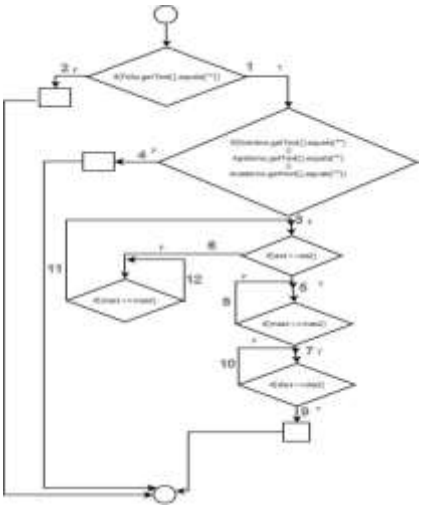


Figura 56 Diagrama de flujo del botón de exportar en la ventana de mediadora.

Botón de todos en la ventana de reporte de mediadora. (Figura 57)

```
1 → if(selecArchivo.showDialog(null,"Exportar")==JFileChooser.APPROVE_OPTION)
    {archivo=selecArchivo.getSelectedFile();
3 → if(archivo.getName().endsWith("xls")||archivo.getName().endsWith("xlsx"))
      {JOptionPane.showMessageDialog(null, modeloE.Exportar(archivo,JTMediadora,JTMediadora1));}
4 → else{JOptionPane.showMessageDialog(null, "Seleccione un formato valido.");}
    }
```

Figura 57 Botón de todos de la ventana de reportes mediadora.

Diagrama de Flujo. (Figura 58)

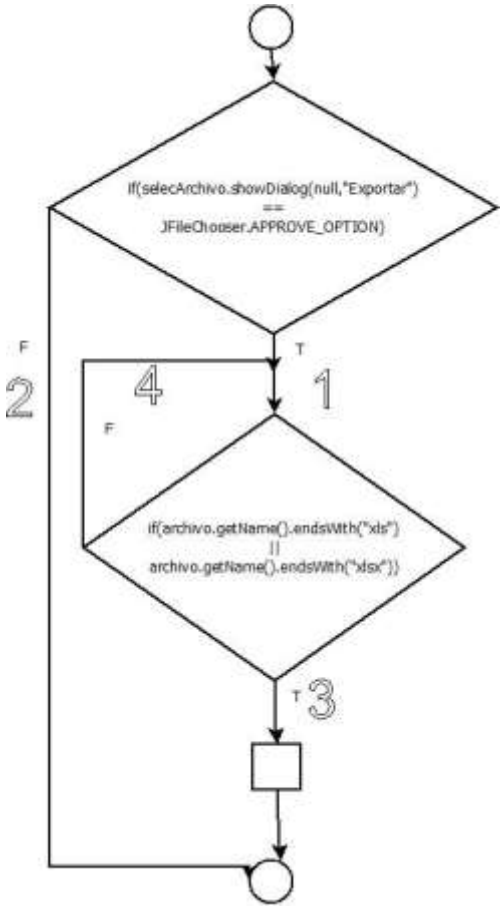


Figura 58 Diagrama de Flujo del botón de todos de la ventana de reportes mediadora.

3.6.5. Tabla de resultados CODOMCYC

Prueba de Caja Blanca			
Acción de botones	**Funciono Correctamente**		**Comentarios**
	Si	**NO**	
Ventana de Bienvenida			
Aceptar	*		
Ventana de Conciliadora			
Guardar	*		
Buscar	*		
Actualizar	*		
Ventana de Mediadora			
Guardar	*		
Buscar	*		
Actualizar	*		
Agregar	*		
Actualizar	*		
Ventana de Citas			
Guardar	*		
Buscar	*		
Actualizar	*		
Ventana de Reportes Citas			
Buscar	*		
Todos	*		
Exportar			
Ventana de Reportes Conciliadora.			
Exportar	*		
Todos	*		
Ventana de Reportes Mediadora			
Exportar	*		
Todos	*		

3.6.6. Modificaciones en Vistas CODOMCYC

La mejor opción sería que también marcara minutos, porque no todos los registros se hacen en horas cerradas. (Figura 59)

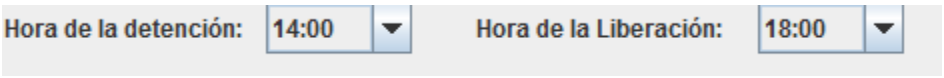


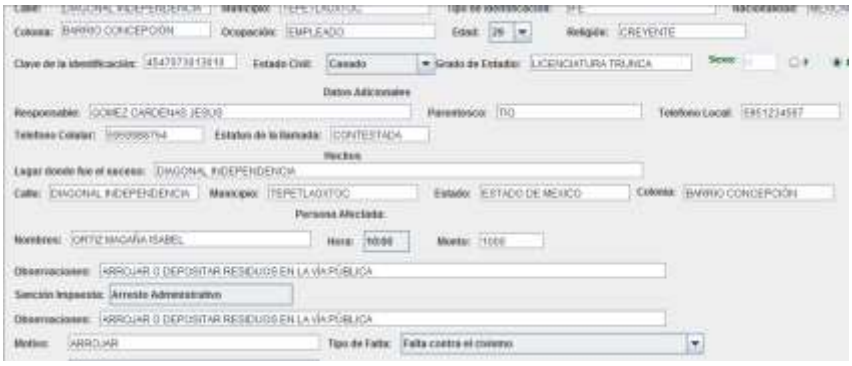
Figura 59 Modificación 1.

Solución**.** (Figura 60)

Figura 60 Solución a la modificación 1.

Modificar vista

Me sería más útil que las opciones de estado civil, sanción impuesta y tipo de falta. También se muestre en mayúsculas. (Figura 61)



Figuración 61 Modificación 2.

Solución**.** (Figura 62)

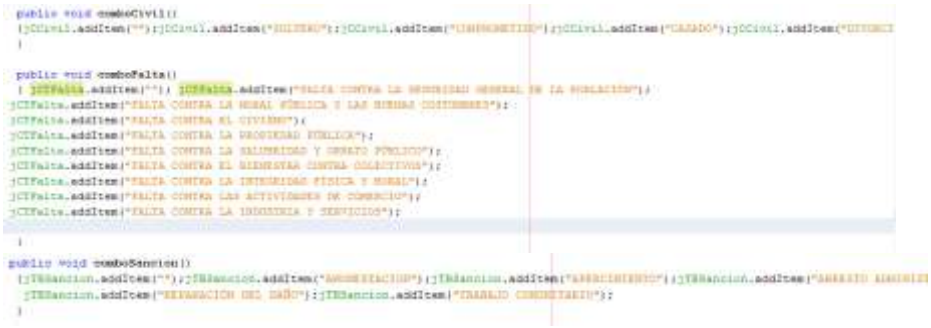


Figura 62 Solución de la modificación 2.

Modificar vista

Requiero que se muestre el texto completo en el cuadro, para una mejor visualización. (Figura 63)

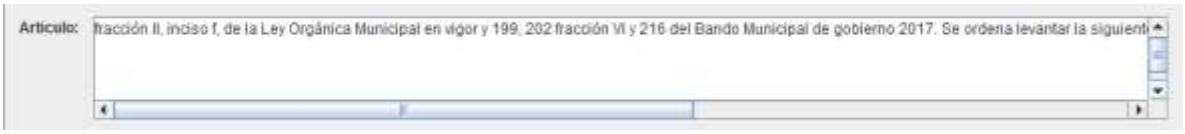


Figura 63 Modificación 3.

Solución. (Figura 64)

Figura 64 Solución de la modificación 3.

Mantenimiento

Esta etapa indica los módulos que requieren de adaptación a especificaciones nuevas.

3.7.1 Módulos Recomendados

Se recomienda hacer una copia de la base de datos cada mes y reiniciar la base de datos, para que su funcionalidad se la adecuada. Ya que se registran al día 15 actas y en el año anterior se crearon 547documentos.

Capítulo 4 Documentación

Se le proporciona un manual de usuario, en donde especifica cómo opera el sistema y la interacción que tiene. **Anexo 3 Manual de Usos.**

4.1 Conclusiones

El sistema registra las actas y se pueden guardan en formato Word, se registran las citas, dando los reportes en tiempo real, por persona o por rango de fechas, exportando a un archivo de Excel.

La implementación del patrón MVC en CODOMCyC, permite que la programación del sistema sea más eficiente y fácil de modificar si esto es necesario.

El sistema CODOMCyC logra iniciar con una administración de expedientes y diversos casos que la oficialía recibe, con el sistema se pueden obtener de manera rápida y eficiente consultas sobre los casos que el ciudadano ha iniciado. En los municipios no existe un sistema como el que se ha desarrollado en este trabajo de tesis, nuestra aportación ha sido relevante en este sentido.

4.2 Trabajos Futuros

Si en un determinado tiempo el área crece se puede migrar el sistema a una página web, por medio de una intranet, en donde se le permita ingresar a la base de datos multiusuarios, en distintos equipos de cómputo.

El sistema CODOMCyC puede ser considerado la detección de violencia familiar, así como problemas con equidad de género, principalmente en las mujeres. De usarse de manera regular, se pueden obtener información para la toma de decisiones en la continuidad de procesos de carácter legal. De implementarse en los distintos municipios del Estado de México, se tendría un mejor control de diversos tipos de violencia de género que las instancias de gobierno de mayor rango como: Procuraduría de Justicia a nivel estatal podrían tomar como antecedente administrativo.

Bibliografía

Arguello, F. (2011). Sistemas de reconocimiento basados en la imagen facial. Revista Avances en Sistemas e Informática. 18(3). Medellín.

Botella, A., Muñoz, A & Olivella, R, (2011), Sistema de información geografica y geotelematica, Mexico: Editarial: UOC.

Bakkas, J. & Bahaj, M. (2013). Generating of RDF graph from a relational database using JENA API. International Journal of Engineering and Technology. 5(2). (IJET).

Coetzer, W. Moodley, D. & Gerber, A. (2013). A Case-Study of Ontology-Driven Semantic Mediation of Flower-Visiting Data from Heterogeneous Data-Stores in Three South Africa

Cuevas, Alma-Delia (2006) "Unión de ontologías usando propiedades semánticas", Tesis Doctoral, Centro de Investigación en Computación del IPN, México.

Dadjo, M. & Kheirkhah E. (2015). An Approach For Transforming of Relational Database to OWL Ontology. International Journal of Web & Semantic Technology 6(1). (IJWesT).

Ortega, M (2017), Manual de Procedimientos 2017, Mexico. Editorial.?

E. KENDALL, K. & E. KENDALL, J.,(2005),Analisis y Diseño de Sistemas, México: Editorial: PEARSON, Prentice Hall

Fawiler, M., Scott,R. & Scott, K.,(2000),UML gota a gota, Mexico: Editorial=?

Garzon & Alexander,(2010),maestrosdelweb, consultado el 09 de Febrero del 2010. En http://www.maestrosdelweb.com/uml-diagramas-desarrollo-gran-escala/

Hinojosa, C. & Silvia, M.,(2012),Desarrollo de software de administración fiscal para MiPymes, México: Editorial: UAEM

Kalimuthu M. & Krishnamurthi I. (2015). Semantic –Based Facial Image-Retrieval System with Aid of Adaptive Particle Swarm Optimization and Squared Euclidian Distance. Journal of Applied Mathematics. Hindawi Publishing Corporation.

Karasneh, Y. Ibrahim.H. Othman, M. & Yaakob, R. (2009) .Integrating Schemas of Heterogeneous Relational Databases through Schema Matching. Proceedings of the 11th International Conference on Information Integration and Web-based Applications & Service. (iiWWAS2009). ACM 978-60558-660-1/09/0012.

Lewis, J. & Chase, J.,(2006),Estructura de datos con Java, Diseño de estructura y algoritmos,MADRID: Editorial: PEARSON ADDISON WESLEY

Martínez, E.,(2015),informática jurídica.com, consultado el 15 de Enero del 2017. En http://www.informatica-juridica.com/trabajos/procedimiento-realizar-pruebas-caja-blanca/

Moreno, A.,(2000),El enfoque relacional, consultado el 20 de Mayo del 2017. En http://elies.rediris.es/elies9/4-2-3.htm

R.M.Stair, R & G.W,Reynolds.,(2000), Principios de sistemas de información: enfoque administrativo, Mexico: Editorial Thomson.

Rob, P. & Coronel, C.,(2003),Sistemas de Bases de datos, México: Editorial=?

Sooksawaddee N., & Sasiporn U. (2013). Information Retrieval by Image Recognition using ontology Mapping with Neural Networks. International Journal of Computer Science and Electronics Engineering 1(4). (IJCSEE).

Willian, P.(2009),Implementación, consultado en 10 de Junio del 2017. En http://william-programasweb.blogspot.mx/2009/07/implementacion.html

(2001), EL modelo de datos relacional, consultado el 12 de Mayo del 2017. En http://www.cs.us.es/cursos/bd-2001/temas/modelo_relacional.html

Glosario

APIs= Interfaz de programación de aplicaciones (IPA) o API (del inglés Application Programming Interface) es el conjunto de funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software.
BD= Base de datos.
BDH= Base de datos Heterogéneas.
Bidimenionalidad= se utiliza para calificar a aquello que tiene dos dimensiones (2D). Un cuerpo que se proyecta a lo largo y a lo ancho, por ejemplo, cuenta con dos dimensiones.
Caso de Uso = Es una descripción del comportamiento de un sistema.
CODOMCyC = Control de Documentos del área de la Oficialía Mediadora, Conciliadora y Calificadora.
CORR= Análisis de subregiones, correlación.
DBMS = Sistema de Administración de Base de Datos.
DLL= Permiten crear y definir nuevas bases de datos, campos e índices.
DML= Permiten generar consultas para ordenar, filtrar y extraer datos de la base de datos.
Fisherfaces= Análisis de discriminantes lineales de Fisher.
Hardware = Es la parte física de un ordenador o sistema informático.
ICA= Análisis de componentes independientes
IODBC= Conectividad de bases de datos abierta independiente
JCBD= Conectividad a bases de datos Java
LPP= Conservación de proyecciones locales
MVC= Modelo de vista controlador.
ODBC= Conectividad de bases de datos abierta.
OLEDB= Objetos de enlaces y bases de datos embebidas.
Randomefaces= Proyecciones aleatorias
RN= Redes neuronales artificiales
SFIR= Sistema para recuperar imagen facial a base de semántica
SI= Sistema de Información.
Software = Es un término informático que hace referencia a un programa o conjunto de programas de cómputo que incluye datos, procedimientos y pautas que permiten realizar distintas tareas en un sistema informático.
SQL = (Structured Query Language), Lenguaje de consulta estructurado.
UML= Lenguaje de Modelado Unificado.

.

Anexo 1 Liberación de Proyecto de Sistema de Computo

H. Ayuntamiento Constitucional de Tepetlaoxtoc Estado de México 2016 - 2018

"2018, Año del Bicentenario del Natalicio de Ignacio Ramírez Calzada, El Nigromante"

DEPENDENCIA: SECRETARIA MUNICIPAL
OFICIO: TEPE/SM/082/2018

ASUNTO: **Liberación de Proyecto de Sistema de Cómputo**

Tepetlaoxtoc, Estado de México a 22 de Febrero del 2018

DR. EN DER. RICARDO COLIN GARCÍA.
DIRECTOR DEL CENTRO UNIVERSITARIO TEXCOCO
UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MEXICO.

PRESENTE.

Por medio del presente, le envió un cordial y afectuoso saludo, así mismo, hago constar que el C. **Miguel Ángel Romero Pérez** con número de cuenta **9021420,** de la Licenciatura en **Informática Administrativa** ha desarrollado el sistema de cómputo **RESOLUCIÓN DE INCONSISTENCIAS EN BASE DE DATOS RELACIONADOS PARA EL CONTROL DE DOCUMENTOS DEL ÁREA DE OFICIALÍA MEDIADORA CONCILIADORA Y CALIFICADORA (CODOMCYC) PARA EL MUNICIPIO DE TEPETLAOXTOC, ESTADO DE MÉXICO** en el periodo del 2017 al 2018. Mismo que será utilizado de manera regular para el control del área de **Oficialía Mediadora Conciliadora**. Una vez que las pruebas que se han realizado han sido de éxito.

En conclusión el sistema desarrollado resulta de utilidad debido a que nos ayudará a organizar la información generada, programar las citas y que la elaboración de las actas sea más eficiente, por lo que agradece la aportación profesional reflejado en el sistema.

Sin otro particular, reciba la seguridad de mi consideración y respeto, quedando a sus órdenes para cualquier duda o aclaración.

ATENTAMENTE

Prof. José Antonio Ortega Ayala
Presidente Municipal por Ministerio de Ley del H. Ayuntamiento de Tepetlaoxtoc.

Correo electrónico:
preMuntep.1618@hotmail.com

Lic. Juan Antonio López Morales
Secretario del H. Ayuntamiento de Tepetlaoxtoc

Correo electrónico:
secretariatepetlaoxtoc@gmail.com

Lcda. Mayra Azucena Ortega Espinosa
Oficial Calificador del H. Ayuntamiento de Tepetlaoxtoc.

Correo electrónico:
maortegae@derecho.unam.mx

C.C.P ARCHIVO

Palacio Municipal s/n Colonia Centro, Tepetlaoxtoc de Hidalgo, Tepetlaoxtoc, Edo. de México C.P. 56070
Tel.: (595) 923-0010 y (595) 923-0044

Anexo 2 Codificación de la base de datos

```
-- MySQL Administrator dump 1.4
--
-- ------------------------------------------------------
-- Server version 6.0.10-alpha-community


/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8 */;

/*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;
/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
FOREIGN_KEY_CHECKS=0 */;
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='NO_AUTO_VALUE_ON_ZERO'
*/;


--
-- Create schema seguridad
--

CREATE DATABASE IF NOT EXISTS seguridad;
USE seguridad;

--
-- Definition of table `cita`
--

DROP TABLE IF EXISTS `cita`;
CREATE TABLE `cita` (
  `IdCita` int(10) unsigned NOT NULL AUTO_INCREMENT,
  `Persona` varchar(35) NOT NULL,
  `Fecha` varchar(10) NOT NULL,
  `Estatus` varchar(40) NOT NULL,
  `Hora` varchar(10) NOT NULL,
  PRIMARY KEY (`IdCita`)
) ENGINE=InnoDB AUTO_INCREMENT=4 DEFAULT CHARSET=latin1;

--
-- Dumping data for table `cita`
--

/*!40000 ALTER TABLE `cita` DISABLE KEYS */;
/*!40000 ALTER TABLE `cita` ENABLE KEYS */;


--
-- Definition of table `cm`
--
```

```
DROP TABLE IF EXISTS `cm`;
CREATE TABLE `cm` (
  `IdOM` varchar(20) NOT NULL,
  `IdPersona1` int(10) unsigned NOT NULL,
  `IdPersona2` int(10) unsigned NOT NULL,
`IdPersona3` int(10) unsigned NOT NULL,
  `IdPersona4` int(10) unsigned NOT NULL,
  `IdPersona5` int(10) unsigned NOT NULL,
  `IdPersona6` int(10) unsigned NOT NULL,
  `IdPersona7` int(10) unsigned NOT NULL,
  `IdPersona8` int(10) unsigned NOT NULL,
  `IdPersona9` int(10) unsigned NOT NULL,
  `IdPersona10` int(10) unsigned NOT NULL,
  `TActa` varchar(50) NOT NULL,
  `FSuceso` varchar(20) NOT NULL,
  `HSuceso` varchar(20) NOT NULL,
  PRIMARY KEY (`IdOM`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table `cm`
--

/*!40000 ALTER TABLE `cm` DISABLE KEYS */;
/*!40000 ALTER TABLE `cm` ENABLE KEYS */;


--
-- Definition of table `oc`
--

DROP TABLE IF EXISTS `oc`;
CREATE TABLE `oc` (
  `IdOC` varchar(15) NOT NULL,
  `IdPersona` varchar(10) NOT NULL,
  `TIdentificacion` varchar(50) NOT NULL,
  `CIdentificacion` varchar(50) NOT NULL,
  `Religion` varchar(50) NOT NULL,
  `Responsable` varchar(200) NOT NULL,
  `Parentesco` varchar(100) NOT NULL,
  `TLocal` varchar(10) NOT NULL,
  `TCelular` varchar(13) NOT NULL,
`SLlamada` varchar(40) NOT NULL,
  `LSuceso` varchar(255) NOT NULL,
  `Calle` varchar(50) NOT NULL,
  `Municipio` varchar(50) NOT NULL,
  `Colonia` varchar(50) NOT NULL,
  `Estado` varchar(50) NOT NULL,
  `NPA` varchar(200) NOT NULL,
`Ob1` varchar(255) NOT NULL,
  `TSancion` varchar(50) NOT NULL,
  `Hora` varchar(10) NOT NULL,
  `Monto` varchar(50) NOT NULL,
```

```
  `Ob2` varchar(255) NOT NULL,
  `Fecha` varchar(10) NOT NULL,
  `HDetencion` varchar(8) NOT NULL,
  `HLiberacion` varchar(8) NOT NULL,
  `Motivo` varchar(150) NOT NULL,
  `TipoFalta` varchar(100) NOT NULL,
  `Fraccion` varchar(250) NOT NULL,
  `Articulo` varchar(999) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table `oc`
--

/*!40000 ALTER TABLE `oc` DISABLE KEYS */;
/*!40000 ALTER TABLE `oc` ENABLE KEYS */;


--
-- Definition of table `personas`
--

DROP TABLE IF EXISTS `personas`;
CREATE TABLE `personas` (
  `IdPersona` int(10) unsigned NOT NULL,
  `Nombre` varchar(45) NOT NULL,
  `APaterno` varchar(45) NOT NULL,
  `AMaterno` varchar(45) NOT NULL,
  `Edad` varchar(2) NOT NULL,
  `Calle` varchar(50) NOT NULL,
  `Colonia` varchar(50) NOT NULL,
  `Municipio` varchar(50) NOT NULL,
  `Estado` varchar(50) NOT NULL,
  `Nacionalidad` varchar(50) NOT NULL,
  `Sexo` char(1) NOT NULL,
  `GE` varchar(50) NOT NULL,
  `Ocupacion` varchar(80) NOT NULL,
  `EC` varchar(50) NOT NULL,
  PRIMARY KEY (`IdPersona`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table `personas`
--

/*!40000 ALTER TABLE `personas` DISABLE KEYS */;
INSERT INTO `personas`
(`IdPersona`,`Nombre`,`APaterno`,`AMaterno`,`Edad`,`Calle`,`Colonia`,`Mun
icipio`,`Estado`,`Nacionalidad`,`Sexo`,`GE`,`Ocupacion`,`EC`) VALUES
 (0,'','','','','','','','','','','','','');
/*!40000 ALTER TABLE `personas` ENABLE KEYS */;
```

```
/*!40101 SET SQL_MODE=@OLD_SQL_MODE */;
/*!40014 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS */;
/*!40014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS */;
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
```

Anexo 3 Diseño de Pantallas

Pantalla de Usuario y contraseña. (A3 65)

A3 65 Pantalla de Usuario y contraseña.

Pantalla de Menú inicial. (A3 66)

A3 66 Pantalla de Menú inicial.

Pantalla de Oficialía Conciliadora. (A3 67)

A3 67 Pantalla de Oficialía Conciliadora.

Pantalla de Oficialía Mediadora. (A3 68)

A3 68 Pantalla de Oficialía Mediadora.

Pantalla de Citas. (A3 69)

A3 69 Pantalla de Citas.

Pantalla de Reporte de Citas. (A3 70)

A3 70 Pantalla de Reporte de Citas.

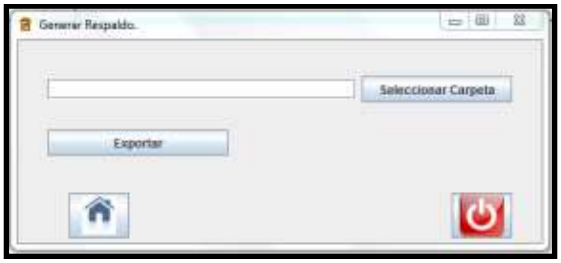
Pantalla de Reporte Mediadora. (A3 71)

A3 71 Pantalla de Reporte Mediadora.

Pantalla de Reporte Calificadora. (A3 72)

A3 72 Pantalla de Reporte Calificadora.

Pantalla Generar Respaldo. (A3 73)



A3 73 Pantalla Generar Respaldo.

Pantalla para Restaurar Respaldo. (A3 74)

A3 74 Pantalla para Restaurar Respaldo.

Pantalla para Exportar en Word. (A3 75)

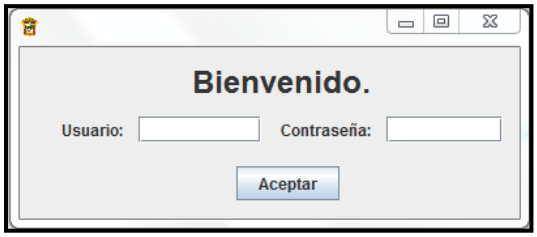
A3 75 Pantalla para Exportar en Word.

Pantalla para Exportar en Excel. (A3 76)

A3 76 Pantalla para Exportar en Excel.

Anexo 4 Manual del Usuario

Al ingresar al sistema mostrará la venta principal, donde debe colocar el usuario (Oficialia) y contraseña (Admin1) (A4 78), después se mostrará una ventada donde va a mostrar las citas del día. Tomando como referencia la fecha del sistema operativo, con los datos ID registro de la cita, nombre de la persona, fecha de la cita, el status y la hora. (A4 77)



A4 77 Usuario y Contraseña.

A4 78 Menú Principal.

	Botón para salir del sistema de información
	Botón para realizar la exportación a Excel de la tabla de consulta del menú principal
	Menús principal del sistema

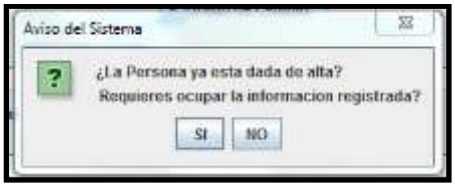
En el menú de Archivo / Oficialía Conciliadora

Para dar de alta un documento de Oficialía calificadora, se tiene las opciones de guardar, buscar el folio del documento y la actualización de un documento.
Todos los campos de esta ventana deben de ser llenados, en los campos que son caracteres deben de ser en mayúsculas. (A4 79)

A4 79 Oficialía Calificadora.

	Botón para salir del sistema de información
	Botón para guardar la información capturada en la ventana y agregarla en la base de datos.
	Botón para buscar el folio de un documento.
	Botón para actualizar el registro del documento en caso que se realice una modificación a la información.
	Botón para regresar a menú principal.

En caso de que la persona ya existe en la base de datos mandara un mensaje. (A4 80)



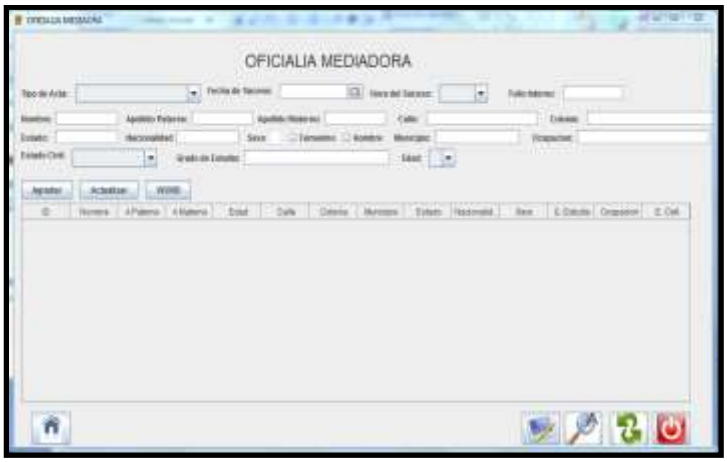
A4 80 Aviso del Sistema.

Si da clic en la opción SI, se registrar los datos de la persona.
Si da clic en a opción NO, se dará un nuevo registro para la persona
Datos que se capturan en la ventana son:

Fecha	Fecha en la que se registró el evento.
Hora de la detención	Hora en la detención de la persona responsable.
Hora de la Liberación	Hora de la liberación de la persona responsables.
Datos Personales	
Nombre, Apellido Paterno, Apellido Materno, Estado, Calle, Municipio, Tipo de Identificación, Nacionalidad, Colonia, Ocupación Edad, Religión, Estado Civil, Grado de Estudios y Sexo.	Se capturan los datos personales de le persona responsable
Datos Adicionales	
Responsable, Parentesco, Teléfono Local, Teléfono Celular, Estatus de la llamada.	Se captura los datos de un familiar o persona que se hará cargo del responsable.
Hechos	
Lugar donde fue el suceso, Calle, Municipio, Estado, Colonia	Se captura la información de donde surgió el suceso.
Persona Afectada	
Nombre, Hora, Monto, Observaciones, Sanción Impuesta, Observaciones, Motivo, Tipo de Falta, Fracción y Artículo.	Se coloca el nombre de la persona afectada, la hora, el monto monetario, agregando una observación. En esta captura se coloca la sanción, el tipo de falta y la fracción que se le impondrá a la persona responsable.

Menú Archivo / Oficialía Mediadora

Para dar de alta un documento de Oficialía Mediadora, se tiene las opciones de guardar, buscar el folio del documento y la actualización de un documento.
Todos los campos de esta ventana deben de ser llenados. (A4 81)



A4 81 Oficialía Mediadora.

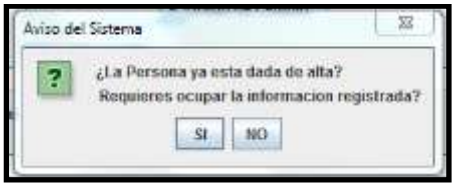
	Botón para salir del sistema de información
	Botón para guardar la información capturada en la ventana y agregarla en la base de datos.
	Botón para buscar el folio de un documento.
	Botón para actualizar el registro del documento en caso que se realice una modificación a la información.
	Botón para regresar a menú principal.
	Botón para agregar los datos de la persona en la tabla y en la base de datos.
	Botón para actualizar los datos de la persona en la tabla y en la base de datos.
	Botón para exportar la información a un archivo de Word con una plantilla correspondiente al tipo de acta que este seleccionada.

Datos que se capturan son.

Tipo de Acta, Fecha de Suceso, Hora del suceso, Folio Interno.	Se captura el tipo de acta seleccionando las opciones correspondientes, se coloca la fecha del suceso y la hora. El folio es de uso interno del área.
Nombre, Apellido Paterno, Apellido Materno, Estado, Calle, Municipio, Nacionalidad, Colonia,	Se coloca los datos personales de las personas involucradas en el documento.

Ocupación Edad, Estado Civil, Grado de Estudios y Sexo.	

Cuando de clic en el campo **Calle** saldrá un mensaje, en caso de que la persona ya existe en la base de datos. (A4 82)



A4 82 Aviso del sistema.

Si da clic en la opción SI, se registrar los datos de la persona.
Si da clic en a opción NO, se dará un nuevo registro para la persona.

Cuando de clic en el botón de **Word** salda la venta, para seleccionar la ruta en donde se guardará la información dependiendo del tipo de acta. (A4 83)

A4 83 Ventana de Word.

Menú Archivo / Citas

Módulo para dar de alta las citas. (A4 84)

A4 84 Citas.

	Botón para salir del sistema de información
	Botón para guardar la información capturada en la ventana y agregarla en la base de datos.
	Botón para buscar el folio de un documento.
	Botón para actualizar el registro del documento en caso de que se realice una modificación a la información.
	Botón para regresar a menú principal.

Datos que se capturan son.

ID Cita	Es la clave de la Cita
Nombre	Nombre de la persona que tendrá la Cita.
Fecha de la Cita	Se coloca la Fecha de la Cita.
Estatus	Se coloca el Estatus de la Cita.
Hora	Se coloca la hora en la cual se presentará en la Cita.

Menú Herramientas / Reportes / Reportes Citas

Módulo para generar los reportes correspondientes a las citas. (A4 85)

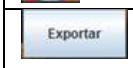
A4 85 Reporte de Citas.

	Botón para salir del sistema de información
	Botón para buscar los documentos por rango de fechas, por nombre o es estatus de la cita.
	Botón para realizar la exportación a Excel de la tabla de consulta.
	Botón para regresar a menú principal.

Menú Herramientas / Reportes / Reportes Mediadora

Módulo para genera los reportes correspondientes de Mediadora. (A4 86)

A4 86 Reporte Mediadora.

	Botón para salir del sistema de información
	Botón para realizar la exportación a Excel de la consulta que se esté realizando por folio, por nombre de la persona o por rengo de fechas.
	Botón para regresar a menú principal.

Menú Herramientas / Reportes / Reportes Calificadora

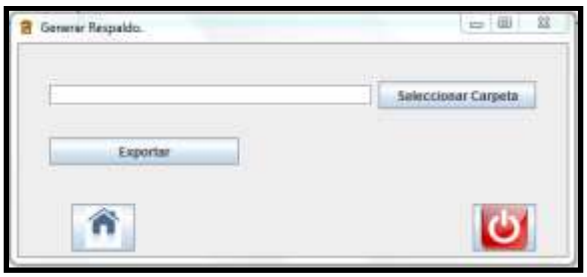
Módulo para realizar los reportes correspondientes de Calificadora. (A4 87)

A4 87 Reporte Calificadora.

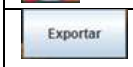
	Botón para salir del sistema de información
	Botón para realizar la exportación a Excel de la consulta que se esté realizando por folio, por nombre de la persona o por rengo de fechas.
	Botón para regresar a menú principal.

Generar Respaldo

Módulo para realizar los respaldos del sistema. (A4 88)



A4 88 Generar Respaldo.

	Botón para salir del sistema de información
	Botón para realizar la exportación a Excel de la consulta que se esté realizando por folio, por nombre de la persona o por rengo de fechas.
	Botón para regresar a menú principal.
	Seleccionar la ruta en donde se va a guardar el respaldo.

Restaurar Respaldo

Módulo para realizar la restauración de respaldo en el sistema. (A4 89)

A4 89 Restauración de Respaldo.

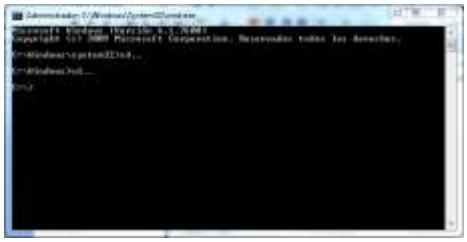
	Botón para salir del sistema de información
	Botón para realizar la exportación a Excel de la consulta que se esté realizando por folio, por nombre de la persona o por rengo de fechas.
	Botón para regresar a menú principal.
	Seleccionar el archivo de la base de datos a restaurar.

Realizar respaldo manual

Tiene que ejecutar cmd como administrador, ir al menú de inicio de Windows a la opción de búsqueda y colocar cmd, dar clic derecho ejecutar como administrador. (A4 90)

A4 90 Buscar cmd.

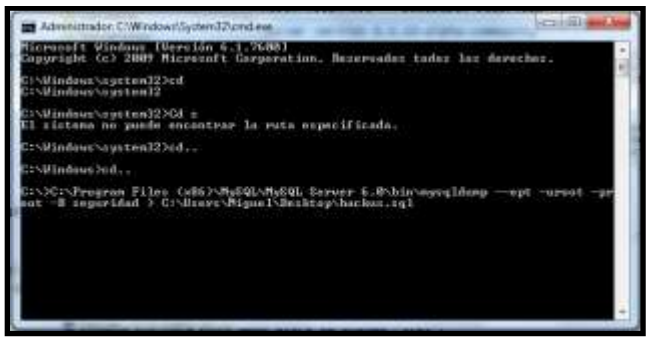
Se dirige a la unidad C con el comando cd.. c: (A3 91)



A4 91 Ventana de cmd.

Guardar la base de datos, coloque la ruta de la carpeta bin de MySQL que está en C:\Program Files (x86)\MySQL\MySQL Server 6.0\bin\ y el comando mysqldump --opt -uroot -proot -B Nombre de la base de datos, el signo >Ruta donde se guarda la base de datos\backus.sql y enter. (A4 92)

Por ejemplo.

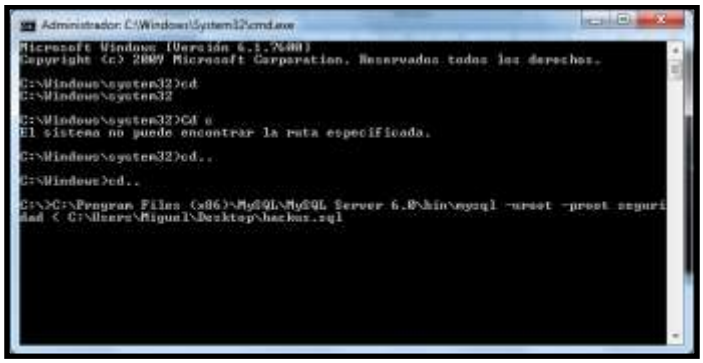


A4 92 Guardar base de datos.

Restaurar respaldo

Coloque la ruta de la carpeta bin de MySQL que está en C:\Program Files (x86)\MySQL\MySQL Server 6.0\bin\ y el comando mysql -uroot -proot Nombre de la base de datos, el signo <Ruta donde se tomara la base de datos\backus.sql y enter. (A4 93)

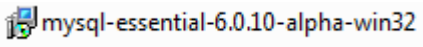
Por ejemplo.

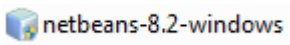


A4 93 Restaurar base de datos.

Anexo 5 Manual de Instalación y Modificación

Programa para instalar.

A. 

B. Para editar el proyecto en Java se instala el programa 

.

C. Instalación de CODOMCYC

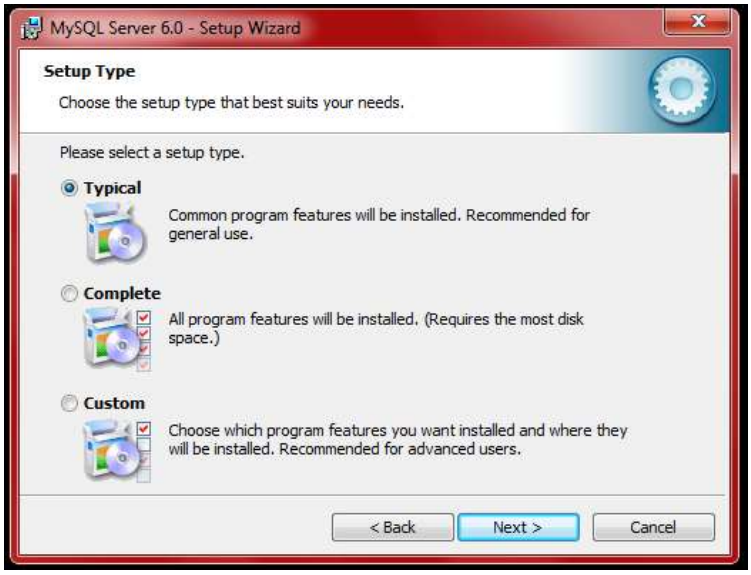
Sistema Operativo a instalar. (A5 94)

A5 94 Requerimientos del Sistema Operativo.

A) Instalación del programa mysql-essential-6.0.10-alpha-win32.
De clic en el botón Next. (A5 95)

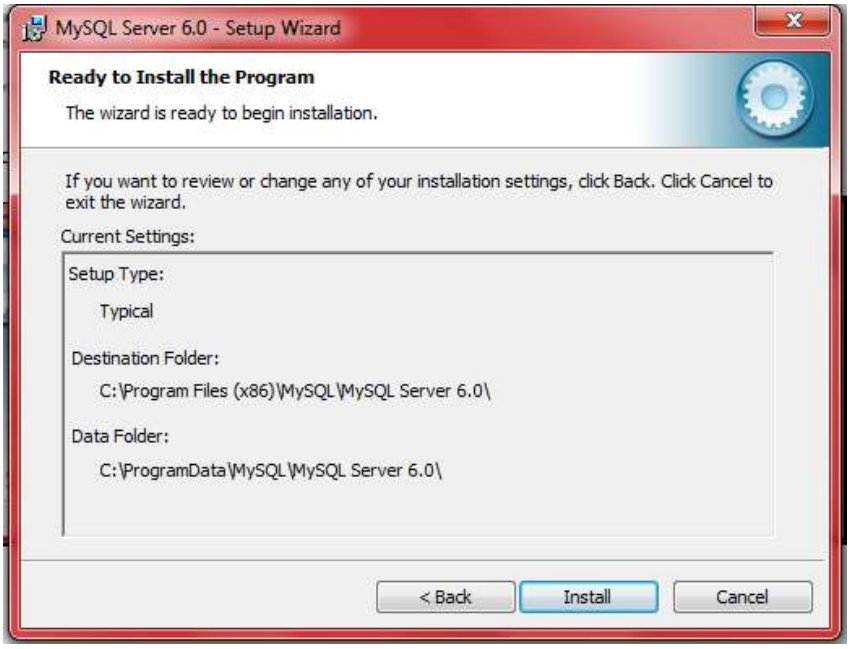
A5 95 Paso 1.

Active la opción de Typical y de clic en Next. (A5 96)



A5 96 Paso 2.

Seleccione instalar. (A5 97)



A5 97 Paso 3.

Espera a que termine de cargar la instalación. (A5 98)

A5 98 Paso 4.

De clic en los botones de Next. (A5 99, A5 100)

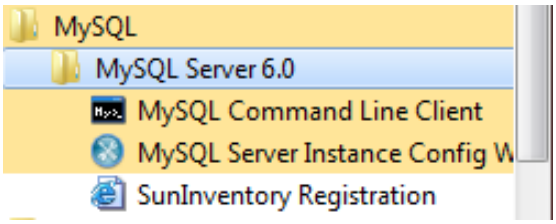
A5 99 Paso 5.

A5 100 Paso 6.

De clic en las opciones Finish, le abrirá una página de internet, cierre la página de internet. (A5 101)

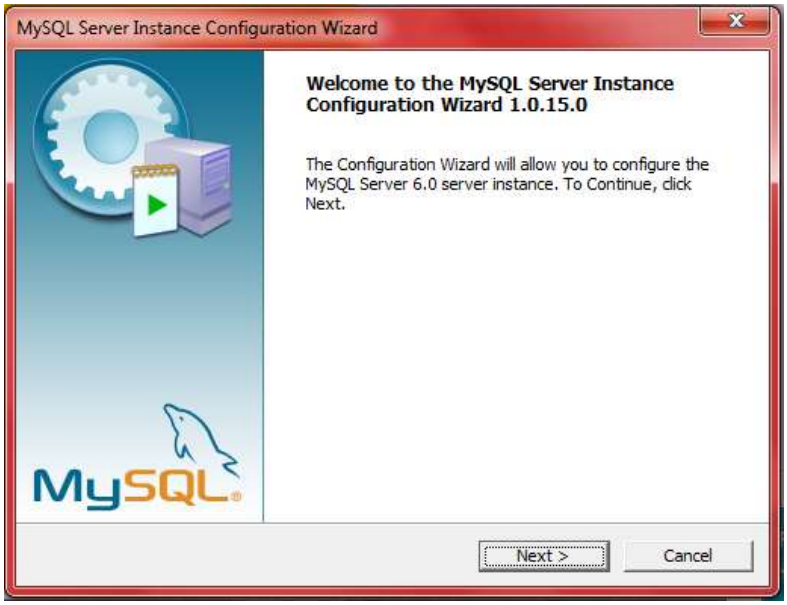
A5 101 Paso 7.

Ir al menú al menú de inicio de Windows / Todos los programas / MySQL dar clic en MySQL Server Instance Config Wizard (A5 102)



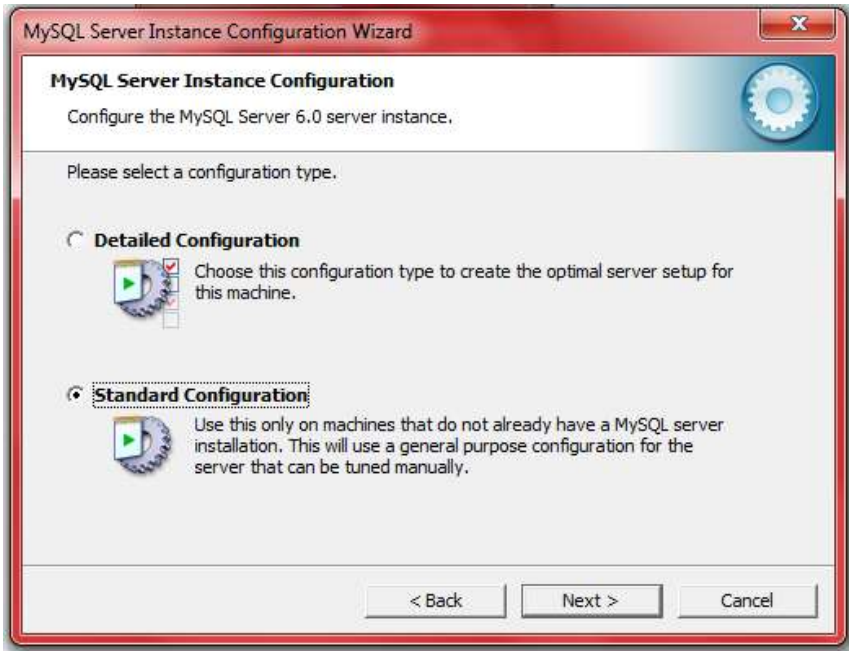
A5 102 MySQL Server Instance Config Wizard.

Le abrirá el asistente en donde tiene que dar clic en Next. (A5 103)



A5 103 Paso 8.

De clic en la opción de Standard Configuration. (A5 104)



A5 104 Paso 9.

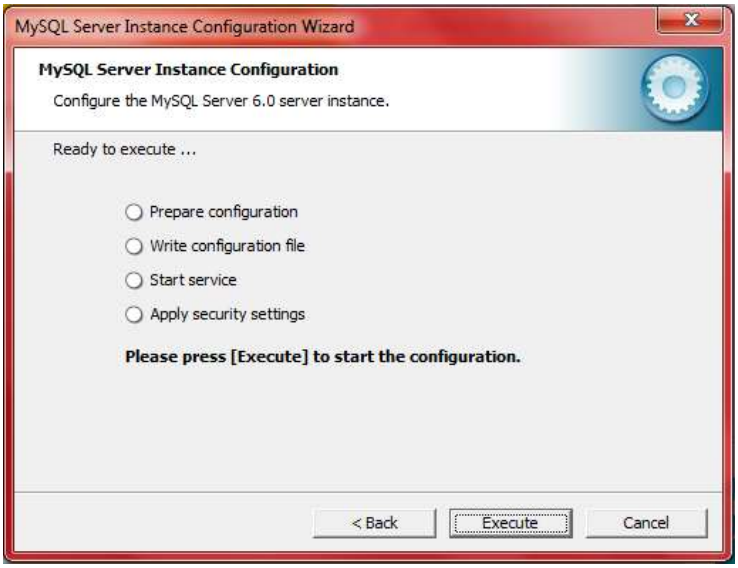
Active las dos opciones y de clic en el botón Next. (A5 105)

A5 105 Paso 10.

Para que se pueda abrir la base de datos del sistema CODOMCYC, tiene que colocar en New root password y en comfirm root, active la opción de Create An Anonymous Account, de clic en el botón de Next. (A5 106)

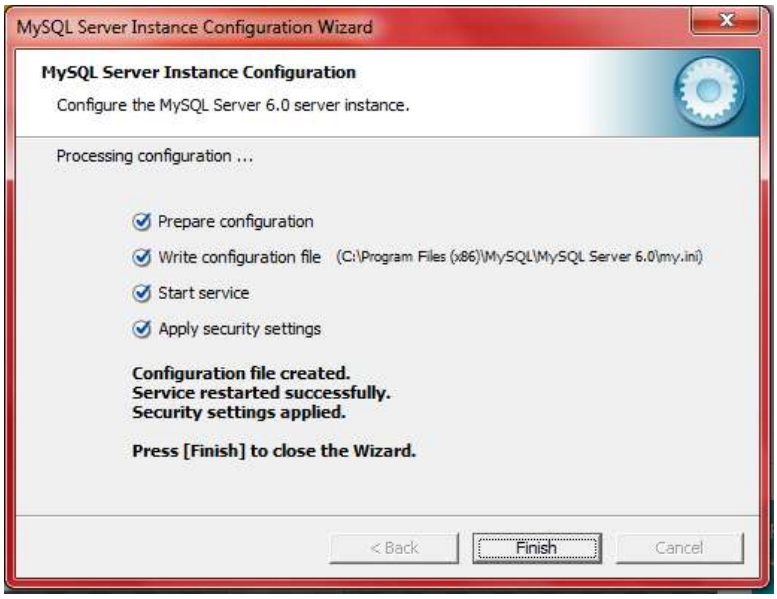
A5 106 Paso 11.

Se realizar la configuración. (A5 107)



A5 107 Paso 12

Cuando termine el sistema le tiene que mandar todas las opciones en paloma, en caso contrario que tenga un tache, tiene que repetir el proceso. De clic en Finish (A5 108)



A5 108 Paso 13.

Ejecutar el programa MySQLCommand Line Client y colocar la contraseña root, esto para identificar que la instalación fue correcta, en caso contrario repetir el proceso. (A5 109)

A5 109 Paso 14.

Se ingresa la contraseña root. (A5 110)

A5 110 Paso 15.

Si manda la información que muestra en la ventana, la configuración esta correcta, en caso contrario tiene que repetir los pasos anteriores. (A5 111)

A5 111 Paso 16.

B) Para la edición del proyecto tiene que instalar el programa de Netbeans.

En caso de que le marque el mensaje siguiente al momento de instalar Netbeans tiene que instalar JDK. (A5 112)

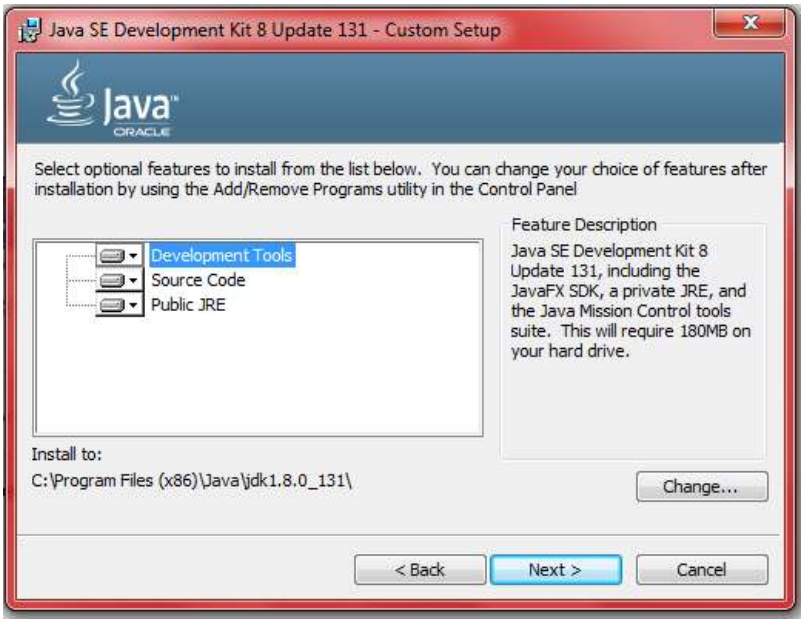
A5 112 Mensaje al instalar Netbeans.

Ejecute el programa jdk-8u131-windows-i586 , esta aplicación la puede descargar desde http://www.oracle.com/technetwork/es/java/javase/downloads/index.html.

De clic en Next. (A5 113)

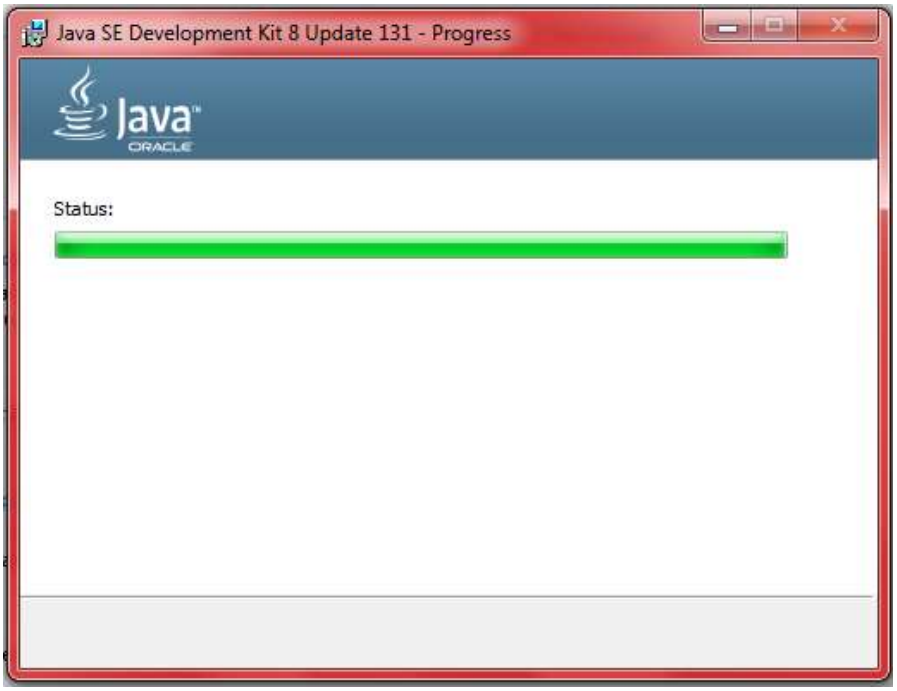
A5 113 Paso 17.

De clic en Next. (A5 114)



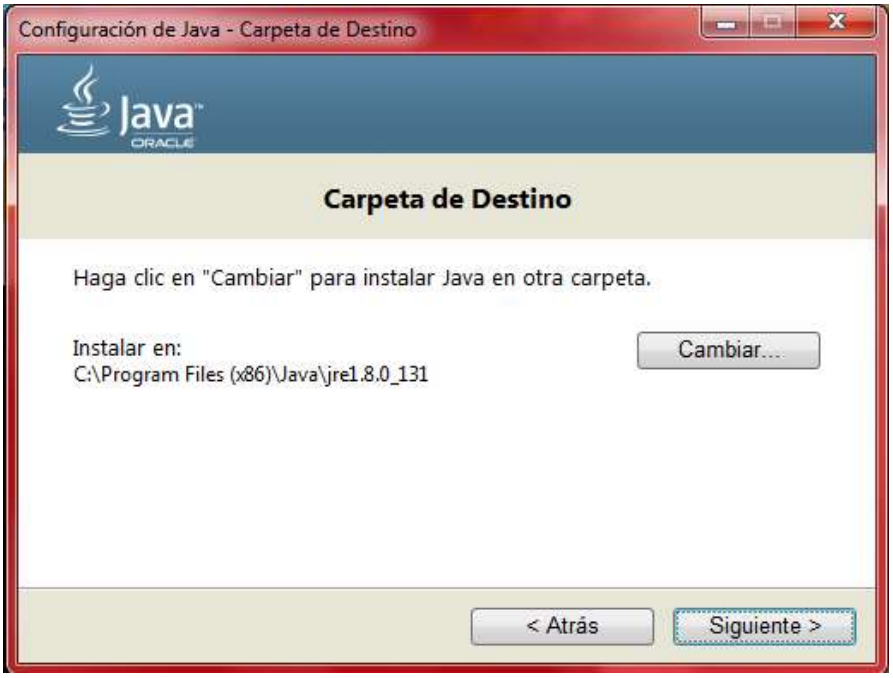
A5 114 Paso 18.

Espera a que termine la instalación. (A5 115)



A5 115 Paso 19.

Cuando termine va a dar clic en el botón de Siguiente. (A5 116)



A5 116 Paso 20.

Espere a que termine la instalación. (A5 117)

A5 117 Paso 21.

De clic en close, para cerrar la instalación. (A5 118)

A5 118 Paso 22.

Instalar Netbeans

Cuando se ejecuta el programa Netbeans manda la siguiente ventana para cargar la aplicación. (A5 119)

A5 119 Cargar Netbeans.

Esperar a que termine la carga. (A5 120)

A5 120 Paso 23.

De clic en Next. (A5 121)

A5 121 Paso 24.

De clic en Next. (A5 122)

A5 122 Paso 25.

De clic en Next. (A5 123)

A5 123 Paso 26.

De clic en Next. (A5 124)

A5 124 Paso 27.

De clic en Instal. (A5 125)

A5 125 Paso 28.

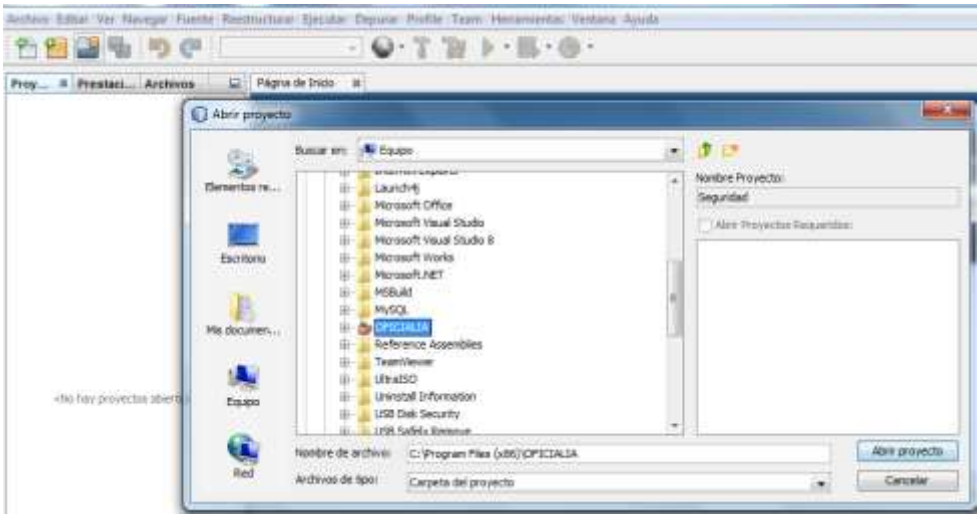
Espere a que termine la instalación. (A5 126)

A5 126 Paso 29.

De clic en Finish. (A5 127)

A5 127 Paso 30.

Ingresar al programa de Netbean y abrir la carpeta del proyecto (Seguridad), que está en lel disco que se le proporciono a la persona responsable.
Esta carpeta la copia y la pega en la carpeta de Documentos.
En Netbeans, ingrese a archivo/ Abrir proyecto. (A5 128)



A5 128 Paso 31.

Una vez editado el archivo debe de generar el archivo .jar y después el archivo de instalación (.exe), ya que el archivo exe tiene la programación actual y no se reflejaran los cambios realizados.

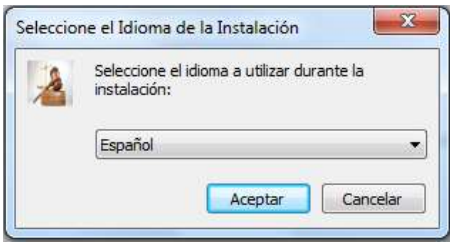
Liga para generar el archivo exe.

https://es.scribd.com/doc/67666115/Crear-un-archivo-de-instalacion-de-un-proyecto-de-NetBeans-o-Visual

C) Instalación de CODOMCYC

Una vez que se tenga el archivo exe de ejecuta y mandara el asistente de instalación.

En la primera ventana, se tiene que seleccionar el idioma de la instalación. (A5 129)

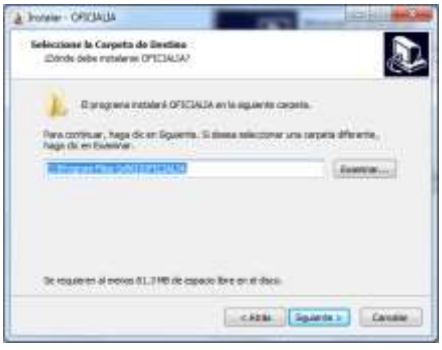


A5 129 Seleccionar Idioma.

En la segunda ventana de clic en el botón de siguiente. (A5 130)

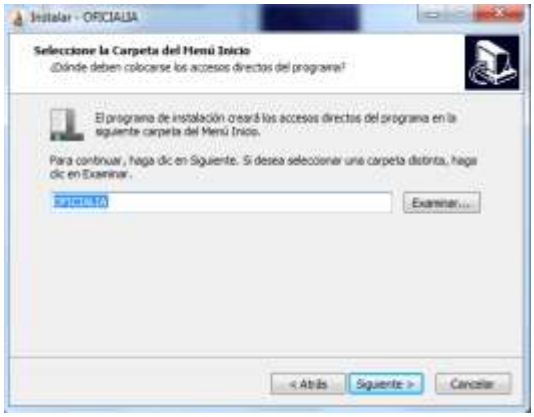
A5 130 Ventana de instalación bienvenido.

Para continuar, haga clic en Siguiente. Si desea seleccionar una carpeta diferente haga clic en Examinar. (A5 131)



A5 131 Seleccionar carpeta de Destino.

Para continuar, haga clic en Siguiente. Si desea seleccionar una carpeta diferente haga clic en Examinar. (A5 132)

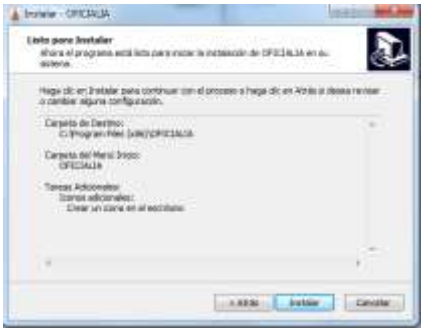


A5 132 Seleccionar carpeta del menú de inicio.

Active la opción de crear un icono en el escritorio y de clic en siguiente. (A5 133)

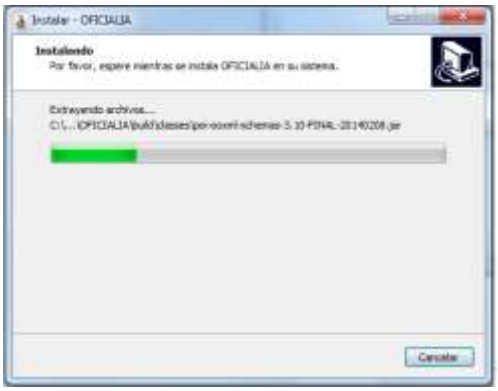
A5 133 Tareas adicionales.

Iniciar el proceso de instalación. (A5 134)



A5 134 Instalar.

Espere a que termine la instalación. (A5 135)



A5 135 Proceso de instalación.

Ejecutar el programa Oficialía. (A5 136)

A5 136 Termino de la instalación.