UNIVERSIDAD AUTÓNOMA DEL ESTADO DE HIDALGO

ESCUELA SUPERIOR DE TLAHUELILPAN

AUTOMATIZACIÓN DE PROCESOS ADMINISTRATIVOS DE LA CLÍNICA Serio

T E S I S

QUE PARA OBTENER EL TÍTULO DE

LICENCIADO EN SISTEMAS COMPUTACIONALES

P R E S E N T A :

ALAMILLA NÁJERA IMELDA.

JUÁREZ MOYA JOAN STEFANY.

DIRECTORES DE LA TESIS:

LIC. GUILLERMO MERA CALLEJAS.

DR. ALEJANDRO FUENTES PENNA.

TLAHUELILPAN DE OCAMPO, HGO. DICIEMBRE 2013

Dedicatorias

A mi Papá

Gracias papá por todo el esfuerzo que hiciste para que yo pudiera terminar mi carrera, por tu amor y comprensión te amo.

A mi Mamá

Gracias Mamá por apoyarme, por las noches de desvelo en las que me acompañaste, pero principalmente por motivarme a salir adelante.

A mi Hermano

Porque eres una de mis principales motivaciones para salir a delante, doy gracias a Dios y a nuestros Padres por haberme dado a un hermano como tú, espero que algún día al leer este trabajo te sientas orgulloso de mi.

A mi Amiga

Gracias Amiga por estar siempre conmigo,este trabajo es fruto de nuestro esfuerzo, y me alegra compartirlo contigo.

A mis Catedráticos

A mis maestros que influyeron con sus lecciones y experiencias en formarme como una persona de bien les dedico cada una de estas páginas de mi tesis.

A ti

Que me inspiraste a ser mejor cada día desde que te conocí, ahora puedo decir que esta tesis lleva mucho de ti, gracias por estar siempre a mi lado, por ser mi más grande inspiración, pero principalmente por estar en mi vida.

Agradecimientos

Primero y antes que nada, dar gracias a Dios, por estar conmigo en cada paso que doy, por fortalecer mi corazón e iluminar mi mente y por haber puesto en mi camino a aquellas personas que han sido mi soporte y compañía durante todo el periodo de estudio.

Agradecer hoy y siempre a mi familia por el esfuerzo realizado por ellos. El apoyo en mis estudios, de ser así no hubiese sido posible. A mis padres y demás familiares ya que me brindan el apoyo, la alegría y me dan la fortaleza necesaria para seguir adelante, a mis Padres, a quien le debo toda mi vida, les agradezco el cariño y su comprensión, a ustedes quienes han sabido formarme con buenos sentimientos, hábitos y valores, lo cual me ha ayudado a salir adelante buscando siempre el mejor camino.

Para mis Hermanos que siempre me han apoyado en momentos difíciles, y nunca me han dejado caer con sus palabras sabias, gracias hermanos por ser quienes son, las quiero mucho

Resumen

Un sistema de información es un conjunto de elementos que interactúan entre sí con el fin de apoyar las actividades de una empresa o negocio. El equipo computacional: el hardware necesario para que el sistema de información pueda operar. El recurso humano que interactúa con el Sistema de Información, el cual está formado por las personas que utilizan el sistema. Los datos o información fuente que son introducidos. Los programas, el Software que hará que los datos procesados arrojen los resultados esperados.

Abstract

An information system is a set of elements that interact with each other in order to support the activities of a company or business. The system equipment: the necessary hardware for the information system to operate. The human resource that interacts with the information system, which is formed by the people who use the system. The source data or information are introduced. Programs, the software that processed data will yield the expected results.

Índice de Contenidos

Lista de cuadros

Lista de figuras

1. Introducción

El presente trabajo, contiene los elementos necesarios para la realización de un sistema de información. Como se sabe un sistema de información es un conjunto de componentes que interactúan entre sí para lograr un objetivo común, Una combinación organizada de: personas, hardware, software, redes de comunicaciones, recursos de datos, políticas y procedimientos, que guarda, recupera, transforma y disemina información en una organización. Es decir recopila, manipula, almacena y crea reportes de información respecto de las actividades de negocio de una empresa, con el fin de ayudar en la administración en el manejo de las operaciones.[1] El objetivo de este trabajo es implementar el sistema que permita satisfacer las necesidades de la clínica Serio, así como ayudar en la eficiencia y calidad de servicio que ofrece.

1.1. Antecedentes

En la clínica de Servicio Integral Odontológico (Serio) no se ha implementado ninguna solución para resolver el problema referente a la automatización de procesos administrativos, el cual afectado por pérdida de expedientes de los pacientes, teniendo que volver a realizar cada uno de los expedientes. Estos procesos se empezaron realizando en hojas escritas a mano, posteriormente en documentos realizados en Word y facturas hechas en Excel, sus citas son almacenadas en una agenda.

1.2. Problema

La clínica Serio requiere de un sistema que ayude a la automatización de sus procesos en cada uno de sus servicios que ofrece.

Dentro de la clínica, se observar que la falta de un sistema provoca los siguientes problemas:

• Pérdida de información de los pacientes que asisten a la clínica.

• Mal control del historial médico y personal de los pacientes así como un mal manejo administrativo.

• Falta de integridad en lo datos, ya que por llevarse en libros, cuadernos e incluso en documento realizados en Word

• Facturas hechas en Excel

• La información es propensa a pérdidas o daños.

Al implementar el sistema dentro de la clínica obtendremos resultados favorables tales como:

• Información actualizada de los pacientes.

• Control detallado de las visitas a la clínica por cada paciente y de sus respectivos historiales médicos.

• Control de los médicos que laboran en la clínica.

• Mejor control administrativo en la cual se manejara mediante facturas.

• Seguridad en la información y datos que se utilicen para la atención de los pacientes y cada uno de sus servicios que oferta la clínica Serio.

Si se implementa el sistema de automatización en la clínica Serio ¿qué ventajas se obtendrá del sistema?

1.3. Justificación

Esta investigación tiene como fundamento la necesidad de implementar un Sistema en el que tenga la capacidad de guardar historiales médicos de los pacientes, llevar un control de citas, controlar y modificar la información de los pacientes, tener una lista de costos de cada uno de sus servicios, control de cada uno de los médicos que labora en la clínica y así poder facilitar a propietarios tener un mejor acceso a la información de cada paciente y de cada uno de sus servicios, como a sus usuarios a tener un cuadro clínico más confiable.

1.4. Objetivos

1.4.1. Objetivo general

Diseñar e implementar un sistema de base de datos como herramienta útil y práctica para llevar el control de cada uno de los servicios, que son atendidos a diario en una clínica odontológica Serio.

1.4.2. Objetivos específicos

1. Análisis y recolección de información para la realización del sistema

2. Crear información general de pacientes y empleados, costos por servicio, citas a programar, tratamientos que se realiza y facturas por medio de una base de datos, para mantener actualizados los datos.

3. Diseñar el sistema que cumpla las necesidades del cliente.

4. Implementar el sistema en la clínica Serio con la finalidad de satisfacer los requerimientos solicitados por el cliente.

1.5. Organización de la obra

En el Capittulo 2 hablaremos de los que es un sistema de información y los elementos que lo conforman, se plantearan temas aceca de base de datos y que conforma una base de datos, el lenguaje de programación que se utiliza para reañizar las interfaces del sistema.

En el Capitulo 3 se realiza una entrevista al Gerecnte de la Clinica Serio conla finalidad de reunir información precisa con la que trabajara el sistema de información.

En el Capitulo 4 se lleva acabo el prototipo del sistema de información, el diseño de la base de datos con base a los diagramas de Entidad-Relación y digrama de Intensión y Extensión.

En el Capitulo 5 se realiza el prototipo del sistema de información y el rediseño de la base de datos, conociendo cada uno de los modulos del sistema de información de la Clinica Serio.

En el capitulo 6 conocemos el funcionamiento del sistema de información una ves implantado dentro de la Clinica Serio, la base de datos a utilizar
.

2. SISTEMA DE INFORMACIÓN

Una base de datos es donde ha de mantener listas de consultas, usuarios y empleados, este tipo de información se le denomina datos. Un gestor de base de datos es un programa el cual permite introducir, almacenar datos, ordenarlos y manipularlos. Ordenarlos de manera significativa para que se pueda obtener información, debe permitir: introducir, almacenar, recuperar y trabajar con los datos. Existen dos tipos de bases de datos planas y relacionales las cuales está compuesta por diferentes elementos como: tablas o ficheros, registros y campos. Un sistema de información son elementos orientados al tratamiento, administración de datos e información, organizados para su uso posterior generados para cubrir un objetivo.

Los sistemas de Información están cambiando la forma en que operan las organizaciones actuales. A través de su uso se logran importantes mejoras, pues automatizan los procesos operativos de la empresa, proporcionan información de apoyo al proceso de toma de decisiones y, lo que es más importante, facilitan el logro de ventajas competitivas a través de su implantación en las empresas.

2.1. Introducción

Para la creación de una BD es necesario conocer los diferentes elementos que la componen, es una entidad en la cual se pueden almacenar datos de manera estructurada, con la menor redundancia posible. Diferentes programas y usuarios deben poder utilizar estos datos.

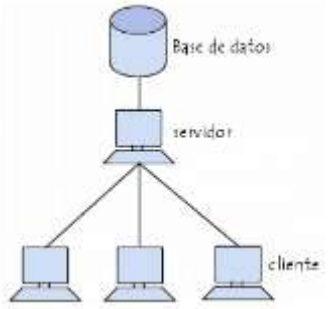


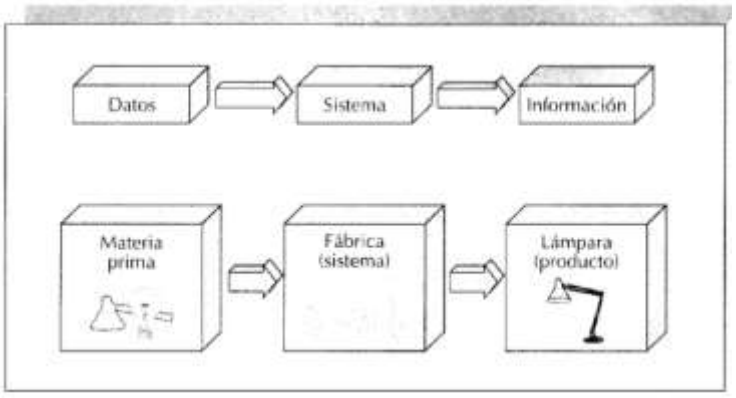
Figura 1.1 Base de datos

Una BD proporciona a los usuarios el acceso a datos, que puedan visualizar, ingresar o actualizar, la cual puede ser local, es decir que solo un usuario en un equipo, o puede ser distribuida, que la información se almacena en equipos remotos y se pueden acceder a ella a través de una red.

Una de sus ventajas es que múltiples usuarios pueden acceder a ella al mismo tiempo.

DATOS, INFORMACION Y SISTEMA.

En ocasiones los términos dato e información se utilizan como sinónimos, lo cual es un error. Datos pueden ser un número, una palabra, una imagen. En el ámbito cotidiano se utilizan en plural “Datos”, los cuales son la materia prima para la producción de información, por su parte, son datos que dentro de un contexto dado tienen un significado para alguien. Finalmente, sistema es el mecanismo por el cual se generan información.

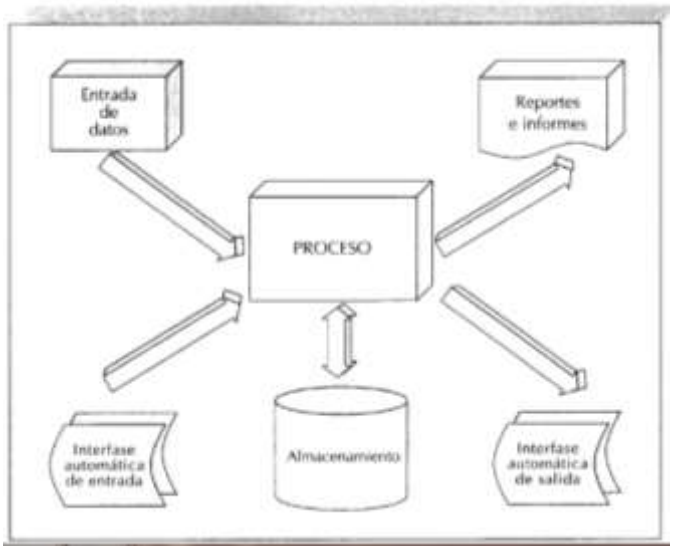


Durante los próximos años, los sistemas de información cumplirán tres objetivos dentro de las organizaciones:

1_ automatizar los procesos operativos.

2_ Proporcionar información que sirva de apoyo al proceso de toma de decisiones.

3_ Lograr ventajas competitivas a través de su implantación y uso.



2.2. Sistema de Información para la clínica *Serio*.

Conjunto de elementos orientados al tratamiento, administración de datos e información, organizados y listos para su uso, generados las necesidades de la organización. Un SI lo conforman las siguientes categorías: Datos, Personas, Actividades y recursos materiales.

Cada uno de los elementos interactúa para procesar los datos y dan lugar a la información, que se distribuye de la manera más adecuada de acuerdo a la función de sus objetivos.

Un SI se compone de diversos elementos, los cuales se desarrollarán más adelante [1].

2.3. Base de Datos.

Es un conjunto de datos almacenados dentro de una tabla las cuales están relacionadas de manera lógica y diseñada para satisfacer los requerimientos de información de manera ordenada.

2.4. Modelo de Datos.

Colección de herramientas conceptúales para describir cada uno de los datos, relacionados entre ellos.

Los modelos de datos se clasifican de diversas maneras:

- Entidad-Relación
- Orientado a objetos
- Binario
- Semántico de datos
- Infológico
- Funcional de datos

Los modelos de Entidad-Relación y orientado a objetos son los más representativos.

2.5. Modelo Relacional.

Los datos y las relaciones entre los datos, se representan mediante una colección de tablas, cada una debe contener un nombre único.

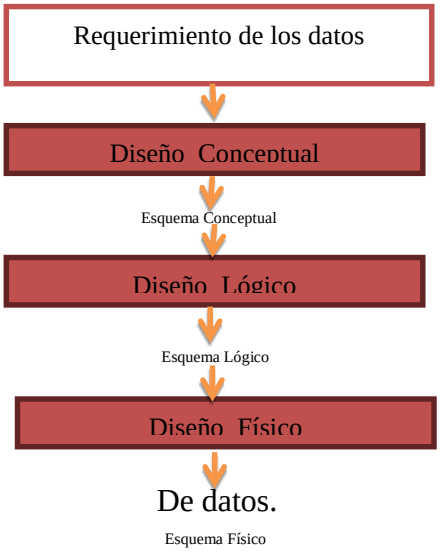
2.6. Componentes de una Base de Datos.

2.6.1. Normalización

Conjunto de reglas para desarrollar un esquema que minimice los problemas de lógica. Se puede definir como el proceso en el cual se transforman los datos complejos a un conjunto de estructuras de datos más pequeñas y fáciles de mantener. [2]

2.6.2. Fases del diseño de datos.

Representa los datos para el SI, la atención se basa en los datos y sus propiedades.



El modelo a utilizar es el de Cascada

2.6.3. Diagrama Entidad Relación.

Modelo basado en la relación cada uno de sus elementos, formando una perspectiva basada en el mundo real en el cual el SI esta desenvuelto.

2.6.3.1. Elementos básicos.

Entidades: Es un objeto que se distingue de otros objetos por medio de un conjunto específico de atributos. Las entidades se representan clases de objetos de la realidad. Se representan gráficamente por rectángulos.

Relaciones: Es una asociación entre varias entidades. Se representa mediante rombos.

Atributos: Representa las propiedades básicas de las entidades. Se representa por elipses.

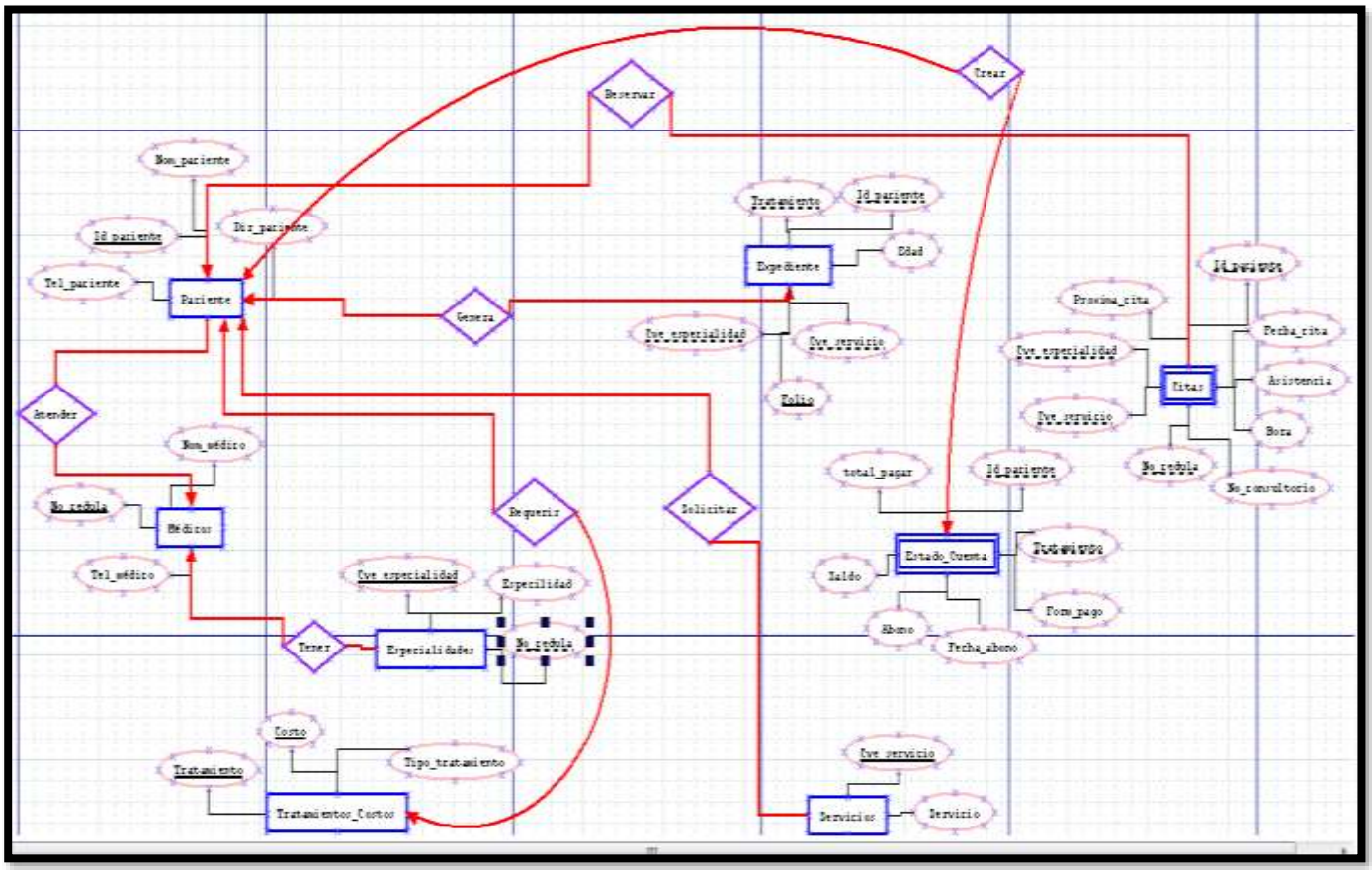
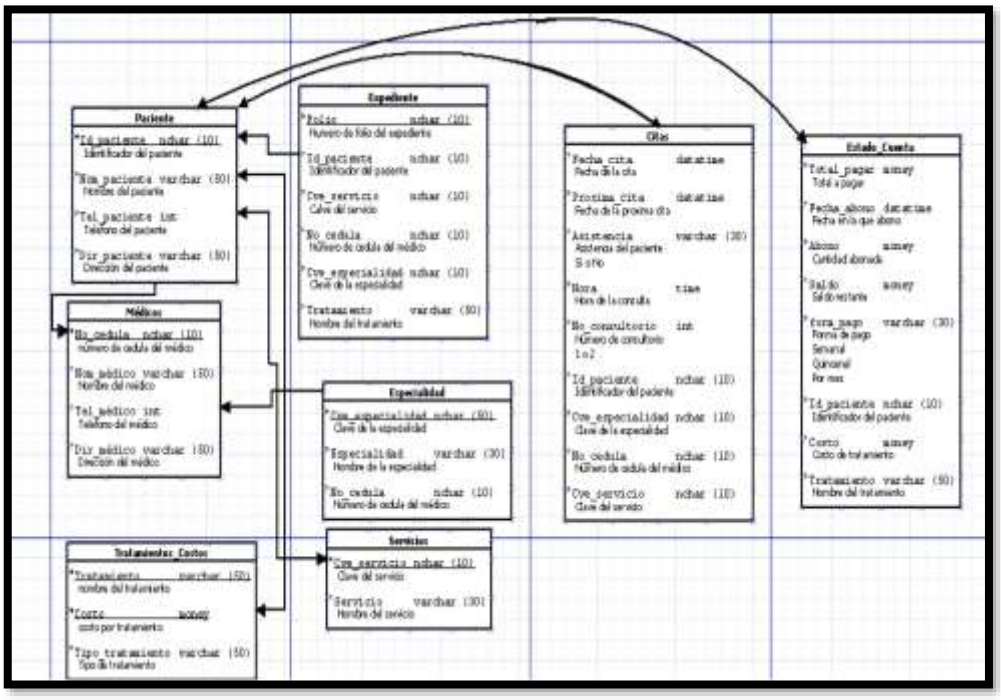


Diagrama y elementos Básicos de Entidad Relación.

2.6.3.2. Diagrama de Intención y Extensión.



2.6.3.3. Redundancia de datos.

Es la constante repetición de datos en los archivos que conforman en la base de datos, puede llevar a inconsistencia de los datos es decir que no sean correctos los datos.

2.6.3.4. Inconsistencia.

Ocurre cuando dentro de la BD existe información contradictoria o incongruente en los datos.

2.6.3.5. Diccionario de Datos.

Almacena información referente a la los datos. Es el conjunto de atributos que describen la base de datos.

2.6.3.6. Diccionario de Metadatos.

Se refiere a la información del dato contenida en el dato.

2.7. Lenguaje de consulta estructurado(SQL)

Lenguaje de acceso a base de datos relacionados permite especificar diversos tipos de operaciones. Maneja algebra y el cálculo relacional con el cual permite efectuar consultas con el fin de recuperar de forma sencilla información de interés de la base de datos.

2.8. Características generales de SQL.

El LDD (Lenguajes de Definición de Datos) de SQL proporciona comandos para la definición de esquema de relación, borrado de relaciones y modificaciones de los esquemas de relación al igual que en la información.

LMD (Lenguaje interactivo de manipulación de datos) incluye de consultas basadas en algebra relacional y calculo relacional de tuplas.

Integridad: El LDD incluye comandos para especificar las restricciones que deben cumplir los datos en la base de datos.

Definición de vistas: El LDD incluye comandos para definir las vistas, mediantes ellas se muestra la información que queramos que el usuario vea.

Control de transacciones: SQL tiene comandos que especifican el comienzo y el final de una transacción.

SQL incorporado y dinámico: Esto quiere decir que se pueden incorporar instrucciones SQL en lenguajes de programación como C++, C, Java, Cobol, Pascal y Fortran.

Autorización: El LDD incluye comandos para especificar los derechos de acceso a las relaciones y a las vistas.[2]

2.9. Programación en NetBeans.

NetBeans es un entorno de desarrollo integrado libre, hecho principalmente para el lenguaje de programación Java. Existe además un número importante de módulos para extenderlo. NetBeans IDE1 es un producto libre y gratuito sin restricciones de uso. La plataforma NetBeans permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software llamados módulos. Un módulo es un archivo Java que contiene clases de java escritas para interactuar con las APIs de NetBeans y un archivo especial (manifest file) que lo identifica como módulo.

3. Análisis del Sistema de Información (Objetivos del Sistema de Información).

Una entrevista es un acto de comunicación que se establece entre dos o más personas con el fin de obtener información, se puede distinguir de dos tipos informativas y psicológicas las cuales consta de tres partes como: Presentación (Se dan a conocer datos y el motivo), Preguntas (Atreves de ellas se obtiene toda la información requerida), Conclusión (Se puede dar por acabada la entrevista con un resumen de lo más importante), Existes distintos tipos de preguntas que se pueden realizar en una entrevista como: Preguntas cerradas (se espera una respuesta muy concreta), abiertas (respuestas extensas), hipotéticas (plantean una situaciones hipotéticas), de sondeo (obtener más información y profundizar en el tema). Las buenas entrevistas dependen del conocimiento del analista así como de la preparación del objetivo de la entrevista específica como de las preguntas a realizar las personas determinadas para la obtención de información.

3.1. Introducción

El término entrevista está ligado al verbo entrevistar acción de desarrollar una plática con una o más personas con un objetivo de abordar ciertos temas y con un fin determinado. La cual nos permite conocer a las personas e información que será útil a la investigación que se realiza, la recolección de los datos se puede realizar de distintas maneras tales como: entrevistas, encuestas, cuestionarios, observaciones, diagrama de flujo, y diccionario de

datos, se refiere al uso de una gran diversidad de técnicas y herramientas que pueden ser utilizadas por el analista para desarrollar los sistemas de información, con la finalidad de buscar información que será útil para una investigación, quienes responden pueden ser clientes o empleados los cuales serán usuarios del sistema, usuarios potenciales del sistema propuesto o aquellos que proporcionaran datos o serán afectados por la aplicación.

3.2. Entrevista: Introducción y Objetivo.

En esta sección se presenta la composición de la entrevista que será aplicada al propietario de la Clínica Serio, obteniendo de ella los requerimientos y necesidades que llevara el SI.

3.3. Entrevista al cliente.

Buenos días Dr. Gerardo Reynoso Fermoso.

1.- ¿Qué es la Clínica Serio?

Una empresa con más de 10 años de experiencia, que trabaja día con día buscando alternativas para que nuestros pacientes depositen en nosotros su confianza, su tranquilidad y el cuidado de su salud bucal.

2.- ¿con que propósito fue creada la Clínica Serio?

Contribuir a mantener la salud bucal de las personas y propiciar una cultura de salud preventiva, mejorando hábitos que deriven en una buena salud integral.

3.- ¿Cuál es la problemática que se tiene?

La clínica Serio (Servicio Integral Odontológico) requiere de un sistema que ayude a la automatización de sus procesos en cada uno de sus servicios que ofrece. Debido que sus procesos los realiza en documentos de Word como los son los expedientes de los pacientes y lista de precios en cada uno de sus servicios, sus facturas se manejan en Excel y las citas programadas en cada consulta las lleva de forma manual en una agenda. Dentro de la clínica hemos podido observar que la falta de un sistema provoca los siguientes problemas:

•Perdida de información de los pacientes que asisten a la clínica.

•Mal control del historial médico y personal de los pacientes así como un mal manejo administrativo.

•Falta de integridad en lo datos, ya que por llevarse en libros, cuadernos e incluso en documento realizados en Word y facturas hechas en Excel son propensos a pérdidas o daños.

4.- ¿Que procesos requiere que el SI realice?

Contar con un sistema el cual contenga una base de datos como herramienta útil y práctica para llevar el control de cada uno de los servicios, que son atendidos a diario en una clínica odontológica.

5.- ¿Con que módulos debe contar el sistema?

Expediente: información médica acerca del paciente

Paciente: Datos generales del paciente

Tratamientos y sus costos: Tratamientos dentro de la clínica

Médicos que laboran en la Clínica: información del Medico

Especialidades que se ofrecen: que especialidades se dan y que médicos las desempeñan

Estados de cuenta de cada uno de los pacientes: Los costos del tratamiento que recibe cada paciente y los pagos que se van abonando en cada cita

Servicios que ofrece

Citas programadas por día: agenda cotidiana de las citas que serán atendidas en la clínica

6.- ¿Con que equipo de trabajo cuenta la clínica?

CLINICA DE ESPECIALIDADES.- Atención en todas las áreas odontológicas con reconocidos especialistas y la tecnología más avanzada.

UNIDAD MOVIL MEDICO-DENTAL.- Para proporcionar un servicio integral hasta el lugar de trabajo.

LABORATORIO MECANICO DENTAL.- Que contribuye a disminuir costos, aplicando lo último en tecnología y calidad de los materiales.

UNIDAD ODONTOLOGICA PORTATIL.- Para proporcionar atención a personas con capacidades diferentes y de la tercera edad, en la comodidad de su domicilio.

7.- Especialidades que ofrece la Clínica

Odontología general.- Amalgamas, extracciones, limpiezas.

Odontogeriatría.- Personas de la tercera edad.

Odontopediatría.- Manejo de niños.

Ortodoncia.- Corrección de dientes mal alineados.

Periodoncia.- Encías sangrantes, inflamadas y movilidad dentaria.

Patología bucal.- Detección oportuna de cáncer, tumores.

Cirugía Maxilofacial.- Cirugía de terceros molares, dientes retenidos, fracturas.

Cirugía Ortognática.- Corrección de labio y paladar hendido.

Implantología.- colocación de dientes en hueso.

8.- ¿Cuántos Médicos laboran en la clínica?

4 Médicos.

9.- ¿Qué especialidades desempeñan?

Cirujano dentista, Edoteriodontologia, Odontopediatría, Ortodoncia.

10.- ¿Cuál es el personal administrativo con el que cuenta la clínica?

Cuenta con una secretaria, 4 Médicos,, 1 asistente.

11.- ¿Quiénes serían los usuarios del sistema?

Todos los que laboran en la clínica.

12.- ¿Cuáles serían las funciones de cada usuario?

Llevar de manera rápida, eficaz y sin pérdida de información los procesos realizados en la clínica.

4. Prototipo del Sistema de información (Diseño de la Base de datos e Interfaces)

El diseño debe implementar todos los requisitos contenidos en el análisis y debe cumplir requerimientos del cliente, proporcionar una idea completa. Una vez concluido el análisis del sistema y el modelado de datos, se prosigue al prototipo de la base de datos basada en una arquitectura cliente-servidor y las interfaces, con el objetivo específico del cambio de procesos manuales a procesos sistematizados.

4.1. Introducción

Una Base de Datos (BD) correctamente diseñada permite obtener acceso a la información exacta. Un diseño correcto es esencial para lograr los objetivos fijados, en este informe se decidirá qué información se necesita, a dividir la información en tablas y columnas adecuadas y a relacionar las tablas entre sí y lograr obtener un conjunto de datos y un conjunto de operaciones, que permitan satisfacer las necesidades de la organización. Así como también una interfaz correctamente diseñada que ayude a facilitar el trabajo de los usuarios que los haga sentir seguro, confiado y tranquilo, guiarlo e incluso condicionarlo de tal forma que se sienta con el control total de la situación.

4.2. Prototipo del SI de la clínica serio.

Con base en lo anterior se procederá al prototipo del SI, desarrollando la BD y el diseño de las interfaces.

4.3. Diseño de la BD.

Para el diseño de la base de datos es necesario la utilización de diversas técnicas y herramientas que ayudarán a obtener un buen análisis sobre el escenario donde se implementará la BD.

4.3.1. Diccionario de Datos y Metadatos.

El DD y el DM, son utilizados para identificar los campos que contendrán la BD y el SI.

4.3.2. Diagrama de E-R, Intención y Extensión.

Modelo basado en la relación cada uno de sus elementos, formando una perspectiva basada en el mundo real en el cual el SI esta desenvuelto.

El Diagrama de Intensión y Extensión es solo una forma más de representar un Diagrama de E-R, la cual no ayuda a saber de qué manera se comportara la BD del sistema a desarrollar.

4.3.3. Diseño del Sistema de Información.

En esta sección se presenta el desarrollo de la BD de acuerdo al análisis desarrollado anteriormente, así como el diseño de las interfaces con la finalidad de analizar el comportamiento que tendrá el SI, y realizar las modificaciones de acuerdo a las necesidades de la organización.

4.3.4. Elementos que conformaran la BD.

Diccionario de Datos en el cual se describe que campos serán utilizados dentro de la BD

Paciente

*Id_paciente	nchar (10)	Identificador del paciente
*Nom_paciente	varchar (50)	Nombre del paciente
*Tel_paciente	int	Teléfono del paciente
*Dir_paciente	varchar (50)	Dirección del paciente

Expediente

*Folio	nchar (10)	Numero de folio del expediente
*Id_paciente	nchar (10)	Identificador del paciente
*Cve_servicio	nchar (10)	Clave del servicio
*No_cedula	nchar (10)	Número de cedula del médico
*Cve_especialidad	nchar (10)	Clave de la especialidad
*Tratamiento	varchar (50)	Nombre del tratamiento

Médicos

*No_cedula	nchar (10)	número de cedula del médico
*Nom_médico	varchar (50)	Nombre del médico
*Tel_médico	int	Teléfono del médico
*Dir_médico	varchar (50)	Dirección del médico

Citas

*Fecha_cita	datatime	Fecha de la cita
*Proxima_cita	datatime	Fecha de la proxima cita
*Asistencia	varchar (30)	Asistencia del paciente Si o No
*Hora	time	Hora de la consulta
*No_consultorio	int	Número de consultorio 1 o 2
*Id_paciente	nchar (10)	Identificador del paciente
*Cve_especialidad	nchar (10)	Clave de la especialidad
*No_cedula	nchar (10)	Número de cedula del médico
*Cve_servicio	nchar (10)	Clave del servicio

Estado_Cuenta

*Total_pagar	money	Total a pagar
*Fecha_abono	datatime	Fecha en la que abono
*Abono	money	Cantidad abonada
*Saldo	money	Saldo restante
*form_pago	varchar (30)	Forma de pago Semanal Quincenal Por mes
*Id_paciente	nchar (10)	Identificador del paciente
*Costo	money	Costo de tratamiento
*Tratamiento	varchar (50)	Nombre del tratamiento

Servicios

*Cve_servicio	nchar (10)	Clave del servicio
*Servicio	varchar (30)	Nombre del servicio

Tratamientos_Costos

*Tratamiento	varchar (50)	nombre del tratamiento
*Costo	money	costo por tratamiento
*Tipo_tratamiento	varchar (50)	Tipo de tratamiento

Especialidad

*Cve_especialidad	nchar (50)	Clave de la especialidad
*Especialidad	varchar (30)	Nombre de la especialidad
*No_cedula	nchar (10)	Número de cedula del médico

Diagrama E-R En base al Diccionario de Datos se desarrolla el diagrama el cual consiste en el comportamiento que tendrá la BD, cómo las entidades estarán relacionadas entre sí y que campos tendrá

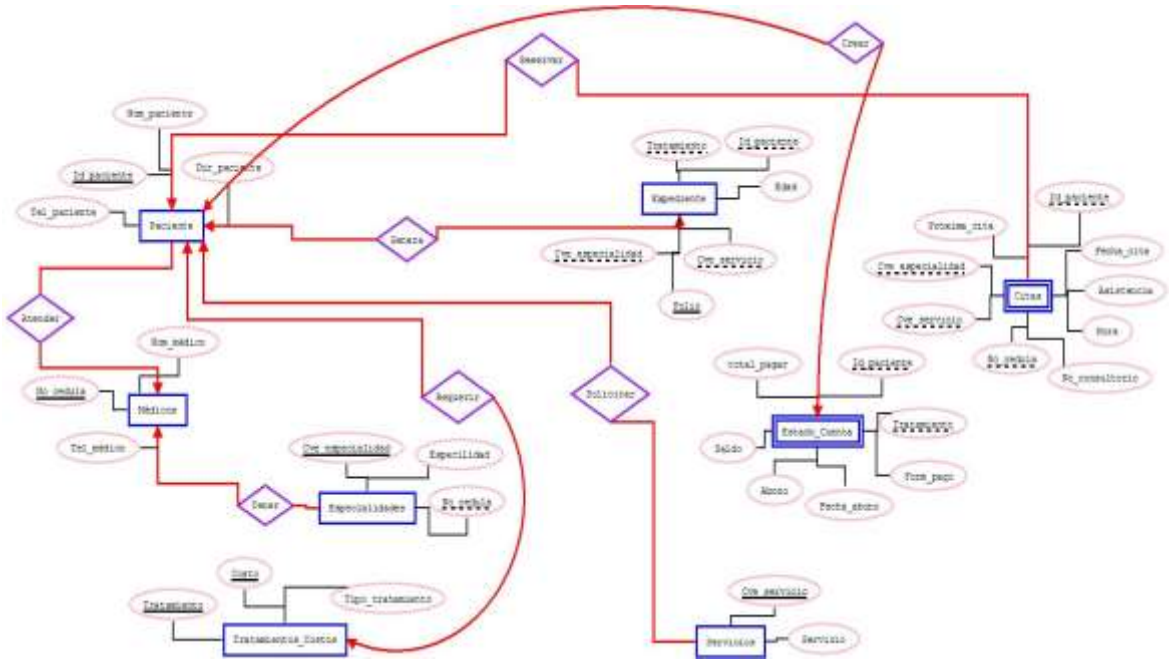


Diagrama de Intención Extensión.

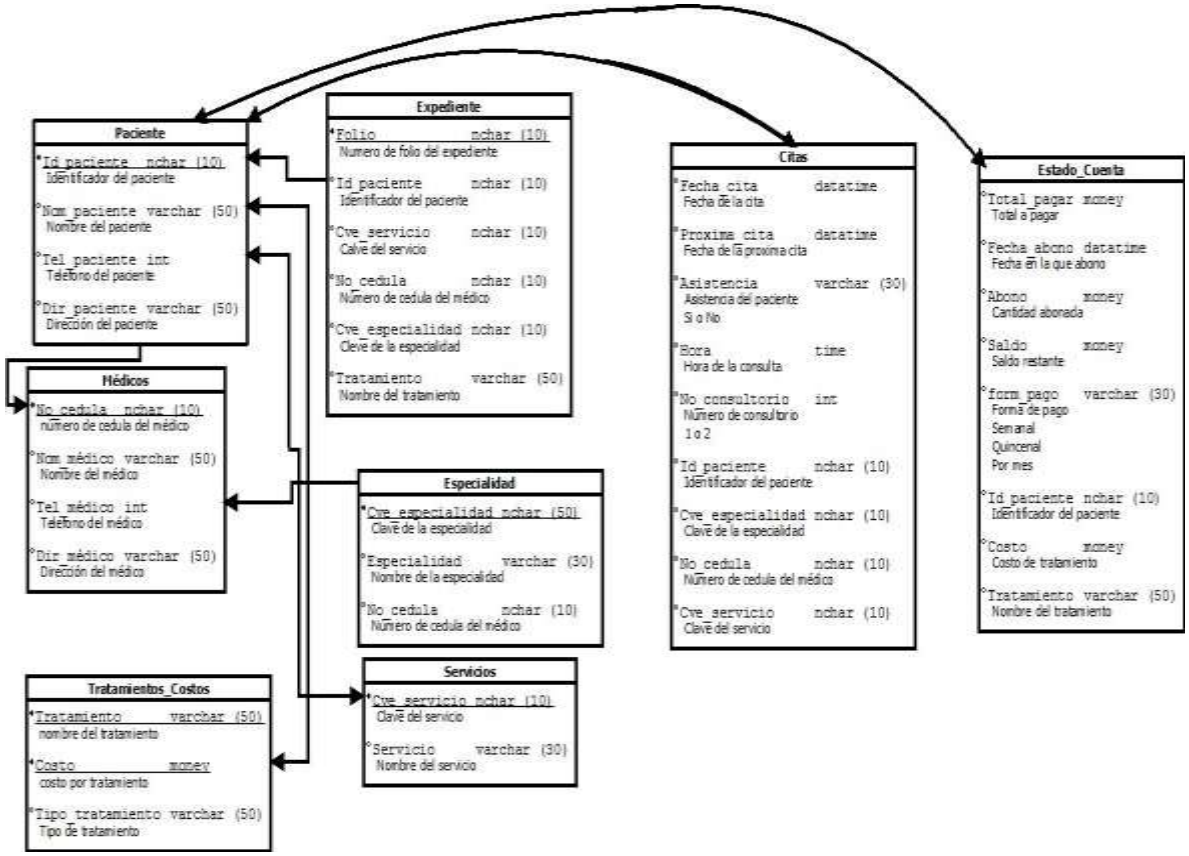
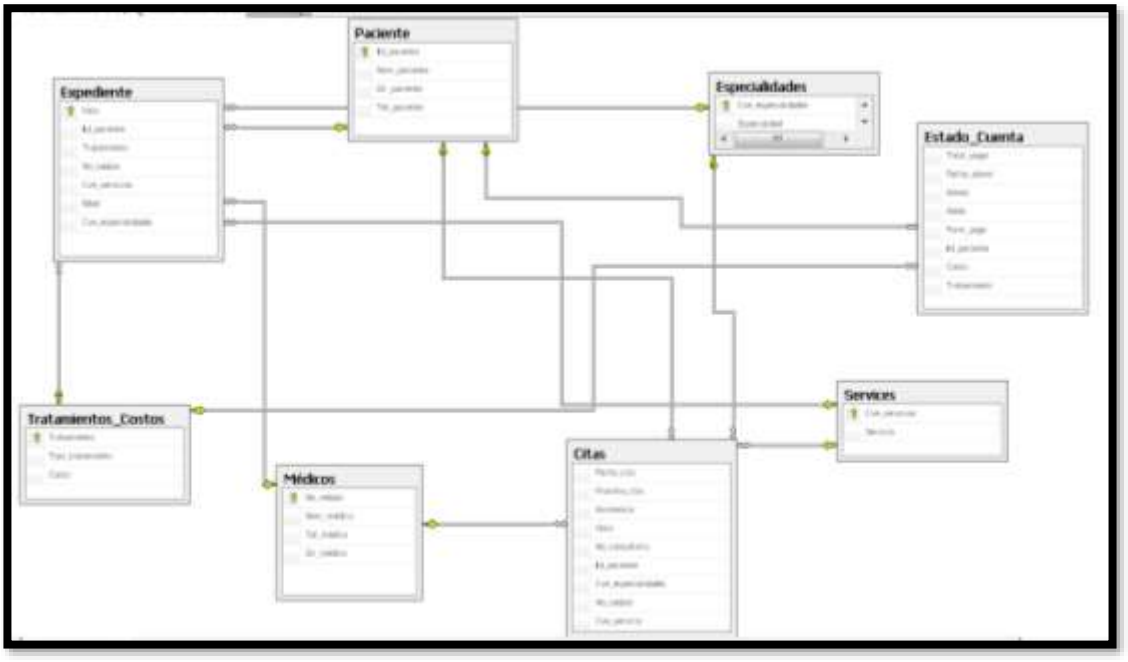


Diagrama Físico. El diagrama físico consiste una vez creada la base de datos genera la relación que existe entre las tablas dentro de la base de datos.



4.4. Diseño de las interfaces.

La primera interfaz será la de inicio donde se encontraran todos los módulos manejados en la clínica.

Cada módulo envía a cada una de las tablas de la bases de datos.

Registro de Pacientes

Médicos.

Expediente

Especialidades.

Tratamientos y Costos

Citas

Servicio.

Forma de Pago

5. Prototipo del Sistema de información (Rediseño Lógico de la Base de datos e Interfaces).

Se creó el diseño de la base de datos y las interfaces de acuerdo a cada uno de los módulos requeridos por la organización, tomando en cuenta el análisis y requerimientos obtenidos en los informes anteriores, las cuales conforman las operaciones llevadas a cabo cotidianamente en la clínica Serio.

5.1. Introducción

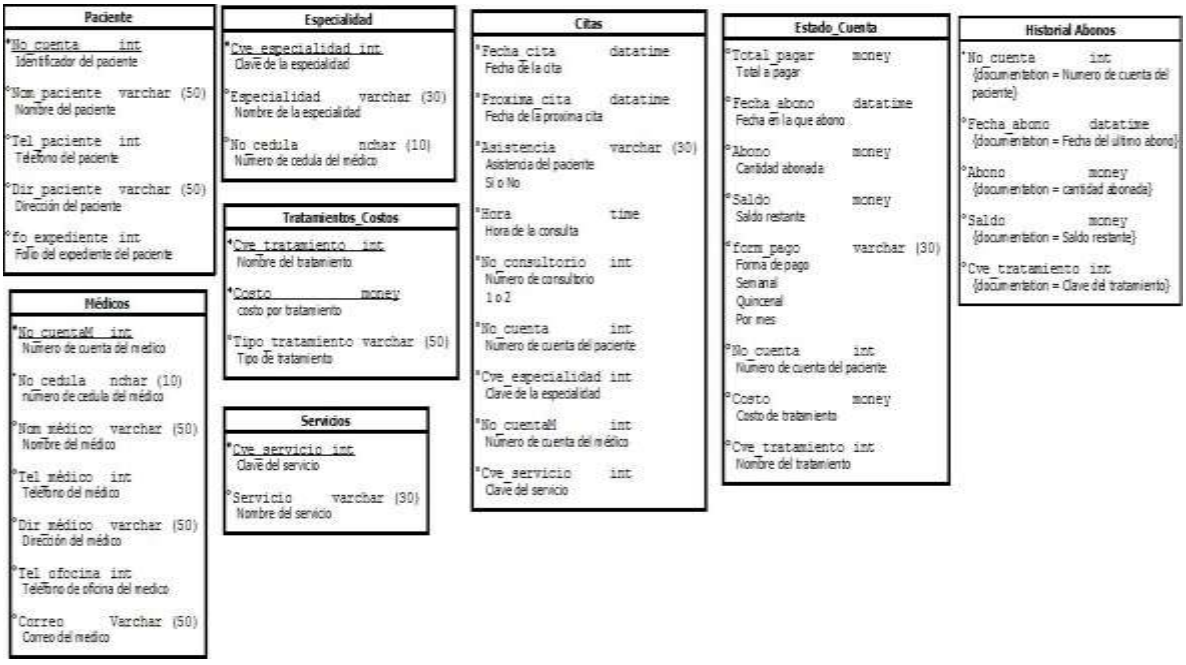
Un modelo de datos es algo conocido como contenedor de datos (algo en donde se guarda la información), así como de los métodos para almacenar y recuperar información de esos contenedores. Los modelos de datos no son cosas físicas: son abstracciones que permiten la implementación de un sistema eficiente de base de datos. Asi como un sistema de información (SI) representa todos los elementos que forman parte de la administración, el procesamiento, el transporte y la distribución de la información dentro de la empresa.

5.2. Rediseño de la BD.

Datos almacenados dentro de una tabla en los cuales existen relaciones lógicas diseñada para satisfacer los requerimientos de una empresa u organización. En una base de datos, también almacena la descripción de los datos

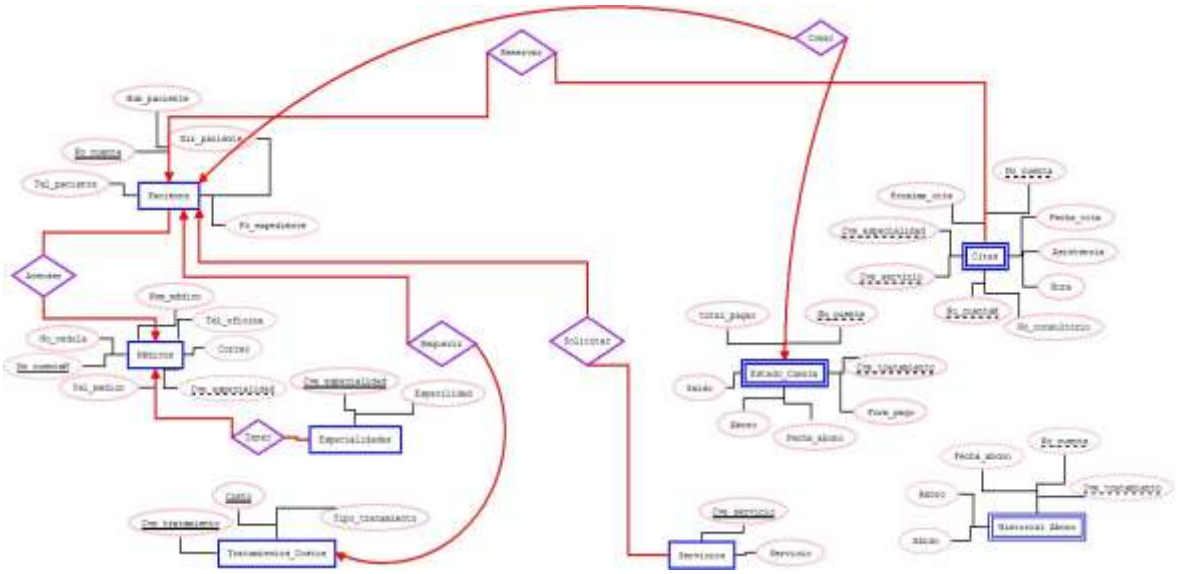
5.2.1. Diccionario de datos y Metadatos.

Contiene la descripción de cada elemento de la base de datos. Y se refiere a los datos del dato.



5.2.2. Diagrama de E-R.

Permite representar las entidades más importantes de un sistema de información así como sus interrelaciones y propiedades en cada una de ellas.

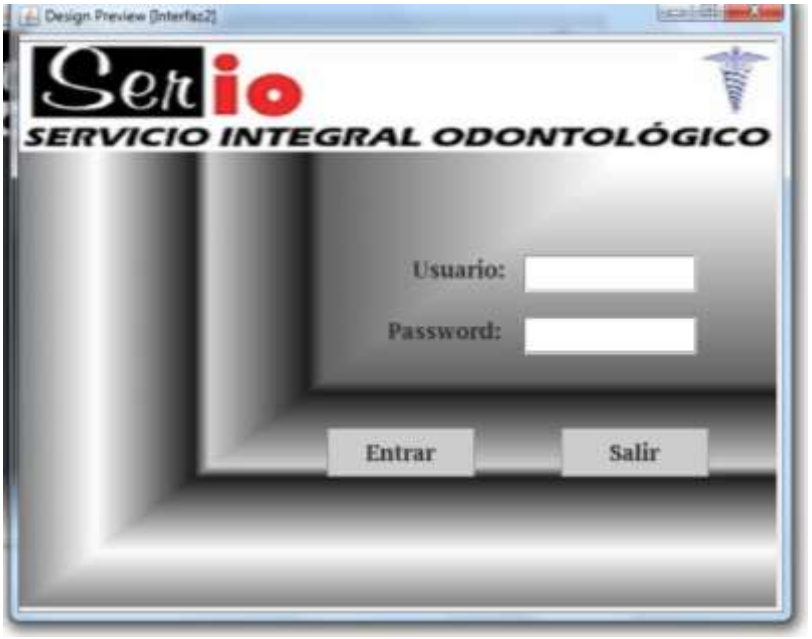


5.3. Interfaces.

La primera interfaz será la de inicio de sesión en la cual se tendrá dos tipos de usuarios.

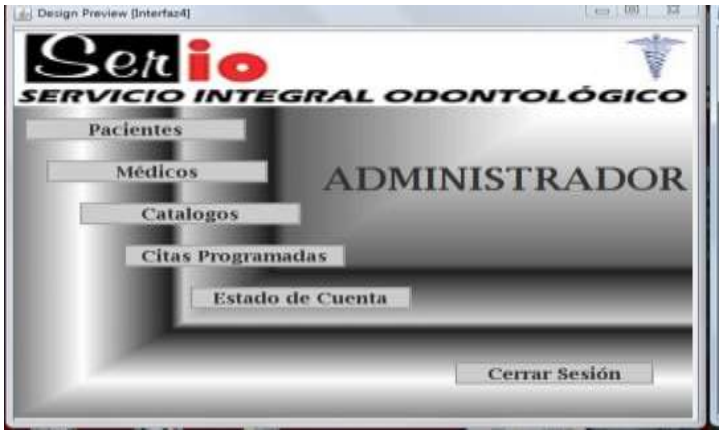
5.3.1. Inicio de sesión.

En las siguientes pantallas se muestra el inicio de sesión dependiendo del tipo de usuario que sea.



5.3.2. Módulos del Sistema

. En la siguiente interfaz se manejas los distintos módulos a los que tendrá acceso cada usuario.



5.3.3. Módulo de paciente.

En esta interfaz se manejara el registro del paciente, asignándole un número de cuenta, el paciente tendrá un estado (Activo) si es que el paciente sigue llevando tratamientos o (Inactivo) si es que termino los tratamientos, sin eliminarlo de la base de datos. Toda la información se medica se almacenara en un documento de Word el cual será almacenado dentro de la base de datos.

5.3.4. Modulo médicos.

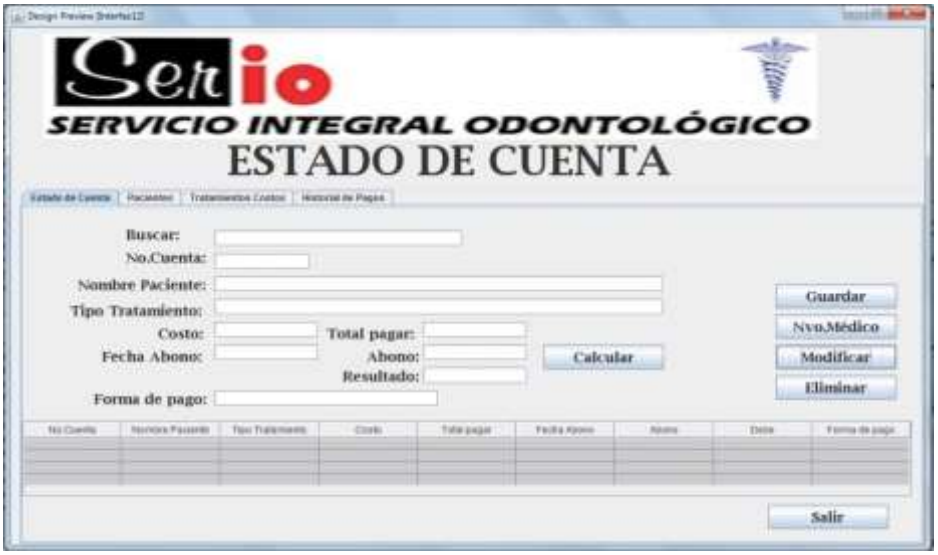
Esta interfaz maneja el registro de los médicos, asignándole un número de cuenta de tal manera que con el puedan acceder al sistema.

5.3.5. Módulo Catálogos.

Las siguientes interfaces son catálogos de servicios que la clínica presta.

5.3.6. Modulo Estado de Cuenta.

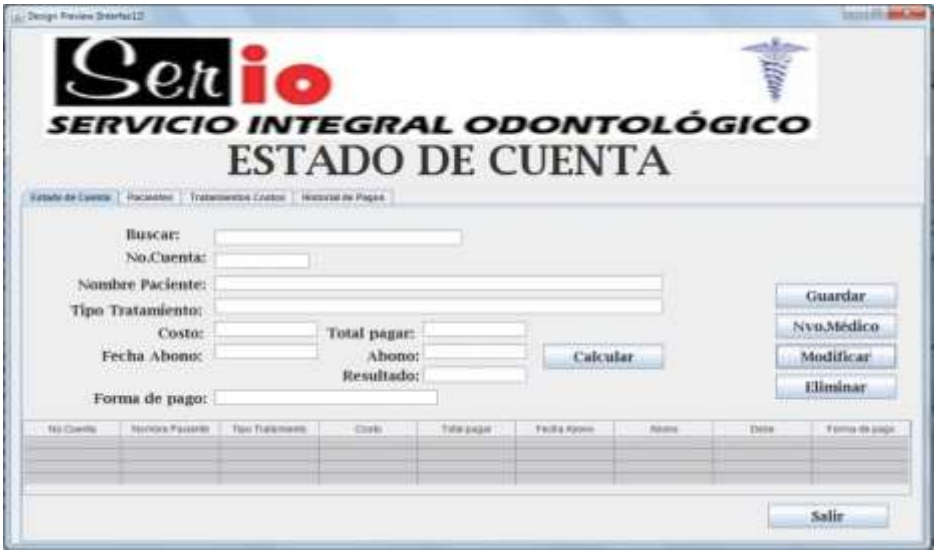
La interfaz de Estado de cuenta se mostrara que tratamientos ha tenido el paciente y el costo de cada uno de ellos, así como el total a pagar, la forma en que se pagara los tratamientos, la monto de abono que dejara en cada pago especificando la fecha y a su vez el saldo restante de la cuenta. Imprimiendo un ticket.



5.3.7. Modulo Historial

A su vez todos los pagos serán guardados, mostrados y almacenados en un historial el cual contendrá los siguientes campos.

.



5.3.8. Modulo citas Programadas.

La siguiente interfaz es la agenda que la clínica llevara cotidianamente almacenando en ella las citas durante el día de cada médico y el servicio que solita el paciente.

6. Base de datos y Sistema de Información de la Clínica Serio

Una BD es una entidad en la cual se pueden almacenar datos de manera estructurada, con la menor redundancia posible. Diferentes programas y diferentes usuarios deben poder utilizar estos datos. Por lo tanto, el concepto de base de datos generalmente está relacionado con el de red ya que se debe poder compartir esta información. De allí el término base. "Sistema de información" es el término general utilizado para la estructura global que incluye todos los mecanismos para compartir datos que se han instalado. Una base de datos proporciona a los usuarios el acceso a datos, que pueden visualizar, ingresar o actualizar, en concordancia con los derechos de acceso que se les hayan otorgado. Se convierte más útil a medida que la cantidad de datos almacenados crece.

6.1. Introducción

Como se sabe un sistema de información es un conjunto de componentes que interactúan entre sí para lograr un objetivo común, Una combinación organizada de: personas, hardware, software, redes de comunicaciones, recursos de datos, políticas y procedimientos, que guarda, recupera, transforma y disemina información en una organización. Es decir recopila, manipula, almacena y crea reportes de información respecto de las actividades de negocio de una empresa, con el fin de ayudar en la administración en el manejo de las operaciones. [1] El objetivo de este trabajo es

implementar el sistema que permita satisfacer las necesidades de la clínica Serio, así como ayudar en la eficiencia y calidad de servicio que ofrece.

6.2. Base de datos y Sistemas de Información.

Se realizara la base de datos y vistas para poder desarrollar las interfaces de acuerdo a las necesidades de la organización..

6.3. Base de Datos (Tablas).

En las siguientes imágenes se muestra la creación de cada una de las entidades que se manejara en el SI. La primera tabla corresponde a los pacientes en la cual estarán contenidos los principales datos del paciente para así poder generar un expediente.

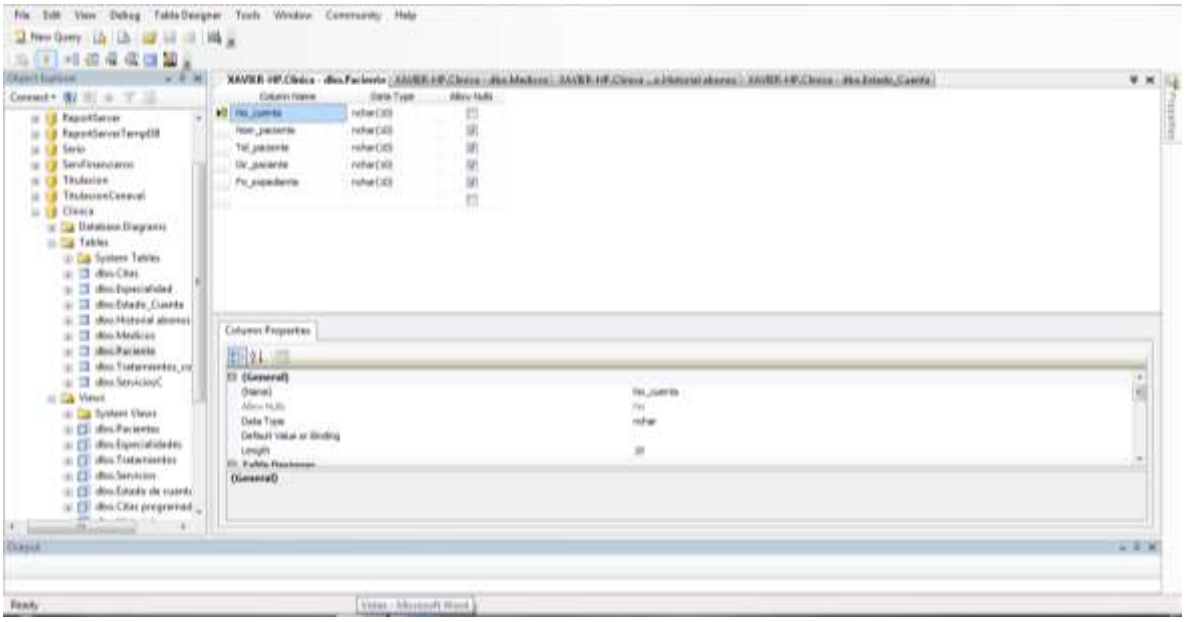


Figura 6.3.1 Tabla Pacientes.

La siguiente pantalla corresponde a los datos del Médico que labora en la clínica

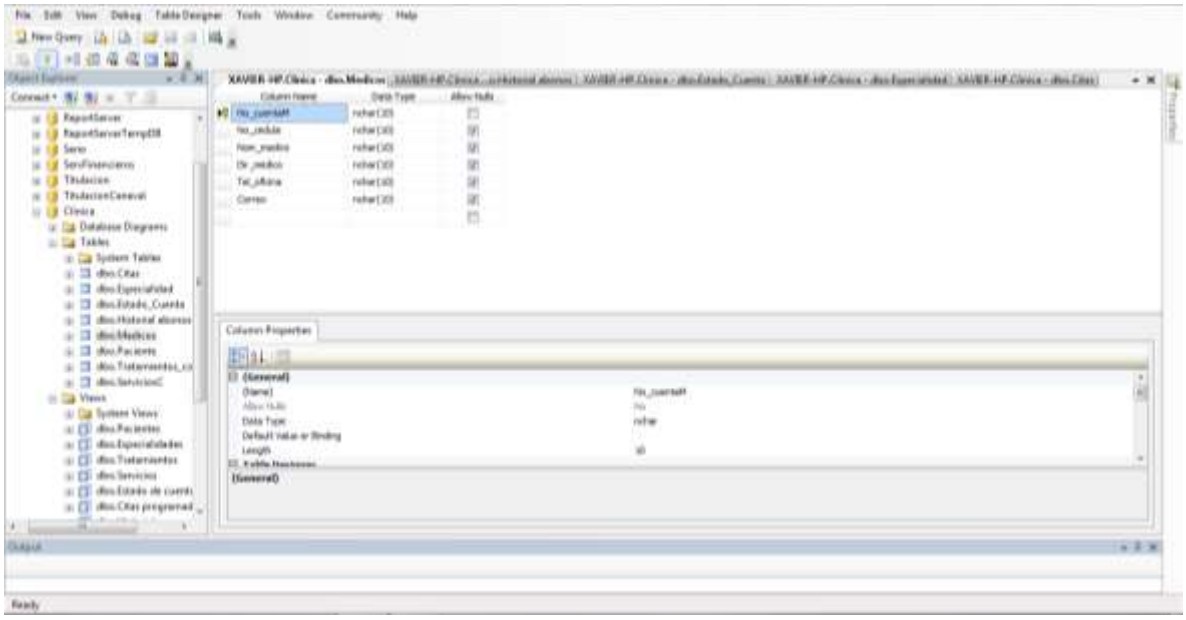


Figura 6.3.2 Tabla Médicos

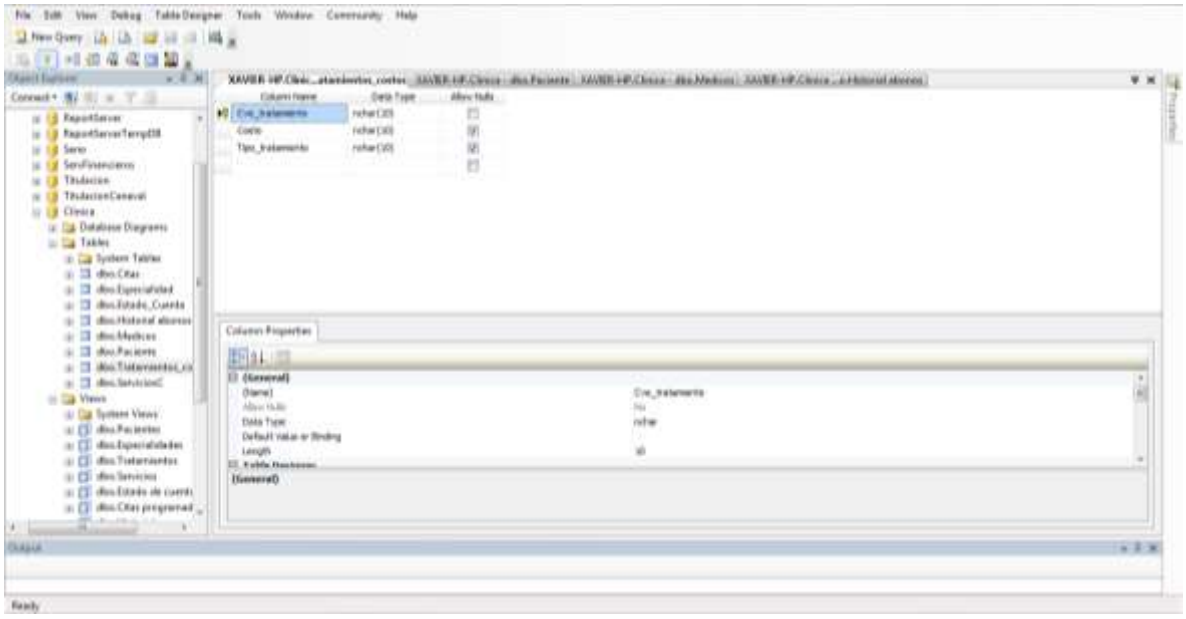


Figura 6.3.3 Tabla Tratamientos Costos

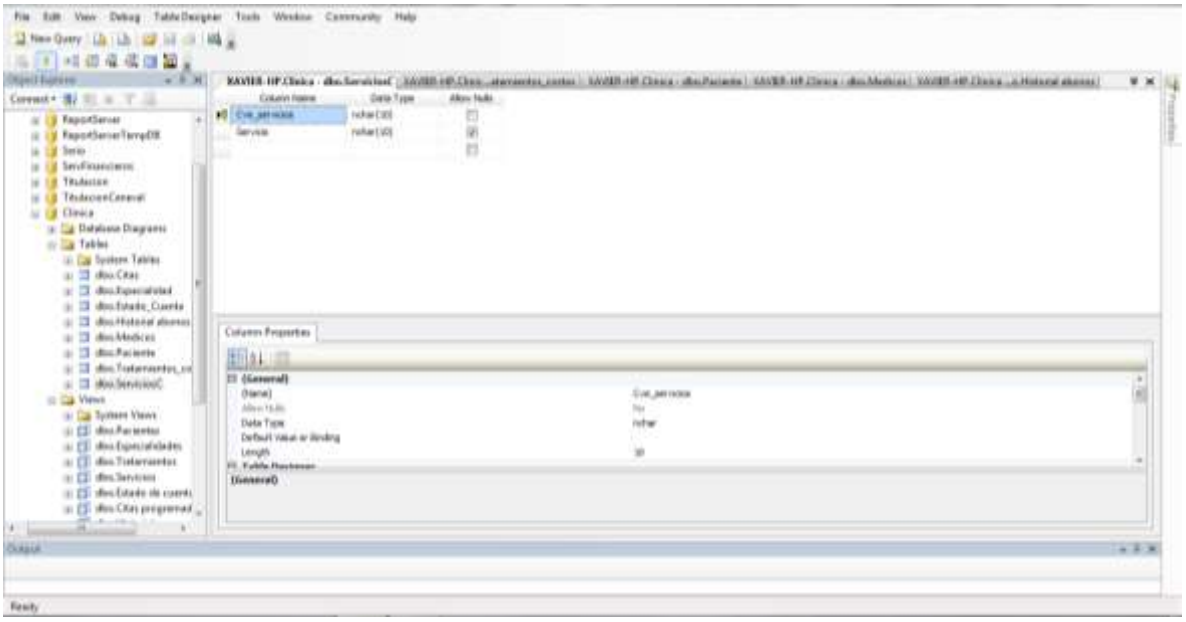


Figura 6.3.4 Tabla Servicios

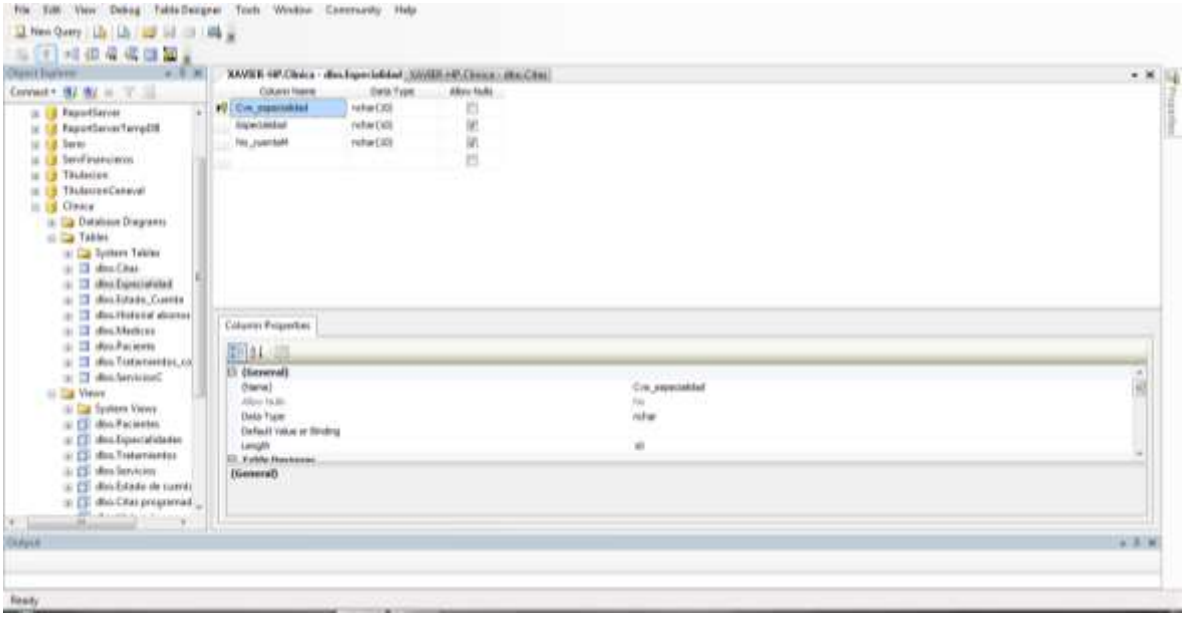


Figura 6.3.5 Tabla Especialidades

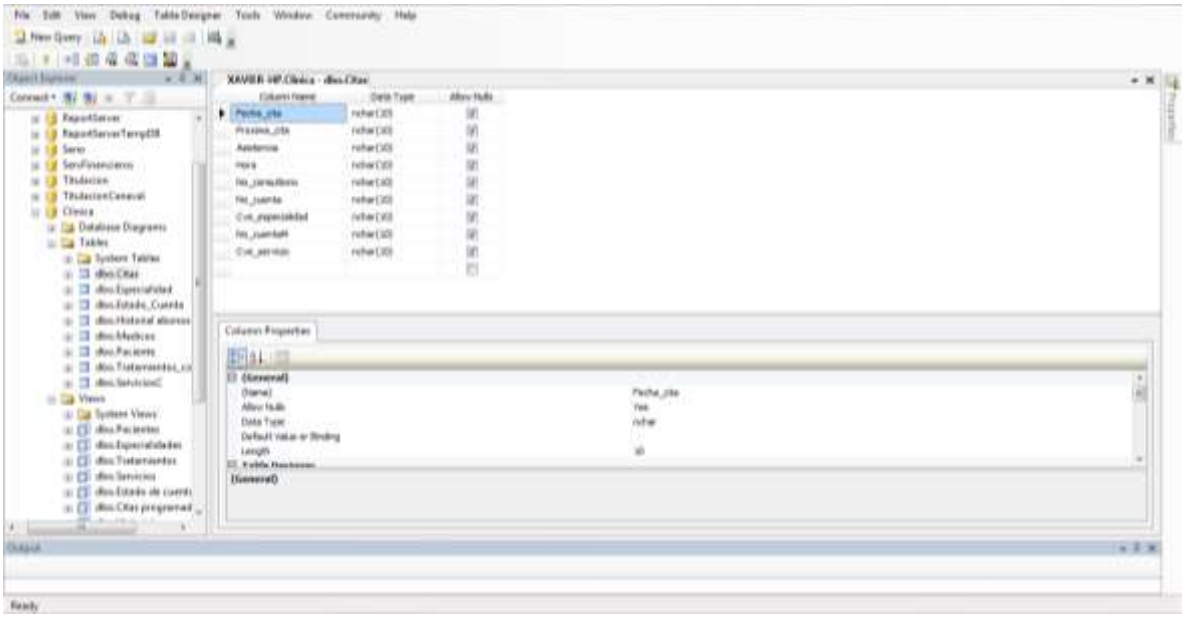


Figura 6.3.6 Tabla Citas Programadas

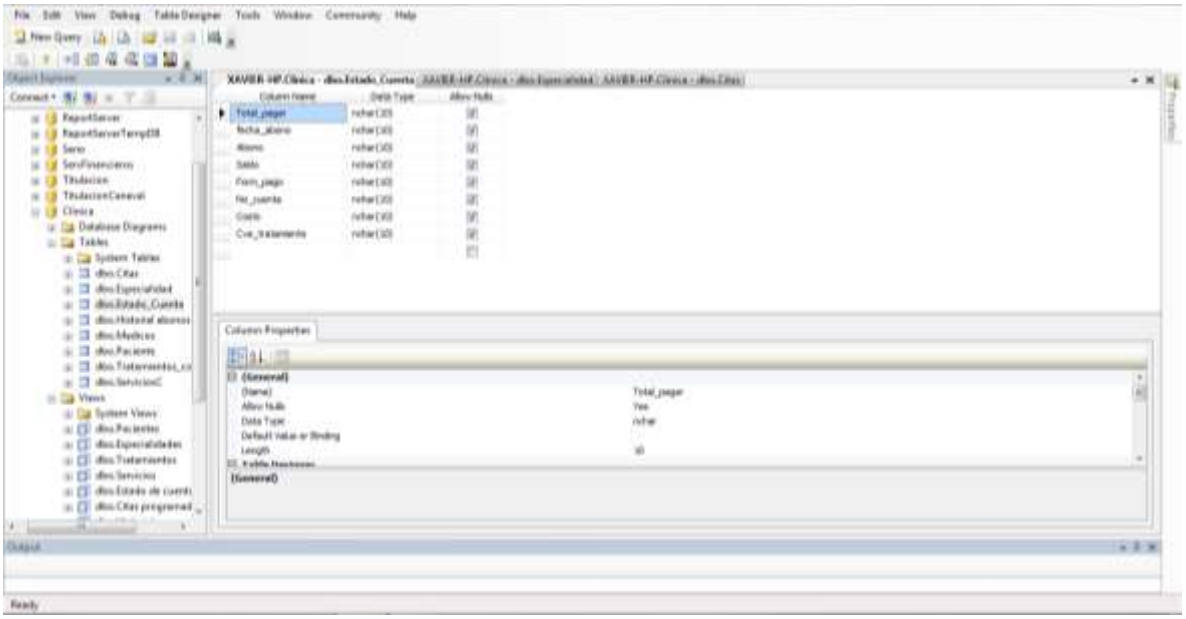


Figura 6.3.7 Tabla Estados de Cuenta

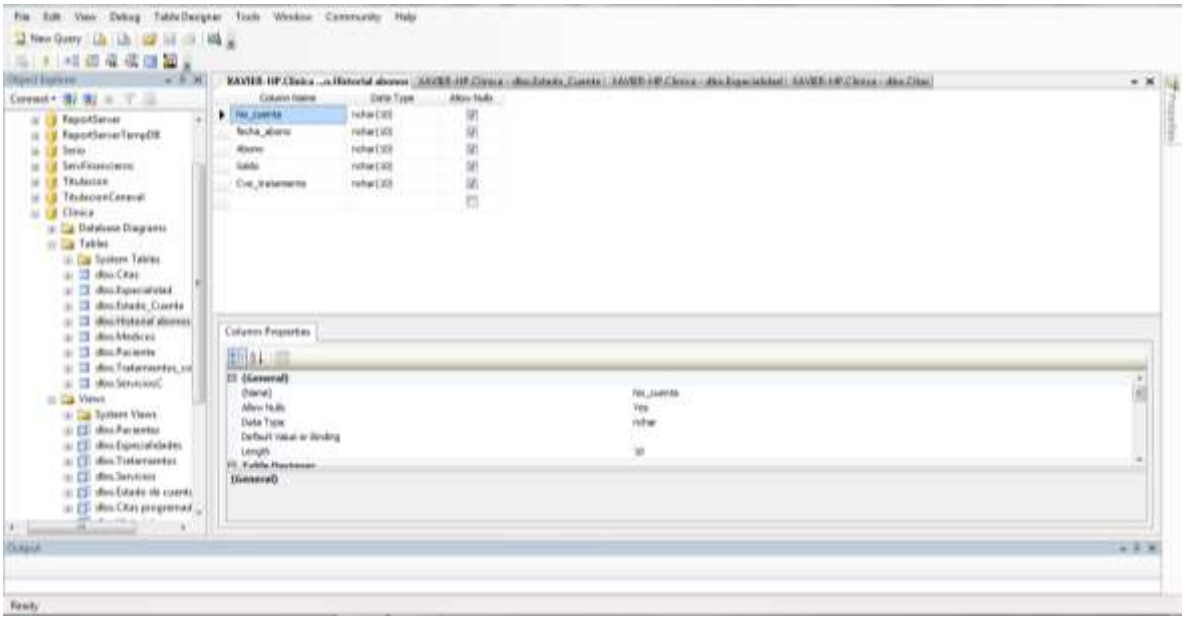


Figura 6.3. 8 Tabla Historial de Abonos

6.4. Vistas.

Una Vista es aquella en la que solo mostrara los datos más relevantes de una o más tablas de la BD con la finalidad de no repetir los datos, las cuales son las llaves foráneas.

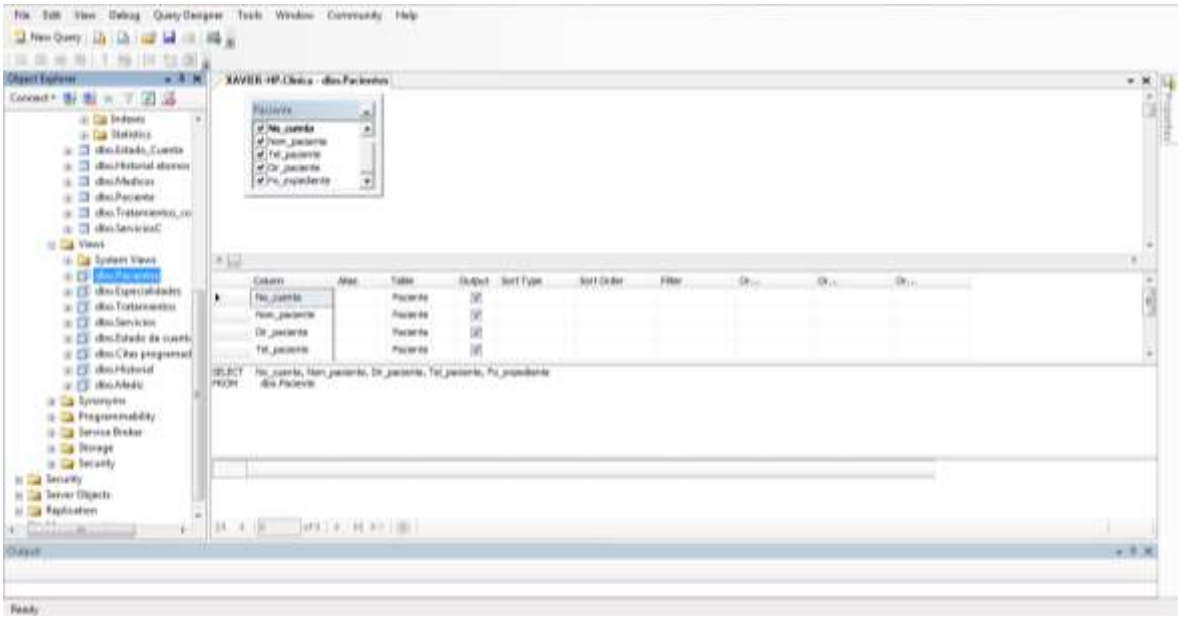


Figura 6.4.1 Vista Paciente.

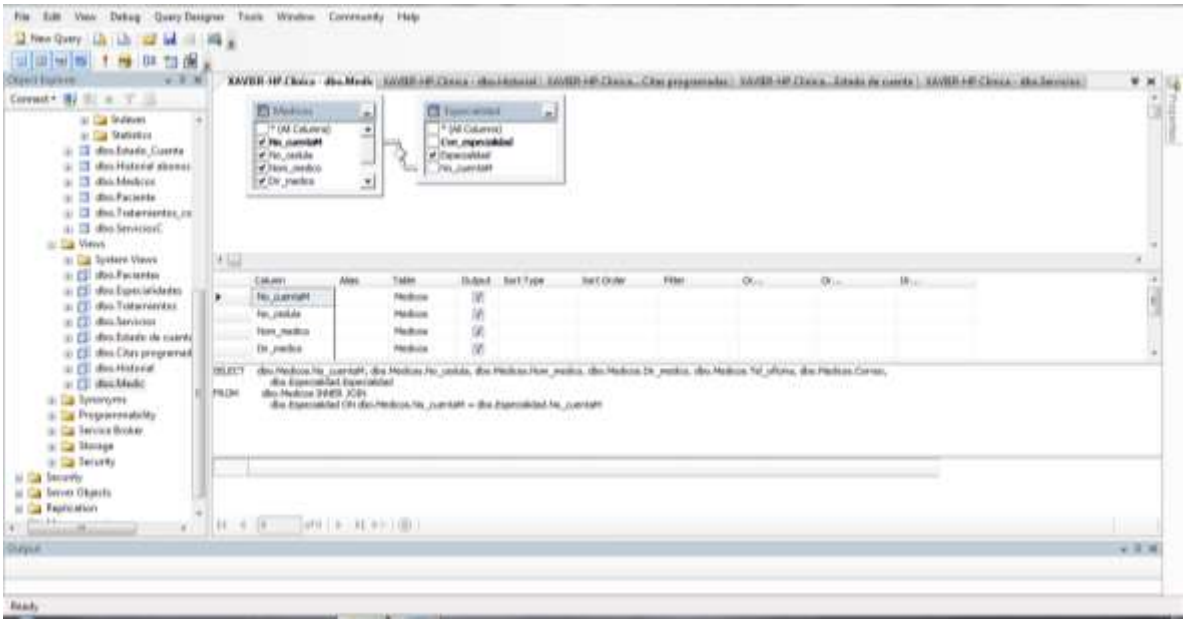


Figura 6.4.2 Vista Médicos

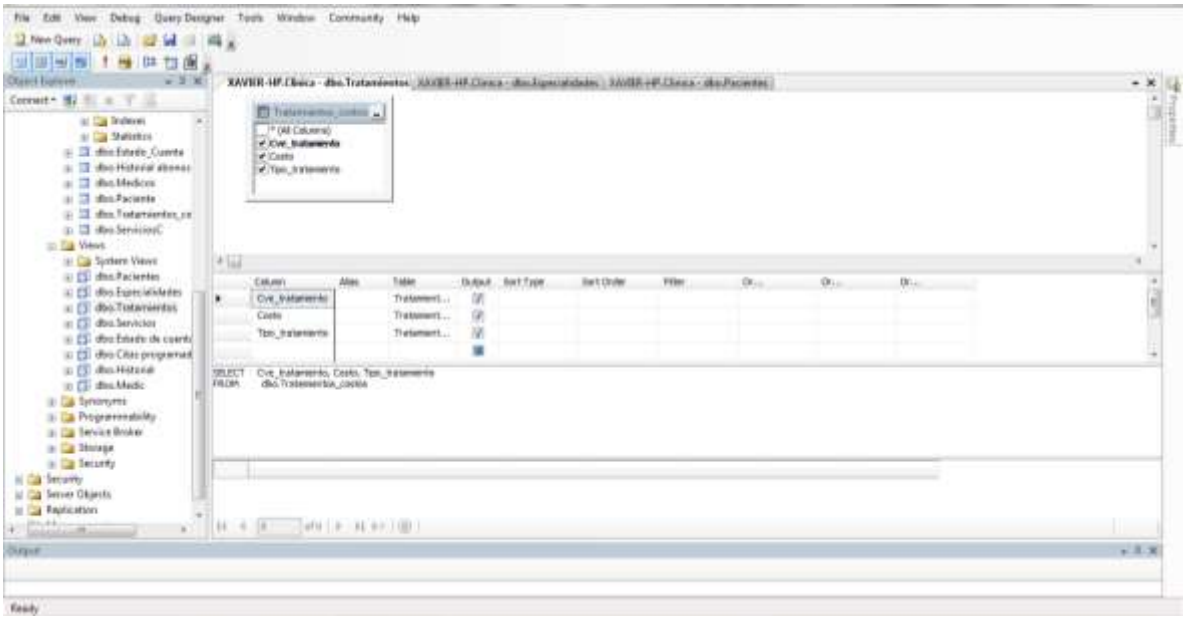


Figura 6.4.3 Vista Tratamientos Costos

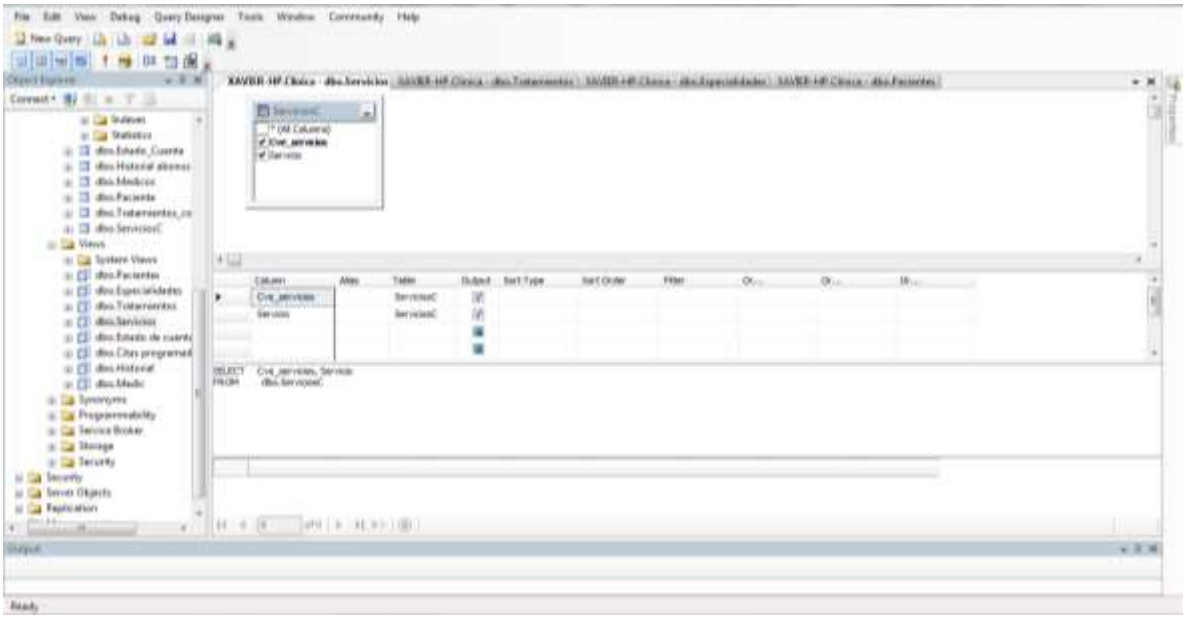


Figura 6.4.4 Vista Servicios

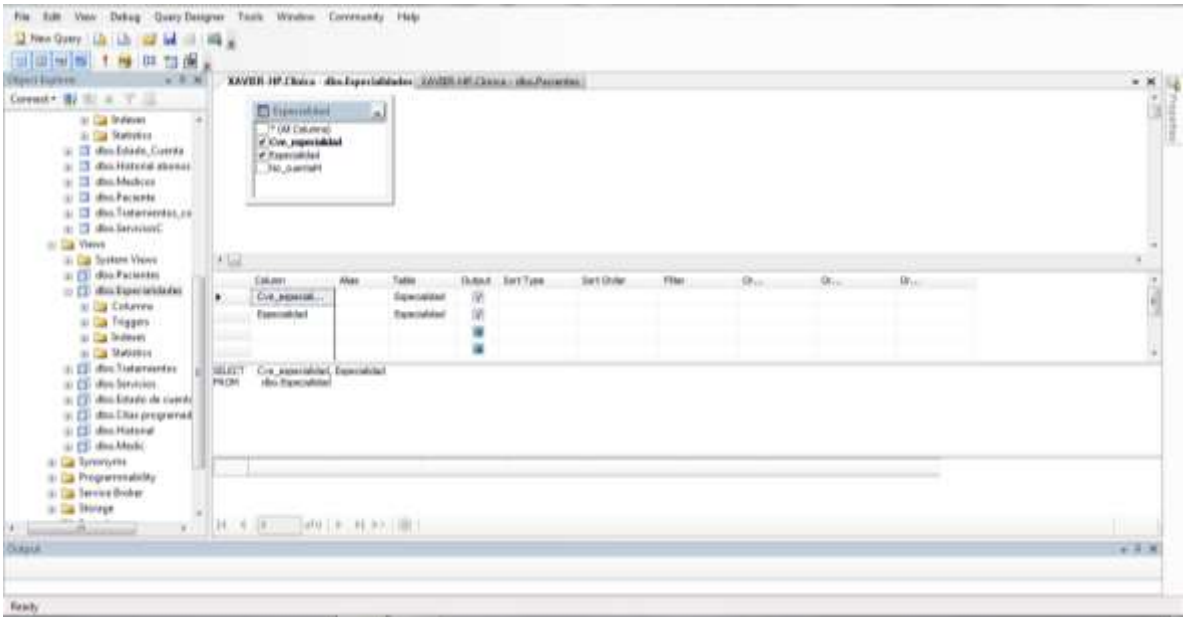


Figura 6.4.5 Vista Especialidades

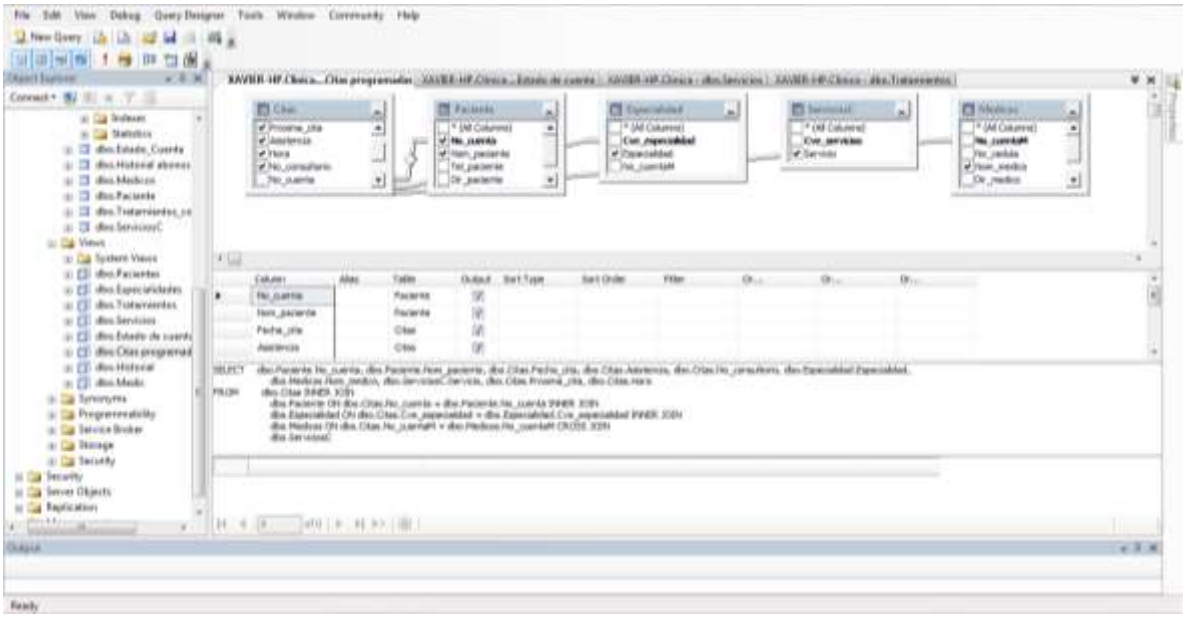


Figura 6.4.6 Vista Citas Programadas

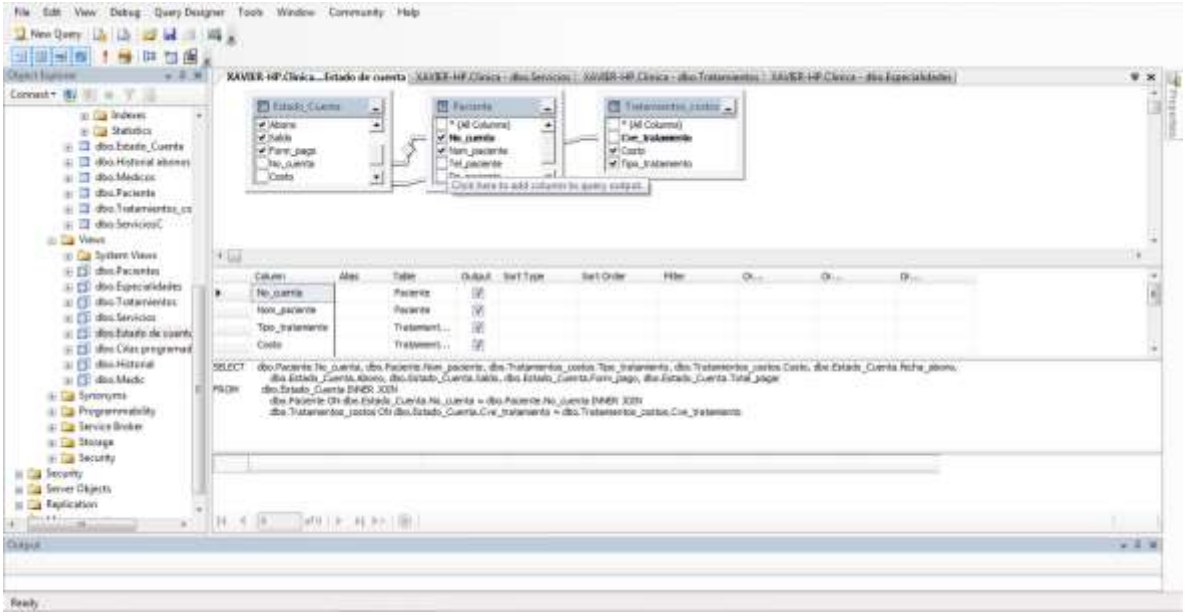


Figura 6.4.7 Vista Estado de Cuenta

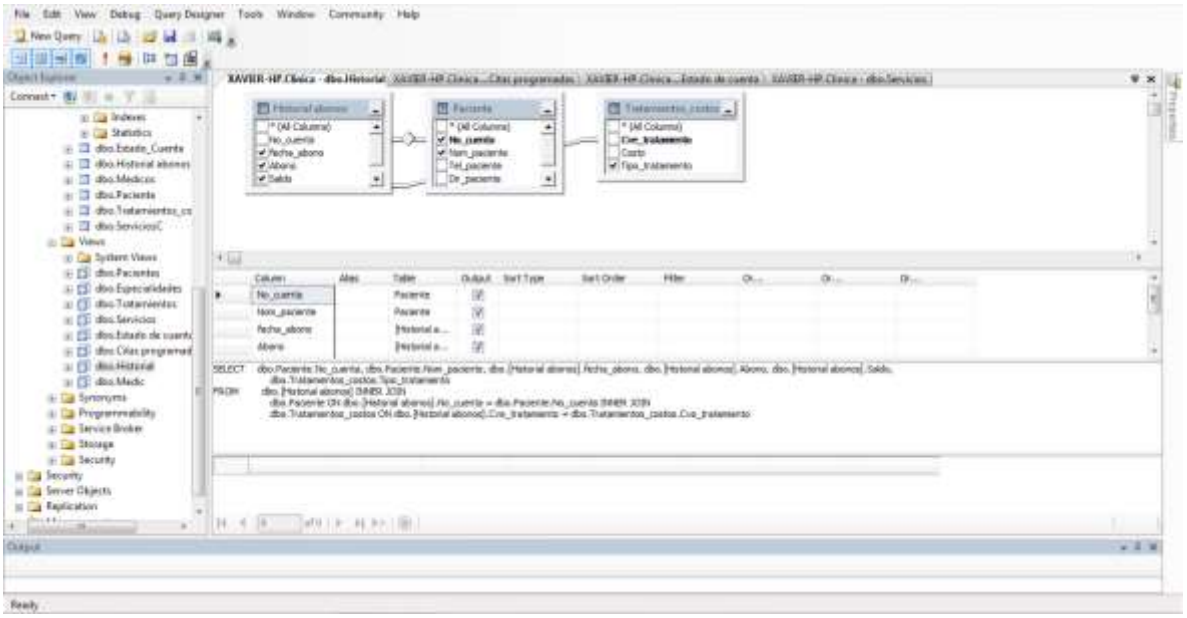


Figura 6.4.8 Vista Historial Abono

6.5. Diagrama Físico de la BD.

Es la forma física en que está relacionada las entidades de la BD

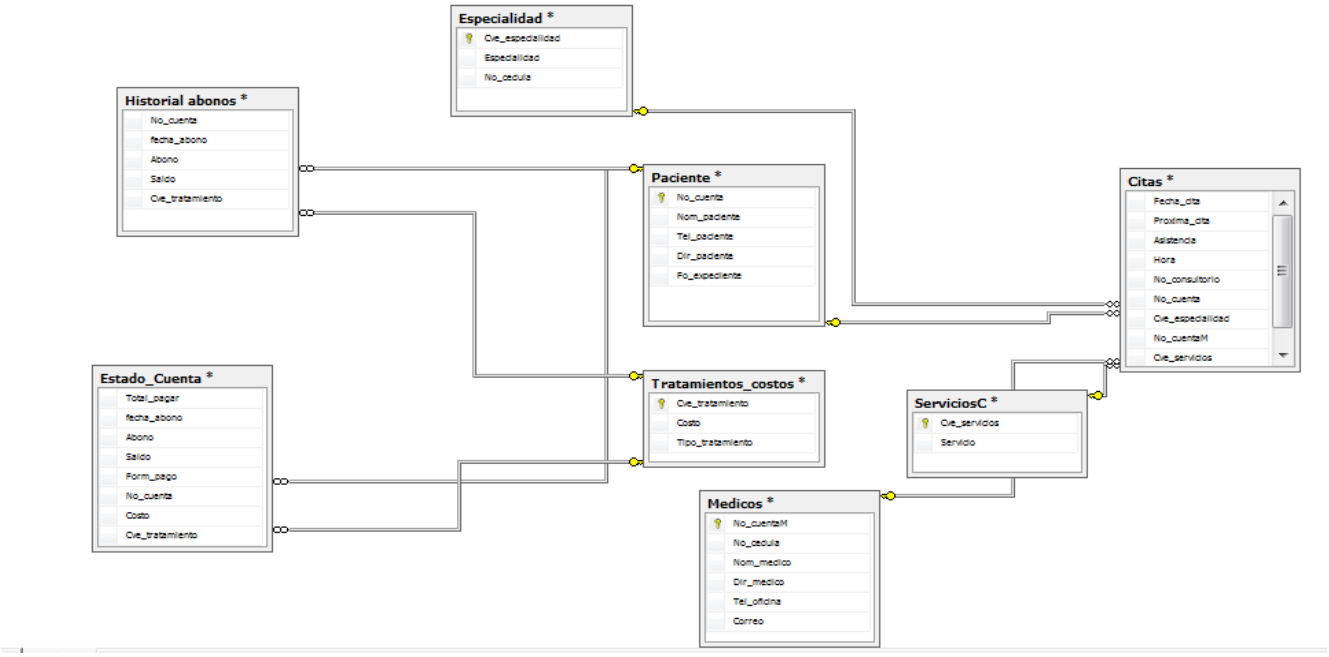


Figura 6.5.1 Diagrama Físico

6.6. Sistema de información.

Se presenta el funcionamiento del SI, con la finalidad de saber cómo funciona y si contiene errores sean corregidos.

Figura 6.6.1 Interfaz principal

```
*---------------------------------->>Librerias------------------------------------------>>>*/
import java.sql.*;
import javax.swing.JOptionPane;
/*
 *sirve para la conexion con sql*/
public class Interfaz1 extends javax.swing.JFrame {
    Connection con;
    ResultSet r;
    CallableStatement cst;
    /**
         */
    public Interfaz1() {
        initComponents();
```

```
    setLocationRelativeTo(null);
        try {
        Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
        con=DriverManager.getConnection("jdbc:odbc:conexionBDS");
    } catch (Exception e) {
    }
}
/**
     */
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {
    jButton1 = new javax.swing.JButton();
    jLabel2 = new javax.swing.JLabel();
    jLabel1 = new javax.swing.JLabel();
    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
    setBackground(new java.awt.Color(255, 255, 255));
    getContentPane().setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());
    jButton1.setBackground(new java.awt.Color(204, 204, 204));
    jButton1.setFont(new java.awt.Font("Lucida Bright", 1, 18)); // NOI18N
    jButton1.setText(" Iniciar Sesión...");
    jButton1.setToolTipText("De clic para iniciar sesión");

jButton1.setBorder(javax.swing.BorderFactory.createCompoundBorder(javax.swing.Borde
rFactory.createEtchedBorder(java.awt.Color.white, new java.awt.Color(204, 204, 255)),
javax.swing.BorderFactory.createEtchedBorder(new java.awt.Color(204, 204, 255),
java.awt.Color.lightGray)));
    jButton1.setCursor(new java.awt.Cursor(java.awt.Cursor.DEFAULT_CURSOR));
    jButton1.setDebugGraphicsOptions(javax.swing.DebugGraphics.NONE_OPTION);
    jButton1.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
```

```
            jButton1ActionPerformed(evt);
          }
        });
        getContentPane().add(jButton1, new
org.netbeans.lib.awtextra.AbsoluteConstraints(430, 290, 160, 50));
        jLabel2.setIcon(new javax.swing.ImageIcon("H:\\Serio\\Imagenes\\img1.jpg")); //
NOI18N
        getContentPane().add(jLabel2, new org.netbeans.lib.awtextra.AbsoluteConstraints(10,
100, -1, 330));
        jLabel1.setIcon(new javax.swing.ImageIcon("H:\\Serio\\Imagenes\\fondo6.jpg")); //
NOI18N
        getContentPane().add(jLabel1, new org.netbeans.lib.awtextra.AbsoluteConstraints(0,
0, 620, 470));
        pack();
    }// </editor-fold>
    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        // codigo para entrar a la cuenta administrador
        Interfaz2 I2 =new Interfaz2();
        I2.setVisible(true);
        this.setVisible(false);
    }
    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
        /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and
feel.
         * For details see
http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
```

```
        */
        try {
            for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(Interfaz1.class.getName()).log(java.util.logging.Level.S
EVERE, null, ex);
        } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(Interfaz1.class.getName()).log(java.util.logging.Level.S
EVERE, null, ex);
        } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(Interfaz1.class.getName()).log(java.util.logging.Level.S
EVERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(Interfaz1.class.getName()).log(java.util.logging.Level.S
EVERE, null, ex);
        }
        //</editor-fold>
        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new Interfaz1().setVisible(true);
```

```
        }
    });
}
// Variables declaration - do not modify
private javax.swing.JButton jButton1;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
// End of variables declaration
```

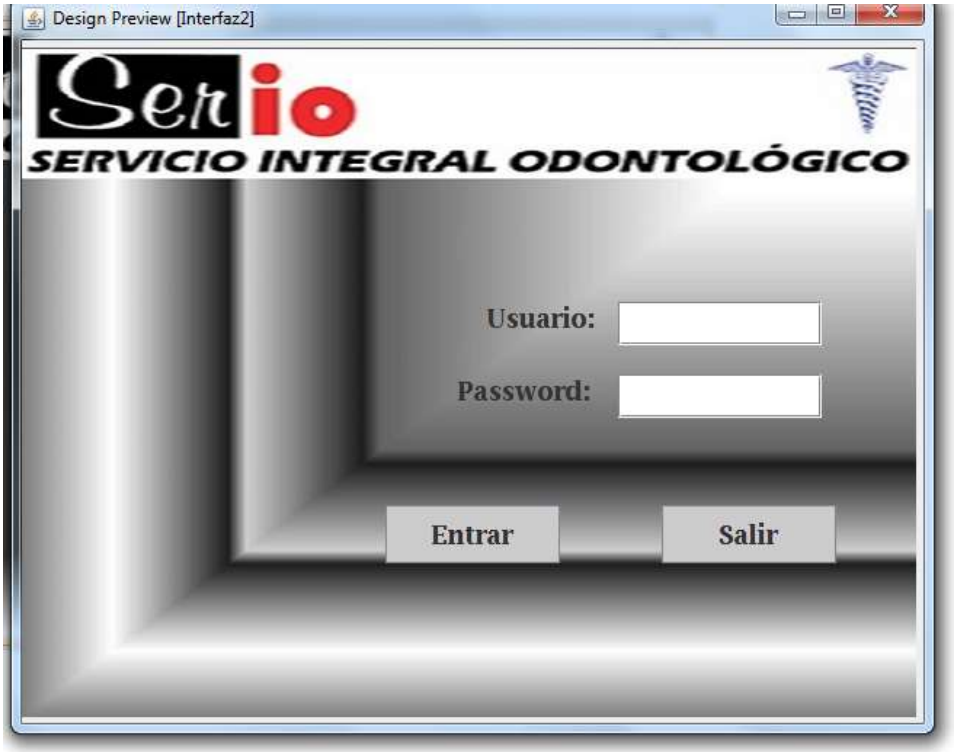


Figura 6.6.2 Inicio de Sesión

```
*/------------------------------>>>-Librerias-------------------------->>> */
import java.sql.*;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;
```

```
/**
 ----------------Declaracion de variables------------------- */
public class Interfaz2 extends javax.swing.JFrame {
    String usuario;
    Connection con;
    ResultSet r;
    CallableStatement cst;

    /**
     * ------------------Conexion con sql y centra la interfaz--------
     */
    public Interfaz2() {
        initComponents();
        setLocationRelativeTo(null);
            try {
            Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
            try {
                con=DriverManager.getConnection("jdbc:odbc:serio");

            } catch (SQLException ex) {
            }
                    }
            catch (ClassNotFoundException ex) {
                Logger.getLogger(Interfaz2.class.getName()).log(Level.SEVERE, null, ex);
            }
    }

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        // Accion del boton para ingresar a Menú del Administrador:
        usuario = jTextField1.getText();
        String password=pass.getText();
```

```
if((password.isEmpty() && usuario.isEmpty()))
{
   JOptionPane.showMessageDialog(null, "Ingrese su nombre de usuario y contraseña");
}
else
{
    try {
      Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
   } catch (ClassNotFoundException ex) {
      Logger.getLogger(Interfaz2.class.getName()).log(Level.SEVERE, null, ex);
   }
   try {
      con=DriverManager.getConnection("jdbc:odbc:serio");

   } catch (SQLException ex) {
   }

}

   try {
      Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
   } catch (ClassNotFoundException ex) {
      Logger.getLogger(Interfaz2.class.getName()).log(Level.SEVERE, null, ex);
   }
   try {
      con=DriverManager.getConnection("jdbc:odbc:serio");
        } catch (SQLException ex) {
   }
            try
```

```
{
    Statement st = con.createStatement();
    ResultSet rs = st.executeQuery("SELECT * FROM Medicos Where No_cuentaM='"+usuario+"'and Passwordm='"+password+"'");
    rs.next();
    int encontrado = rs.getRow();
    this.Cargo.setText(rs.getString("Tipo_usuario"));

   if (encontrado == 1)
    {
      JOptionPane.showMessageDialog(null, "Logueo Exitoso","Usuario y Contrasena Correcta",JOptionPane.INFORMATION_MESSAGE);
       if(Cargo.getText().equals("Admin"))
      {
        Interfaz4 I4 = new  Interfaz4();
      I4.setVisible(true);

      hide();
      }
       if(Cargo.getText().equals("Usuario"))
     {
      Interfaz5 I5 = new Interfaz5();
         I5.setVisible(true);
           hide();
      }
    }
    else
    {
      JOptionPane.showMessageDialog(null, "Usuario o contrasena Incorrecta","Usuario o contrasena Incorrecta",JOptionPane.INFORMATION_MESSAGE);
    }
```

```
    }
    catch (SQLException ex)
   {
    }
}
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    // codigo para salir de la cuenta y regresar al interfaz principal
    Interfaz1 I1 =new Interfaz1();
    I1.setVisible(true);
    this.setVisible(false);}
```

Figura 6.6.3 Módulos de Sesión Administrador

```
* ------------------Librerias-------------------
 */
import java.sql.*;
import javax.swing.JOptionPane;
```

```
/**

 */
public class Interfaz4 extends javax.swing.JFrame {
    Connection con;
    ResultSet r;
    CallableStatement cst;

    /**
     ----------Conexion a la base de datos----------
     */
    public Interfaz4() {
        initComponents();
        setLocationRelativeTo(null);
        try {
            Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
            con=DriverManager.getConnection("jdbc:odbc:conexionBDS");
        } catch (Exception e) {
        }
    }

    /**
         */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jButton1 = new javax.swing.JButton();
        jButton2 = new javax.swing.JButton();
        jButton4 = new javax.swing.JButton();
```

```
        jButton6 = new javax.swing.JButton();
        jButton7 = new javax.swing.JButton();
        jButton9 = new javax.swing.JButton();
        jLabel2 = new javax.swing.JLabel();
        jLabel1 = new javax.swing.JLabel();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
        setBackground(new java.awt.Color(204, 204, 204));
        getContentPane().setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

        jButton1.setBackground(new java.awt.Color(204, 204, 204));
        jButton1.setFont(new java.awt.Font("Lucida Bright", 1, 18)); // NOI18N
        jButton1.setText("Pacientes");
        jButton1.setToolTipText("De clic para registrar un nvo paciente");

jButton1.setBorder(javax.swing.BorderFactory.createEtchedBorder(java.awt.Color.lightGr
ay, java.awt.Color.gray));
        jButton1.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton1ActionPerformed(evt);
            }
        });
        getContentPane().add(jButton1, new
org.netbeans.lib.awtextra.AbsoluteConstraints(10, 110, 200, -1));

        jButton2.setBackground(new java.awt.Color(204, 204, 204));
        jButton2.setFont(new java.awt.Font("Lucida Bright", 1, 18)); // NOI18N
        jButton2.setText("Médicos");
        jButton2.setToolTipText("Clic para regristar un nvo médico");
```

```
jButton2.setBorder(javax.swing.BorderFactory.createEtchedBorder(java.awt.Color.lightGray, java.awt.Color.gray));
        jButton2.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton2ActionPerformed(evt);
            }
        });
        getContentPane().add(jButton2, new org.netbeans.lib.awtextra.AbsoluteConstraints(30, 160, 200, -1));

        jButton4.setBackground(new java.awt.Color(204, 204, 204));
        jButton4.setFont(new java.awt.Font("Lucida Bright", 1, 18)); // NOI18N
        jButton4.setText("Catalogos");
        jButton4.setToolTipText("De clic para ver servicios, Tratamientos y Costos, Especialidades");

jButton4.setBorder(javax.swing.BorderFactory.createEtchedBorder(java.awt.Color.lightGray, java.awt.Color.gray));
        jButton4.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton4ActionPerformed(evt);
            }
        });
        getContentPane().add(jButton4, new org.netbeans.lib.awtextra.AbsoluteConstraints(60, 210, 200, -1));

        jButton6.setBackground(new java.awt.Color(204, 204, 204));
        jButton6.setFont(new java.awt.Font("Lucida Bright", 1, 18)); // NOI18N
        jButton6.setText("Estado de Cuenta");
        jButton6.setToolTipText("De clic para entrar");
```

```
jButton6.setBorder(javax.swing.BorderFactory.createEtchedBorder(java.awt.Color.lightGr
ay, java.awt.Color.gray));
        jButton6.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton6ActionPerformed(evt);
            }
        });
        getContentPane().add(jButton6, new
org.netbeans.lib.awtextra.AbsoluteConstraints(160, 310, 200, -1));

        jButton7.setBackground(new java.awt.Color(204, 204, 204));
        jButton7.setFont(new java.awt.Font("Lucida Bright", 1, 18)); // NOI18N
        jButton7.setText("Citas Programadas");
        jButton7.setToolTipText("De clic para agregar una nva cita");

jButton7.setBorder(javax.swing.BorderFactory.createEtchedBorder(java.awt.Color.lightGr
ay, java.awt.Color.gray));
        jButton7.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton7ActionPerformed(evt);
            }
        });
        getContentPane().add(jButton7, new
org.netbeans.lib.awtextra.AbsoluteConstraints(100, 260, 200, -1));

        jButton9.setBackground(new java.awt.Color(204, 204, 204));
        jButton9.setFont(new java.awt.Font("Lucida Bright", 1, 18)); // NOI18N
        jButton9.setText("Cerrar Sesión");
        jButton9.setToolTipText("De clic para regresar a la pantalla principal");
```

```
jButton9.setBorder(javax.swing.BorderFactory.createEtchedBorder(java.awt.Color.lightGray, java.awt.Color.gray));
        jButton9.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton9ActionPerformed(evt);
            }
        });
        getContentPane().add(jButton9, new org.netbeans.lib.awtextra.AbsoluteConstraints(400, 400, 180, -1));

        jLabel2.setFont(new java.awt.Font("Lucida Bright", 1, 36)); // NOI18N
        jLabel2.setText("ADMINISTRADOR");
        getContentPane().add(jLabel2, new org.netbeans.lib.awtextra.AbsoluteConstraints(280, 160, -1, -1));

        jLabel1.setIcon(new javax.swing.ImageIcon("C:\\Users\\Erick\\Documents\\NetBeansProjects\\Serio\\Imagenes\\fondo6.jpg")); // NOI18N
        getContentPane().add(jLabel1, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, 620, 470));

        pack();
    }// </editor-fold>

    private void jButton9ActionPerformed(java.awt.event.ActionEvent evt) {
        // código para salir y regresar a la interfaz 1
        Interfaz1 I1 =new Interfaz1();
        I1.setVisible(true);
        this.setVisible(false);
    }
```

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
   // Código para entrar a la interfaz de registro de paciente:
   Interfaz6u1 I6u =new Interfaz6u1();
   I6u.setVisible(true);
   this.setVisible(false);
}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
   // Código para entrar a la interfaz de registro de Médicos:
   Interfaz7 I7 =new Interfaz7();
   I7.setVisible(true);
   this.setVisible(false);
}

private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {
   // Código para ingresar a la interfaz de Catalogos
   Interfaz8u1 I8u =new Interfaz8u1();
   I8u.setVisible(true);
   this.setVisible(false);
}

private void jButton6ActionPerformed(java.awt.event.ActionEvent evt) {
   // Manda al Estado de Cuenta
   Interfaz12 I12 =new Interfaz12();
   I12.setVisible(true);
   this.setVisible(false);
}

private void jButton7ActionPerformed(java.awt.event.ActionEvent evt) {
   // Manda a la interfaz de Citas Programadas
```

```
        Interfaz11 I11 =new Interfaz11();
        I11.setVisible(true);
        this.setVisible(false);
    }
```

Figura 6.6.4 Modulo de Sesión Usuario

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        // Manda a la interfaz de registro de Pacientes:
        Interfaz6 I6 =new Interfaz6();
        I6.setVisible(true);
        this.setVisible(false);
    }

    private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {
        // Manda a la interfaz de Catalogos
```

```
    Interfaz8u I8 =new Interfaz8u();
    I8.setVisible(true);
    this.setVisible(false);
}
private void jButton7ActionPerformed(java.awt.event.ActionEvent evt) {
    // Manda a la interfaz de Citas programadas
    Interfaz11u I11u =new Interfaz11u();
    I11u.setVisible(true);
    this.setVisible(false);
}
private void jButton9ActionPerformed(java.awt.event.ActionEvent evt) {
    //Regresa a la interfaz de Tipo de usuario
    Interfaz1 I1 =new Interfaz1();
    I1.setVisible(true);
    this.setVisible(false);
}
```

En la siguientes interfaces creamos métodos los cuales sirven para hacer más fácil la programación y ahorrar código. El siguiente código pertenece a la programación del administrador. En la interfaz del usuario solo podrá hacer búsquedas, ya que no tendrá permisos para modificar o eliminar algún registro de la base de datos.

El siguiente código pertenece a cada una de las pestañas de la interfaz de Catálogos.

Figura 6.6.5 Modulo Médicos

```
----------->>>Librerias---------->>>
 */
import java.sql.*;
import java.util.ArrayList;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;

/**
 ------>>>>Declaración de variables---------->>>>
 */
public final class Interfaz7 extends javax.swing.JFrame {
    DefaultTableModel model;
    Connection con4;
     Connection con;
    CallableStatement cst;
    ResultSet r;

   /**
    ---------->>>Conexión a SQL-------->>>>
    */
   public Interfaz7() {
       initComponents();
       setLocationRelativeTo(null);
      try {
          Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
          try {
              con4=DriverManager.getConnection("jdbc:odbc:serio");
          } catch (SQLException ex) {
              Logger.getLogger(Interfaz7.class.getName()).log(Level.SEVERE, null, ex);
```

```
            }
        } catch (ClassNotFoundException ex) {
            Logger.getLogger(Interfaz7.class.getName()).log(Level.SEVERE, null, ex);
        }

        cargar("");
        limpiar();

    }
    /*-------->>>Limpia los registros------->>>*/
    void limpiar()
{
    jTextField1.setText("");
    jTextField2.setText("");
    jTextField3.setText("");
    jTextField4.setText("");
    jTextField5.setText("");
    jTextField6.setText("");
    jTextField7.setText("");
    jTextField8.setText("");
    jTextField9.setText("");
    jTextField10.setText("");
    pass.setText("");

}
   /*------->>>Carga los registros y los actualiza--------->>>*/
    void cargar(String consul)
    {
```

```
String[] titulos ={"No.Cuenta","No.Cedula","Password","Médico","Dirección","Teléfono","Tel.Oficina","Correo electrónico","Especialidad"};
String[] registros=new String[9];

String sql ="SELECT * FROM  Medicos where Nom_medico LIKE '%"+consul+"%'";

model= new DefaultTableModel(null,titulos);

try {

Statement st = con4.createStatement();
ResultSet rs= st.executeQuery(sql);

while(rs.next())
{
registros[0]=rs.getString("No_cuentaM");
registros[1]=rs.getString("No_cedula");
registros[2]=rs.getString("Passwordm");
registros[3]=rs.getString("Nom_medico");
registros[4]=rs.getString("Dir_medico");
registros[5]=rs.getString("Tel_medico");
registros[6]=rs.getString("Tel_oficina");
registros[7]=rs.getString("Correo");
registros[8]=rs.getString("Especialidad");
//registros[8]=rs.getString("Tipo_usuario");



model.addRow(registros);
}
```

```
            jTable1.setModel(model);

        } catch (SQLException ex) {
            JOptionPane.showMessageDialog(null,ex);
        }

    }
private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
    // ------------------->>>Modifica los registros------------>>>
        try {
            Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
            try {
                con=DriverManager.getConnection("jdbc:odbc:serio");
            } catch (SQLException ex) {
                Logger.getLogger(Interfaz7.class.getName()).log(Level.SEVERE, null, ex);
            }
        } catch (ClassNotFoundException ex) {
            Logger.getLogger(Interfaz7.class.getName()).log(Level.SEVERE, null, ex);
        }
        String No_cuentaM=jTextField1.getText();
        String Nom_medico=jTextField2.getText();
        String No_cedula=jTextField3.getText();
        String Tipo_usuario=(String)jComboBox1.getSelectedItem();
        String Tel_medico=jTextField4.getText();
        String Tel_oficina=jTextField5.getText();
        String Dir_medico=jTextField6.getText();
        String Correo=jTextField7.getText();
        String Especialidad=jTextField8.getText();
        String Passwordm=pass.getText();
```

```
String Sql;
Sql="UPDATE Medicos SET No_cuentaM=?,No_cedula=?,Passwordm=?,Nom_medico=?,Dir_medico=?,Tel_medico=?,Tel_oficina=?,Correo=?,Especialidad=?,Tipo_usuario=? WHERE No_cedula="+No_cedula+"";
      try {
        // Statement st = con4.createStatement();
         //ResultSet rs= st.executeQuery(Sql);
         PreparedStatement go = con.prepareStatement(Sql);
         go.setString(1,No_cuentaM);
         go.setString(2,No_cedula);
         go.setString(3,Passwordm);
         go.setString(4,Nom_medico);
         go.setString(5,Dir_medico);
         go.setString(6,Tel_medico);
         go.setString(7,Tel_oficina);
         go.setString(8,Correo);
         go.setString(9,Especialidad);
         go.setString(10,Tipo_usuario);

         int r=go.executeUpdate();

         if (r>0)
         {
            JOptionPane.showMessageDialog(null, "Registro Exitoso");

         }
      } catch (SQLException ex) {
         Logger.getLogger(Interfaz6u1.class.getName()).log(Level.SEVERE, null, ex);
      }
      limpiar();
```

```
        cargar("");



    }

    private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {
        //------------------->>>Limpia los registros------------>>>
        limpiar();
    }

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        // ----------------->>>Guarda los regsitros-------------->>>
        String No_cuentaM=jTextField1.getText();
        String Nom_medico=jTextField2.getText();
        String No_cedula=jTextField3.getText();
        String Tel_medico=jTextField4.getText();
        String Tel_oficina=jTextField5.getText();
        String Dir_medico=jTextField6.getText();
        String Correo=jTextField7.getText();
        String Especialidad=jTextField8.getText();
        String Tipo_usuario=(String)jComboBox1.getSelectedItem();
        String Passwordm=pass.getText();

String Sql;
Sql="INSERT INTO Medicos (No_cuentaM, No_cedula, Passwordm, Nom_medico,
Dir_medico, Tel_medico, Tel_oficina, Correo, Especialidad,
Tipo_usuario)values(?,?,?,?,?,?,?,?,?,?)";
        try {
            PreparedStatement g = con4.prepareStatement(Sql);

            //g.setString(1,Identificador);
```

```
            g.setString(1,No_cuentaM);
            g.setString(2,No_cedula);
            g.setString(3,Passwordm);
            g.setString(4,Nom_medico);
            g.setString(5,Dir_medico);
            g.setString(6,Tel_medico);
            g.setString(7,Tel_oficina);
            g.setString(8,Correo);
            g.setString(9,Especialidad);
            g.setString(10,Tipo_usuario);

            int r=g.executeUpdate();

            if (r>0)
            {
               JOptionPane.showMessageDialog(null, "Registro Exitoso");

            }
        } catch (SQLException ex) {
            Logger.getLogger(Interfaz6u1.class.getName()).log(Level.SEVERE, null, ex);
        }
        limpiar();
        cargar("");

    }
private void jTable1MouseClicked(java.awt.event.MouseEvent evt) {
        // --->>>Carga los registros en los campos de la interfaz--->>>
        try {
            Statement st = con4.createStatement();

            int row =jTable1.getSelectedRow();
```

```
String TableC=(jTable1.getModel().getValueAt(row,0).toString());
String sql ="SELECT * FROM  Medicos where No_cuentaM= '"+TableC+"'";
 ResultSet rs= st.executeQuery(sql);
 if(rs.next()){
    String T1 = rs.getString("No_cuentaM");
    jTextField1.setText(T1);
    String T4 = rs.getString("No_cedula");
    jTextField3.setText(T4);
    String T2 = rs.getString("Passwordm");
    pass.setText(T2);
    String T3 = rs.getString("Nom_medico");
    jTextField2.setText(T3);
    String T5 = rs.getString("Dir_medico");
    jTextField6.setText(T5);
    String T6 = rs.getString("Tel_medico");
    jTextField4.setText(T6);
    String T7 = rs.getString("Tel_oficina");
    jTextField5.setText(T7);
    String T8 = rs.getString("Correo");
    jTextField7.setText(T8);
    String T9 = rs.getString("Especialidad");
    jTextField8.setText(T9);
    String T10 = rs.getString("Tipo_usuario");
    jTextField9.setText(T10);

 }

} catch (Exception ex) {
  JOptionPane.showMessageDialog(null,ex);
```

```
    }

    }

    private void jTextField10KeyReleased(java.awt.event.KeyEvent evt) {
        // ----------->>>Realiza una busqueda indexada------------>>>
        cargar(jTextField10.getText());
    }

    private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {
        // ---------------->>>Elimina los registros--------->>>
        String No_cuentaM= jTextField1.getText();
       String sql="DELETE FROM Medicos WHERE No_cuentaM="+No_cuentaM+"";
        try {
            Statement st=con4.createStatement();
            int i = st.executeUpdate(sql);
            if(i==1){
            JOptionPane.showMessageDialog(this,
"Eliminado","Aviso",JOptionPane.INFORMATION_MESSAGE);
            }else {
                JOptionPane.showMessageDialog(this,          "Error          al
elminar","Aviso",JOptionPane.INFORMATION_MESSAGE);
            }

        } catch (SQLException ex) {
            Logger.getLogger(Interfaz6u1.class.getName()).log(Level.SEVERE, null, ex);
        }

       limpiar();
       jTextField10.setText("");
```

```
    cargar("");
  }

  private void jButton6ActionPerformed(java.awt.event.ActionEvent evt) {
    // ---->>>Regresa a la interfaz del menú del administrador----->>>
    Interfaz4 I4=new Interfaz4();
    I4.setVisible(true);
    this.setVisible(false);
  }
```

Figura 6.6.6 Módulo Paciente

```
*----------->>>Librerias---------->>>*/

import java.sql.*;
import java.util.ArrayList;
import java.util.logging.Level;
```

```
import java.util.logging.Logger;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;
/**
---------->>>Declaración de variables---------->>>
 */
public class Interfaz6 extends javax.swing.JFrame {
DefaultTableModel model;
Connection con;
    CallableStatement cst;
    ResultSet r;

    /*
     ---------->>>Conexión a SQL---------->>>
     */
    public Interfaz6() {
        initComponents();
        setLocationRelativeTo(null);
       try {
            Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
            try {
                con=DriverManager.getConnection("jdbc:odbc:serio");
            } catch (SQLException ex) {
                Logger.getLogger(Interfaz7.class.getName()).log(Level.SEVERE, null, ex);
            }
        } catch (ClassNotFoundException ex) {
            Logger.getLogger(Interfaz7.class.getName()).log(Level.SEVERE, null, ex);
        }

        cargar("");
        limpiar();
```

```
    }
    /*------------>>> Metodo para limpiar los campos---------------->>>*/
    void limpiar()
{
    jTextField1.setText("");
    jTextField2.setText("");
    jTextField3.setText("");
    jTextField4.setText("");
    jTextField5.setText("");
    jTextField6.setText("");
    pass.setText("");

}
    /*----------->>>Actulaiza los registros---------------->>>*/
void cargar(String consul)
    {

        String[] titulos ={"No.Cuenta","Nombre Paciente","Dirección","Teléfono","Folio.Expediente"};
        String[] registros=new String[5];

     String sql ="SELECT * FROM Pacientes where Nom_paciente LIKE '%"+consul+"%'";

        model= new DefaultTableModel(null,titulos);

            try {

         Statement st = con.createStatement();
         ResultSet rs= st.executeQuery(sql);
```

```
          while(rs.next())
          {
           registros[0]=rs.getString("No_cuenta");
           registros[1]=rs.getString("Nom_paciente");

           registros[2]=rs.getString("Dir_paciente");
           registros[3]=rs.getString("Tel_paciente");
           registros[4]=rs.getString("Fo_expediente");




           model.addRow(registros);
           }

           jTable1.setModel(model);

       } catch (SQLException ex) {
           JOptionPane.showMessageDialog(null,ex);
       }

   }
private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
       // ---------->>>Modifica los registros---------->>>
       String No_cuenta=jTextField1.getText();
       String Nom_paciente=jTextField2.getText();
       String Estado=(String)jComboBox1.getSelectedItem();
       String Tel_paciente=jTextField4.getText();
       String Fo_expediente=jTextField5.getText();
       String Dir_paciente=jTextField6.getText();
       String Password=pass.getText();
```

```
String Sql;
Sql="UPDATE Pacientes SET Password=?,Nom_paciente=?,Dir_paciente=?,Tel_paciente=?, Fo_expediente=?, Estado=? WHERE No_cuenta="+No_cuenta+"";
      try {
         PreparedStatement g = con.prepareStatement(Sql);
         //g.setString(1,No_cuenta);
         g.setString(1,Password);
         g.setString(2,Nom_paciente);
         g.setString(3,Dir_paciente);
         g.setString(4,Tel_paciente);
         g.setString(5,Fo_expediente);
         g.setString(6,Estado);

         int r=g.executeUpdate();

         if (r>0)
         {
            JOptionPane.showMessageDialog(null, "Registro Exitoso");

         }
      } catch (SQLException ex) {
         Logger.getLogger(Interfaz6u1.class.getName()).log(Level.SEVERE, null, ex);
      }
      limpiar();
      cargar("");

   }
private void jTable1MouseClicked(java.awt.event.MouseEvent evt) {
      // ----->>>Carga los registros en los campos de la interfaz---->>>
       try {
```

```
        Statement st = con.createStatement();

       int row =jTable1.getSelectedRow();
       String TableC=(jTable1.getModel().getValueAt(row,0).toString());
       String sql ="SELECT * FROM  Pacientes where No_cuenta= '"+TableC+"'";
        ResultSet rs= st.executeQuery(sql);
        if(rs.next()){
           String T1 = rs.getString("No_cuenta");
           jTextField1.setText(T1);
           String T2 = rs.getString("Password");
           pass.setText(T2);
           String T3 = rs.getString("Nom_paciente");
           jTextField2.setText(T3);
           String T4 = rs.getString("Dir_paciente");
           jTextField6.setText(T4);
           String T5 = rs.getString("Tel_paciente");
           jTextField4.setText(T5);
           String T6 = rs.getString("Fo_expediente");
           jTextField5.setText(T6);
           String T7 = rs.getString("Estado");
           jTextField3.setText(T7);

        }

    } catch (Exception ex) {
       JOptionPane.showMessageDialog(null,ex);

}
}

private void jTextField7KeyReleased(java.awt.event.KeyEvent evt) {
```

```
        //-------------->>>Realiza una busqueda indexada---------->>>
        cargar(jTextField7.getText());
    }

    private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {
        // ----------->>>Limpia los campos------------>>>
        limpiar();
    }

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        // ------------->>>Guarda los registros---------------->>>
        String No_cuenta=jTextField1.getText();
        String Nom_paciente=jTextField2.getText();
        String Estado=(String)jComboBox1.getSelectedItem();
        String Tel_paciente=jTextField4.getText();
        String Fo_expediente=jTextField5.getText();
        String Dir_paciente=jTextField6.getText();
        String Password=pass.getText();

        String Sql;
        Sql="INSERT INTO Pacientes (No_cuenta, Password, Nom_paciente, Dir_paciente,
Tel_paciente, Fo_expediente, Estado)values(?,?,?,?,?,?,?)";
        try {
            PreparedStatement g = con.prepareStatement(Sql);
            //g.setString(1,Identificador);
            g.setString(1,No_cuenta);
            g.setString(2,Password);
            g.setString(3,Nom_paciente);
            g.setString(4,Dir_paciente);
            g.setString(5,Tel_paciente);
            g.setString(6,Fo_expediente);
```

```
            g.setString(7,Estado);

            int r=g.executeUpdate();

            if (r>0)
            {
               JOptionPane.showMessageDialog(null, "Registro Exitoso");

            }
        } catch (SQLException ex) {
            Logger.getLogger(Interfaz6u1.class.getName()).log(Level.SEVERE, null, ex);
        }
        limpiar();
        cargar("");
    }

    private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {
        // ------------>>>Elimina los registros------------------>>>
        String No_cuenta= jTextField1.getText();
       String sql="DELETE FROM Pacientes WHERE No_cuenta="+No_cuenta+"";
        try {
            Statement st=con.createStatement();
            int i = st.executeUpdate(sql);
            if(i==1){
            JOptionPane.showMessageDialog(this,
"Eliminado","Aviso",JOptionPane.INFORMATION_MESSAGE);
            }else {
                JOptionPane.showMessageDialog(this,          "Error          al
elminar","Aviso",JOptionPane.INFORMATION_MESSAGE);
            }
```

```
    } catch (SQLException ex) {
        Logger.getLogger(Interfaz6u1.class.getName()).log(Level.SEVERE, null, ex);
    }

  limpiar();
  jTextField7.setText("");
  cargar("");
}

private void jButton6ActionPerformed(java.awt.event.ActionEvent evt) {
    // ---------->>>Regresa al Menú del administrador-------->>>
    Interfaz5 I5=new Interfaz5();
    I5.setVisible(true);
    this.setVisible(false);

}
```

Figura 6.6.7 Módulos Catálogos

```
----------Librerias----------
 */
import java.sql.*;
import java.util.ArrayList;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;

/** ----------Declaración de variables---------
 */
public class Interfaz8u extends javax.swing.JFrame {
    DefaultTableModel model;
    DefaultTableModel model1;
    DefaultTableModel model2;
    Connection con;
    Connection con1;
    Connection con2;
    CallableStatement cst;
    ResultSet r;

    /**
     * ----------Conexión a SQL----------
     */
    public Interfaz8u() {
        initComponents();
        setLocationRelativeTo(null);
       try {
            Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
            try {
                con=DriverManager.getConnection("jdbc:odbc:serio");
```

```
            con1=DriverManager.getConnection("jdbc:odbc:serio");
            con2=DriverManager.getConnection("jdbc:odbc:serio");
          } catch (SQLException ex) {
            Logger.getLogger(Interfaz7.class.getName()).log(Level.SEVERE, null, ex);
          }
        } catch (ClassNotFoundException ex) {
          Logger.getLogger(Interfaz8u.class.getName()).log(Level.SEVERE, null, ex);
        }
          /*Metodos */
        cargar1("");
        limpiar1();
        cargar2("");
        cargar3("");

    }
    /*----------------->>>>> Este metodo limpia los regsitros*/
    void limpiar1()
{
    jTextField1.setText("");
    jTextField2.setText("");
    jTextField3.setText("");
    jTextField4.setText("");
    jTextField5.setText("");
    jTextField6.setText("");
    jTextField7.setText("");
    jTextField8.setText("");
    jTextField9.setText("");
    jTextField11.setText("");

}
    /* ------------>>>Con este metodo Actualiza los registros*/
```

```
void cargar1(String consul)
   {


      String[] titulos ={"Cve.Especialidad","Especialidad"};
      String[] registros=new String[2];

    String sql ="SELECT * FROM  Especialidades where Especialidad LIKE '%"+consul+"%'";

      model= new DefaultTableModel(null,titulos);

            try {

       Statement st = con.createStatement();
       ResultSet rs= st.executeQuery(sql);

        while(rs.next())
        {
         registros[0]=rs.getString("Cve_especialidad");
         registros[1]=rs.getString("Especialidad");

         model.addRow(registros);
         }

         jTable1.setModel(model);

      } catch (SQLException ex) {
         JOptionPane.showMessageDialog(null,ex);
      }
```

```
    }
void cargar2(String consul)
    {

        String[] titulos ={"Cve.Tratamiento","Tratamiento","Costo"};
        String[] registros=new String[3];

      String sql ="SELECT * FROM  TratamientosC where Tipo_tratamiento LIKE '%"+consul+"%'";

        model1= new DefaultTableModel(null,titulos);


          try {

         Statement st = con1.createStatement();
         ResultSet rs= st.executeQuery(sql);

          while(rs.next())
          {
           registros[0]=rs.getString("Cve_tratamiento");
           registros[1]=rs.getString("Tipo_tratamiento");
           registros[2]=rs.getString("Costo");


           model1.addRow(registros);
           }

           jTable2.setModel(model1);

        } catch (SQLException ex) {
```

```
            JOptionPane.showMessageDialog(null,ex);
        }

    }
void cargar3(String consul)
    {


        String[] titulos ={"Cve.Servicio","Servicio"};
        String[] registros=new String[2];

      String sql ="SELECT * FROM  Servicios where Servicio LIKE '%"+consul+"%'";

        model2= new DefaultTableModel(null,titulos);


          try {

         Statement st = con2.createStatement();
         ResultSet rs= st.executeQuery(sql);

          while(rs.next())
          {
           registros[0]=rs.getString("Cve_servicio");
           registros[1]=rs.getString("Servicio");


           model2.addRow(registros);
           }

           jTable3.setModel(model2);
```

```
    } catch (SQLException ex) {
        JOptionPane.showMessageDialog(null,ex);
    }

}

/**
 * This method is called from within the constructor to initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is always
 * regenerated by the Form Editor.
 */
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

    jLabel2 = new javax.swing.JLabel();
    jButton6 = new javax.swing.JButton();
    jLabel1 = new javax.swing.JLabel();
    jTabbedPane1 = new javax.swing.JTabbedPane();
    jPanel1 = new javax.swing.JPanel();
    jScrollPane1 = new javax.swing.JScrollPane();
    jTable1 = new javax.swing.JTable();
    jLabel4 = new javax.swing.JLabel();
    jTextField2 = new javax.swing.JTextField();
    jLabel3 = new javax.swing.JLabel();
    jTextField1 = new javax.swing.JTextField();
    jLabel9 = new javax.swing.JLabel();
    jTextField3 = new javax.swing.JTextField();
    jButton1 = new javax.swing.JButton();
    jButton3 = new javax.swing.JButton();
```

```
jButton4 = new javax.swing.JButton();
jButton5 = new javax.swing.JButton();
jPanel2 = new javax.swing.JPanel();
jTextField6 = new javax.swing.JTextField();
jLabel6 = new javax.swing.JLabel();
jLabel10 = new javax.swing.JLabel();
jTextField5 = new javax.swing.JTextField();
jTextField4 = new javax.swing.JTextField();
jLabel5 = new javax.swing.JLabel();
jLabel7 = new javax.swing.JLabel();
jTextField7 = new javax.swing.JTextField();
jButton8 = new javax.swing.JButton();
jButton9 = new javax.swing.JButton();
jButton10 = new javax.swing.JButton();
jButton11 = new javax.swing.JButton();
jScrollPane2 = new javax.swing.JScrollPane();
jTable2 = new javax.swing.JTable();
jPanel3 = new javax.swing.JPanel();
jLabel8 = new javax.swing.JLabel();
jTextField8 = new javax.swing.JTextField();
jLabel11 = new javax.swing.JLabel();
jTextField9 = new javax.swing.JTextField();
jLabel13 = new javax.swing.JLabel();
jTextField11 = new javax.swing.JTextField();
jButton13 = new javax.swing.JButton();
jButton14 = new javax.swing.JButton();
jButton15 = new javax.swing.JButton();
jButton16 = new javax.swing.JButton();
jScrollPane3 = new javax.swing.JScrollPane();
jTable3 = new javax.swing.JTable();
```

```
        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
        getContentPane().setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

        jLabel2.setBackground(new java.awt.Color(0, 0, 0));
        jLabel2.setFont(new java.awt.Font("Lucida Bright", 1, 48)); // NOI18N
        jLabel2.setText("    CATALOGOS");
        getContentPane().add(jLabel2, new
org.netbeans.lib.awtextra.AbsoluteConstraints(350, 150, 430, 50));

        jButton6.setFont(new java.awt.Font("Lucida Bright", 1, 18)); // NOI18N
        jButton6.setText("Salir");
        jButton6.setToolTipText("Da clic para salir");
        jButton6.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton6ActionPerformed(evt);
            }
        });
        getContentPane().add(jButton6, new
org.netbeans.lib.awtextra.AbsoluteConstraints(970, 560, 150, -1));

        jLabel1.setForeground(new java.awt.Color(0, 153, 255));
        jLabel1.setIcon(new
javax.swing.ImageIcon("C:\\Users\\Erick\\Documents\\NetBeansProjects\\Serio\\Imagenes\
\fondol.jpg")); // NOI18N
        getContentPane().add(jLabel1, new
org.netbeans.lib.awtextra.AbsoluteConstraints(100, 0, 970, 150));

        jPanel1.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

        jTable1.setAutoCreateRowSorter(true);
        jTable1.setBackground(new java.awt.Color(204, 204, 204));
```

```
jTable1.setBorder(new javax.swing.border.MatteBorder(null));
jTable1.setFont(new java.awt.Font("Lucida Bright", 0, 12)); // NOI18N
jTable1.setModel(new javax.swing.table.DefaultTableModel(
   new Object [][] {
      {null, null},
      {null, null},
      {null, null},
      {null, null}
   },
   new String [] {
      "Cve.Especialidad", "Especialidad"
   }
));
jTable1.addMouseListener(new java.awt.event.MouseAdapter() {
   public void mouseClicked(java.awt.event.MouseEvent evt) {
      jTable1MouseClicked(evt);
   }
});
jScrollPane1.setViewportView(jTable1);

jPanel1.add(jScrollPane1, new org.netbeans.lib.awtextra.AbsoluteConstraints(10, 130,
890, 100));

jLabel4.setFont(new java.awt.Font("Lucida Bright", 1, 18)); // NOI18N
jLabel4.setText("Buscar:");
jPanel1.add(jLabel4, new org.netbeans.lib.awtextra.AbsoluteConstraints(90, 30, 70,
30));

jTextField2.setFont(new java.awt.Font("Lucida Bright", 0, 14)); // NOI18N
jTextField2.addKeyListener(new java.awt.event.KeyAdapter() {
   public void keyReleased(java.awt.event.KeyEvent evt) {
```

```
            jTextField2KeyReleased(evt);
        }
    });
    jPanel1.add(jTextField2, new org.netbeans.lib.awtextra.AbsoluteConstraints(170, 30, 440, -1));

    jLabel3.setFont(new java.awt.Font("Lucida Bright", 1, 18)); // NOI18N
    jLabel3.setText("Cve.Especilidad:");
    jPanel1.add(jLabel3, new org.netbeans.lib.awtextra.AbsoluteConstraints(10, 70, 190, 30));

    jTextField1.setFont(new java.awt.Font("Lucida Bright", 0, 14)); // NOI18N
    jPanel1.add(jTextField1, new org.netbeans.lib.awtextra.AbsoluteConstraints(170, 70, 120, -1));

    jLabel9.setFont(new java.awt.Font("Lucida Bright", 1, 18)); // NOI18N
    jLabel9.setText("Especialidad:");
    jPanel1.add(jLabel9, new org.netbeans.lib.awtextra.AbsoluteConstraints(300, 80, -1, -1));

    jTextField3.setFont(new java.awt.Font("Lucida Bright", 0, 14)); // NOI18N
    jTextField3.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            jTextField3ActionPerformed(evt);
        }
    });
    jPanel1.add(jTextField3, new org.netbeans.lib.awtextra.AbsoluteConstraints(430, 70, 560, -1));

    jButton1.setFont(new java.awt.Font("Lucida Bright", 1, 18)); // NOI18N
    jButton1.setText("Guardar");
```

```
        jButton1.setToolTipText("De clic para guardar un nvo registro");
        jButton1.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton1ActionPerformed(evt);
            }
        });
        jPanel1.add(jButton1, new org.netbeans.lib.awtextra.AbsoluteConstraints(930, 110, 150, 30));

        jButton3.setFont(new java.awt.Font("Lucida Bright", 1, 18)); // NOI18N
        jButton3.setText("Modificar");
        jButton3.setToolTipText("Da clic para modificar un registro");
        jButton3.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton3ActionPerformed(evt);
            }
        });
        jPanel1.add(jButton3, new org.netbeans.lib.awtextra.AbsoluteConstraints(930, 190, 150, 30));

        jButton4.setFont(new java.awt.Font("Lucida Bright", 1, 18)); // NOI18N
        jButton4.setText("Eliminar");
        jButton4.setToolTipText("Da clic para eliminar un registro");
        jButton4.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton4ActionPerformed(evt);
            }
        });
        jPanel1.add(jButton4, new org.netbeans.lib.awtextra.AbsoluteConstraints(930, 230, 150, 30));
```

```
        jButton5.setFont(new java.awt.Font("Lucida Bright", 1, 18)); // NOI18N
        jButton5.setText("Nuevo");
        jButton5.setToolTipText("De clic para limpiar los registros");
        jButton5.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton5ActionPerformed(evt);
            }
        });
        jPanel1.add(jButton5, new org.netbeans.lib.awtextra.AbsoluteConstraints(930, 150, 150, 30));

        jTabbedPane1.addTab("Especialidades", jPanel1);

        jPanel2.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

        jTextField6.setFont(new java.awt.Font("Lucida Bright", 0, 14)); // NOI18N
        jTextField6.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jTextField6ActionPerformed(evt);
            }
        });
        jPanel2.add(jTextField6, new org.netbeans.lib.awtextra.AbsoluteConstraints(430, 70, 560, -1));

        jLabel6.setFont(new java.awt.Font("Lucida Bright", 1, 18)); // NOI18N
        jLabel6.setText("Cve.Tratamiento:");
        jPanel2.add(jLabel6, new org.netbeans.lib.awtextra.AbsoluteConstraints(10, 70, 190, 30));

        jLabel10.setFont(new java.awt.Font("Lucida Bright", 1, 18)); // NOI18N
        jLabel10.setText("Tratamiento:");
```

```
jPanel2.add(jLabel10, new org.netbeans.lib.awtextra.AbsoluteConstraints(300, 80, -1, -1));

jTextField5.setFont(new java.awt.Font("Lucida Bright", 0, 14)); // NOI18N
jPanel2.add(jTextField5, new org.netbeans.lib.awtextra.AbsoluteConstraints(170, 70, 120, -1));

jTextField4.setFont(new java.awt.Font("Lucida Bright", 0, 14)); // NOI18N
jTextField4.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyReleased(java.awt.event.KeyEvent evt) {
        jTextField4KeyReleased(evt);
    }
});
jPanel2.add(jTextField4, new org.netbeans.lib.awtextra.AbsoluteConstraints(170, 30, 440, -1));

jLabel5.setFont(new java.awt.Font("Lucida Bright", 1, 18)); // NOI18N
jLabel5.setText("Buscar:");
jPanel2.add(jLabel5, new org.netbeans.lib.awtextra.AbsoluteConstraints(100, 30, 70, 30));

jLabel7.setFont(new java.awt.Font("Lucida Bright", 1, 18)); // NOI18N
jLabel7.setText("Costo:");
jPanel2.add(jLabel7, new org.netbeans.lib.awtextra.AbsoluteConstraints(110, 110, 60, 30));

jTextField7.setFont(new java.awt.Font("Lucida Bright", 0, 14)); // NOI18N
jPanel2.add(jTextField7, new org.netbeans.lib.awtextra.AbsoluteConstraints(170, 110, 120, -1));

jButton8.setFont(new java.awt.Font("Lucida Bright", 1, 18)); // NOI18N
```

```
        jButton8.setText("Guardar");
        jButton8.setToolTipText("De clic para guardar un nvo registro");
        jButton8.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton8ActionPerformed(evt);
            }
        });
        jPanel2.add(jButton8, new org.netbeans.lib.awtextra.AbsoluteConstraints(940, 140, 150, 30));

        jButton9.setFont(new java.awt.Font("Lucida Bright", 1, 18)); // NOI18N
        jButton9.setText("Modificar");
        jButton9.setToolTipText("Da clic para modificar un registro");
        jButton9.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton9ActionPerformed(evt);
            }
        });
        jPanel2.add(jButton9, new org.netbeans.lib.awtextra.AbsoluteConstraints(940, 220, 150, 30));

        jButton10.setFont(new java.awt.Font("Lucida Bright", 1, 18)); // NOI18N
        jButton10.setText("Eliminar");
        jButton10.setToolTipText("Da clic para eliminar un registro");
        jButton10.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton10ActionPerformed(evt);
            }
        });
        jPanel2.add(jButton10, new org.netbeans.lib.awtextra.AbsoluteConstraints(940, 260, 150, 30));
```

```
jButton11.setFont(new java.awt.Font("Lucida Bright", 1, 18)); // NOI18N
jButton11.setText("Nuevo");
jButton11.setToolTipText("De clic para limpiar los registros");
jButton11.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton11ActionPerformed(evt);
    }
});
jPanel2.add(jButton11, new org.netbeans.lib.awtextra.AbsoluteConstraints(940, 180, 150, 30));

jTable2.setAutoCreateRowSorter(true);
jTable2.setBackground(new java.awt.Color(204, 204, 204));
jTable2.setBorder(new javax.swing.border.MatteBorder(null));
jTable2.setFont(new java.awt.Font("Lucida Bright", 0, 12)); // NOI18N
jTable2.setModel(new javax.swing.table.DefaultTableModel(
    new Object [][] {
        {null, null, null},
        {null, null, null},
        {null, null, null},
        {null, null, null}
    },
    new String [] {
        "Cve.Tratamiento", "Tratamiento", "Costo"
    }
));
jTable2.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        jTable2MouseClicked(evt);
    }
```

```
        });
        jScrollPane2.setViewportView(jTable2);
        jTable2.getColumnModel().getColumn(2).setHeaderValue("Costo");

        jPanel2.add(jScrollPane2, new org.netbeans.lib.awtextra.AbsoluteConstraints(10, 150,
920, 150));

        jTabbedPane1.addTab("Tratamientos Costos", jPanel2);

        jPanel3.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

        jLabel8.setFont(new java.awt.Font("Lucida Bright", 1, 18)); // NOI18N
        jLabel8.setText("Buscar:");
        jPanel3.add(jLabel8, new org.netbeans.lib.awtextra.AbsoluteConstraints(60, 30, 70,
30));

        jTextField8.setFont(new java.awt.Font("Lucida Bright", 0, 14)); // NOI18N
        jTextField8.addKeyListener(new java.awt.event.KeyAdapter() {
            public void keyReleased(java.awt.event.KeyEvent evt) {
                jTextField8KeyReleased(evt);
            }
        });
        jPanel3.add(jTextField8, new org.netbeans.lib.awtextra.AbsoluteConstraints(130, 30,
440, -1));

        jLabel11.setFont(new java.awt.Font("Lucida Bright", 1, 18)); // NOI18N
        jLabel11.setText("Cve.Servicio:");
        jPanel3.add(jLabel11, new org.netbeans.lib.awtextra.AbsoluteConstraints(10, 70, 130,
30));

        jTextField9.setFont(new java.awt.Font("Lucida Bright", 0, 14)); // NOI18N
```

```
        jPanel3.add(jTextField9, new org.netbeans.lib.awtextra.AbsoluteConstraints(140, 70, 120, -1));

        jLabel13.setFont(new java.awt.Font("Lucida Bright", 1, 18)); // NOI18N
        jLabel13.setText("Servicio:");
        jPanel3.add(jLabel13, new org.netbeans.lib.awtextra.AbsoluteConstraints(270, 70, -1, -1));

        jTextField11.setFont(new java.awt.Font("Lucida Bright", 0, 14)); // NOI18N
        jTextField11.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jTextField11ActionPerformed(evt);
            }
        });
        jPanel3.add(jTextField11, new org.netbeans.lib.awtextra.AbsoluteConstraints(360, 70, 560, -1));

        jButton13.setFont(new java.awt.Font("Lucida Bright", 1, 18)); // NOI18N
        jButton13.setText("Guardar");
        jButton13.setToolTipText("De clic para guardar un nvo registro");
        jButton13.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton13ActionPerformed(evt);
            }
        });
        jPanel3.add(jButton13, new org.netbeans.lib.awtextra.AbsoluteConstraints(910, 110, 150, 30));

        jButton14.setFont(new java.awt.Font("Lucida Bright", 1, 18)); // NOI18N
        jButton14.setText("Modificar");
        jButton14.setToolTipText("Da clic para modificar un registro");
```

```
        jButton14.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton14ActionPerformed(evt);
            }
        });
        jPanel3.add(jButton14, new org.netbeans.lib.awtextra.AbsoluteConstraints(910, 190, 150, 30));

        jButton15.setFont(new java.awt.Font("Lucida Bright", 1, 18)); // NOI18N
        jButton15.setText("Eliminar");
        jButton15.setToolTipText("Da clic para eliminar un registro");
        jButton15.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton15ActionPerformed(evt);
            }
        });
        jPanel3.add(jButton15, new org.netbeans.lib.awtextra.AbsoluteConstraints(910, 230, 150, 30));

        jButton16.setFont(new java.awt.Font("Lucida Bright", 1, 18)); // NOI18N
        jButton16.setText("Nuevo");
        jButton16.setToolTipText("De clic para limpiar los registros");
        jButton16.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton16ActionPerformed(evt);
            }
        });
        jPanel3.add(jButton16, new org.netbeans.lib.awtextra.AbsoluteConstraints(910, 150, 150, 30));

        jTable3.setAutoCreateRowSorter(true);
```

```
        jTable3.setBackground(new java.awt.Color(204, 204, 204));
        jTable3.setBorder(new javax.swing.border.MatteBorder(null));
        jTable3.setFont(new java.awt.Font("Lucida Bright", 0, 12)); // NOI18N
        jTable3.setModel(new javax.swing.table.DefaultTableModel(
            new Object [][] {
                {null, null},
                {null, null},
                {null, null},
                {null, null}
            },
            new String [] {
                "Cve.Servicio", "Servicio"
            }
        ));
        jTable3.addMouseListener(new java.awt.event.MouseAdapter() {
            public void mouseClicked(java.awt.event.MouseEvent evt) {
                jTable3MouseClicked(evt);
            }
        });
        jScrollPane3.setViewportView(jTable3);

        jPanel3.add(jScrollPane3, new org.netbeans.lib.awtextra.AbsoluteConstraints(30, 120,
860, 100));

        jTabbedPane1.addTab("Servicios", jPanel3);

        getContentPane().add(jTabbedPane1,                                    new
org.netbeans.lib.awtextra.AbsoluteConstraints(30, 200, 1100, 350));

        pack();
    }// </editor-fold>
```

```
    private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
        // ---------------->>>Modifica los registros-------->>>:
        String Cve_especialidad=jTextField1.getText();
        String Especialidad=jTextField3.getText();

String Sql;
Sql="UPDATE Especialidades SET Cve_especialidad=?,Especialidad=? WHERE Cve_especialidad="+Cve_especialidad+"";
        try {
            PreparedStatement g = con.prepareStatement(Sql);
            //g.setString(1,No_cuenta);
            g.setString(1,Cve_especialidad);
            g.setString(2,Especialidad);

            int r=g.executeUpdate();

            if (r>0)
            {
                JOptionPane.showMessageDialog(null, "Registro Exitoso");

            }
        } catch (SQLException ex) {
            Logger.getLogger(Interfaz6u1.class.getName()).log(Level.SEVERE, null, ex);
        }
        limpiar1();
        cargar1("");

       }

    private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {
```

```
        // ----------------->>>Limpia los registros------>>>:
        limpiar1();
    }

    private void jButton9ActionPerformed(java.awt.event.ActionEvent evt) {
        //---------------->>>Modifica los registros---------------------->>>:
        String Cve_tratamiento=jTextField5.getText();
        String Tipo_tratamiento=jTextField6.getText();
        String Costo=jTextField7.getText();

String Sql;
Sql="UPDATE TratamientosC SET Cve_tratamiento=?,Tipo_tratamiento=?, Costo=?
WHERE Cve_tratamiento="+Cve_tratamiento+"";
        try {
            PreparedStatement g = con1.prepareStatement(Sql);
            //g.setString(1,No_cuenta);
            g.setString(1,Cve_tratamiento);
            g.setString(2,Tipo_tratamiento);
            g.setString(3,Costo);

            int r=g.executeUpdate();

            if (r>0)
            {
                JOptionPane.showMessageDialog(null, "Registro Exitoso");

            }
        } catch (SQLException ex) {
            Logger.getLogger(Interfaz6u1.class.getName()).log(Level.SEVERE, null, ex);
        }
        limpiar1();
```

```
        cargar2("");
    }

    private void jButton11ActionPerformed(java.awt.event.ActionEvent evt) {
        // ---------->>>Limpia los regristros-------->>>
        limpiar1();
    }

    private void jButton6ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        Interfaz5 I5 =new Interfaz5();
        I5.setVisible(true);
        this.setVisible(false);
    }

    private void jTextField11ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
    }

    private void jButton14ActionPerformed(java.awt.event.ActionEvent evt) {
        // ---------->>>Modifica los registros---------->>>
        String Cve_servicio=jTextField9.getText();
        String Servicio=jTextField11.getText();

String Sql;
Sql="UPDATE Servicios SET Cve_servicio=?,Servicio=? WHERE
Cve_servicio="+Cve_servicio+"";
        try {
            PreparedStatement g = con2.prepareStatement(Sql);
            //g.setString(1,No_cuenta);
            g.setString(1,Cve_servicio);
```

```
        g.setString(2,Servicio);

        int r=g.executeUpdate();

        if (r>0)
        {
           JOptionPane.showMessageDialog(null, "Registro Exitoso");

        }
    } catch (SQLException ex) {
        Logger.getLogger(Interfaz6u1.class.getName()).log(Level.SEVERE, null, ex);
    }
    limpiar1();
    cargar3("");
}

private void jButton16ActionPerformed(java.awt.event.ActionEvent evt) {
    // ----------------->>>Limpia los registros---------------------->>>
    limpiar1();
}

private void jTextField2KeyReleased(java.awt.event.KeyEvent evt) {
    // --------->>>Realiza una busqueda indexada---------------->>>:
    cargar1(jTextField2.getText());
}

private void jTable1MouseClicked(java.awt.event.MouseEvent evt) {
    // -->>>Carga los datos en los campos de acuerdo a la busqueda que realizo--->>>:
    try {
        Statement st = con.createStatement();
```

```
            int row =jTable1.getSelectedRow();
            String TableC=(jTable1.getModel().getValueAt(row,0).toString());
            String sql ="SELECT * FROM  Especialidades where Cve_especialidad= '"+TableC+"'";
             ResultSet rs= st.executeQuery(sql);
             if(rs.next()){
                String T1 = rs.getString("Cve_especialidad");
                jTextField1.setText(T1);
                String T2 = rs.getString("Especialidad");
                jTextField3.setText(T2);

             }

        } catch (Exception ex) {
            JOptionPane.showMessageDialog(null,ex);


              }
    }

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        // ------------------>>>Guarda los registros---------------->>>:
        String Cve_especialidad=jTextField1.getText();
        String Especialidad=jTextField3.getText();

String Sql;
Sql="INSERT INTO Especialidades (Cve_especialidad, Especialidad)values(?,?)";
        try {
            PreparedStatement g = con.prepareStatement(Sql);
            //g.setString(1,Identificador);
            g.setString(1,Cve_especialidad);
```

```
            g.setString(2,Especialidad);

            int r=g.executeUpdate();

            if (r>0)
            {
               JOptionPane.showMessageDialog(null, "Registro Exitoso");

            }
        } catch (SQLException ex) {
            Logger.getLogger(Interfaz6u1.class.getName()).log(Level.SEVERE, null, ex);
        }
        limpiar1();
        cargar1("");
    }

    private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {
        // ------------------>>>Elimina los regristros----------------->>>:
        String Cve_especialidad= jTextField1.getText();
       String sql="DELETE FROM Especialidades WHERE
Cve_especialidad="+Cve_especialidad+"";
        try {
            Statement st=con.createStatement();
            int i = st.executeUpdate(sql);
            if(i==1){
            JOptionPane.showMessageDialog(this,
"Eliminado","Aviso",JOptionPane.INFORMATION_MESSAGE);
            }else {
                JOptionPane.showMessageDialog(this, "Error al
elminar","Aviso",JOptionPane.INFORMATION_MESSAGE);
            }
```

```
        } catch (SQLException ex) {
            Logger.getLogger(Interfaz6u1.class.getName()).log(Level.SEVERE, null, ex);
        }

      limpiar1();
      jTextField2.setText("");
      cargar1("");
    }

    private void jTextField4KeyReleased(java.awt.event.KeyEvent evt) {
        // Realiza una busqueda indexada
        cargar2(jTextField4.getText());
    }

    private void jTable2MouseClicked(java.awt.event.MouseEvent evt) {
        // ------->>> Carga los registros en los campos----->>>
        try {
            Statement st = con1.createStatement();

            int row =jTable2.getSelectedRow();
            String TableC=(jTable2.getModel().getValueAt(row,0).toString());
            String sql ="SELECT * FROM  TratamientosC where Cve_tratamiento=
'"+TableC+"'";
            ResultSet rs= st.executeQuery(sql);
            if(rs.next()){
                String T1 = rs.getString("Cve_tratamiento");
                jTextField5.setText(T1);
                String T2 = rs.getString("Tipo_tratamiento");
                jTextField6.setText(T2);
                String T3 = rs.getString("Costo");
```

```
            jTextField7.setText(T3);

          }

      } catch (Exception ex) {
         JOptionPane.showMessageDialog(null,ex);

   }
   }

   private void jButton8ActionPerformed(java.awt.event.ActionEvent evt) {
      // ---------->>>Guarda registros---------->>>:
      String Cve_tratamiento=jTextField5.getText();
      String Tipo_tratamiento=jTextField6.getText();
      String Costo=jTextField7.getText();

String Sql;
Sql="INSERT INTO TratamientosC (Cve_tratamiento, Tipo_tratamiento,
Costo)values(?,?,?)";
      try {
         PreparedStatement g = con1.prepareStatement(Sql);
         //g.setString(1,Identificador);
         g.setString(1,Cve_tratamiento);
         g.setString(2,Tipo_tratamiento);
         g.setString(3,Costo);

         int r=g.executeUpdate();

         if (r>0)
         {
            JOptionPane.showMessageDialog(null, "Registro Exitoso");
```

```
            }
        } catch (SQLException ex) {
            Logger.getLogger(Interfaz6u1.class.getName()).log(Level.SEVERE, null, ex);
        }
        limpiar1();
        cargar2("");

    }

    private void jButton10ActionPerformed(java.awt.event.ActionEvent evt) {
        // ------------>>>Elimina los registros----------->>>
        String Cve_tratamiento= jTextField5.getText();
       String sql="DELETE FROM TratamientosC WHERE Cve_tratamiento="+Cve_tratamiento+"";
        try {
            Statement st=con1.createStatement();
            int i = st.executeUpdate(sql);
            if(i==1){
            JOptionPane.showMessageDialog(this, "Eliminado","Aviso",JOptionPane.INFORMATION_MESSAGE);
            }else {
                JOptionPane.showMessageDialog(this, "Error al elminar","Aviso",JOptionPane.INFORMATION_MESSAGE);
            }

        } catch (SQLException ex) {
            Logger.getLogger(Interfaz6u1.class.getName()).log(Level.SEVERE, null, ex);
        }

       limpiar1();
```

```
  jTextField4.setText("");
  cargar2("");
}

private void jTextField8KeyReleased(java.awt.event.KeyEvent evt) {
   // ----------->>>Realiza una busqueda indexada----->>>
   cargar3(jTextField8.getText());
}

private void jTable3MouseClicked(java.awt.event.MouseEvent evt) {
   // ---------->>>Carga los registros en los campos---->>>
    try {
        Statement st = con2.createStatement();

       int row =jTable3.getSelectedRow();
       String TableC=(jTable3.getModel().getValueAt(row,0).toString());
       String sql ="SELECT * FROM  Servicios where Cve_servicio= '"+TableC+"'";
        ResultSet rs= st.executeQuery(sql);
        if(rs.next()){
            String T1 = rs.getString("Cve_servicio");
            jTextField9.setText(T1);
            String T2 = rs.getString("Servicio");
            jTextField11.setText(T2);

        }

   } catch (Exception ex) {
      JOptionPane.showMessageDialog(null,ex);


         }
```

```
    }

    private void jButton13ActionPerformed(java.awt.event.ActionEvent evt) {
        // ---------->>>Guarda los registros---------->>>
        String Cve_servicio=jTextField9.getText();
        String Servicio=jTextField11.getText();

String Sql;
Sql="INSERT INTO Servicios (Cve_servicio, Servicio)values(?,?)";
        try {
            PreparedStatement g = con2.prepareStatement(Sql);
            //g.setString(1,Identificador);
            g.setString(1,Cve_servicio);
            g.setString(2,Servicio);

            int r=g.executeUpdate();

            if (r>0)
            {
                JOptionPane.showMessageDialog(null, "Registro Exitoso");

            }
        } catch (SQLException ex) {
            Logger.getLogger(Interfaz6u1.class.getName()).log(Level.SEVERE, null, ex);
        }
        limpiar1();
        cargar3("");

    }

    private void jButton15ActionPerformed(java.awt.event.ActionEvent evt) {
```

```
        // ------------------>>>Elimina los registros------------------>>>
        String Cve_servicio= jTextField9.getText();
       String sql="DELETE FROM Servicios WHERE Cve_servicio="+Cve_servicio+"";
        try {
            Statement st=con2.createStatement();
            int i = st.executeUpdate(sql);
            if(i==1){
            JOptionPane.showMessageDialog(this,
"Eliminado","Aviso",JOptionPane.INFORMATION_MESSAGE);
            }else {
                JOptionPane.showMessageDialog(this,                "Error                al
elminar","Aviso",JOptionPane.INFORMATION_MESSAGE);
            }

        } catch (SQLException ex) {
            Logger.getLogger(Interfaz6u1.class.getName()).log(Level.SEVERE, null, ex);
        }

       limpiar1();
       jTextField8.setText("");
       cargar3("");
    }
```

Figura 6.6.8 Modulo Citas programadas

```
*------------>>>Librerias--------------->>>
 */
import java.sql.*;
import java.util.ArrayList;
import java.util.Calendar;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;
/**
------------>>>Declaración de variables-------------->>>
 */
public class Interfaz11 extends javax.swing.JFrame {
    DefaultTableModel model;
    DefaultTableModel model1;
    DefaultTableModel model2;
    DefaultTableModel model3;
```

```
DefaultTableModel model4;
Connection con;
Connection con5;
Connection con2;
Connection con3;
Connection con4;

CallableStatement cst;
ResultSet r;
/**
 --------->>>Conexión a SQL-------->>>
 */
public Interfaz11() {
    initComponents();
   setLocationRelativeTo(null);
  try {
      Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
      try {
         con=DriverManager.getConnection("jdbc:odbc:serio");
         con5=DriverManager.getConnection("jdbc:odbc:serio");
         con2=DriverManager.getConnection("jdbc:odbc:serio");
         con3=DriverManager.getConnection("jdbc:odbc:serio");
         con4=DriverManager.getConnection("jdbc:odbc:serio");
      } catch (SQLException ex) {
         Logger.getLogger(Interfaz7.class.getName()).log(Level.SEVERE, null, ex);
      }
   } catch (ClassNotFoundException ex) {
      Logger.getLogger(Interfaz7.class.getName()).log(Level.SEVERE, null, ex);
   }
   cargar("");
   cargar1("");
```

```
        cargar2("");
        cargar3("");
        cargar4("");
        limpiar1();
        Calendar cal=Calendar.getInstance();
 String fecha=cal.get(cal.YEAR)+"-"+"11"+"-"+cal.get(cal.DATE);
 String
hora=cal.get(cal.HOUR_OF_DAY)+":"+cal.get(cal.MINUTE)+":"+cal.get(cal.SECOND);
jTextField5.setText(hora);
jTextField2.setText(fecha);
    }
void limpiar1()
{
   jTextField1.setText("");
   jTextField2.setText("");
   jTextField3.setText("");
   jTextField4.setText("");
   jTextField5.setText("");
   jTextField6.setText("");
   jTextField7.setText("");
   jTextField8.setText("");
   jTextField9.setText("");
   jTextField10.setText("");
   jTextField11.setText("");
   jTextField12.setText("");
   jTextField13.setText("");
   jTextField14.setText("");
   jTextField15.setText("");

     }
/*---------->>>Actualiza los registros despues de cada operación------>>>*/
```

```
void cargar(String consul)
   {

      String[] titulos ={"No.Cuenta","Nombre Paciente","Fecha Cita","Proxima Cita","Hora","Asistencia","No.Consultorio","Nombre Medico","Especialidad","Servicio"};
      String[] registros=new String[10];

     String sql ="SELECT * FROM  Citas where No_cuenta LIKE '%"+consul+"%'";

      model= new DefaultTableModel(null,titulos);

        try {

       Statement st = con.createStatement();
       ResultSet rs= st.executeQuery(sql);

        while(rs.next())
        {
         registros[0]=rs.getString("No_cuenta");
         registros[1]=rs.getString("Nom_paciente");
         registros[2]=rs.getString("Fecha_cita");
         registros[3]=rs.getString("Proxima_cita");
         registros[4]=rs.getString("Hora");
         registros[5]=rs.getString("Asistencia");
         registros[6]=rs.getString("No_consultorio");
         registros[7]=rs.getString("Nom_medico");
         registros[8]=rs.getString("Especialidad");
         registros[9]=rs.getString("Servicio");

         model.addRow(registros);
         }
```

```
        jTable1.setModel(model);

      } catch (SQLException ex) {
         JOptionPane.showMessageDialog(null,ex);
      }

   }
 void cargar1(String consul)
   {


      String[] titulos ={"No.Cuenta","Nombre Paciente"};
      String[] registros=new String[2];

    String sql ="SELECT * FROM  Pacientes where Nom_paciente LIKE '%"+consul+"%'";

      model1= new DefaultTableModel(null,titulos);



        try {

       Statement st = con5.createStatement();
       ResultSet rs= st.executeQuery(sql);

        while(rs.next())
        {
         registros[0]=rs.getString("No_cuenta");
         registros[1]=rs.getString("Nom_paciente");
```

```
        model1.addRow(registros);
        }

        jTable5.setModel(model1);

    } catch (SQLException ex) {
        JOptionPane.showMessageDialog(null,ex);
    }

 }
void cargar2(String consul)
 {


    String[] titulos ={"Nombre Medico"};
    String[] registros=new String[2];

   String sql ="SELECT * FROM  Medicos where Nom_medico LIKE '%"+consul+"%'";

    model2= new DefaultTableModel(null,titulos);

      try {

      Statement st = con2.createStatement();
      ResultSet rs= st.executeQuery(sql);

       while(rs.next())
       {
        registros[0]=rs.getString("Nom_medico");
```

```
        model2.addRow(registros);
        }

        jTable2.setModel(model2);

      } catch (SQLException ex) {
        JOptionPane.showMessageDialog(null,ex);
      }

  }
  void cargar3(String consul)
  {

      String[] titulos ={"Especialidad"};
      String[] registros=new String[1];

    String sql ="SELECT * FROM  Especialidades where Especialidad LIKE '%"+consul+"%'";

      model3= new DefaultTableModel(null,titulos);

        try {

       Statement st = con3.createStatement();
       ResultSet rs= st.executeQuery(sql);

        while(rs.next())
        {
         registros[0]=rs.getString("Especialidad");
```

```
        model3.addRow(registros);
        }

        jTable3.setModel(model3);

    } catch (SQLException ex) {
        JOptionPane.showMessageDialog(null,ex);
    }

}
void cargar4(String consul)
{

    String[] titulos ={"Servicio"};
    String[] registros=new String[1];

  String sql ="SELECT * FROM  Servicios where Servicio LIKE '%"+consul+"%'";

    model4= new DefaultTableModel(null,titulos);

      try {

     Statement st = con4.createStatement();
     ResultSet rs= st.executeQuery(sql);

      while(rs.next())
      {
       registros[0]=rs.getString("Servicio");
```

```
            model4.addRow(registros);
            }

            jTable4.setModel(model4);

        } catch (SQLException ex) {
            JOptionPane.showMessageDialog(null,ex);
        }

    }
private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
        // ---------->>>Modifica los regristros---------->>>
        String No_cuenta=jTextField1.getText();
        String Nom_paciente=jTextField3.getText();
        String Fecha_cita=jTextField2.getText();
        String Proxima_cita=jTextField6.getText();
        String Hora=jTextField5.getText();
        String Asistencia=jTextField4.getText();
        String No_consultorio=jTextField7.getText();
        String Nom_medico=jTextField8.getText();
        String Especialidad=jTextField9.getText();
        String Servicio=jTextField10.getText();

String Sql;
Sql="UPDATE Citas SET No_cuenta=?,Nom_paciente=?, Fecha_cita=?, Proxima_cita=?,
Hora=?, Asistencia=?, No_consultorio=?, Nom_medico=?, Especialidad=?, Servicio=?
WHERE No_cuenta="+No_cuenta+"";
        try {
            PreparedStatement g = con.prepareStatement(Sql);
            //g.setString(1,No_cuenta);
```

```
        g.setString(1,No_cuenta);
        g.setString(2,Nom_paciente);
        g.setString(3,Fecha_cita);
        g.setString(4,Proxima_cita);
        g.setString(5,Hora);
        g.setString(6,Asistencia);
        g.setString(7,No_consultorio);
        g.setString(8,Nom_medico);
        g.setString(9,Especialidad);
        g.setString(10,Servicio);


        int r=g.executeUpdate();

        if (r>0)
        {
            JOptionPane.showMessageDialog(null, "Registro Exitoso");

        }
    } catch (SQLException ex) {
        Logger.getLogger(Interfaz6u1.class.getName()).log(Level.SEVERE, null, ex);
    }
    limpiar1();
    cargar("");
}

private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {
    // ---------->>>Limpia los registros---------->>>
    limpiar1();
}
```

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    // ---------->>>Guarda los registros---------->>>
    String No_cuenta=jTextField1.getText();
    String Nom_paciente=jTextField3.getText();
    String Fecha_cita=jTextField2.getText();
    String Proxima_cita=jTextField6.getText();
    String Hora=jTextField5.getText();
    String Asistencia=jTextField4.getText();
    String No_consultorio=jTextField7.getText();
    String Nom_medico=jTextField8.getText();
    String Especialidad=jTextField9.getText();
    String Servicio=jTextField10.getText();


String Sql;
Sql="INSERT INTO Citas (No_cuenta, Nom_paciente,Fecha_cita, Proxima_cita, Hora,
Asistencia, No_consultorio, Nom_medico, Especialidad,
Servicio)values(?,?,?,?,?,?,?,?,?,?)";
    try {
        PreparedStatement g = con.prepareStatement(Sql);
        //g.setString(1,Identificador);
        g.setString(1,No_cuenta);
        g.setString(2,Nom_paciente);
        g.setString(3,Fecha_cita);
        g.setString(4,Proxima_cita);
        g.setString(5,Hora);
        g.setString(6,Asistencia);
        g.setString(7,No_consultorio);
        g.setString(8,Nom_medico);
        g.setString(9,Especialidad);
        g.setString(10,Servicio);
```

```
            int r=g.executeUpdate();

            if (r>0)
            {
               JOptionPane.showMessageDialog(null, "Registro Exitoso");

            }
        } catch (SQLException ex) {
            Logger.getLogger(Interfaz11.class.getName()).log(Level.SEVERE, null, ex);
        }
        limpiar1();
        cargar("");
    }
private void jTable1MouseClicked(java.awt.event.MouseEvent evt) {
        // ---------------->>>Carga los registros en los campos de la interfaz--->>>
        try {
             Statement st = con.createStatement();

            int row =jTable1.getSelectedRow();
            String TableC=(jTable1.getModel().getValueAt(row,0).toString());
            String sql ="SELECT * FROM  Citas where No_cuenta= '"+TableC+"'";
             ResultSet rs= st.executeQuery(sql);
             if(rs.next()){
                String T1 = rs.getString("No_cuenta");
                jTextField1.setText(T1);
                String T2 = rs.getString("Nom_paciente");
                jTextField3.setText(T2);
                String T3 = rs.getString("Fecha_cita");
                jTextField2.setText(T3);
```

```
            String T4 = rs.getString("Proxima_cita");
            jTextField6.setText(T4);
            String T5 = rs.getString("Hora");
            jTextField5.setText(T5);
            String T6 = rs.getString("Asistencia");
            jTextField4.setText(T6);
            String T7 = rs.getString("No_consultorio");
            jTextField7.setText(T7);
            String T8 = rs.getString("Nom_medico");
            jTextField8.setText(T8);
            String T9 = rs.getString("Especialidad");
            jTextField9.setText(T9);
            String T10 = rs.getString("Servicio");
            jTextField10.setText(T10);

         }

    } catch (Exception ex) {
       JOptionPane.showMessageDialog(null,ex);

}
}

private void jTable5MouseClicked(java.awt.event.MouseEvent evt) {
    // ------------>>>Carga los registros en los campos------>>>
     try {
       Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
       try {
          con5=DriverManager.getConnection("jdbc:odbc:serio");
       } catch (SQLException ex) {
          Logger.getLogger(Interfaz11.class.getName()).log(Level.SEVERE, null, ex);
```

```
        }
    } catch (ClassNotFoundException ex) {
        Logger.getLogger(Interfaz11.class.getName()).log(Level.SEVERE, null, ex);
    }
    try {
         Statement st = con5.createStatement();

        int row =jTable5.getSelectedRow();
        String TableC=(jTable5.getModel().getValueAt(row,0).toString());
        String sql ="SELECT * FROM  Pacientes where No_cuenta= '"+TableC+"'";
         ResultSet rs= st.executeQuery(sql);
         if(rs.next()){
            String T1 = rs.getString("No_cuenta");
            jTextField1.setText(T1);
            String T2 = rs.getString("Nom_paciente");
            jTextField3.setText(T2);

         }

    } catch (Exception ex) {
        JOptionPane.showMessageDialog(null,ex);



}
  cargar1(jTextField15.getText());
}

private void jTextField15KeyReleased(java.awt.event.KeyEvent evt) {
    // -------->>>Realiza una busqueda indexada---------->><
    cargar1(jTextField15.getText());
```

```
}

private void jTable2MouseClicked(java.awt.event.MouseEvent evt) {
   // ---->>>Carga los registros en los campos de la interfaz----->>>
   try {
       Statement st = con2.createStatement();

      int row =jTable2.getSelectedRow();
      String TableC=(jTable2.getModel().getValueAt(row,0).toString());
      String sql ="SELECT * FROM  Medicos where Nom_medico= '"+TableC+"'";
       ResultSet rs= st.executeQuery(sql);
       if(rs.next()){
          String T1 = rs.getString("Nom_medico");
          jTextField8.setText(T1);

       }

   } catch (Exception ex) {
      JOptionPane.showMessageDialog(null,ex);

}
}

private void jTextField12KeyReleased(java.awt.event.KeyEvent evt) {
   // ------------>>>Realiza una busqueda indexada------------>>>
   cargar2(jTextField12.getText());
}

private void jTable3MouseClicked(java.awt.event.MouseEvent evt) {
   // ---------->>>Carga los registros en los campos-------->>>
   try {
```

```
        Statement st = con3.createStatement();

       int row =jTable3.getSelectedRow();
       String TableC=(jTable3.getModel().getValueAt(row,0).toString());
       String sql ="SELECT * FROM  Especialidades where Especialidad= '"+TableC+"'";
        ResultSet rs= st.executeQuery(sql);
        if(rs.next()){
           String T1 = rs.getString("Especialidad");
           jTextField9.setText(T1);

        }

   } catch (Exception ex) {
      JOptionPane.showMessageDialog(null,ex);

}
}

private void jTextField13KeyReleased(java.awt.event.KeyEvent evt) {
   // ---------------->>>Realiza una busqueda indexada---------->>>
    cargar3(jTextField13.getText());
}

private void jTable4MouseClicked(java.awt.event.MouseEvent evt) {
   // -------->>>Carga los registros en los campos---------->>>
    try {
       Statement st = con4.createStatement();

       int row =jTable4.getSelectedRow();
       String TableC=(jTable4.getModel().getValueAt(row,0).toString());
       String sql ="SELECT * FROM  Servicios where Servicio= '"+TableC+"'";
```

```
            ResultSet rs= st.executeQuery(sql);
            if(rs.next()){
               String T1 = rs.getString("Servicio");
               jTextField10.setText(T1);

            }

        } catch (Exception ex) {
           JOptionPane.showMessageDialog(null,ex);

    }
    }

    private void jTextField14KeyReleased(java.awt.event.KeyEvent evt) {
        // --------->>>Realiza una busqueda indexada------------>>>
         cargar4(jTextField14.getText());
    }

    private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {
        // ------------->>>Elimina los registros------------>>>
        String No_cuenta= jTextField1.getText();
       String sql="DELETE FROM Citas WHERE No_cuenta="+No_cuenta+"";
        try {
           Statement st=con.createStatement();
           int i = st.executeUpdate(sql);
           if(i==1){
           JOptionPane.showMessageDialog(this,
"Eliminado","Aviso",JOptionPane.INFORMATION_MESSAGE);
           }else {
               JOptionPane.showMessageDialog(this,          "Error          al
elminar","Aviso",JOptionPane.INFORMATION_MESSAGE);
```

```
        }

    } catch (SQLException ex) {
        Logger.getLogger(Interfaz6u1.class.getName()).log(Level.SEVERE, null, ex);
    }

  limpiar1();
  jTextField11.setText("");
  cargar("");
}

private void jButton6ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    Interfaz4 I4 =new Interfaz4();
    I4.setVisible(true);
    this.setVisible(false);

}
```

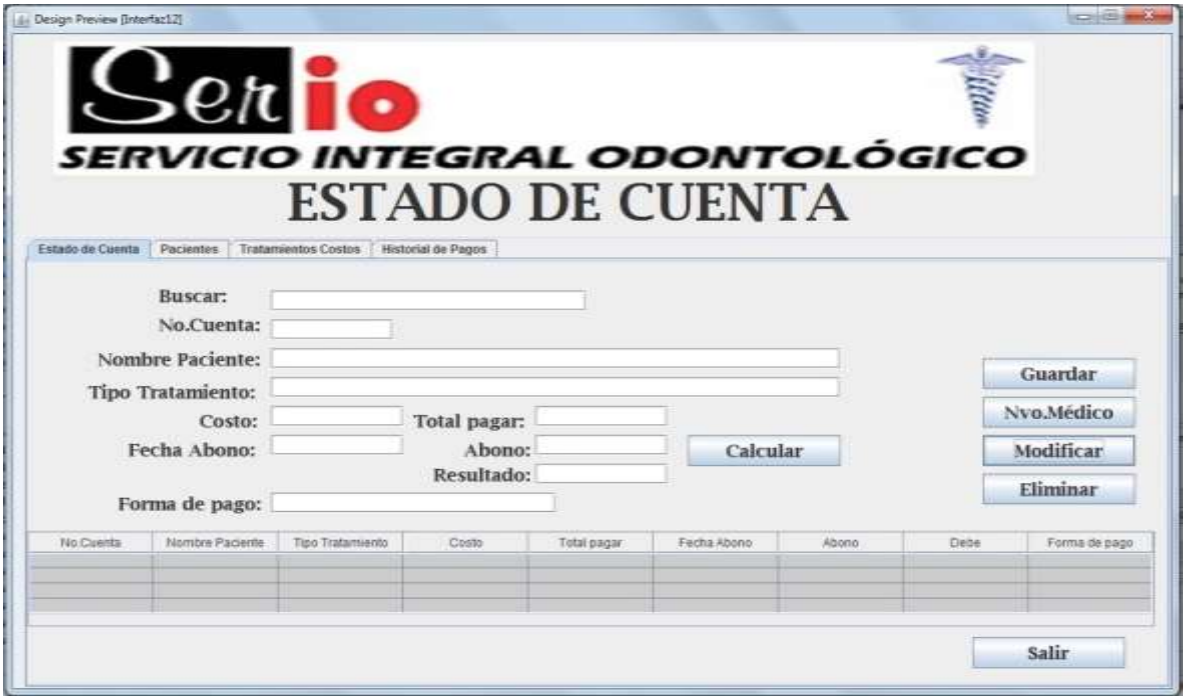


Figura 6.6.9 Modulo Estado de Cuenta

```
---------->>>Librerias---------->>>
 */
import java.sql.*;
import java.util.ArrayList;
import java.util.Calendar;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;

/**
 ---------->>>Declaración de variables---------->>>
 */
public class Interfaz12 extends javax.swing.JFrame {
    DefaultTableModel model;
    DefaultTableModel model1;
    DefaultTableModel model2;
```

```
DefaultTableModel model3;
Connection con;
Connection con1;
Connection con2;
Connection con3;
Connection con4;
CallableStatement cst;
ResultSet r;

/**
 ------->>>Conexión a SQL------>>>
 */
public Interfaz12() {
    initComponents();
    setLocationRelativeTo(null);
   try {
        Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
        try {
            con=DriverManager.getConnection("jdbc:odbc:serio");
            con1=DriverManager.getConnection("jdbc:odbc:serio");
            con2=DriverManager.getConnection("jdbc:odbc:serio");
            con3=DriverManager.getConnection("jdbc:odbc:serio");
            con4=DriverManager.getConnection("jdbc:odbc:serio");
        } catch (SQLException ex) {
            Logger.getLogger(Interfaz7.class.getName()).log(Level.SEVERE, null, ex);
        }
    } catch (ClassNotFoundException ex) {
        Logger.getLogger(Interfaz7.class.getName()).log(Level.SEVERE, null, ex);
    }

    cargar("");
```

```
        cargar1("");
        cargar2("");
        cargar3("");
        limpiar();
         Calendar cal=Calendar.getInstance();
 String fecha=cal.get(cal.YEAR)+"-"+"11"+"-"+cal.get(cal.DATE);
 //String
hora=cal.get(cal.HOUR_OF_DAY)+":"+cal.get(cal.MINUTE)+":"+cal.get(cal.SECOND);
//jTextField4.setText(hora);
jTextField4.setText(fecha);
    }
    void limpiar()
{
    jTextField1.setText("");
    jTextField2.setText("");
    jTextField3.setText("");
    jTextField4.setText("");
    jTextField5.setText("");
    jTextField6.setText("");
    jTextField7.setText("");
    jTextField8.setText("");
    jTextField9.setText("");
    jTextField10.setText("");
    jTextField11.setText("");
   jTextField12.setText("");
   jTextField13.setText("");
}
    void cargar(String consul)
    {
```

```
String[] titulos ={"No.Cuenta","Nombre Paciente"};
String[] registros=new String[2];

String sql ="SELECT * FROM Pacientes where Nom_paciente LIKE '%"+consul+"%'";

model= new DefaultTableModel(null,titulos);



 try {

Statement st = con.createStatement();
ResultSet rs= st.executeQuery(sql);

 while(rs.next())
 {
  registros[0]=rs.getString("No_cuenta");
  registros[1]=rs.getString("Nom_paciente");



  model.addRow(registros);
  }

  jTable2.setModel(model);

} catch (SQLException ex) {
   JOptionPane.showMessageDialog(null,ex);
}

}
```

```
void cargar1(String consul)
   {


       String[] titulos ={"Tipo Tratamiento","Costo"};
       String[] registros=new String[2];

     String sql ="SELECT * FROM  TratamientosC where Tipo_tratamiento LIKE '%"+consul+"%'";

       model1= new DefaultTableModel(null,titulos);

         try {

        Statement st = con.createStatement();
        ResultSet rs= st.executeQuery(sql);

         while(rs.next())
         {
          registros[0]=rs.getString("Tipo_tratamiento");
          registros[1]=rs.getString("Costo");



          model1.addRow(registros);
          }

       jTable3.setModel(model1);

       } catch (SQLException ex) {
          JOptionPane.showMessageDialog(null,ex);
```

```
      }

   }
void cargar2(String consul)
   {

      String[] titulos ={"No.Cuenta","Nombre Paciente","Tipo Tratamiento","Costo","Total pagar","Fecha Abono","Abono","Debe","Forma de pago"};
      String[] registros=new String[9];

    String sql ="SELECT * FROM  EstadoCuenta where Nom_paciente LIKE '%"+consul+"%'";

      model2= new DefaultTableModel(null,titulos);

        try {

       Statement st = con.createStatement();
       ResultSet rs= st.executeQuery(sql);

        while(rs.next())
        {
         registros[0]=rs.getString("No_cuenta");
         registros[1]=rs.getString("Nom_paciente");
         registros[2]=rs.getString("Tipo_tratamiento");
         registros[3]=rs.getString("Costo");
         registros[4]=rs.getString("Total_pagar");
         registros[5]=rs.getString("Fecha_abono");
         registros[6]=rs.getString("Abono");
         registros[7]=rs.getString("Saldo");
         registros[8]=rs.getString("Form_pago");
```

```
        model2.addRow(registros);
        }

        jTable1.setModel(model2);

      } catch (SQLException ex) {
        JOptionPane.showMessageDialog(null,ex);
      }

  }
void cargar3(String consul)
  {


      String[] titulos ={"No.Cuenta","Nombre Paciente","Tipo Tratamiento","Costo","Total pagar","Fecha Abono","Abono","Debe","Forma de pago"};
      String[] registros=new String[9];

    String sql ="SELECT * FROM  EstadoCuenta where Nom_paciente LIKE '%"+consul+"%'";

      model3= new DefaultTableModel(null,titulos);

        try {

       Statement st = con4.createStatement();
       ResultSet rs= st.executeQuery(sql);

        while(rs.next())
```

```
        {
          registros[0]=rs.getString("No_cuenta");
          registros[1]=rs.getString("Nom_paciente");
          registros[2]=rs.getString("Tipo_tratamiento");
          registros[3]=rs.getString("Costo");
          registros[4]=rs.getString("Total_pagar");
          registros[5]=rs.getString("Fecha_abono");
          registros[6]=rs.getString("Abono");
          registros[7]=rs.getString("Saldo");
          registros[8]=rs.getString("Form_pago");


          model3.addRow(registros);
          }

          jTable4.setModel(model3);

      } catch (SQLException ex) {
          JOptionPane.showMessageDialog(null,ex);
      }

   }
private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
      // ---------->>>Modifica los registros---------->>>
      String No_cuenta=jTextField1.getText();
      String Nom_paciente=jTextField3.getText();
      String Tipo_tratamiento=jTextField2.getText();
      String Costo=jTextField6.getText();
      String Total_pagar=jTextField5.getText();
      String Fecha_abono=jTextField4.getText();
      String Abono=jTextField7.getText();
```

```
String Resultado=jTextField8.getText();
String Form_pago=jTextField9.getText();
//String Servicio=jTextField10.getText();

String Sql;
Sql="UPDATE EstadoCuenta SET No_cuenta=?,Nom_paciente=?, Tipo_tratamiento=?, Costo=?, Total_pagar=?, Fecha_abono=?, Abono=?, Saldo=?, Form_pago=? WHERE No_cuenta="+No_cuenta+"";
try {
   PreparedStatement g = con.prepareStatement(Sql);
   //g.setString(1,No_cuenta);
   g.setString(1,No_cuenta);
   g.setString(2,Nom_paciente);
   g.setString(3,Tipo_tratamiento);
   g.setString(4,Costo);
   g.setString(5,Total_pagar);
   g.setString(6,Fecha_abono);
   g.setString(7,Abono);
   g.setString(8,Resultado);
   g.setString(9,Form_pago);
   //g.setString(10,Servicio);


   int r=g.executeUpdate();

   if (r>0)
   {
      JOptionPane.showMessageDialog(null, "Registro Exitoso");

   }
} catch (SQLException ex) {
```

```
            Logger.getLogger(Interfaz12.class.getName()).log(Level.SEVERE, null, ex);
        }
        limpiar();
        cargar("");
    }

    private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {
        // ------------Limpia los registros------->>>
        limpiar();
    }

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        // --------->>>Guarda los registros---------->>>
        String No_cuenta=jTextField1.getText();
        String Nom_paciente=jTextField3.getText();
        String Tipo_tratamiento=jTextField2.getText();
        String Costo=jTextField6.getText();
        String Total_pagar=jTextField5.getText();
        String Fecha_abono=jTextField4.getText();
        String Abono=jTextField7.getText();
        String Resultado=jTextField8.getText();
        String Form_pago=jTextField9.getText();


String Sql;
Sql="INSERT INTO EstadoCuenta (No_cuenta, Nom_paciente,Tipo_tratamiento, Costo,
Total_pagar, Fecha_abono, Abono, Saldo, Form_pago)values(?,?,?,?,?,?,?,?,?)";
        try {
            PreparedStatement g = con.prepareStatement(Sql);
            //g.setString(1,Identificador);
            g.setString(1,No_cuenta);
```

```
            g.setString(2,Nom_paciente);
            g.setString(3,Tipo_tratamiento);
            g.setString(4,Costo);
            g.setString(5,Total_pagar);
            g.setString(6,Fecha_abono);
            g.setString(7,Abono);
            g.setString(8,Resultado);
            g.setString(9,Form_pago);
            //g.setString(10,Servicio);

            int r=g.executeUpdate();

            if (r>0)
            {
                JOptionPane.showMessageDialog(null, "Registro Exitoso");

            }
        } catch (SQLException ex) {
            Logger.getLogger(Interfaz12.class.getName()).log(Level.SEVERE, null, ex);
        }
        limpiar();
        cargar2("");
    }
private void jTable1MouseClicked(java.awt.event.MouseEvent evt) {
        //---------->>>Carga los registros en los campos---------->>>
         String T1;
         int costo;
        try {
            Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
            try {
                con3=DriverManager.getConnection("jdbc:odbc:serio");
```

```
        } catch (SQLException ex) {
            Logger.getLogger(Interfaz7.class.getName()).log(Level.SEVERE, null, ex);
        }
    } catch (ClassNotFoundException ex) {
        Logger.getLogger(Interfaz7.class.getName()).log(Level.SEVERE, null, ex);
    }
    try {
         Statement st = con3.createStatement();

        int row =jTable1.getSelectedRow();
        String TableC=(jTable1.getModel().getValueAt(row,0).toString());
        String sql ="SELECT * FROM  EstadoCuenta where No_cuenta= '"+TableC+"'";
         ResultSet rs= st.executeQuery(sql);
         if(rs.next()){
            T1 = rs.getString("Total_pagar");
              costo=Integer.parseInt(T1);
              int tratamiento=Integer.parseInt(jTextField6.getText());
            int total= costo+tratamiento;

             jTextField5.setText(String.valueOf(total));
             //jTextField5.setText(T1);
             String T2 = rs.getString("Form_pago");
             jTextField9.setText(T2);
         }

    } catch (Exception ex) {
        JOptionPane.showMessageDialog(null,ex);

}

}
```

```
private void jTable2MouseClicked(java.awt.event.MouseEvent evt) {
   // ---------->>>Carga los registros en los campos---------->>>
   try {
      Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
      try {
         con3=DriverManager.getConnection("jdbc:odbc:serio");
      } catch (SQLException ex) {
         Logger.getLogger(Interfaz7.class.getName()).log(Level.SEVERE, null, ex);
      }
   } catch (ClassNotFoundException ex) {
      Logger.getLogger(Interfaz7.class.getName()).log(Level.SEVERE, null, ex);
   }
   try {
       Statement st = con3.createStatement();

      int row =jTable2.getSelectedRow();
      String TableC=(jTable2.getModel().getValueAt(row,0).toString());
      String sql ="SELECT * FROM  Pacientes where No_cuenta= '"+TableC+"'";
       ResultSet rs= st.executeQuery(sql);
       if(rs.next()){
          String T1 = rs.getString("No_cuenta");
          jTextField1.setText(T1);
          String T2 = rs.getString("Nom_paciente");
          jTextField3.setText(T2);

       }

   } catch (Exception ex) {
      JOptionPane.showMessageDialog(null,ex);
```

```
    }
     cargar2(jTextField3.getText());
    }

    private void jTable3MouseClicked(java.awt.event.MouseEvent evt) {
        // ---------->>>Carga los registros en los campos----->>>
        try {
            Statement st1 = con2.createStatement();

           int row =jTable3.getSelectedRow();
           String TableC=(jTable3.getModel().getValueAt(row,0).toString());
           String sql ="SELECT * FROM  TratamientosC where Tipo_tratamiento= '"+TableC+"'";
            ResultSet rs= st1.executeQuery(sql);
            if(rs.next()){
               String T1 = rs.getString("Tipo_tratamiento");
               jTextField2.setText(T1);
               String T2 = rs.getString("Costo");
               jTextField6.setText(T2);

            }

        } catch (Exception ex) {
           JOptionPane.showMessageDialog(null,ex);

    }

    }

private void jButton6ActionPerformed(java.awt.event.ActionEvent evt) {
        // ---------->>>Realiza los calculos nemericos---------->>>
```

```
        int resultado;
        int abono;
        int debe;
              debe=Integer.parseInt(jTextField5.getText());
               abono=Integer.parseInt(jTextField8.getText());
              resultado= debe-abono;
               jTextField7.setText(String.valueOf(resultado));

    }
private void jTextField12KeyReleased(java.awt.event.KeyEvent evt) {
        //--------->>>Realiza una busqueda indexada---------->>>
        cargar2(jTextField12.getText());
    }

    private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {
        //---------->>>Elimina los registros---------->>>
        String No_cuenta= jTextField1.getText();
       String sql="DELETE FROM EstadoCuenta WHERE No_cuenta="+No_cuenta+"";
        try {
            Statement st=con.createStatement();
            int i = st.executeUpdate(sql);
            if(i==1){
            JOptionPane.showMessageDialog(this,
"Eliminado","Aviso",JOptionPane.INFORMATION_MESSAGE);
            }else {
                JOptionPane.showMessageDialog(this,          "Error          al
elminar","Aviso",JOptionPane.INFORMATION_MESSAGE);
            }

        } catch (SQLException ex) {
            Logger.getLogger(Interfaz6u1.class.getName()).log(Level.SEVERE, null, ex);
```

```
    }

   limpiar();
   jTextField12.setText("");
   cargar("");
}

private void jButton7ActionPerformed(java.awt.event.ActionEvent evt) {
    // --------->>>Regresa al Menú del administrador---------->>>
    Interfaz4 I4 =new Interfaz4();
    I4.setVisible(true);
    this.setVisible(false);

}

private void jTable4MouseClicked(java.awt.event.MouseEvent evt) {
    //---------->>>Carga los registros en los campos------->>>
    try {
        Statement st = con4.createStatement();

       int row =jTable4.getSelectedRow();
       String TableC=(jTable4.getModel().getValueAt(row,0).toString());
       String sql ="SELECT * FROM  Citas where Nom_paciente= '"+TableC+"'";
        ResultSet rs= st.executeQuery(sql);
        if(rs.next()){
            String T1 = rs.getString("No_cuenta");
            jTextField1.setText(T1);
            String T2 = rs.getString("Nom_paciente");
            jTextField3.setText(T2);
            String T3 = rs.getString("Tipo_tratamiento");
            jTextField2.setText(T3);
```

```
            String T4 = rs.getString("Costo");
            jTextField6.setText(T4);
            String T5 = rs.getString("Total_pagar");
            jTextField5.setText(T5);
            String T6 = rs.getString("Fecha_abono");
            jTextField4.setText(T6);
            String T7 = rs.getString("Abono");
            jTextField7.setText(T7);
            String T8 = rs.getString("Resultado");
            jTextField8.setText(T8);
            String T9 = rs.getString("Form_pago");
            jTextField9.setText(T9);


          }

        } catch (Exception ex) {
           JOptionPane.showMessageDialog(null,ex);

    }
    }
private void jTextField13KeyReleased(java.awt.event.KeyEvent evt) {
        //---------->>>Realiza una busqueda indexada---------->>>
        cargar3(jTextField13.getText());
    }
```

Conclusiones

En el desarrollo de la Tesis se creó la base de datos y las interfaces de acuerdo a cada uno de los módulos requeridos por la organización, tomando en cuenta el análisis y requerimientos obtenidos durante la investigación, las cuales conforman las operaciones llevadas a cabo cotidianamente en la clínica Serio y todos los requerimientos que debe tener el SI para la Clínica, deberán satisfacer y resolver las necesidades del cliente.

Trabajo futuro

Estará basado en la realización de los manuales técnicos y de usuario, así como la instalación del SI y la BD en la Clínica Serio. Se impartirá una capacitación a los integrantes de la Clínica con la finalidad de que hagan buen uso del SI.

Referencias

[1] C. DATE, Introducción a los Sistemas de base de datos, Pearson Prentice Hall.

[2]Abraham Silberschatz & Fernando Sáez Pérez. Fundamentos de Base de Datos Quinta edición. Editorial McGraw Hill (España).

[3] T.GROUSSARD, Visual Basic.NET (VB.NET) Programe con Visual Studio 2008

[4] Kenneth E. Kendall, Julie E. Kendall *sistemas de información* *2005*, México.

[5] Kenneth E. Kendall, Julie E. Kendall *sistemas de información* *2005*, México.

[6] IICA, MAG, BID, Sistema de Información MAG: Diseño,(2007)

[7] V.H. Galván Zavala, "Control No Lineal de un Robot Bípedo", Tesis de MaesAnálisis de Sistemas-mundo: Una Introducciónтría, Asesor: Dr. Luis Enrique Ramos Velasco, Universidad Autónoma del Estado de Hidalgo, México. (2007)

[8] http://www3.uaem.mx/posgrado/mcruz/cursos/miic/diseño.pdf

Anexos

I Tabla comparativa de Precios, Materiales y tratamientos.