



INSTITUTO TECNOLÓGICO
"CORDILLERA"

CARRERA DE ANÁLISIS DE SISTEMAS

**“GESTIÓN DE HISTORIAS CLÍNICAS Y CONTROL DE CITAS MÉDICAS
PARA ADENTAL CLÍNICA ODONTOLÓGICA DE LA CIUDAD DE
QUITO”**

Proyecto de investigación previo a la obtención del título de

Tecnólogo Analista de Sistemas.

AUTOR: GUEVARA CHANGOTAXI NELSON PATRICIO

TUTOR: ING. JOHNNY CORONEL

QUITO, Abril 2014



DECLARATORIA

Declaro que la investigación es absolutamente original, auténtica, personal, que se han citado las fuentes correspondientes y que en su ejecución se respetaron las disposiciones legales que protegen los derechos de autor vigentes. Las ideas, doctrinas, resultados y conclusiones a los que he llegado son de mi absoluta responsabilidad.

Nelson Patricio Guevara Changotaxi

CC 100281313-5



CONTRATO DE CESIÓN SOBRE DERECHOS PROPIEDAD

INTELECTUAL

Comparecen a la celebración del presente contrato de cesión y transferencia de derechos de propiedad intelectual, por una parte, el estudiante **GUEVARA CHANGOTAXI NELSON PATRICIO**, por sus propios y personales derechos, a quien en lo posterior se le denominará el "CEDENTE"; y, por otra parte, el INSTITUTO SUPERIOR TECNOLÓGICO CORDILLERA, representado por su Rector el Ingeniero Ernesto Flores Córdova, a quien en lo posterior se lo denominará el "CESIONARIO". Los comparecientes son mayores de edad, domiciliados en esta ciudad de Quito Distrito Metropolitano, hábiles y capaces para contraer derechos y obligaciones, quienes acuerdan al tenor de las siguientes cláusulas:

PRIMERA: ANTECEDENTE.- a) El Cedente dentro del pensum de estudio en la carrera de análisis de sistemas que imparte el Instituto Superior Tecnológico Cordillera, y con el objeto de obtener el título de Tecnólogo en Análisis de Sistemas, el estudiante participa en el proyecto de grado denominado

"GESTIÓN DE HISTORIAS CLÍNICAS, Y CONTROL DE CITAS MÉDICAS PARA ADENTAL CLÍNICA ODONTOLÓGICA DE LA CIUDAD DE QUITO", el cual incluye la creación y desarrollo del programa de ordenador o software, para lo cual ha implementado los conocimientos adquiridos en su calidad de alumno. **b)** Por iniciativa y responsabilidad del Instituto Superior Tecnológico Cordillera se desarrolla la creación del programa de ordenador, motivo por el cual se regula de forma clara la cesión de los derechos de autor que genera la obra literaria y que es producto del proyecto de grado, el mismo que culminado es de plena aplicación técnica, administrativa y de reproducción.



SEGUNDA: CESIÓN Y TRANSFERENCIA.- Con el antecedente indicado, el Cedente libre y voluntariamente cede y transfiere de manera perpetua y gratuita todos los derechos patrimoniales del programa de ordenador descrito en la clausula anterior a favor del Cesionario, sin reservarse para sí ningún privilegio especial (código fuente, código objeto, diagramas de flujo, planos, manuales de uso, etc.). El Cesionario podrá explotar el programa de ordenador por cualquier medio o procedimiento tal cual lo establece el Artículo 20 de la Ley de Propiedad Intelectual, esto es, realizar, autorizar o prohibir, entre otros: a) La reproducción del programa de ordenador por cualquier forma o procedimiento; b) La comunicación pública del software; c) La distribución pública de ejemplares o copias, la comercialización, arrendamiento o alquiler del programa de ordenador; d) Cualquier transformación o modificación del programa de ordenador; e) La protección y registro en el IEPI el programa de ordenador a nombre del Cesionario; f) Ejercer la protección jurídica del programa de ordenador; g) Los demás derechos establecidos en la Ley de Propiedad Intelectual y otros cuerpos legales que normen sobre la cesión de derechos de autor y derechos patrimoniales.

TERCERA: OBLIGACIÓN DEL CEDENTE.- El cedente no podrá transferir a ningún tercero los derechos que conforman la estructura, secuencia y organización del programa de ordenador que es objeto del presente contrato, como tampoco emplearlo o utilizarlo a título personal, ya que siempre se deberá guardar la exclusividad del programa de ordenador a favor del Cesionario.

CUARTA: CUANTIA.- La cesión objeto del presente contrato, se realiza a título gratuito y por ende el Cesionario ni sus administradores deben cancelar valor alguno o regalías por este contrato y por los derechos que se derivan del mismo.



QUINTA: PLAZO.- La vigencia del presente contrato es indefinida.

SEXTA: DOMICILIO, JURISDICCIÓN Y COMPETENCIA.- Las partes fijan como su domicilio la ciudad de Quito. Toda controversia o diferencia derivada de éste, será resuelta directamente entre las partes y, si esto no fuere factible, se solicitará la asistencia de un Mediador del Centro de Arbitraje y Mediación de la Cámara de Comercio de Quito. En el evento que el conflicto no fuere resuelto mediante este procedimiento, en el plazo de diez días calendario desde su inicio, pudiendo prorrogarse por mutuo acuerdo este plazo, las partes someterán sus controversias a la resolución de un árbitro, que se sujetará a lo dispuesto en la Ley de Arbitraje y Mediación, al Reglamento del Centro de Arbitraje y Mediación de la Cámara de comercio de Quito, y a las siguientes normas: a) El árbitro será seleccionado conforme a lo establecido en la Ley de Arbitraje y Mediación; b) Las partes renuncian a la jurisdicción ordinaria, se obligan a acatar el laudo arbitral y se comprometen a no interponer ningún tipo de recurso en contra del laudo arbitral; c) Para la ejecución de medidas cautelares, el árbitro está facultado para solicitar el auxilio de los funcionarios públicos, judiciales, policiales y administrativos, sin que sea necesario recurrir a juez ordinario alguno; d) El procedimiento será confidencial y en derecho; e) El lugar de arbitraje serán las instalaciones del centro de arbitraje y mediación de la Cámara de Comercio de Quito; f) El idioma del arbitraje será el español; y, g) La reconvención, caso de haberla, seguirá los mismos procedimientos antes indicados para el juicio principal.

SÉPTIMA: ACEPTACIÓN.- Las partes contratantes aceptan el contenido del presente contrato, por ser hecho en seguridad de sus respectivos intereses.



En aceptación firman a los 17 días del mes de Abril del dos mil catorce.

f) _____

C.C. N° 1002813135

CEDENTE

f) _____

Instituto Superior Tecnológico Cordillera

CESIONARIO



AGRADECIMIENTO

En primer lugar a Dios por haberme guiado por el camino de la felicidad hasta ahora; en segundo lugar a cada uno de los que son parte de mi familia, a mi MADRE, todos mis tíos; por siempre haberme dado su fuerza y apoyo incondicional que me han ayudado y llevado hasta donde estoy ahora y a mi director de tesis quién me ayudó en todo momento, Ing. Johnny Coronel.

Son muchas las personas que han formado parte de mi vida profesional a las que les encantaría agradecerles su amistad, consejos, apoyo, ánimo y compañía en los momentos más difíciles de mi vida. Algunas están aquí conmigo y otras en mis recuerdos y en mi corazón, sin importar en donde estén quiero darles las gracias por formar parte de mí, por todo lo que me han brindado y por todas sus bendiciones.



DEDICATORIA

Dedico este proyecto a Dios y a mi Madre, a Dios porque ha estado conmigo a cada paso que doy, cuidándome y dándome fortaleza para continuar, a mi Madre, quien a lo largo de mi vida ha velado por mi bienestar y educación siendo mi apoyo en todo momento depositando su entera confianza en cada reto que se me presentaba sin dudar ni un solo momento en mi inteligencia y capacidad. Es por ello que soy lo que soy ahora.



ÍNDICE GENERAL

Titulo	Página
DECLARATORIA.....	ii
AGRADECIMIENTO.....	vii
DEDICATORIA	viii
RESUMEN EJECUTIVO	xvii
ABSTRACT	xix
Capítulo I: Antecedentes	1
1.01 Contexto.....	1
1.02 Justificación.....	2
1.03 MATRIZ T	3
Capítulo II: Análisis de Involucrados	5
2.01 Análisis de los involucrados.....	5
2.02. MATRIZ DE ANALISIS DE INVOLUCRADOS	7
Matriz de involucrados.....	7
Capítulo III: Problemas y Objetivos	8
3.01 Construcción del árbol de problemas.....	8
3.02 Construcción del árbol de objetivos.....	8
ARBOL DE PROBLEMAS	9
Capítulo IV: Análisis de Alternativas	11
4.01 Matriz de análisis de alternativas	11
Matriz de análisis de alternativas	11
4.02 Matriz de análisis del impacto de los objetivos	12
Titulo	Página



4.03 Diagrama de estrategias	13
4.04 Matriz del marco lógico	14
04.05 Matriz del marco lógico	15
Matriz Marco Lógico	15
Capítulo V: Propuesta	16
5.01 Justificación del software	16
5.02 Análisis y diseño	16
5.02.01 Diagramas de caso de uso	16
Caso de uso General.....	18
Caso de uso general.....	18
5.02.02. Diagrama de Realización	19
Asignar Turno	19
REGISTRAR PAGO CONSULTA	20
CREAR HISTORIA CLINICA.....	21
REGISTRAR DIAGNOSTICO	22
REGISTRAR TRATAMIENTO.....	23
5.02.03 Diagramas de secuencia	24
Diagrama de Secuencia Asignar Turno.....	25
5.02.04 Diagramas de Colaboración	30
5.02.05 Diagrama de componentes	35
5.02.06 Diagrama físico y lógico.	37
Modelo lógico de la base de datos	38
Modelo físico de la base de datos.	39
5.03 Desarrollo.....	40
5.03.01 Arquitectura de software.....	40



Titulo	Página
5.03.01.01. Arquitectura basada en capas.....	40
5.03.01.02. Arquitectura de N-Capas / 3-Capas.....	41
5.03.01.04. Requerimientos de software.....	42
5.03.02 Estándares de base de datos.	43
Estándares de base de datos	45
5.03.03 Estándares de programación.	45
5.03.04 Diseño de interfaces.	47
5.04 Pruebas	58
5.04.01 Casos de pruebas	58
Historial de Revisiones	58
Alcances	59
Requerimientos de pruebas	59
5.04.02 Pruebas de Caso de Uso	64
Capítulo VI: Aspectos Administrativos	71
6.01 Recursos	71
6.02 Presupuesto	72
6.03 Cronograma.....	73
Capítulo VII: Conclusiones y Recomendaciones.....	74
7.01 Conclusiones	74
7.02 Recomendaciones.....	75
ANEXOS.....	76
A.01. Matriz de análisis del impacto de los objetivos.....	77
Bibliografía	78
Manual de Usuario del Sistema.....	79



Titulo	Página
Diccionario de Datos.....	89
Código Fuente	93
Script de la base de datos.	113
MANUAL TECNICO DEL SISTEMA.....	132



INDICE DE TABLAS

Titulo	Página
Tabla 1. Análisis de fuerza T	4
Tabla 2. Matriz de Involucrados	7
Tabla 3. Matriz de Análisis de Alternativas.....	11
Tabla 4. Matriz Marco Lógico	15
Tabla 5. Caso de uso general.....	18
Tabla 6. Asignar Turno	19
Tabla 7. Registrar Pago Consulta.....	20
Tabla 8. Crear Historia Clínica	21
Tabla 9. Registrar Diagnóstico.....	22
Tabla 10. Registrar Tratamiento	23
Tabla 11. Estándares de Base de Datos.....	45
Tabla 12. Historial de Revisiones	58
Tabla 13. CP_01_E01	64
Tabla 14. CP_01_E02	65
Tabla 15. CP_01_E03	65
Tabla 16. CP_01_E04	65
Tabla 17. CP_02_E02	66
Tabla 18. CP_02_E03	66
Tabla 19. CP_02_E04	66
Tabla 20. CP_02_E05	67
Tabla 21. CP_03_E03	67
Tabla 22. CP_03_E03	68
Tabla 23. CP_03_E03	68



Titulo	Página
Tabla 24. CP_03_E03	68
Tabla 25. CP_03_E03	69
Tabla 26. CP_03_E03	69
Tabla 27. CP_04_E04	69
Tabla 28. CP_04_E04	70
Tabla 29. Pacientes.....	89
Tabla 30. Médicos.....	90
Tabla 31. Historia Clínica	90
Tabla 32. Usuarios	91
Tabla 33. Área.....	91
Tabla 34. Cargo.....	91
Tabla 35. Especialidad	91
Tabla 36. Grupo Diente.....	92
Tabla 37. Horario	92
Tabla 38.....	92



INDICE DE FIGURAS

Titulo	Página
Figura .1. Mapa de involucrados.....	5
Figura 2. Arbol de Problemas	9
Figura 3. Arbol de Objetivos.....	10
Figura 4.	13
Figura. 5. Caso de uso general	18
Figura 6. D.R. Asignar Turno.	20
Figura 7. D.R. Registrar pago consulta medica	21
Figura 8. D.R. Crear Historia Clinica	22
Figura 9. D.R. Registrar Diagnostico.....	23
Figura 10. D.R. REGISTRAR TRATAMIENTO.....	24
Figura. 11. D.S. Asignar de Turno	25
Figura. 12. D.S. Pago de Consulta	26
Figura. 13. D.S. Crear Historia Clínica.....	27
Figura. 14. D.S. Registrar Diagnóstico	28
Figura. 15. D.S. Registrar Tratamiento	29
Figura. 16. D.C. Asignar de Turno.....	30
Figura. 17. DCS. Pago de Consulta.....	31
Figura. 18. DCS. Crear Historia Clínica	32
Figura. 19. D.C. Registrar Diagnóstico.....	33
Figura. 20. D.C. Registrar Tratamiento.....	34
Figura 21. Diagrama de Componentes.....	35
Figura 22. Diagrama de Componentes.....	36
Figura 23. Modelo Lógico.....	38



Titulo	Página
Figura 24. Modelo Físico	39
Figura 25. Arquitectura basada en capas.....	40
Figura 26.	42
Figura 27. Ingreso "Login-Odontológico"	47
Figura 28. Interfaz Pantalla Principal.....	48
Figura 29. Interfaz Usuario	49
Figura 30. Interfaz Usuario	50
Figura 31. Lista de médicos agregados	51
Figura 32. Registro Nuevo Medico	51
Figura 33. Listado de Pacientes	52
Figura 34. Ingresar nuevo paciente	52
Figura 35. Genera turno por especialidad	53
Figura 36. Agregamos los días para solicitar turno.....	53
Figura 37. Buscar Historia Clínica.....	54
Figura 38. Interfaz Nueva Historia Clínica	54
Figura 39. Interfaz Lista de historias clínicas ingresadas	55
Figura 40. Interfaz Ingresar Tratamiento	55
Figura 41. Registrar tratamiento en Odontograma.....	56
Figura 42. Interfaz Mantenimientos	57
Figura 43. Cronograma de actividades.....	73



RESUMEN EJECUTIVO

Las Historias Clínicas son parte fundamental en el ámbito de la salud tanto pública como privada, ellas contienen información vital relevante para la planificación de tratamientos y la verificación de la evolución de los pacientes a dichos tratamientos, de ahí que la historia clínica debe estar al alcance de los médicos tratantes de una manera eficiente, evitando contratiempos de accesibilidad, pérdida de información, ilegibilidad, falta de confiabilidad y riesgos de confiabilidad.

En la actualidad, muchos países ya cuentan con sistemas de toma de turnos y citas médicas integrales desarrollados de forma eficiente para mejorar el servicio de atención médica, que permite gestionar la solicitud de citas entre los pacientes y su Centro de Salud, en cualquier día, a cualquier hora, proporcionándole un nuevo canal de comunicación con su médico, razón por la cual más personas solicitan dichos turnos porque los pacientes evitan el uso del tiempo empleado a la hora de desplazarse al centro de salud para solicitar una cita con su médico.

En la mayoría de las instituciones médicas no cuentan con un sistema de citas médicas, razón por la cual hoy en día usan el proceso convencional, es decir tradicional, el cual consiste en hacer colas para solicitar una cita médica; por lo que el inconveniente de dicho proceso es que muchas personas se quedan haciendo cola sin tener conocimiento que la citas habían llegado al concluirse y la espera era en vano.

De igual forma se presenta muchos problemas con la atención que ADENTAL Clínica Odontológica tiene con sus línea telefónica para citas, nunca hay una respuesta para otorgar citas, y su excusa es que los parámetros están cerrados que no pueden dar para otro día, que vuelva a llamar causando un malestar mayor ya que igual te toma tiempo hasta que te contesten y para que te den una negativa una vez realizado la llamada, además el costo que implica la llamada, otro solución para ellos es ir a los mismos



centros médicos y llamar desde sus teléfonos que solo son uno máximo dos por cada centro de salud para que la operadora te diga que ya no hay citas, haciendo colas interminables para esperar que te toque tu turno de llamar.



ABSTRACT

OVERVIEW EXECUTIVE

The Medical Records are a fundamental part in the field of public and private health, they contain vital information relevant to treatment planning and verification of patient outcomes such treatments, hence the medical record must be available treating physicians in an efficient manner, avoiding setbacks accessibility, loss of information, illegibility, unreliability and reliability risks. At present , many countries already have systems of turn-taking and comprehensive medical appointments efficiently developed to improve health care service , which allows the application to manage appointments between patients and their health center, on any day, at any time, providing a new channel of communication with your doctor , which is why most people apply these shifts because patients avoid the use of time spent when traveling to the health center for an appointment with your physician.

In most medical institutions do not have an appointments system, which is why today use the conventional process, ie traditional, which involves making queues to request an appointment; so the downside of this process is that many people are queuing without knowledge that the quotations had come to the conclusion and the wait was in vain.

Similarly many problems with attention that has ADENTAL Dental Clinic with telephone line dating is presented , there is never an answer to give quotes, and their excuse is that the parameters are closed they can not give for another day , which again causing further distress call because you just takes time until you answer and so you get a negative once made the call, plus the cost of the call , another solution for them is to go to the same medical centers and call from their phones are only one maximum



two per health center for the operator tells you that there are no quotes, doing endless queues to wait for your turn your turn to call.



Capítulo I: Antecedentes

1.01 Contexto

La Clínica Odontológica "ADENTAL" de la ciudad de Quito, desde hace mucho tiempo viene sirviendo a la sociedad en las diferentes áreas de la Odontología, como son: tipo de tratamientos odontológicos: implantes, estética dental, blanqueamiento, prótesis, ortodoncia, ortodoncia invisible, odontopediatría, cirugía oral, endodoncia, odontología conservadora, periodoncia, entre otras, tiene como finalidad mejorar la atención a los pacientes, para ofrecer un servicio eficiente y eficaz.

El Diseño e Implementación de Software de Manejo de Historias Clínicas y Control de Citas Médicas para ADENTAL Clínica Odontológica de Quito permitirá incrementar el incentivo en el estudiante para el desarrollo y aporte tanto científico como tecnológico en esta institución médica, el mismo que nos admitirá utilizar una metodología que habilite el estudio, descriptivo y explicativo de la bibliografía que nos enseña a descubrir nuevos conocimientos, y nuevas alternativas en el mundo de la informática. Con lo expuesto anteriormente podemos decir que el objetivo principal es implementar un software de manejo de historias clínicas y control de citas médicas que permita elevar el nivel de servicio en la Clínica y llegar como conclusión al fortalecimiento en el proceso de registro de pacientes en la relación teórica - practica a través de la aplicación de un procedimiento que ayude a resolver el problema.

Con esto el autor del presente proyecto se ha propuesto un reto a la investigación y desarrollo orientado a la Odontología y a la informática con el diseño e implementación de Historias Clínicas y Turnos para la Clínica "ADENTAL" de Quito, el mismo que brindará un servicio mejor y oportuno a todos quienes visiten este centro Odontológico.

La gestión de historias clínicas, y control de citas médicas para “ADENTAL” clínica odontológica de la ciudad de Quito, es un sistema que permite automatizar el servicio Odontológico, el mismo que se realizarán a un bajo costo optimizando recursos económicos, humanos y materiales. Éste sistema se desarrolló utilizando la plataforma Punto Net (Visual .Net C#). Es un lenguaje de alto nivel que está preparado para implementar sistemas complejos, validar la entrada de datos, implementar el manejo de errores en la interfaz de usuario, incorporar el manejo de eventos, implementando características de accesibilidad.

La gestión de historias clínicas, y control de citas médicas para “ADENTAL” clínica odontológica se realizará mediante el acceso a datos del paciente a través de su cédula de ciudadanía (*Número de Historia Clínica*) creado por la clínica el mismo que permitirá desplegar en forma oportuna y precisa toda la historia clínica existente, además permitirá mediante un mensaje de texto móvil le permitirá recordar el turno y tomar una cita médica en la especialidad que requiera el usuario.

Este software permite generar y guardar la información en forma permanente para luego imprimir los resultados.

1.02 Justificación

El proceso de atención a los pacientes desde que ingresan a ADENTAL hasta cuando salen es lo que se desea mejorar, a través de la implementación de un software, que almacene la información de los pacientes, control de las historias clínicas y presente análisis estadísticos que ayuden a la toma de decisiones por parte de los doctores.

Se justificara la calidad de los datos y la información brindada orientada a una adecuada toma de decisiones.

Es necesario la implementación de un inventario para poder verificar la información de los insumos y equipos de manera cuidadosa, detallada y segura. La implementación

de este software permitirá solucionar los procesos de búsquedas y el almacenamiento de estas debido a que este software contara con una base de datos donde quedaran todos los registros de los pacientes, como antecedentes personales y familiares programa de formación y a qué tipo de régimen pertenece.

El flujo de información será más rápido y optimo, evitando con ello cuellos de botella y tiempos innecesarios de espera por tener la información a la mano.

1.03 MATRIZ T

La siguiente matriz, muestra la construcción de un árbol de problemas donde se muestra un análisis de la situación actual, describiendo las fuerzas bloqueadoras y las fuerzas impulsoras que ayudan a mejorar la situación actual de la Clínica Odontológica.

En la Matriz T, podemos observar que la apertura de una Historia Clínica es muy obsoleta a eso se pierde tiempo en la atención del Odontólogo por no haber obtenido con anterioridad un turno para así evitar las esperas innecesarias, con el uso de la tecnología podemos optimizar procesos existentes.

Las fuerzas bloqueadoras obstaculizarían el proceso de cambio con lo cual la Clínica tendría un bajo volumen de pacientes y esto demostraría un descontento de los usuarios.

Análisis Fuerzas T

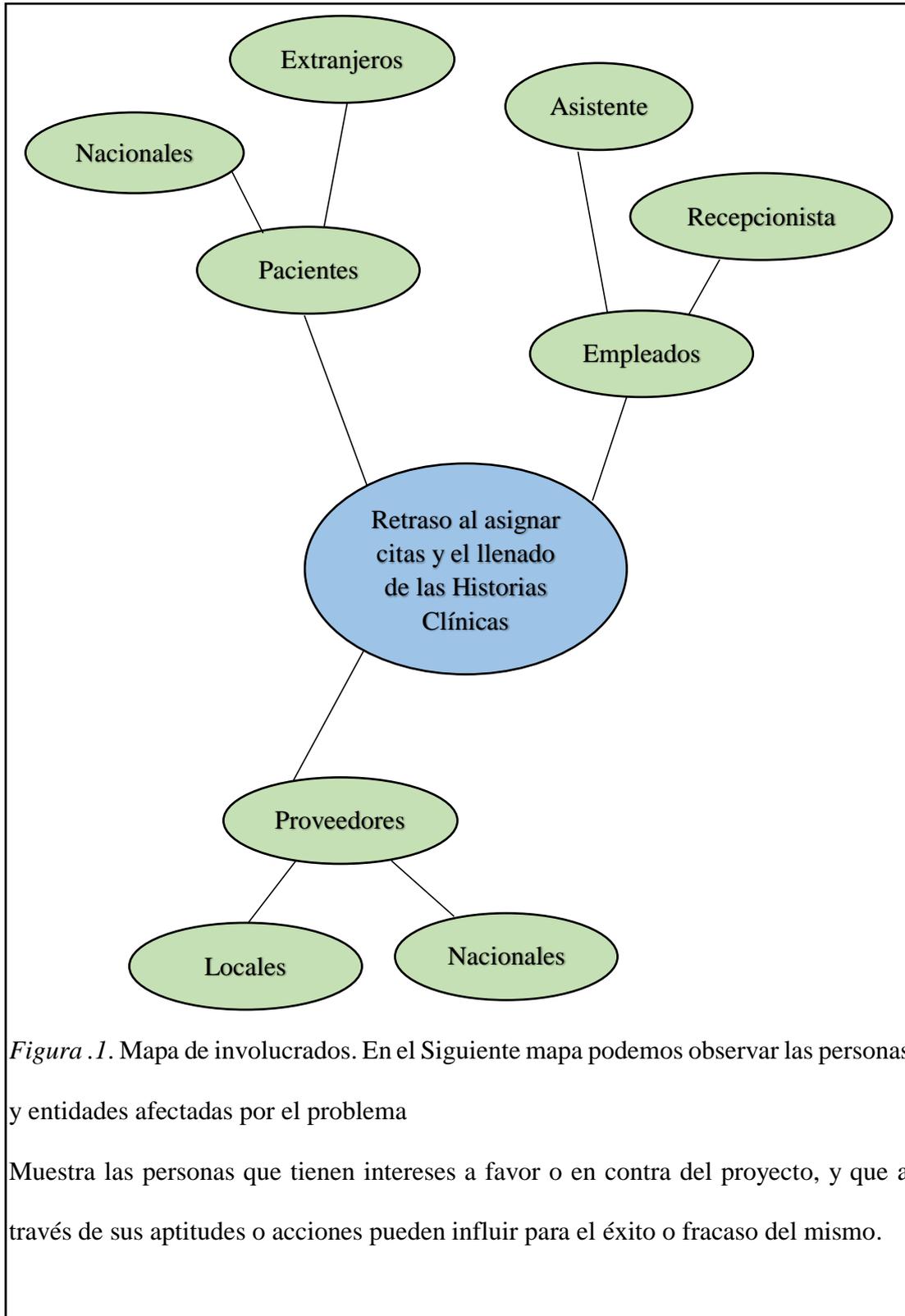
Tabla 1. Análisis de fuerza T

ANALISIS FUERZAS T

Situación Empeorada	Situación Actual				Situación Mejorada
Un masivo incremento de pacientes por mala recepción de toma de las turnos	Alto índice de pacientes en pérdida de tiempo en tomar un turno.				Pacientes muy satisfechos por el nuevo sistema de toma de turnos sin pérdida de tiempo
Fuerzas Impulsadoras	1	PC	1	PC	Fuerzas Bloqueadoras
Atención eficiente hacia los pacientes	3	5	4	2	Una mala atención de la secretaria
Incentivar a los pacientes visitar con más frecuencia al Doctor	1	4	4	4	Bajos Ingresos económicos de las familias
Campaña de automatizar el registro de pacientes	1	4	4	5	Pacientes insatisfechos
Brindar la ayuda necesaria al paciente para que este actualizada sus datos personales.	3	4	5	4	El paciente no facilita la información básica para el registro de la Historia Clínica
Impulsar el uso de una herramienta tecnológica	2	5	5	4	No existe un adecuado seguimiento de registro de datos de pacientes
Aumentar los ingresos de la Clínica Odontológica	4	4	4	4	Poca innovación tecnológica de la Clínica

Capítulo II: Análisis de Involucrados

2.01 Análisis de los involucrados





Pacientes:- Influye en el problema ya que pierde tiempo en ir a la clínica y corre el riesgo de que no sea atendida por la alta afluencia de personas.

Proveedores:- Influye en que acude a dejar los insumos en horas no adecuadas ya que no hay una persona encargada de receptar los productos por lo que se está atendiendo a los pacientes

Empleados:- Son de ayuda primordial en la Clínica por las varias actividades que se hacen en la misma.

2.02. MATRIZ DE ANALISIS DE INVOLUCRADOS

Tabla 2.

Matriz de involucrados

Actores Involucrados	Intereses sobre el problema central	Problemas Percibidos	Recursos, Mandatos y Capacidades	Intereses sobre el Proyecto	Conflictos Potenciales
Empleados	Mejorar en la atención para con los Pacientes	Pérdida de tiempo en la atención	Infraestructura de la empresa	Facilidad entre Clínica y pacientes	Fácil acceso a la información de paciente
Pacientes	Más tiempo en la recepción de información	Falta de información de la recepcionista	Ampliación salas de espera	Paciente satisfecho por la atención	Agilizar tiempos adecuados de espera
Proveedores	Las instituciones no van en el tiempo estipulado de la cita	Demora por la entrega de insumos	Capacitación antes de visitar la clínica	Facilitar el manejo de los registros	Entrega ineficiente

Podemos observar en las figuras, que en la toma de citas y el llenado de las Historias Clínicas con los empleados, pacientes y proveedores y con la comunidad, los cuales están afectados directa o indirectamente por el proyecto. Además indica los mandatos, recursos, políticas y prioridades que son requeridos y deseados para poder dar la solución al problema.



Capítulo III: Problemas y Objetivos

En las siguientes figuras, se muestra un análisis de los problemas y los objetivos que se pretenden resolver o alcanzar con el fin de llevar a cabo el proyecto.

3.01 Construcción del árbol de problemas

La siguiente figura describe el problema central, y analiza la relación entre causa y efecto del problema central y los problemas percibidos por los involucrados. Aquí podemos observar que el retraso en otorgar turnos y el llenado manual de las Historias Clínicas, tiene causas negativas las mismas que tienen efectos que no favorecen a la clínica para que mejoren la asistencia de más pacientes.

3.02 Construcción del árbol de objetivos.

La figura siguiente describe los procesos que se quieren cumplir mediante un árbol de objetivos, convirtiendo el problema central en un propósito, sus causas en medios y sus efectos en fines. Se puede observar que el propósito es generar citas a tiempo y el llenado de las historias clínicas, ayudado de la tecnología y la comunicación.

ARBOL DE PROBLEMAS

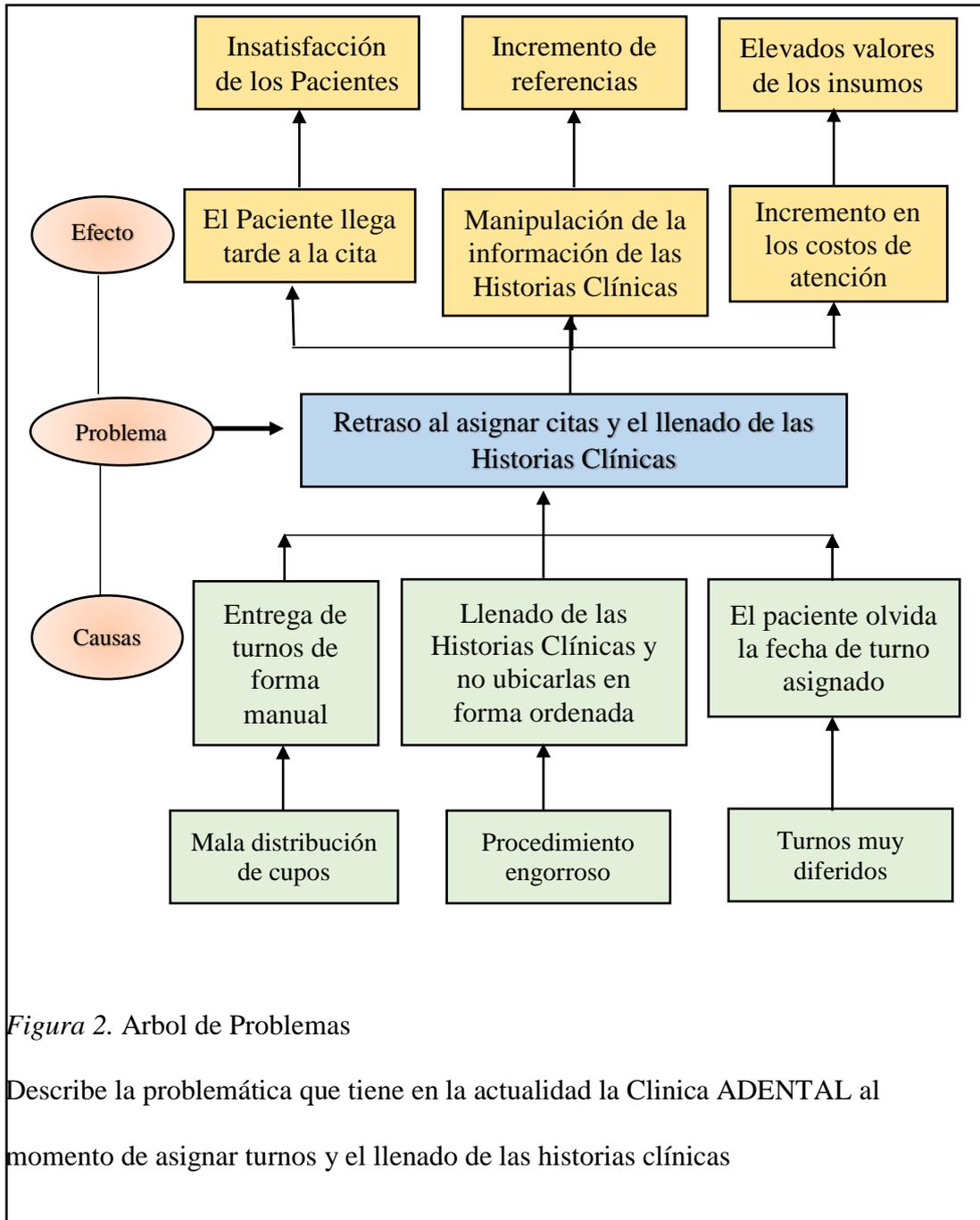
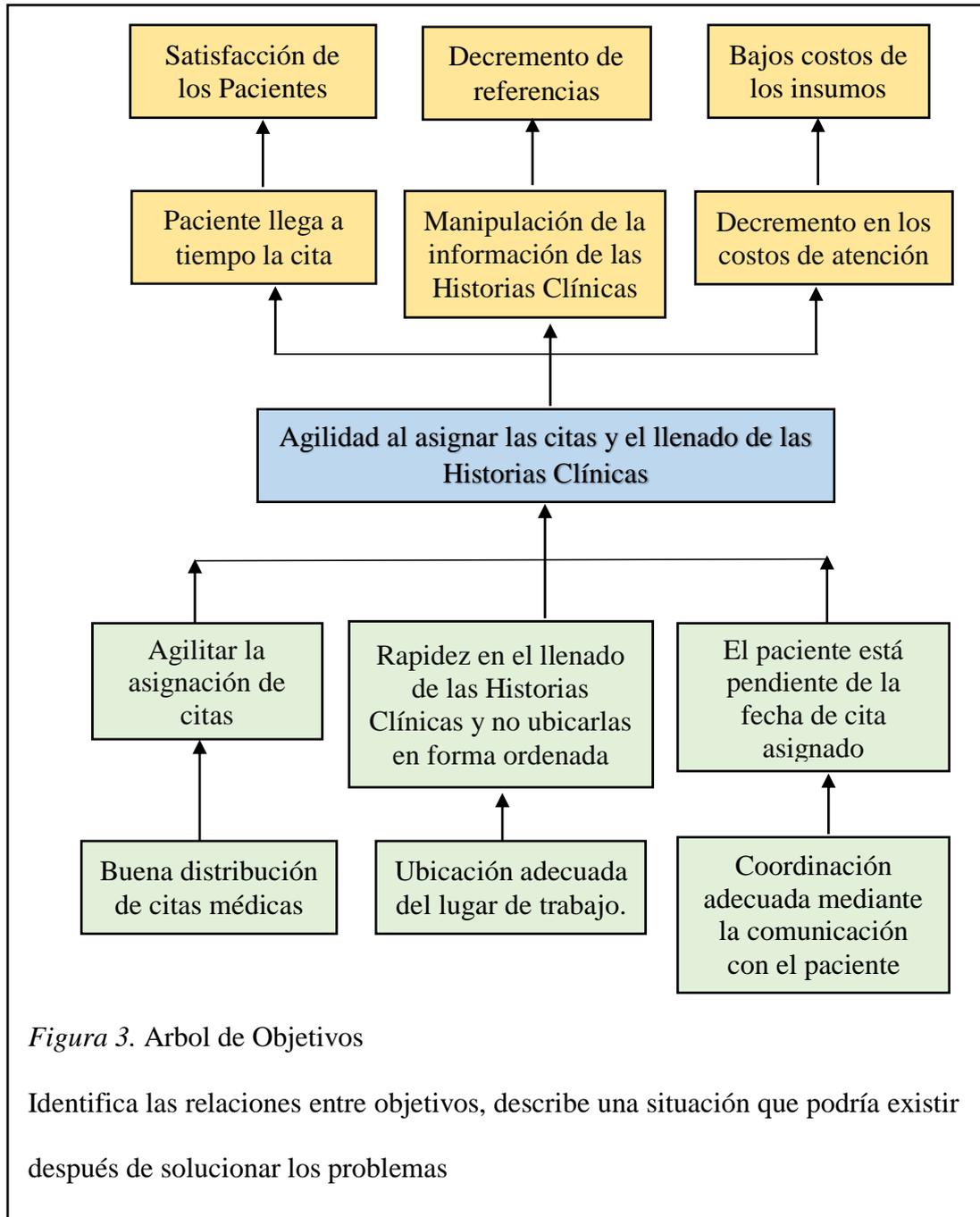


Figura 2. Arbol de Problemas

Describe la problemática que tiene en la actualidad la Clinica ADENTAL al momento de asignar turnos y el llenado de las historias clínicas

ARBOL DE OBJETIVOS



Capítulo IV: Análisis de Alternativas

4.01 Matriz de análisis de alternativas

La matriz siguiente identifica las estrategias y alternativas a partir del árbol de objetivos, que si son ejecutadas, podrían promover el cambio de la situación actual a la situación deseada.

Tabla 3.

Matriz de análisis de alternativas

Objetivos	Impacto sobre el propósito	Factibilidad técnica	Factibilidad Financiera	Factibilidad Social	Factibilidad Política	Total	Categorías
Trámites en General dificultoso	5	4	3	4	4	20	Alta
Tiempo de espera prolongado	5	4	4	4	3	20	Alta
Poca información al paciente sobre proceso de atención	5	4	3	5	3	20	Alta
Equipo informático ineficaz	5	4	5	4	4	22	Alta
Personal no satisface a paciente	4	4	5	5	4	22	Alta
Infraestructura inadecuada	4	5	3	4	5	21	Alta
	28	25	23	26	23	144	

4.02 Matriz de análisis del impacto de los objetivos

La tabla siguiente muestra un análisis de cuál es el impacto de la aplicación en los objetivos del proyecto, donde se obtiene la visión de la situación deseada y las estrategias que se aplicarán para conseguirla.

En la matriz se obtiene un puntaje alto, en factibilidad, impacto ambiental, relevancia y sostenibilidad, además se obtiene un alto grado de participación del género femenino, ayudando a igualdad de género. (Ver Anexo A.01)

4.03 Diagrama de estrategias

A continuación se establece la estructura y alcance de las estrategias que intervienen en el proyecto, expresadas en conjuntos de objetivos que son considerados como factibles de realización, y están dentro de las posibilidades del proyecto.

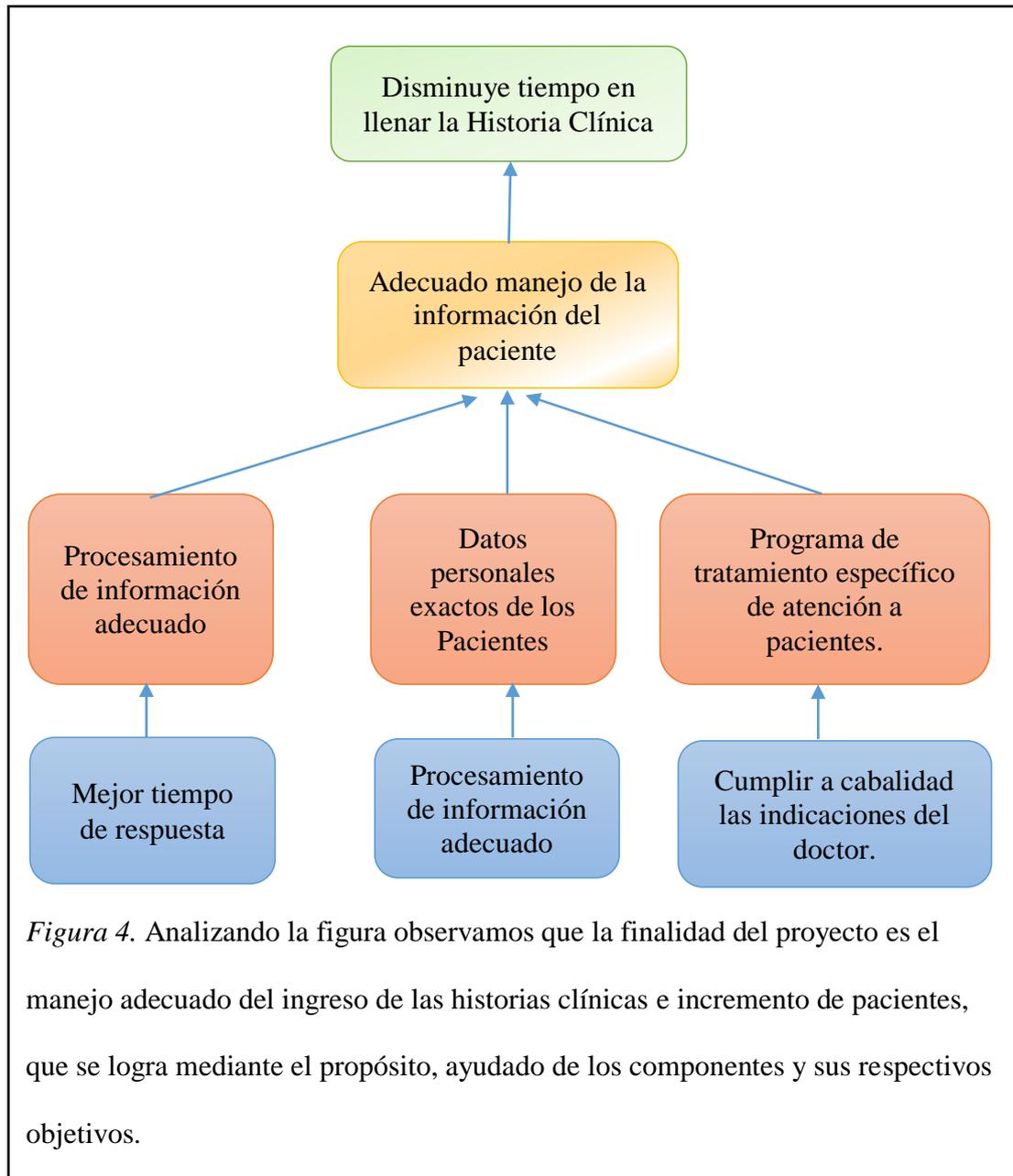


Figura 4. Analizando la figura observamos que la finalidad del proyecto es el manejo adecuado del ingreso de las historias clínicas e incremento de pacientes, que se logra mediante el propósito, ayudado de los componentes y sus respectivos objetivos.



4.04 Matriz del marco lógico

En la siguiente matriz se describe que se desea lograr con el proyecto, ahorrar tiempo en movilidad y disminuir el error odontológico para la intervención desde la expedición de un documento hasta la intervención del doctor, como se alcanzarán el propósito y sus componentes, que factores externos son indispensables para el éxito del mismo, como se pretende medir el cumplimiento de los objetivos y resultados, como se puede obtener la información y finalmente que recursos son necesarios para la ejecución del proyecto.

La matriz describe el fin del proyecto: incremento de pacientes y de tratamientos dentales que asisten con mayor frecuencia, se especifica los indicadores, los medios de verificación y los supuestos necesarios para su realización.

Además se cita el propósito del proyecto, los componentes del proyecto y las actividades del proyecto, evaluando cada una de ellas para poder llevar a cabo la ejecución del proyecto.

04.05 Matriz del marco lógico

Tabla 4.

Matriz Marco Lógico

RESUMEN NARRATIVO	INDICADORES	MEDIOS DE VERIFICACION	SUPUESTO
FIN DEL PROYECTO Reducir los tiempos en asignar turnos y el uso de papel para el registro de pacientes en las Historias Clínicas	- Cantidad de uso de papel - Aumento de ingresos para la Clínica	Informe de Historias clínicas de "ADENTAL" Clínica Odontológica	- El personal no tendrá dificultad de tiempo en asignar un turno al paciente que solicite ese servicio de salud
PROPOSITO DEL PROYECTO Mejorar el tiempo en el registro y asignación de turnos a pacientes	- Los pacientes pueden llegar con facilidad - Aumento de pacientes a la Clínica Odontológica	- Ecuasta a los empleados sobre el uso de herramientas informticas -	- Los precios de los servicios odontológicos se mantienen estables
COMPONENTES DEL PROYECTO Baja del índice de perdida de turnos	- Financiamiento para hacer publicidad. - Conocimiento de los empleados de sistemas informáticos.	- Recolección de datos arrojados por las Historias clínicas de "ADENTAL" Clínica Odontológica	- Aumento de la oferta y calidad de los servicios de Odontología y de la demanda - Concientización a la población acerca de la necesidad de buscar atención Odontológica

Capítulo V: Propuesta

5.01 Justificación del software

La Propuesta del Desarrollo de una Aplicación para el Registro de Historias Clínicas Electrónicas (HCE) para La Clínica Odontológica ADENTAL, se basa en mejorar los procesos actuales y optimizar los recursos internos de la Clínica, como mejorar los procesos del control documentario de las Historias Clínicas (HC). Este sistema será un aplicativo WinForm que nos permitirá registrar la información de la Historias Clínicas (HC).

El registro de la información se realizara a través del Módulo de Historia Clínica, al cual se accederá a través del formulario para aplicación WinForm, de igual podrán acceder las demás áreas Cuerpo Odontológico, Especialistas para el registro o consulta de la información.

5.02 Análisis y diseño

5.02.01 Diagramas de caso de uso

El diagrama de casos de uso representa la forma en como un Cliente (Actor) opera con el sistema en desarrollo, además de la forma, tipo y orden en como los elementos interactúan (operaciones o casos de uso).

Un diagrama de casos de uso consta de los siguientes elementos:

Actor.

Los actores representan un tipo de usuario del sistema. Un actor en un diagrama de casos de uso representa un rol que alguien puede estar jugando, no un individuo particular por lo tanto puede haber personas particulares que puedan estar usando el sistema de formas diferentes en diferentes ocasiones.

Caso de Uso

Es una tarea que debe poder llevarse a cabo con el apoyo del sistema que se está desarrollando.

Relaciones.

Asociación.

Hay una asociación entre un actor y un caso de uso si el actor interactúa con el sistema para llevar a cabo el caso de uso. Dicha relación se denota con una flecha simple.

Dependencia o Instanciación.

Es una forma muy particular de relación entre clases, en la cual una clase depende de otra, es decir, se instancia (se crea). Dicha relación se denota con una flecha punteada.

Generalización.

Este tipo de relación es uno de los más utilizados, cumple una doble función dependiendo de su estereotipo, que puede ser de Uso (<<uses>>) o de Herencia (<<extends>>).

Este tipo de relación está orientado exclusivamente para casos de uso (y no para actores).

El siguiente diagrama de caso de uso, *DCU-001*, expresa el funcionamiento de una forma global que actualmente se encuentra funcionando La Clínica Odontológica ADENTAL.

Caso de uso General

Tabla 5.

Caso de uso general

Nombre

REGLAS DEL NEGOCIO

Descripción

En la regla de negocio posee varios parámetros de entrada de validación y una acción de salida. Este caso de uso es lo que representa dentro de la Clínica Odontológica "ADENTAL" y se realiza todos los días

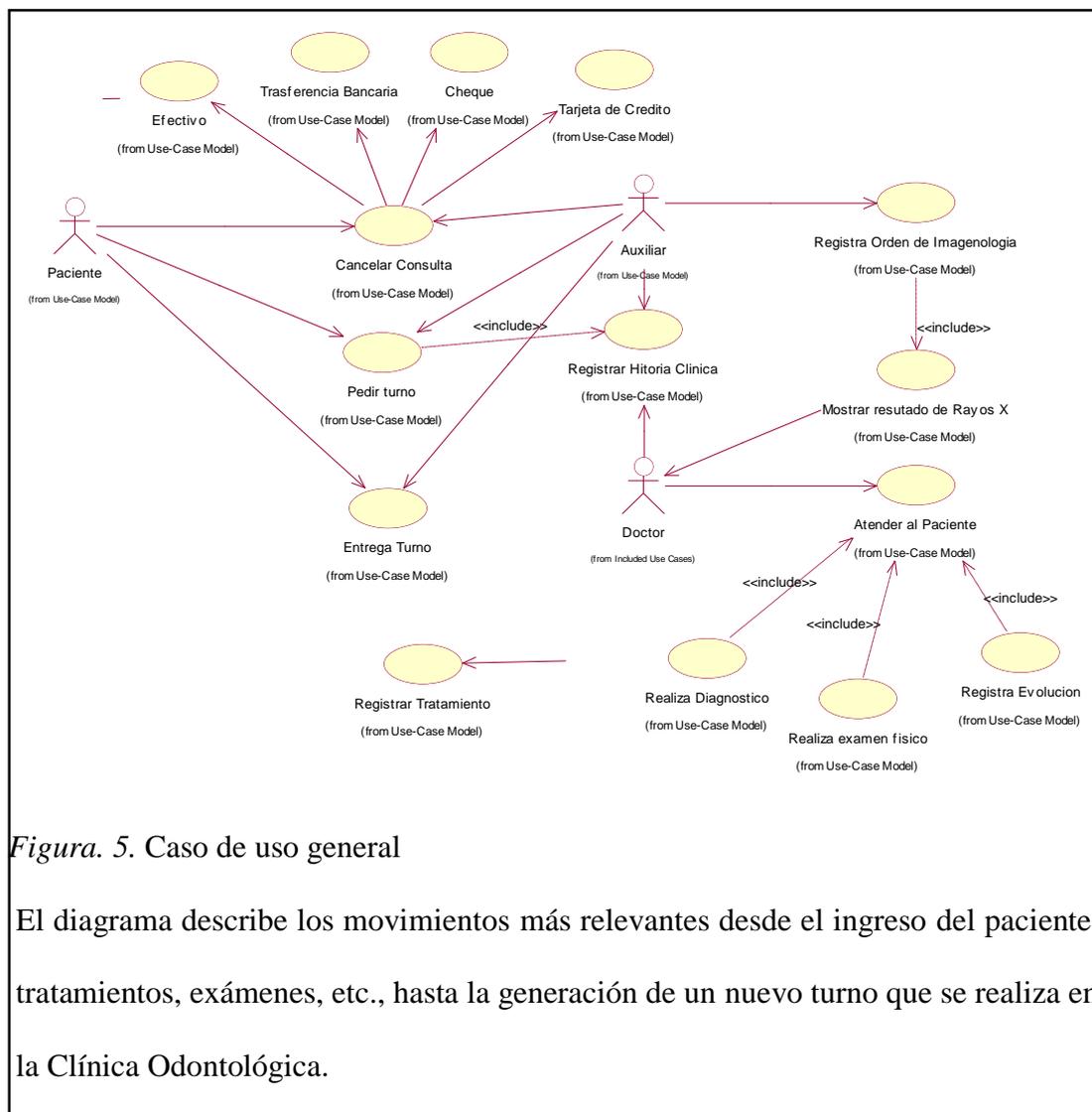


Figura. 5. Caso de uso general

El diagrama describe los movimientos más relevantes desde el ingreso del paciente, tratamientos, exámenes, etc., hasta la generación de un nuevo turno que se realiza en la Clínica Odontológica.

5.02.02. Diagrama de Realización

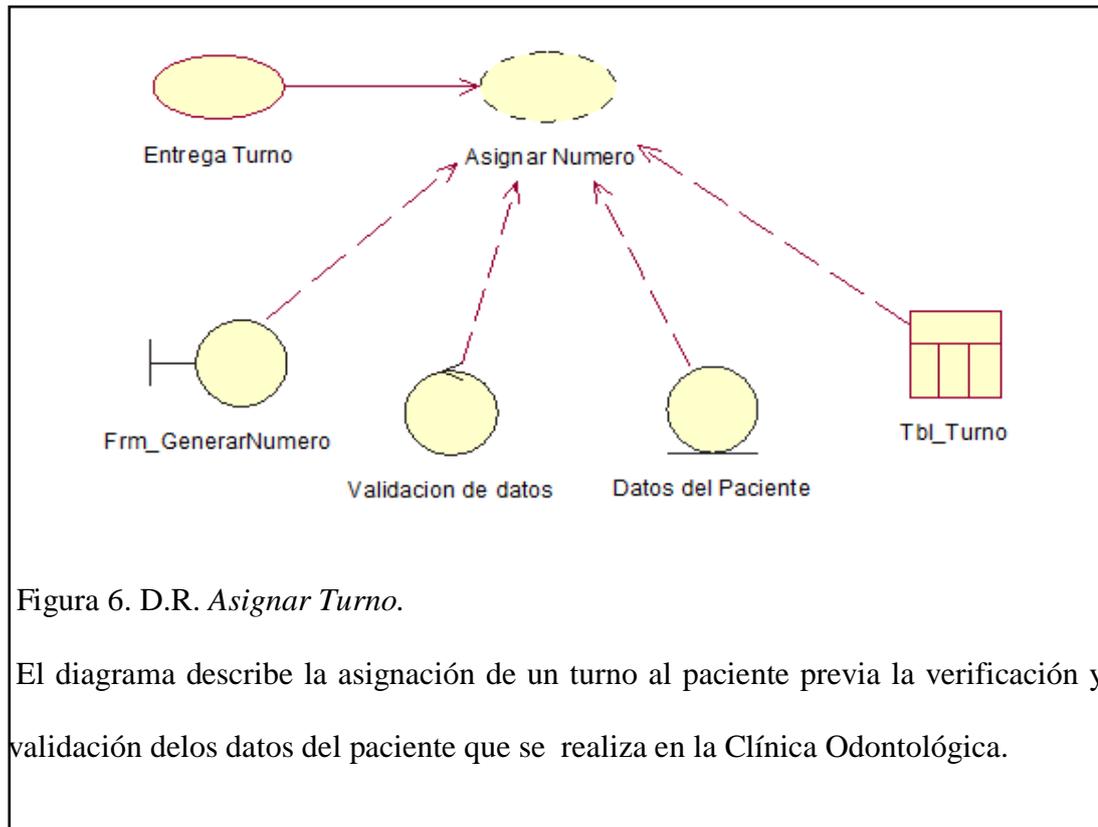
Los componentes son módulos de código, así que los diagramas de componentes vienen a ser análogos físicos a los diagramas de clase. Muestran como está organizado un conjunto de componentes y las dependencias que existe entre ellos.

Determinación de caso de uso ASIGNACIÓN DE TURNO

Tabla 6.

Asignar Turno

Nombre	ASIGNAR TURNO
Actores	Auxiliar
Descripción	La Auxiliar podrá visualizar un formulario, el cual debe llenar para ingresar los datos para generar un turno al paciente.
Flujo de Eventos	<ul style="list-style-type: none"> • La auxiliar debe ingresar el sistema. • Escoger en el menú la opción: "Ingresar Turno"
Post condiciones	<ul style="list-style-type: none"> • Falla del sistema • Mensaje de aviso de éxito o fracaso de la transacción.



Determinación de caso de uso REGISTRAR PAGO CONSULTA

Tabla 7.

REGISTRAR PAGO CONSULTA

Nombre	REGISTRAR PAGO CONSULTA
Actores	Auxiliar
Descripción	La secretaria podrá visualizar un formulario, el cual debe llenar para registrar un nuevo Pago.
Flujo de Eventos	<ul style="list-style-type: none"> • La secretaria debe ingresar el sistema. • Escoger en el menú la opción: "Registrar Pago" • Mensaje de aviso de éxito o fracaso de la transacción
Post condiciones	<ul style="list-style-type: none"> • Falla del sistema • Ingreso de datos no válidos.

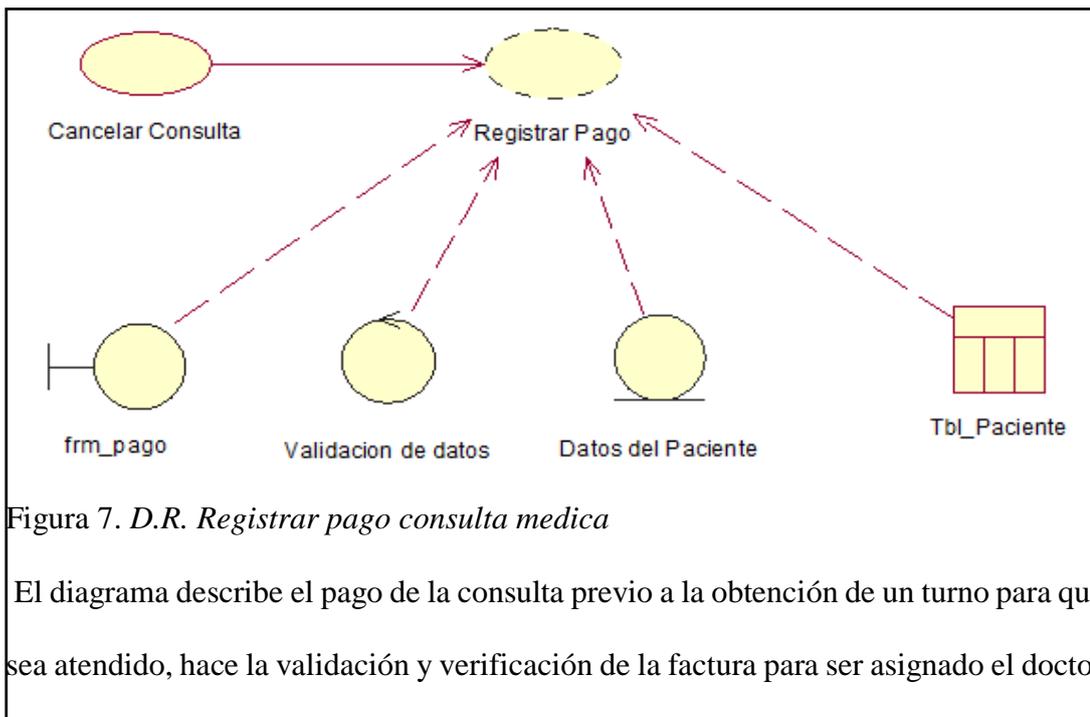


Figura 7. D.R. Registrar pago consulta medica

El diagrama describe el pago de la consulta previo a la obtención de un turno para que sea atendido, hace la validación y verificación de la factura para ser asignado el doctor

Determinación de caso de uso CREAR HISTORIA CLINICA

Tabla 8.

CREAR HISTORIA CLINICA

Nombre	CREAR HISTORIA CLÍNICA
Actores	Odontólogo
Descripción	El médico podrá visualizar un formulario, el cual debe llenar para ingresar los datos de una nueva Historia Clínica
Flujo de Eventos	<ul style="list-style-type: none"> • El Médico debe ingresar el sistema. • Escoger en el menú la opción: “Crear Historia Clínica” • Debe existir un paciente al que le pertenece la nueva Historia Clínica.
Post condiciones	<ul style="list-style-type: none"> • Falla del sistema • Ingreso de datos no validos

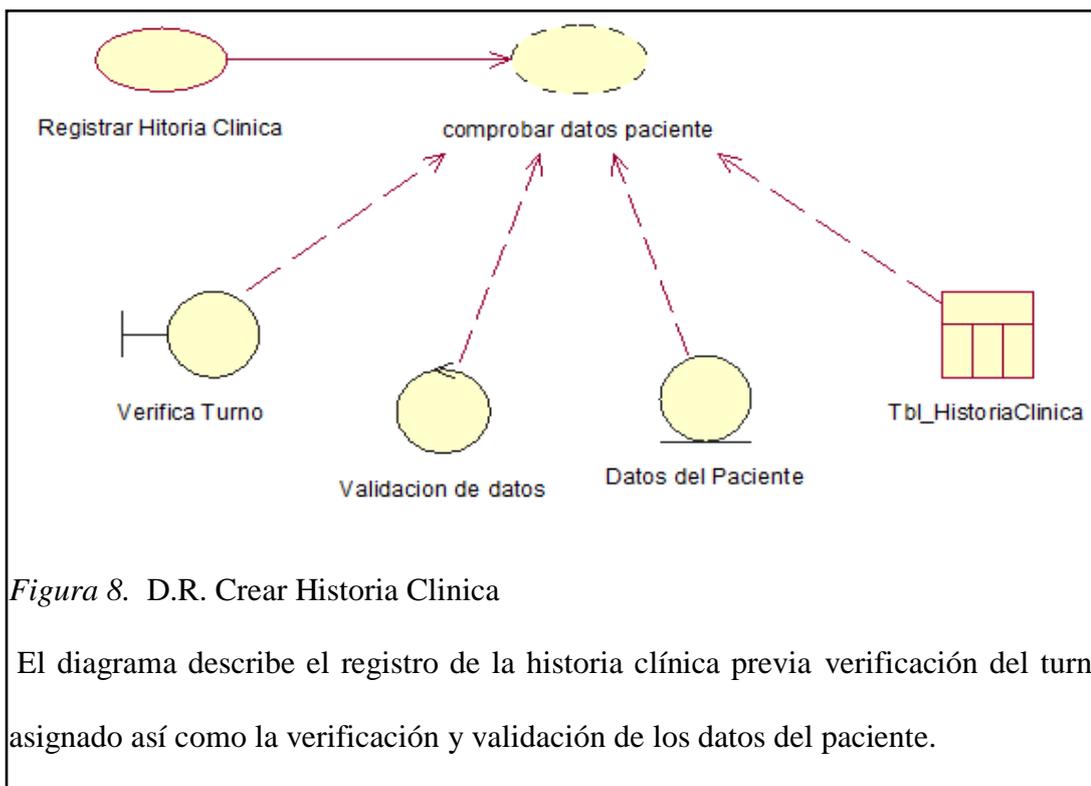


Figura 8. D.R. Crear Historia Clínica

El diagrama describe el registro de la historia clínica previa verificación del turno asignado así como la verificación y validación de los datos del paciente.

Determinación de caso de uso REGISTRAR DIAGNOSTICO

Tabla 9.

REGISTRAR DIAGNOSTICO

Nombre	REGISTRAR DIAGNOSTICO
Actores	Odontólogo
Descripción	Registrar los diagnósticos generados en la consulta médica y consultar historiales anteriores.
Flujo de Eventos	<ul style="list-style-type: none"> • El Odontólogo requiere registrar los diagnósticos generados en la consulta médica. • Los datos no coinciden
Post condiciones	<ul style="list-style-type: none"> • El paciente tiene un turno para consulta externa • Ingreso de datos no válidos.

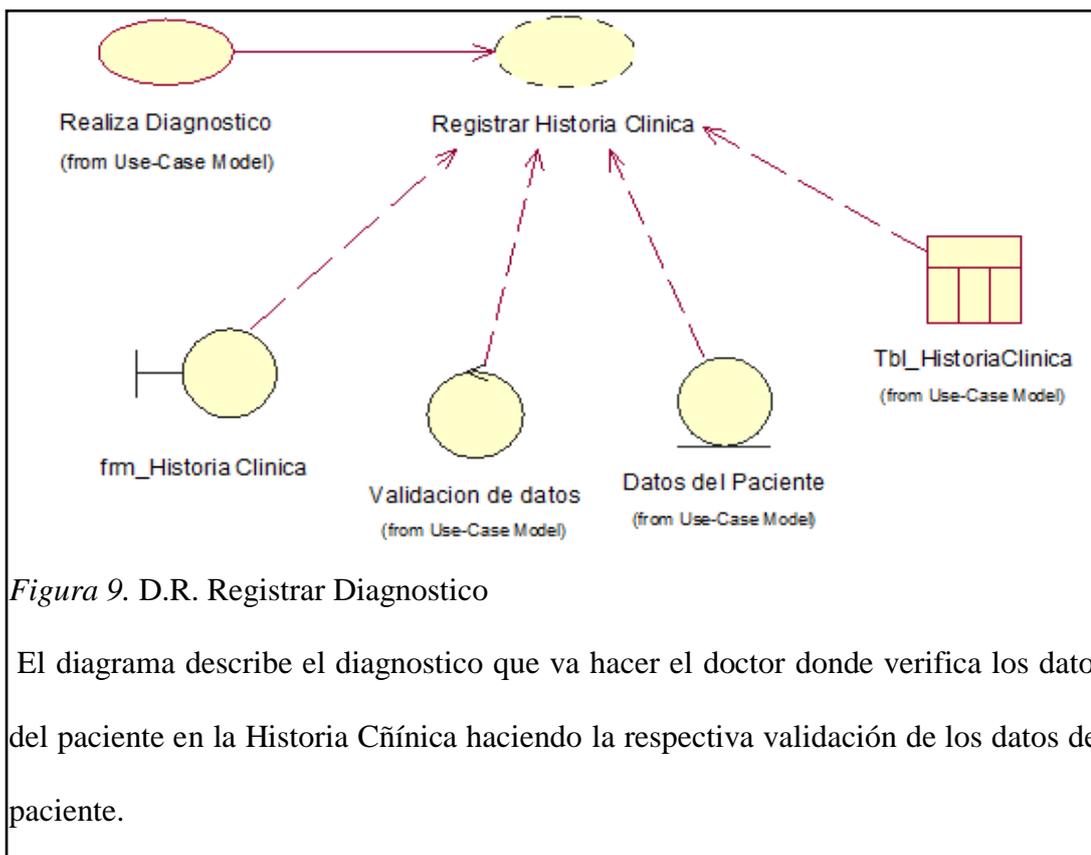


Figura 9. D.R. Registrar Diagnostico

El diagrama describe el diagnostico que va hacer el doctor donde verifica los datos del paciente en la Historia Clínica haciendo la respectiva validación de los datos del paciente.

Determinación de caso de uso REGISTRAR TRATAMIENTO

Tabla 10.

REGISTRAR TRATAMIENTO

Nombre	REGISTRAR TRATAMIENTO
Actores	Odontólogo
Descripción	El odontólogo hace el registro del tratamiento adecuado que se le va a realizar al paciente.
Flujo de Eventos	<p>Registrar tratamiento utiliza el caso de uso registrar diagnostico</p> <ul style="list-style-type: none"> • El odontólogo escoge la opción registrar tratamiento • Digite los datos del tratamiento
Post condiciones	<ul style="list-style-type: none"> • Falla del sistema • Ingreso de datos no válidos.

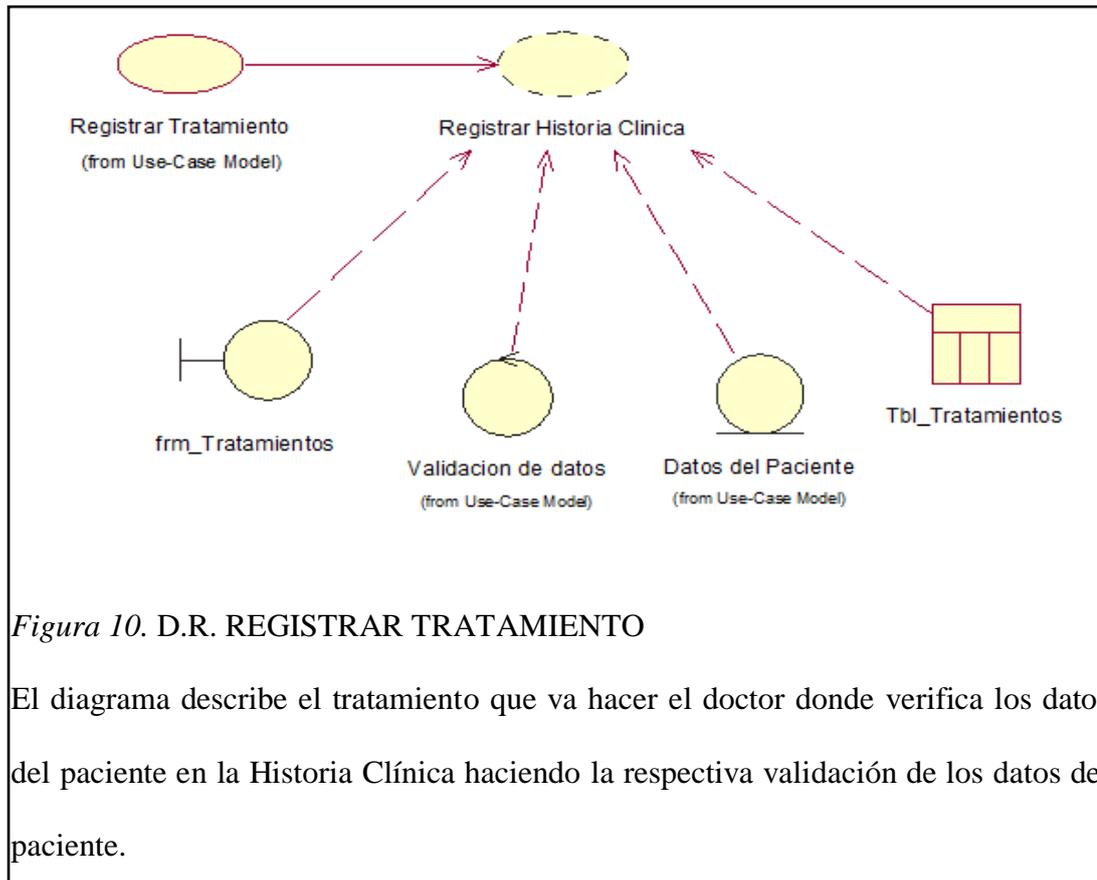


Figura 10. D.R. REGISTRAR TRATAMIENTO

El diagrama describe el tratamiento que va hacer el doctor donde verifica los datos del paciente en la Historia Clínica haciendo la respectiva validación de los datos del paciente.

5.02.03 Diagramas de secuencia

Los diagramas UML de secuencia y de colaboración (llamados diagramas de interacción) se utilizan para modelar los aspectos dinámicos de un sistema. Un diagrama de interacción consiste en un conjunto de objetos y sus relaciones, incluyendo los mensajes que se pueden enviar entre ellos.

Diagrama de Secuencia Asignar Turno

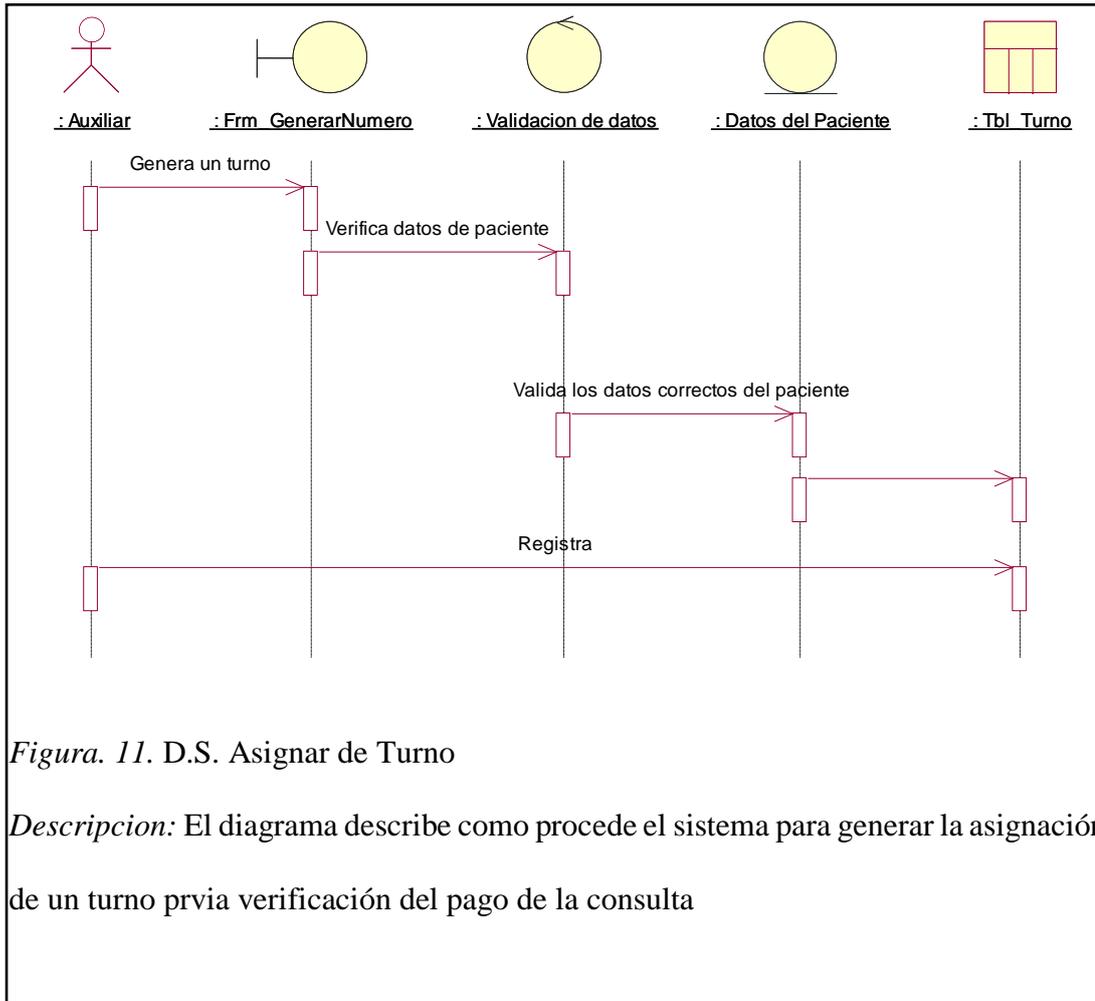


Figura. 11. D.S. Asignar de Turno

Descripcion: El diagrama describe como procede el sistema para generar la asignación de un turno prvia verificación del pago de la consulta

Diagrama de Secuencia Pago de Consulta

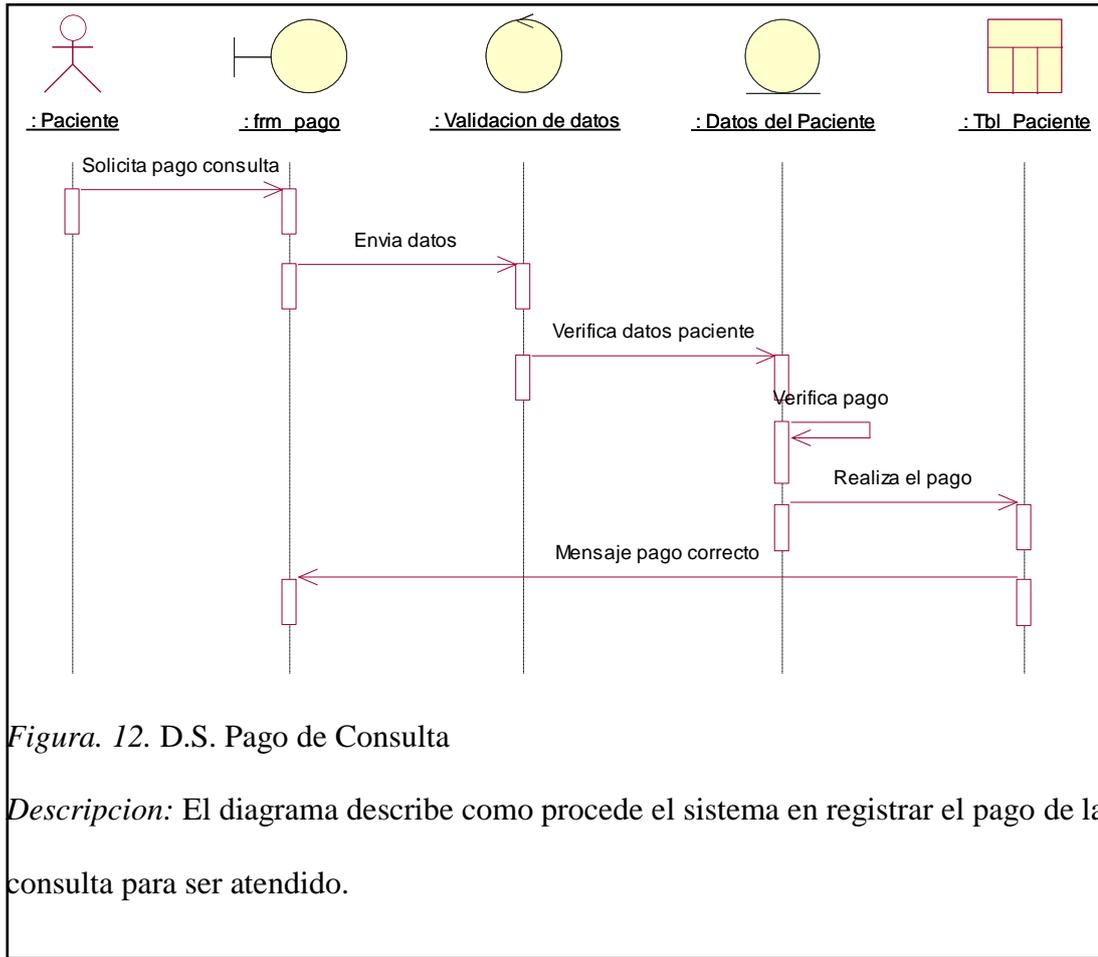


Figura. 12. D.S. Pago de Consulta

Descripcion: El diagrama describe como procede el sistema en registrar el pago de la consulta para ser atendido.

Diagrama de Secuencia Crear Historia Clínica

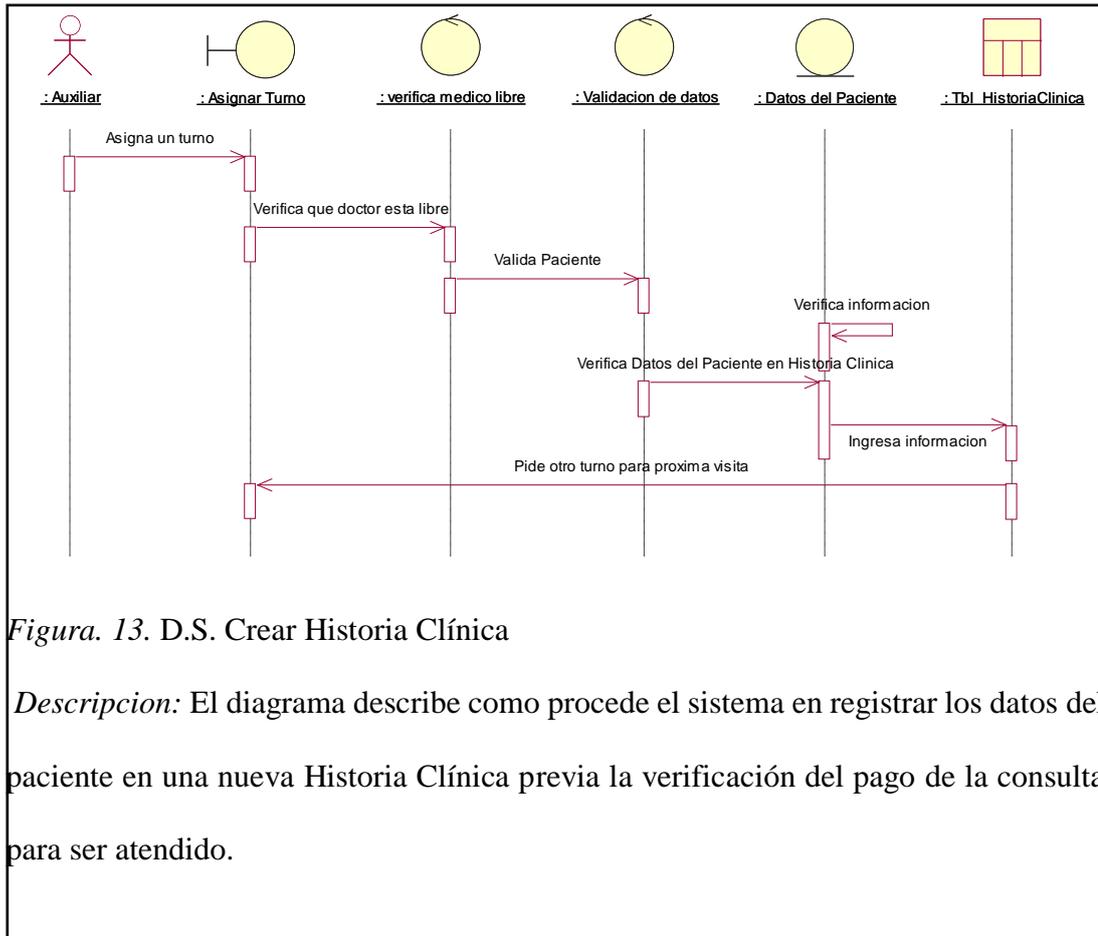


Figura. 13. D.S. Crear Historia Clínica

Descripcion: El diagrama describe como procede el sistema en registrar los datos del paciente en una nueva Historia Clínica previa la verificación del pago de la consulta para ser atendido.

Diagrama de Secuencia Registrar Diagnóstico

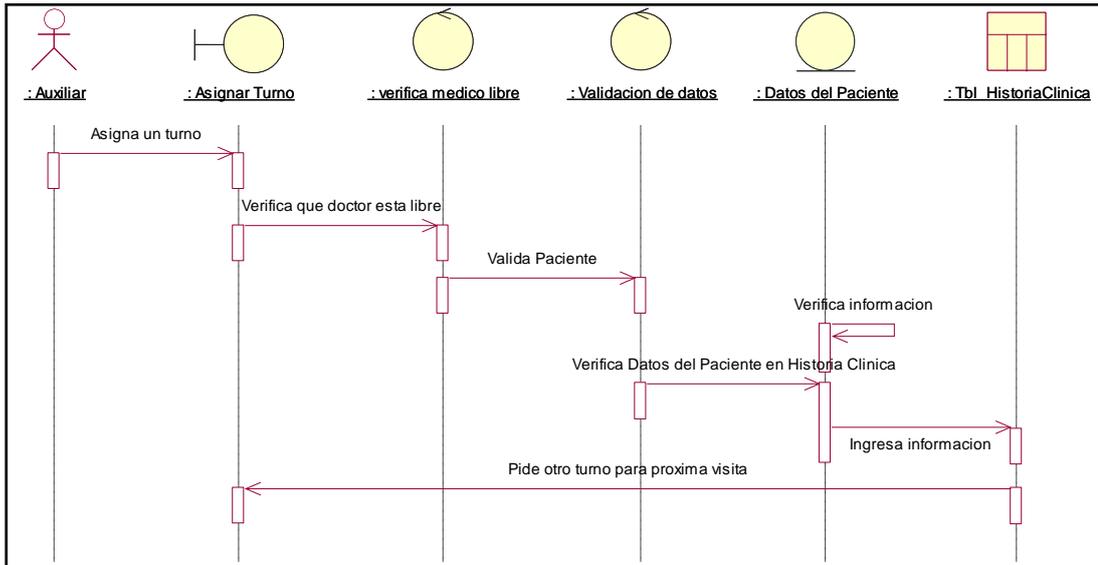


Figura. 14. D.S. Registrar Diagnóstico

Descripcion: El diagrama describe como procede el sistema en registrar el diagnostico que esta realizando el medico verificando los datos del paciente en la Historia Clínica previa la verificación del pago de la consulta para ser atendido.

Diagrama de Secuencia Registrar Tratamiento

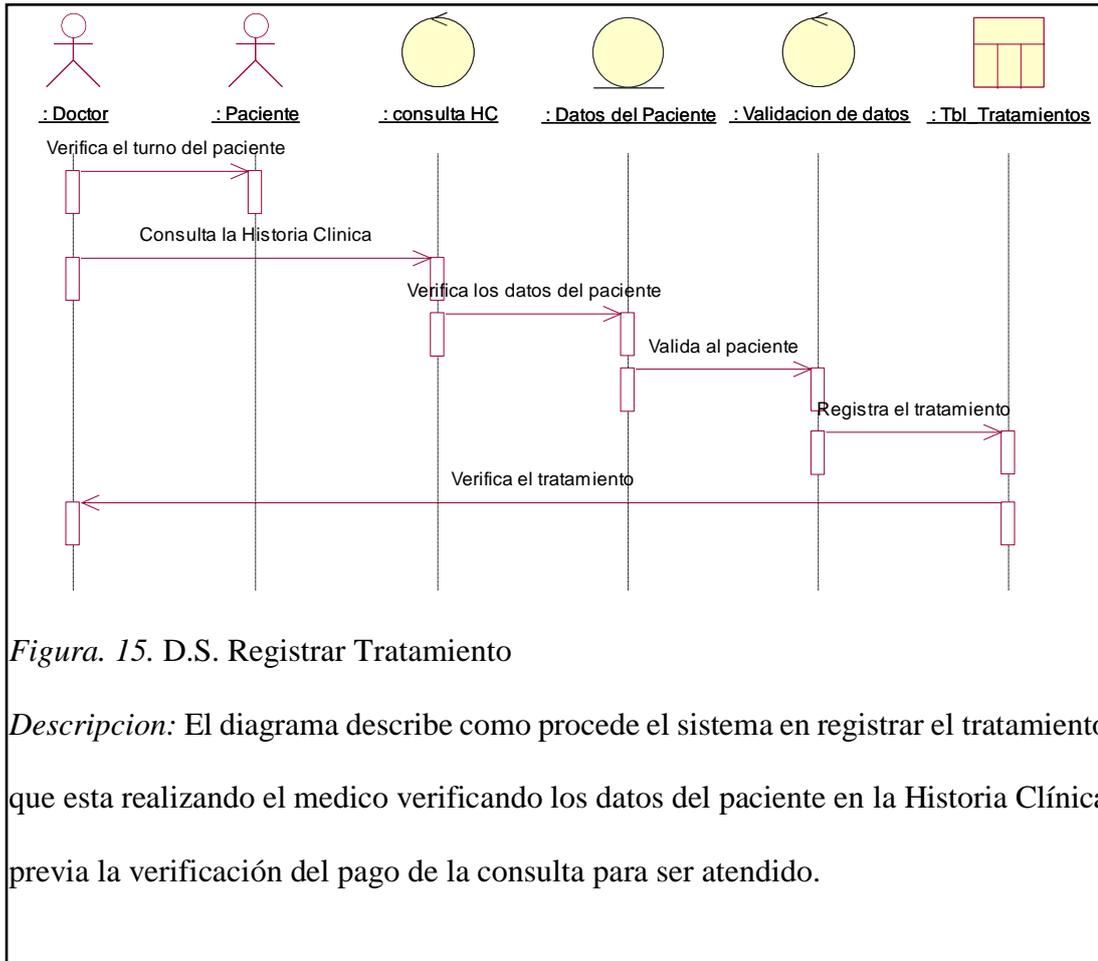


Figura. 15. D.S. Registrar Tratamiento

Descripcion: El diagrama describe como procede el sistema en registrar el tratamiento que esta realizando el medico verificando los datos del paciente en la Historia Clínica previa la verificación del pago de la consulta para ser atendido.

5.02.04 Diagramas de Colaboración

Diagrama de colaboración Asignar Turno

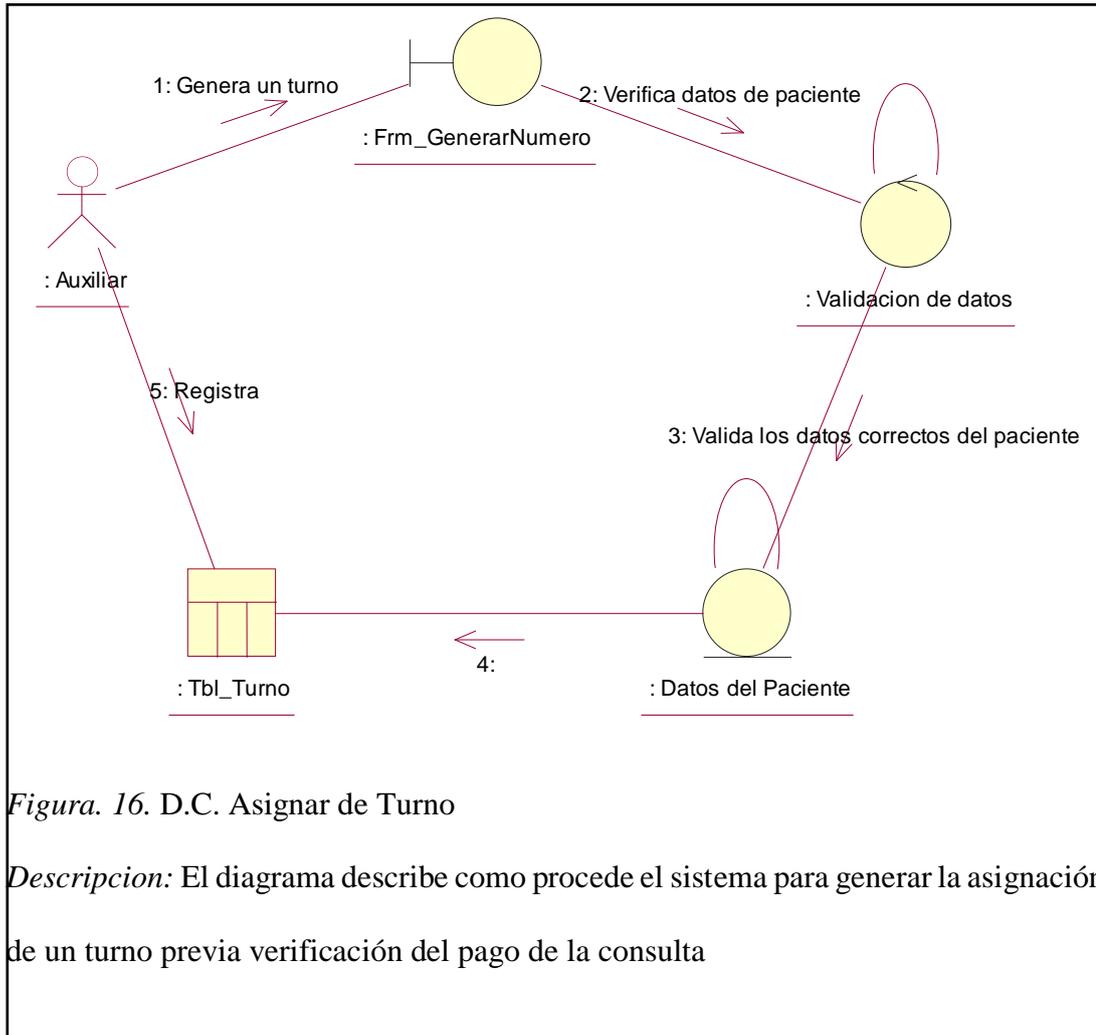


Figura. 16. D.C. Asignar de Turno

Descripcion: El diagrama describe como procede el sistema para generar la asignación de un turno previa verificación del pago de la consulta

Diagrama de colaboración Pago de Consulta

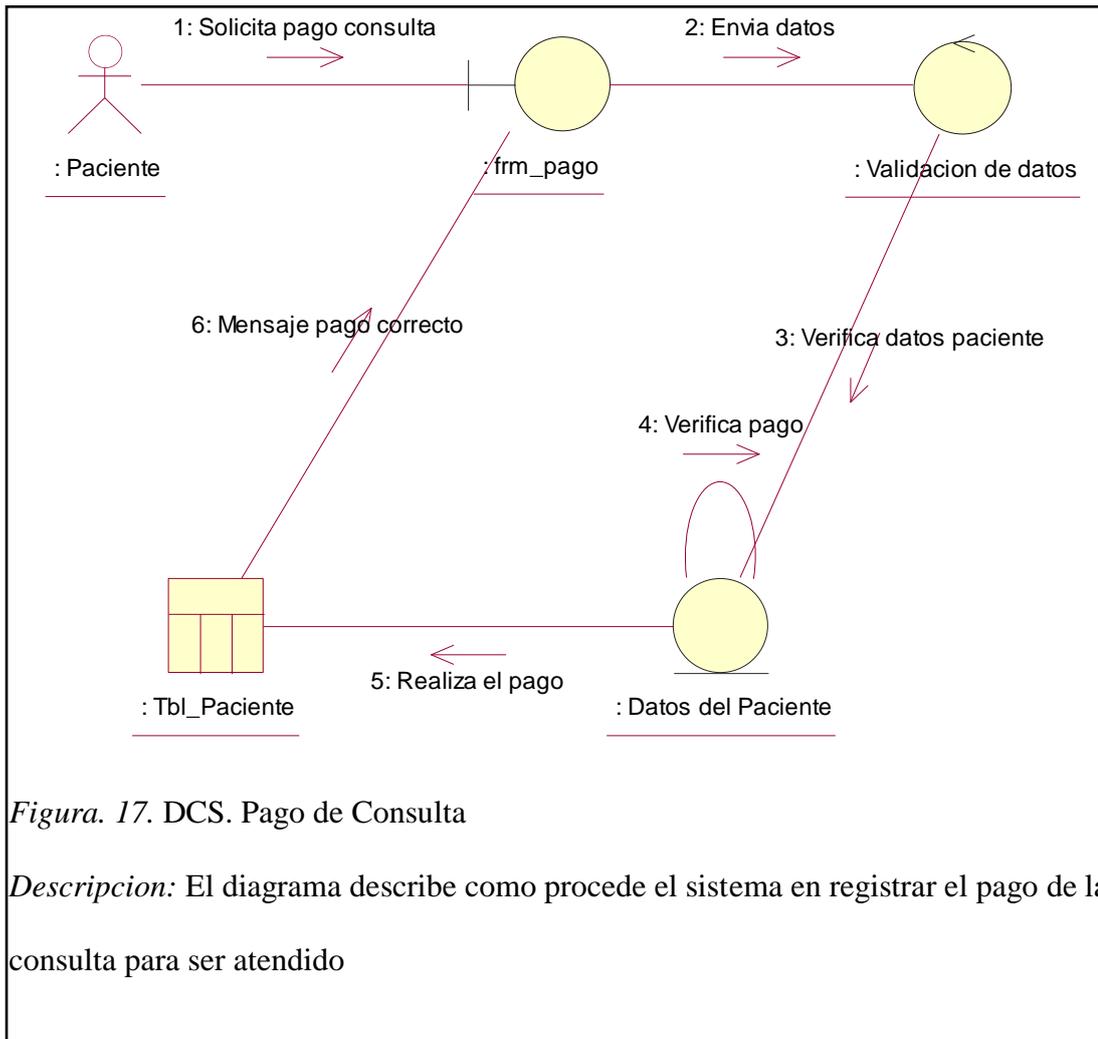


Figura. 17. DCS. Pago de Consulta

Descripcion: El diagrama describe como procede el sistema en registrar el pago de la consulta para ser atendido

Diagrama de colaboración Crear Historia Clínica

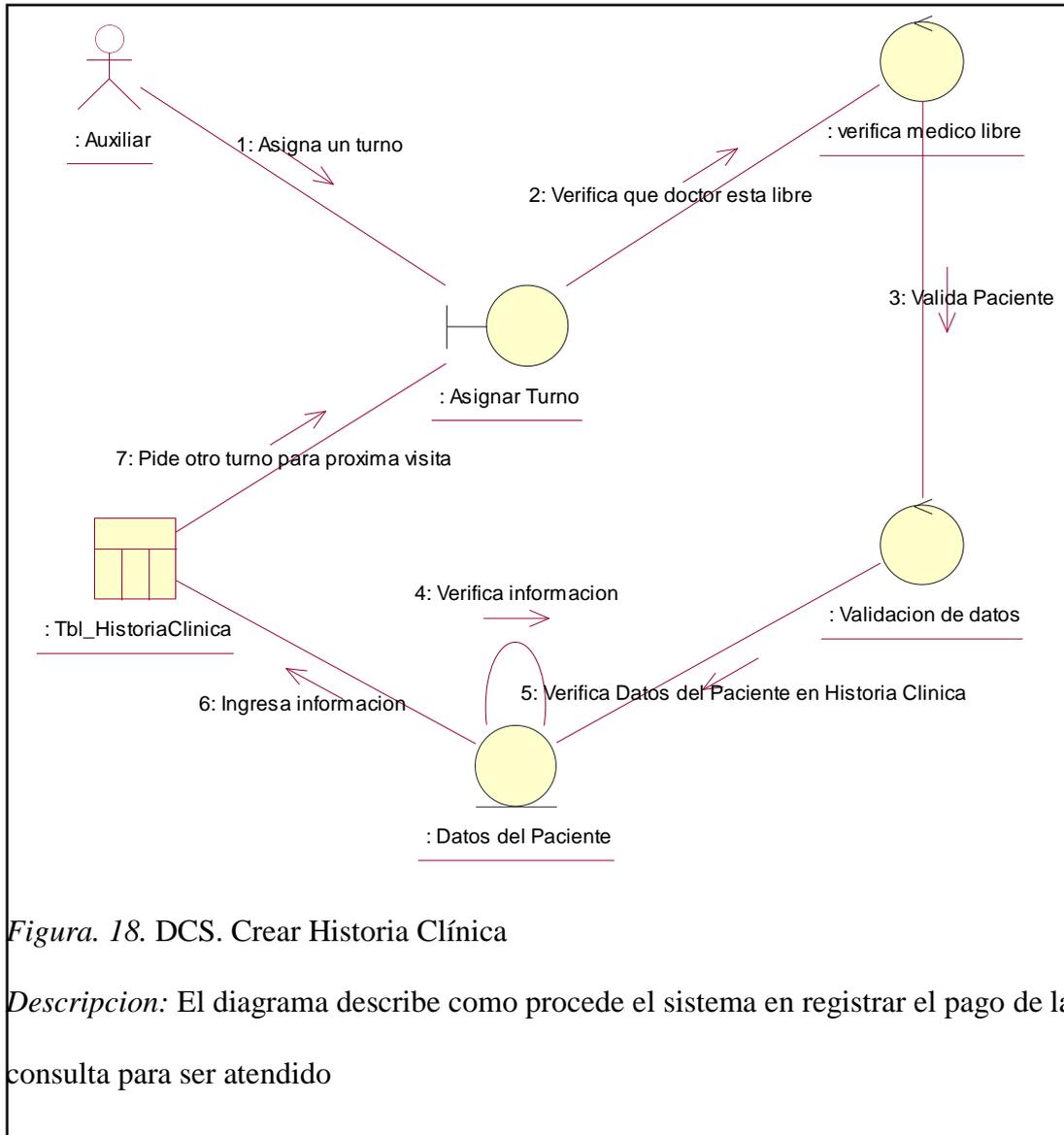


Figura. 18. DCS. Crear Historia Clínica

Descripcion: El diagrama describe como procede el sistema en registrar el pago de la consulta para ser atendido

Diagrama de colaboración Registrar Diagnóstico

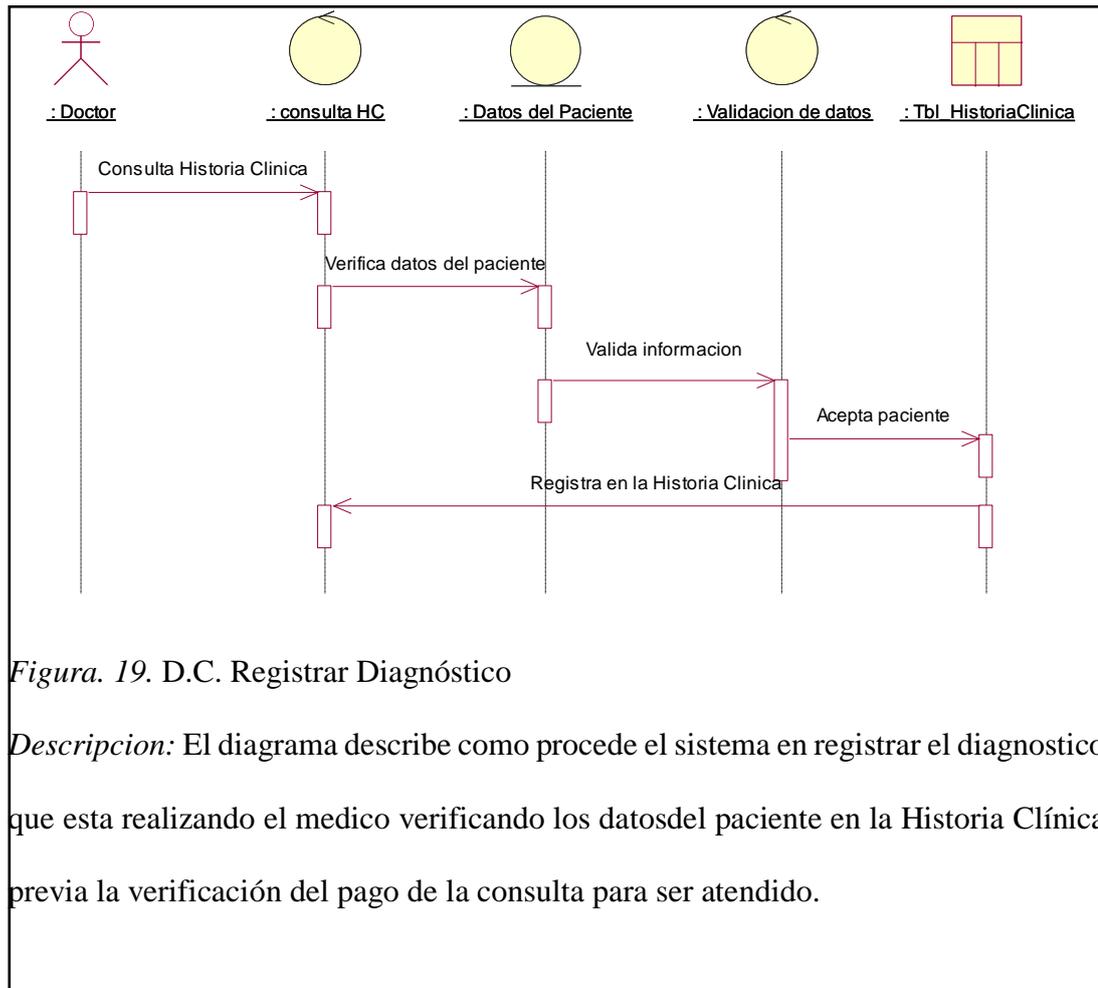


Diagrama de colaboración Registrar Tratamiento

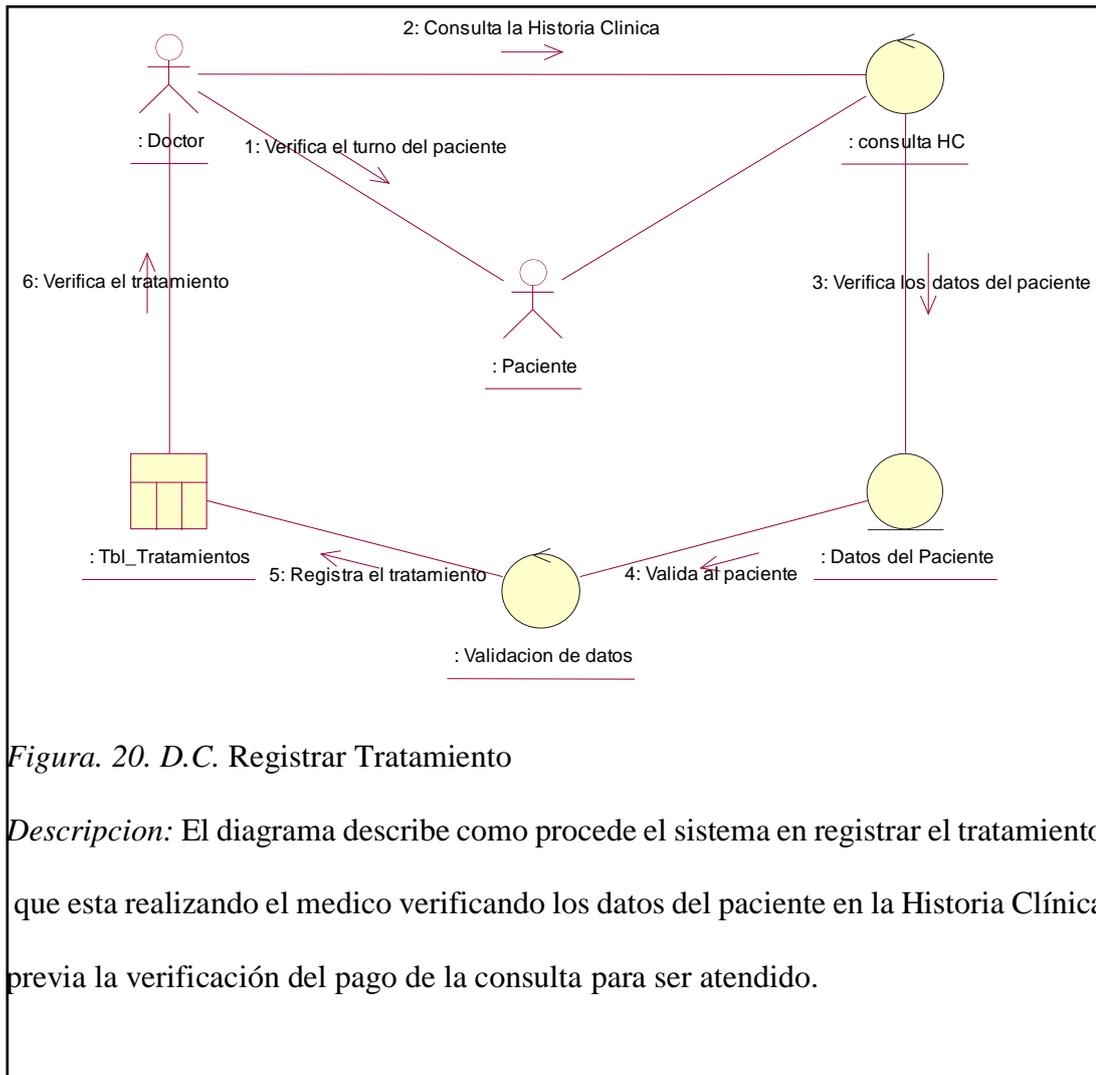
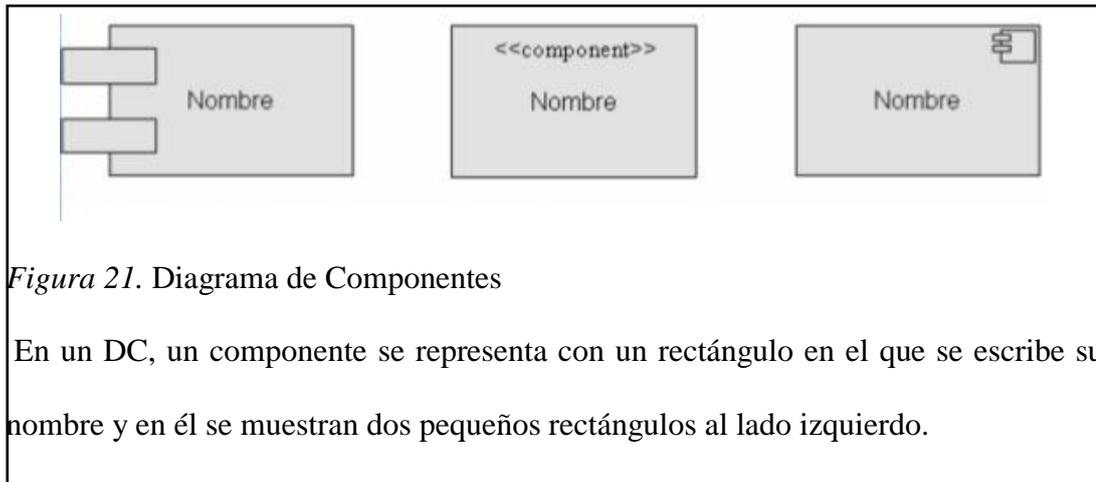


Figura. 20. D.C. Registrar Tratamiento

Descripcion: El diagrama describe como procede el sistema en registrar el tratamiento que esta realizando el medico verificando los datos del paciente en la Historia Clínica previa la verificación del pago de la consulta para ser atendido.

5.02.05 Diagrama de componentes

Un componente es una parte física de un sistema (módulo, base de datos, programa ejecutable, etc.). Se puede decir que un componente es la materialización de una o más clases, porque una abstracción con atributos y métodos pueden ser implementados en los componentes.



Este diagrama, también conocido como diagrama de estructura compuesta, tiene por objetivo principal describir con precisión objetos compuestos. Estos diagramas no sustituyen a los diagramas de clases, sino que los completan. En el diagrama de componentes el objeto compuesto se describe mediante un clasificador, mientras que sus componentes se describen mediante las partes. Un clasificador y una parte están asociados a una clase, cuya descripción completa se realiza en un diagrama de clases. En el diagrama indicado nos permite ver como se encuentra compuesto un sistema, mostrando dependencias de archivos, cabeceras, bibliotecas compartidas, módulos, ejecutables, permitiendo modelar la arquitectura del sistema.

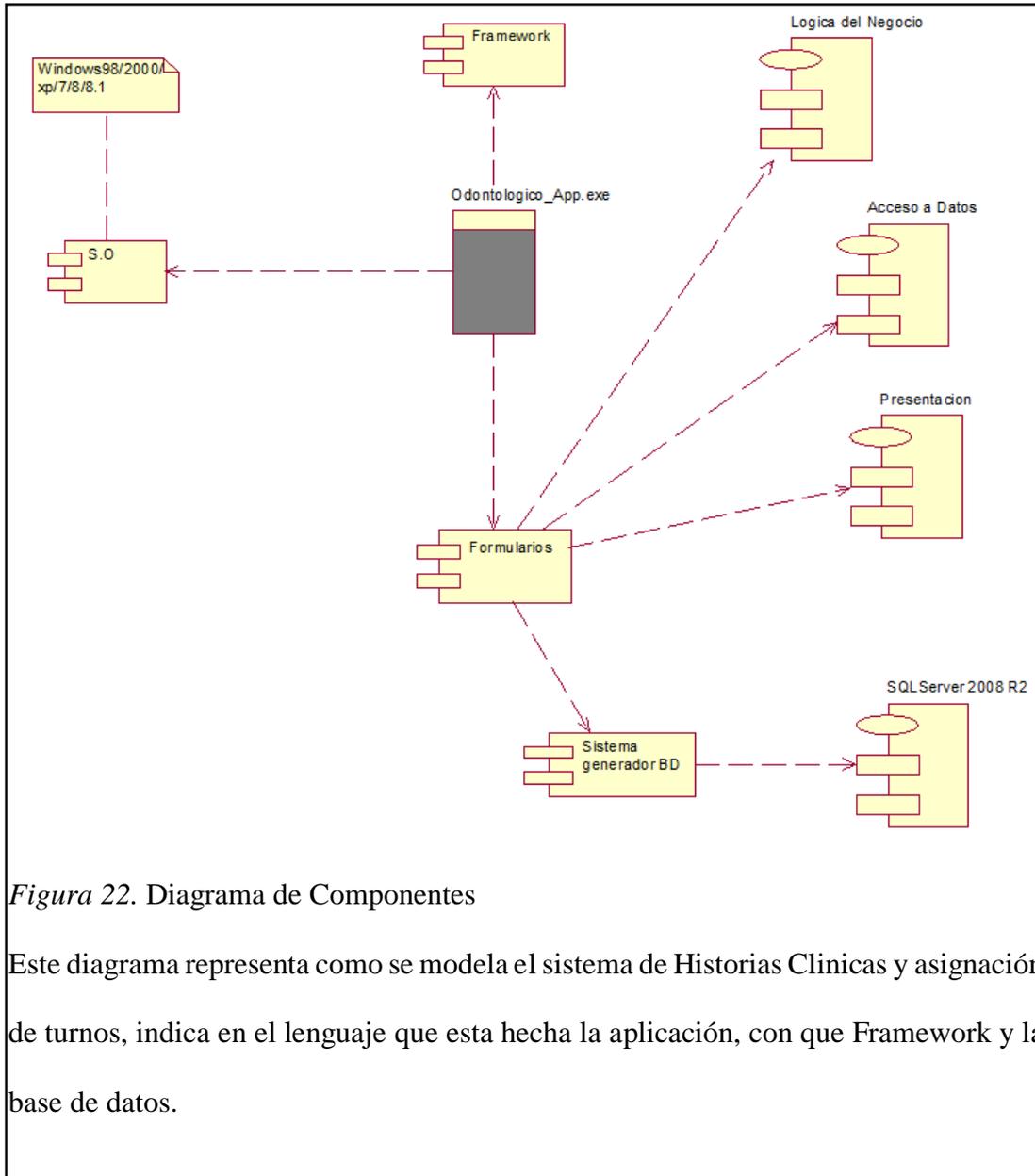


Figura 22. Diagrama de Componentes

Este diagrama representa como se modela el sistema de Historias Clínicas y asignación de turnos, indica en el lenguaje que esta hecha la aplicación, con que Framework y la base de datos.

5.02.06 Diagrama físico y lógico.

El diseño lógico de la base de datos es el proceso que determina la estructura lógica de datos necesaria para soportar los recursos de información en una organización. Este proceso de diseño lógico ayuda a implementar la base de datos para satisfacer los requerimientos del usuario. En esta fase se aplica un diseño lógico, el cual está soportado por medio del análisis relacional de datos o normalización. El análisis relacional de datos permite empezar por definir elementos de datos usados por la organización, para luego normalizar las relaciones en que ellos participan y finalmente obtener un nuevo diagrama lógico de datos (diagrama entidad-relación), el cual validándolo con el obtenido en fase de análisis, servirá para generar el diseño físico de la base de datos.

El diseño físico es el proceso de traducción del modelo lógico abstracto a un diseño técnico específico para el nuevo sistema. Produce las especificaciones reales para el hardware, software y bases de datos físicas, medios de entrada/salida, procedimientos manuales y controles específicos. Proporciona las especificaciones que transforman el diseño lógico abstracto en un sistema de funciones de personas y máquinas.

Entonces el Diseño Físico de la Base de Datos: Es el proceso de producción de una descripción, de una implementación, de un almacenamiento secundario de la Base de Datos, describe el almacenamiento de estructuras y métodos de acceso usados para conseguir el acceso eficiente a los datos.

Modelo lógico de la base de datos

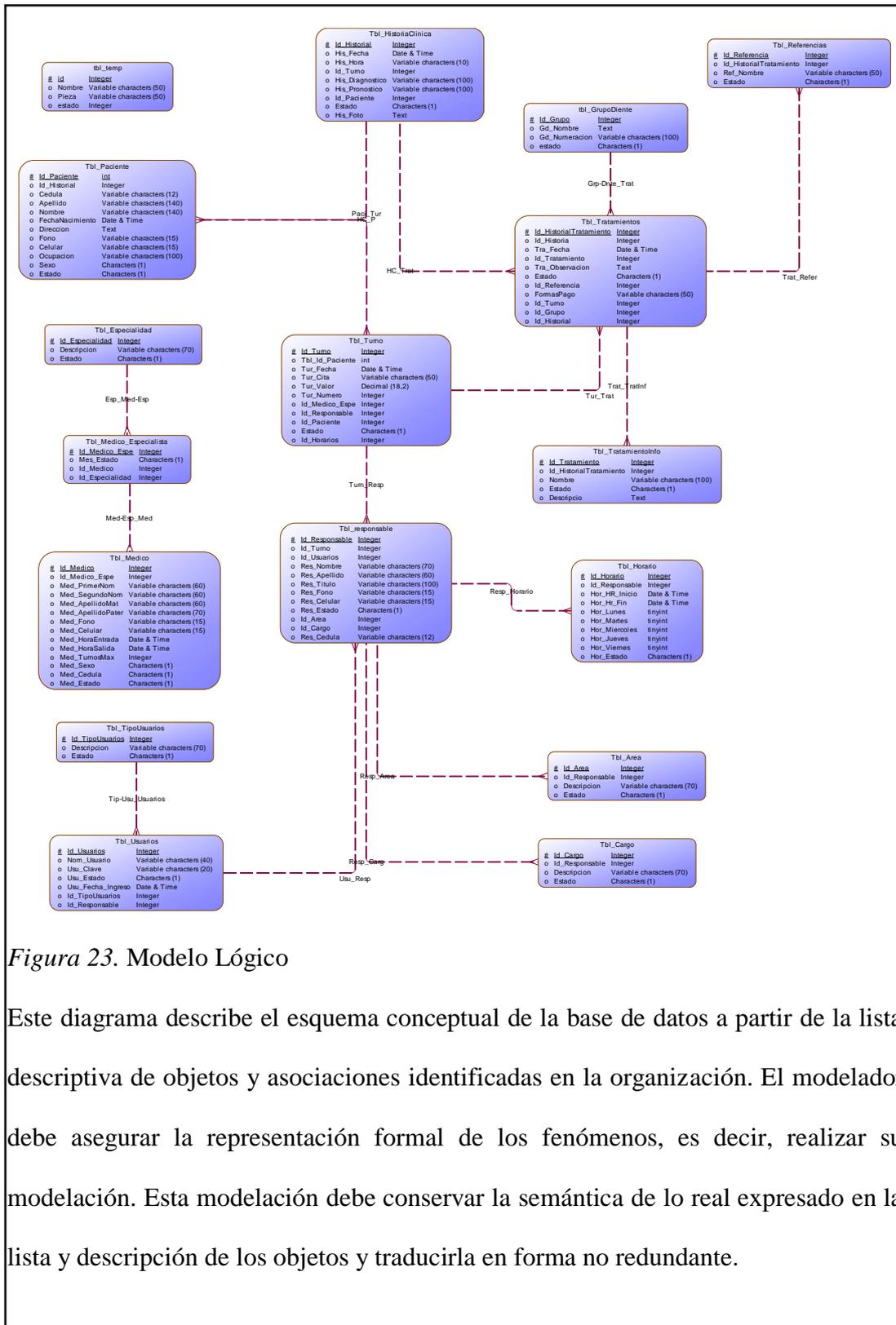


Figura 23. Modelo Lógico

Este diagrama describe el esquema conceptual de la base de datos a partir de la lista descriptiva de objetos y asociaciones identificadas en la organización. El modelador debe asegurar la representación formal de los fenómenos, es decir, realizar su modelación. Esta modelación debe conservar la semántica de lo real expresado en la lista y descripción de los objetos y traducirla en forma no redundante.

5.03 Desarrollo

5.03.01 Arquitectura de software.

5.03.01.01. *Arquitectura basada en capas.*

La arquitectura basada en capas se enfoca en la distribución de roles y responsabilidades de forma jerárquica proveyendo una forma muy efectiva de separación de responsabilidades. El rol indica el modo y tipo de interacción con otras capas, y la responsabilidad indica la funcionalidad que está siendo desarrollada.

Además, permite distribuir el trabajo de creación de una aplicación por niveles; de este modo, cada grupo de trabajo está totalmente abstraído del resto de niveles, de forma que basta con conocer la API que existe entre niveles.

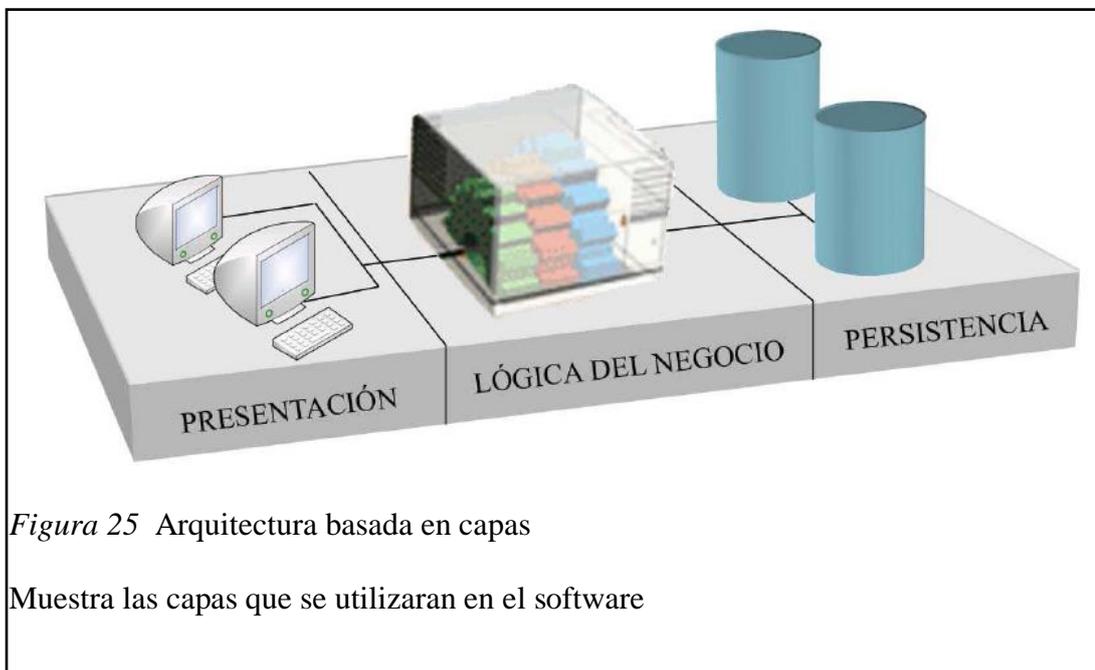


Figura 25 Arquitectura basada en capas

Muestra las capas que se utilizaran en el software

5.03.01.02. Arquitectura de N-Capas / 3-Capas.

Este estilo de despliegue arquitectónico describe la separación de la funcionalidad en segmentos separados de forma muy parecida al estilo de capas, pero en el cual cada segmento está localizado en un computador físicamente separado. Este estilo ha evolucionado desde la aproximación basada en componentes generalmente usando métodos específicos de comunicación asociados a una plataforma en vez de la aproximación basada en mensajes.

1. Arquitectura de 3 capas: se basa más bien en cómo será construido el entorno, una manera de decirlo es romper el clásico concepto Cliente-Servidor para introducir conceptos como Back End (Base de Datos), Middleware (Servidor de Aplicaciones), Front End (Interfaz de Usuario). Este es un concepto grande que no veremos ahora, pero lo remarco para hacer entender que no tiene nada que ver con la programación en capas. Se acerca más a un concepto físico.

2. Programación en 3 (n) capas: este es el tema en cuestión y estira más hacia un concepto lógico. En cómo partimos, agrupamos, clasificamos, optimizamos nuestro código. El mismo introduce conceptos como **Capa de Acceso a Datos** (Esta nos permite conectarnos a la fuente de datos y operar contra ella), **Capa de Negocios** (es la que se encarga de procesar todo, validaciones, etc. la misma suele distribuirse en la aplicación en sí y en la *BBDD*), y **Capa de Presentación** (es más bien lo que el usuario percibe, su interfaz gráfica por lo gral).

Creo que con esos conceptos introductorios ya estamos preparados para comprender mejor ciertos aspectos de este **paradigma**. Para resaltar por último, gracias a la separación en capas quiere decir que podemos cambiar de proveedor de base de datos, y no necesitaremos reescribir toda la aplicación de vuelta, sino solamente esa pequeña capa y reutilizaríamos la interfaz y las reglas de negocios, o también podemos

mantener las reglas de negocios y el motor de base de datos, y fácilmente cambiarnos de una interfaz **WinForm** a **WebForm**, siempre la más dura de cambiar en la de negocios ya que afecta en un nivel mínimo a las otras 2 capas.

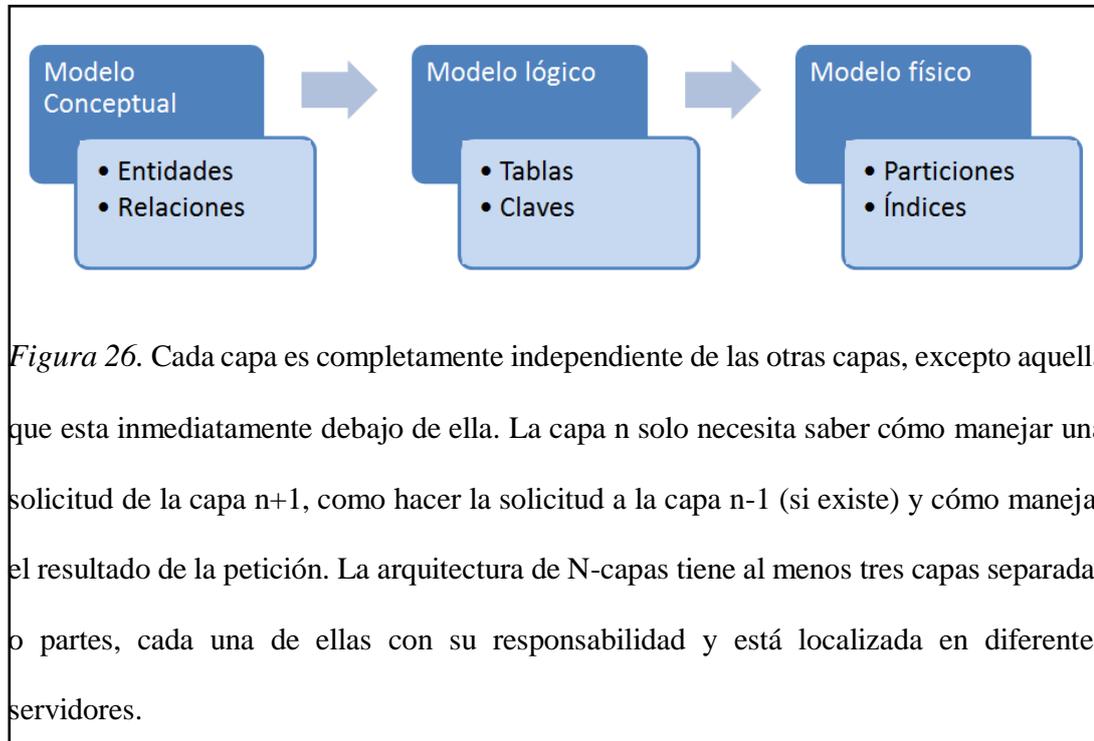


Figura 26. Cada capa es completamente independiente de las otras capas, excepto aquella que esta inmediatamente debajo de ella. La capa n solo necesita saber cómo manejar una solicitud de la capa n+1, como hacer la solicitud a la capa n-1 (si existe) y cómo manejar el resultado de la petición. La arquitectura de N-capas tiene al menos tres capas separadas o partes, cada una de ellas con su responsabilidad y está localizada en diferentes servidores.

5.03.01.04. *Requerimientos de software.*

- Windows 7 Profesional o Windows 8 Pro, con framework 4.0 y Internet Information Services (IIS) 6.0 ó superior.
- Windows Server 2008 R2 o Windows Server 2012, con framework 4.0 y Internet Information Services (IIS) 6.0 ó superior.
- Computador Intel core I3 3.3GHZ Hdd750 4gb Pantalla Led19.
- Conexión Wireles a internet mínimo un megabit por segundo.

5.03.02 Estándares de base de datos.

Base de datos

- Cada Base de Datos será creada como propiedad de un login único para la fase de desarrollo y un login por usuario con los permisos que les corresponda según indicación de los responsables del proyecto.
- Ninguna persona que no sea el DBA utilizará el login " SA " para ningún propósito.
- El tamaño de la Base de Datos deberá ser proyectado al momento de la petición de la creación de la misma, para determinar la ubicación de ésta en base a la disponibilidad de recursos. Asimismo, se recuerda que el tamaño asignado en la etapa de desarrollo será mínimo de acuerdo a un modelo de data que los desarrolladores determinarán para esta fase.
- El código de caracteres que se utilizará para las instalaciones de SQL será el que por default utiliza el producto (ISO); es el más recomendado y el estándar corporativo.
- Los desarrolladores deberán observar con sumo cuidado la recomendación de Microsoft de la NO utilización de palabras reservadas de MS-SQL, bajo responsabilidad.
- Los desarrolladores tendrán el manejo y la responsabilidad de la generación de la estructura interna de las Bases de Datos durante la etapa de desarrollo, durante la etapa de producción los cambios deberán ser realizados por el DBA a petición escrita y debidamente justificada bajo un formato prediseñado.

Índices

- Dado que éste es el punto más crítico dentro de todo desarrollo, se deberán indexar las columnas que sean estrictamente necesarias.

- Una tabla deberá tener como máximo 4 a 5 índices.

El desarrollador tendrá que tener la seguridad de que los índices definidos estén siendo utilizados por el SQL SERVER, de lo contrario tendrán que ser borrados.

- No olvidar que los índices nos ayudan para consultar información, pero, retarda la actualización de la misma.

Conexiones a servidor

- Todos los sistemas a desarrollarse en SQL SERVER deberán de optimizar el uso de conexiones por usuario.

Nomenclatura

- El nombre de las Bases de Datos deberá ser corto, descriptivo y que permita determinar fácilmente su propósito, *por ejemplo*: Sistema de Presupuesto – BD_PRESUPUESTO.
- En el presente estándar, no se permiten los espacios en blanco como parte de un identificador.
- Utilizar para los nombres de objetos palabras en singular.
- Para la definición de nombre de objetos de base de datos de acuerdo al caso se usará el caracter underscore “_” para separar las palabras_del_nombre.

Triggers.

Un trigger no tiene sentido fuera de una tabla y un trigger tiene asociada siempre una operación, por lo que dicha información debe estar asociada al nombre del trigger.

<Tabla>_<operacion>

Vistas.

Las vistas deben nombrarse con la misma notación definida para nombrar tablas, pero prefijadas usando VW_.

Primary keys.

La clave primaria es un conjunto de campos que identifica de forma única un registro en una tabla. Son un caso particular de un índice. La nomenclatura es la siguiente:

pk_<tabla>

Foreign keys.

Como patrón para la nomenclatura de la foreign key elegimos el siguiente.

Fk_<tabla_que_referencia>+<campo_que_referencia>_<tabla_referenciada>+<campo_referenciado>

Estándares de base de datos

Tabla 11.

Estándares de base de datos

TIPO	PREFIJO
Base Datos	BD_
Tablas	Tbl_
Campos	Tabla_Nombre
Vistas	VW_
Triggers	TablaOperacion
Primary keys	PK_
Foreign keys	FK_
Indexes	IDX_

5.03.03 Estándares de programación.

Para implementar estándares de programación, no existe "terminología" o "estilo" que sea mejor que otro. La valoración de dichas "terminologías" se basa no en lo que al programador le guste, sino primordialmente en el uso adecuado de un "terminología"



específica. Esto es lo que denominamos "estándares de programación", que no es más que el usar y seguir ciertas reglas de notación y nomenclatura durante la fase de implementación (codificación) de una aplicación.

Se utilizara la nomenclatura Húngara basada en los siguientes principios:

- Tipología (Mayúsculas/Minúsculas) pero evitando acentos (tildes), espacios y otros caracteres especiales.
- Todos los nombres deben escribirse en singular.
- No se permiten nombres que utilicen exclusivamente mayúsculas.
- No se utilizara el símbolo “_” para separación de partes del Nombre, una letra Mayúscula realizara esta separación.
- Al nombre descriptivo se le debe anteponer un prefijo de tres letras en minúsculas o un identificador que indicara el tipo de concepto.
- Nombre de las capas
- Tipos de variables más utilizados.

5.03.04 Diseño de interfaces.

Interfaz de Ingreso "Login-Odontológico"

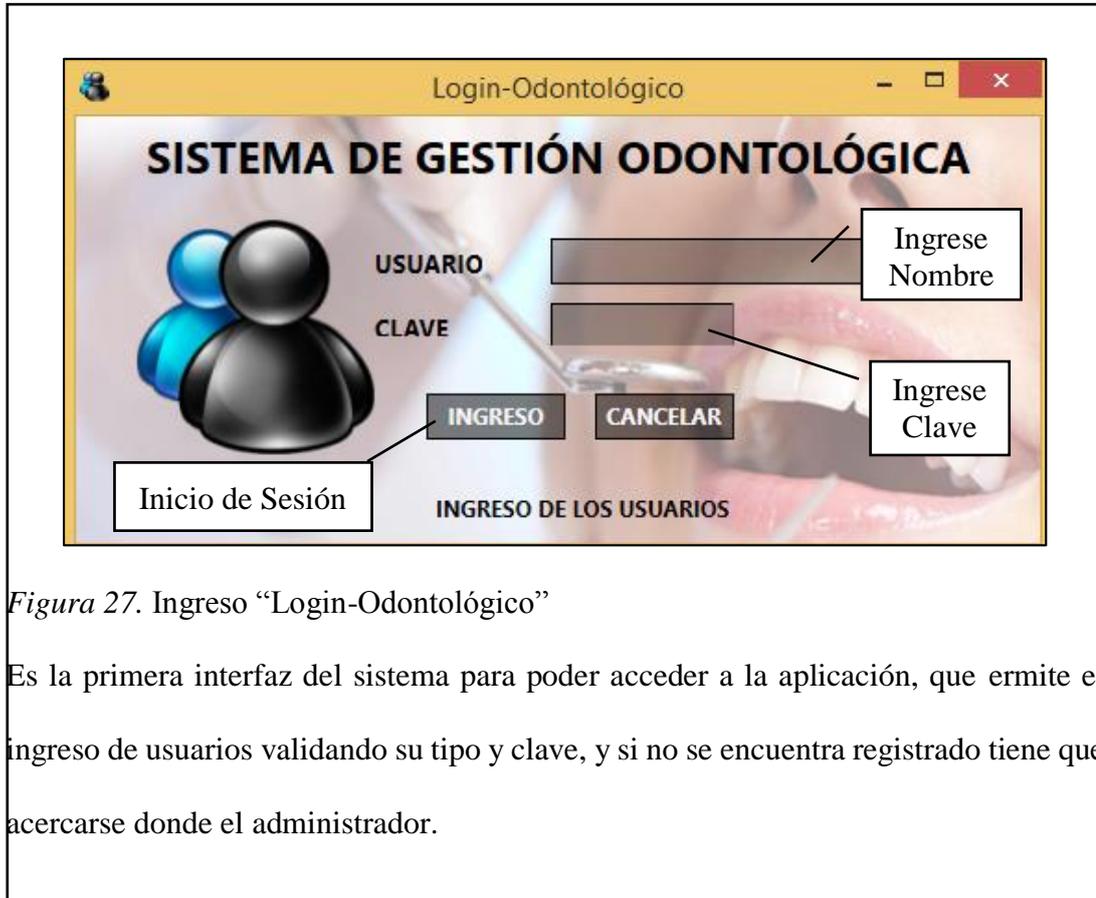


Figura 27. Ingreso "Login-Odontológico"

Es la primera interfaz del sistema para poder acceder a la aplicación, que permite el ingreso de usuarios validando su tipo y clave, y si no se encuentra registrado tiene que acercarse donde el administrador.

Interfaz Menú Principal

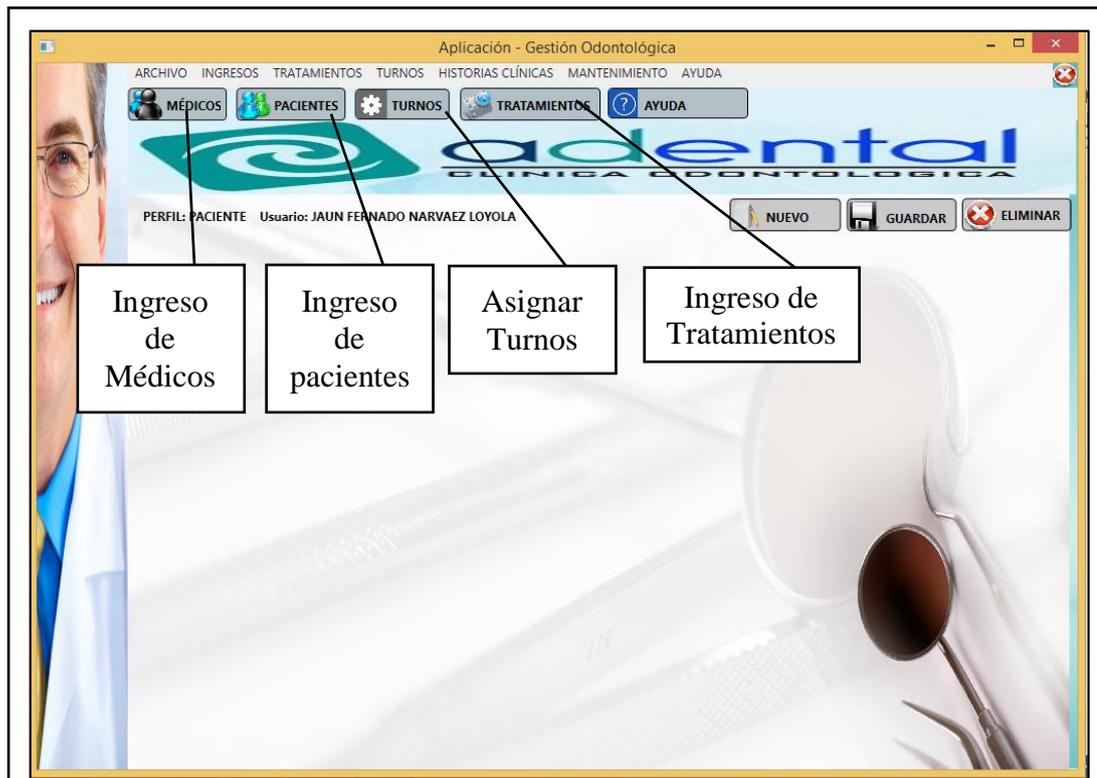


Figura 28. Interfaz Pantalla Principal

En el menú principal encontramos las herramientas para el ingreso de pacientes, doctores, historias clínicas y entrega de turnos.

Interfaz Usuario

Figura 29 shows a web interface for adding a user. At the top, it displays the current user: "PACIENTE Usuario: JAUN FERNADO NARVAEZ LOYOLA". There are three buttons: "NUEVO" (New), "GUARDAR" (Save), and "ELIMINAR" (Delete). Below this is a section titled "BÚSQUEDA DEL RESPONSABLE POR MEDIO DE SU CÉDULA" with a "LISTA" button. The main form has the following fields:

- RESPONSABLE:** A search field with a green checkmark icon and a "BUSCAR" button.
- NOMBRES:** Input field containing "Nelson Patricio".
- APELLIDOS:** Input field containing "Guevara Changotaxi".
- USUARIO:** Input field containing "nguervera".
- CLAVE:** Input field containing "nguevara".
- TIPO USUARIO:** A dropdown menu with the text "Seleccione Tipo Usuario". The dropdown is open, showing options: "ADMINISTRADOR", "MEDICO", "PACIENTE", and "RESPONSABLE".
- FECHA INGRESO:** A field for the user's start date.

Two callout boxes are present:

- One pointing to the "USUARIO" and "CLAVE" fields with the text: "Ingresar: Nombre usuario Clave".
- Another pointing to the "TIPO USUARIO" dropdown with the text: "Selccionar tipo de usuario".

Figura 29. Interfaz Usuario

Ingresamos los datos del usuario y el tipo de usuario que va a utilizar el sistema tanto como el administrador, el médico y el responsable (auxiliar).

Ingreso del Responsable (Auxiliar) al sistema

The screenshot shows a web application interface for entering data for a responsible person. At the top, it displays the user name 'PACIENTE Usuario: JAUN FERNADO NARVAEZ LOYOLA' and three buttons: 'NUEVO' (New), 'GUARDAR' (Save), and 'ELIMINAR' (Delete). Below this is a search section titled 'BÚSQUEDA DEL RESPONSABLE POR MEDIO DE SU CÉDULA' with a 'LISTA' button. The form contains several input fields: 'CÉDULA', 'NOMBRES', 'APELLIDOS', 'TÍTULO', 'TELÉFONO', and 'CELULAR'. At the bottom, there are two dropdown menus for 'ÁREA' (Area) and 'CARGO' (Position), both with the text 'Seleccione' (Select).

Figura 30. Interfaz Usuario

En esta interfaz ingresamos los datos de la persona responsable la cual va a utilizar las herramientas que es desde el ingreso de las historias clínicas y entrega de turnos del sistema.

Lista Médicos Agregados



Cédula	Apellidos	Nombre	Teléfono	Celular	HoraEntrada	HoraSalida	TurnosMax
1734562234	MANTILLA MANTILLA	JORGE ALVERTO	022545678	095443211	8:30	16:30	6
0502631393	AREQUIPA TOAQUIZA	SEGUNDO RUBEN	022464722	0986573763	8:30	17:30	4

Figura 31. Lista de médicos agregados

Indica el listado de los médicos que están ingresados en el sistema

Ingresar Médico Nuevo

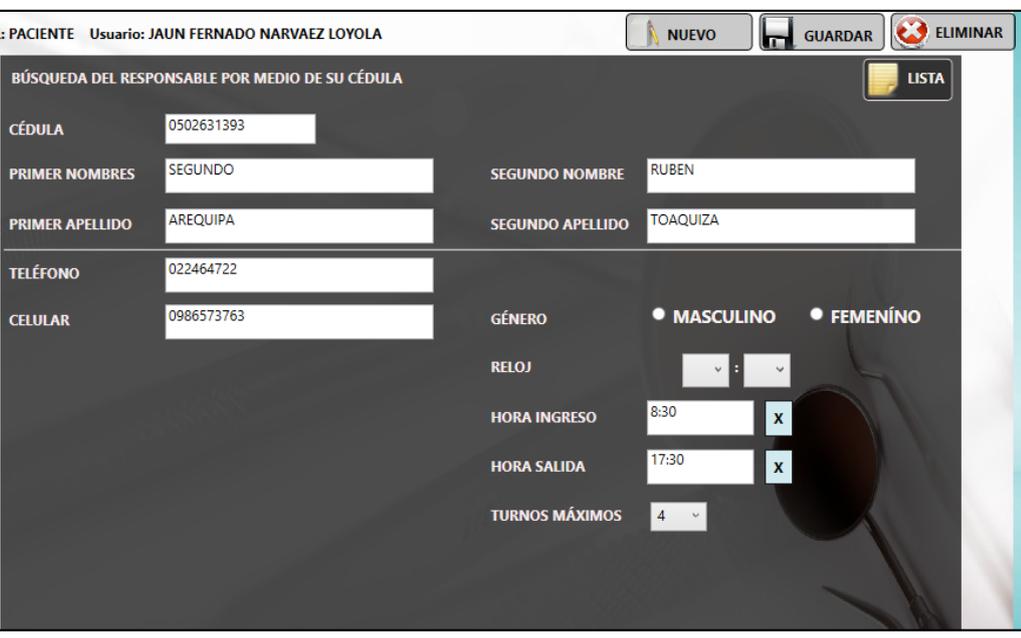


Figura 32. Registro Nuevo Medico

Muestra los datos de un nuevo Doctor que está siendo ingresando al sistema.

Lista de Pacientes

Cédula	Nombre	Apellido	Teléfono	Celular	Dirección
1312597733	ELIANA	RAMONA	098765432	098765432	CONOCOTO
1002813135	Nelson patricio	guevara changotaxi	2285218	0997309363	el dorado
1002813135	Nelson Patricio	Guevara Changotaxi	2285218	0983202160	El Dorado
1312597733	ELIANA	RAMONA	098765432	098765432	CONOCOTO

Figura 33. Listado de Pacientes

Muestra el listado de los pacientes agregados

Ingresar nuevo paciente

INFORMACIÓN DEL PACIENTE

CÉDULA: 1312597733

NOMBRES: ELIANA APELLIDOS: RAMONA

DIRECCIÓN: CONOCOTO

TELÉFONO: 098765432

CELULAR: 098765432

FECHA NACIMIENTO: 27/03/2014

OCUPACIÓN: SECRETARIA 32

GÉNERO: MASCULINO FEMENINO

Figura 34. Ingresar nuevo paciente

Muestra los datos generales del paciente nuevo que está siendo ingresando al sistema.

Generar Turno

USUARIO: PACIENTE Usuario: JAUN FERNADO NARVAEZ LOYOLA

FECHA NACIMIENTO: 08/04/2014

CITA: Dr Ruben Arequipa

VALOR: 30

NÚMERO TURNO: 3

ESPECIALIDAD: PROTESIS DENTA

Medico Especialidad

- PROTESIS DENTAL
- IMPLANTOLOGIA
- ODONTOPEDIATRIA
- ODONTOGERIATRIA
- HIGIENE DENTAL
- ESTETICA DENTAL
- BLANQUEAMIENTO DENTAL
- PROSTODONCIA
- ODONTOLOGIA PREVENTIVA
- ODONTOLOGIA RESTAURADORA
- CIRUGIA BUCOMAXILOFACIAL
- RADIOLOGIA DENTAL
- REHABILITACION ORAL

Cédula	Nombres
1312597733	RAMONA ELIANA
1002813135	guevara changotaxi nelson patricio
1002813135	Guevara Changotaxi Nelson Patricio
1312597733	RAMONA ELIANA

HORARIOS

Figura 35. Genera turno por especialidad

Muestra el nombre de doctor el cual va ser atendido así como la especialidad y el número de turno de atención, además tiene el valor que tiene que cancelar.

Horarios de turno para ser atendido

USUARIO: PACIENTE Usuario: JAUN FERNADO NARVAEZ LOYOLA

HORARIOS

FECHA: 08/04/2014

HORA INICIO: [] HORA FIN: []

BUSCAR: []

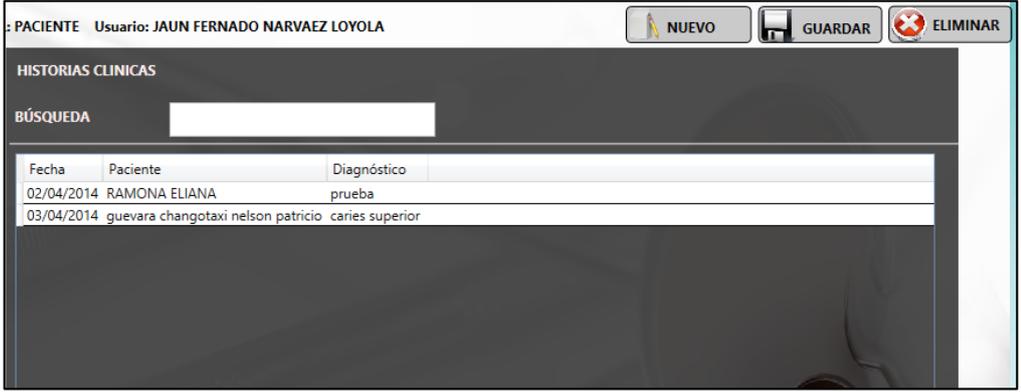
Horainicio	HoraFin	Lunes	Martes	Miércoles	Jueves	Viernes
		<input type="checkbox"/>				

GUARDAR SEMANA

Figura 36. Agregamos los días para solicitar turno

Muestra los días disponibles para ser atendido a elección del paciente.

Buscar Historia Clínica



PERFIL: PACIENTE Usuario: JAUN FERNADO NARVAEZ LOYOLA

HISTORIAS CLINICAS

BÚSQUEDA

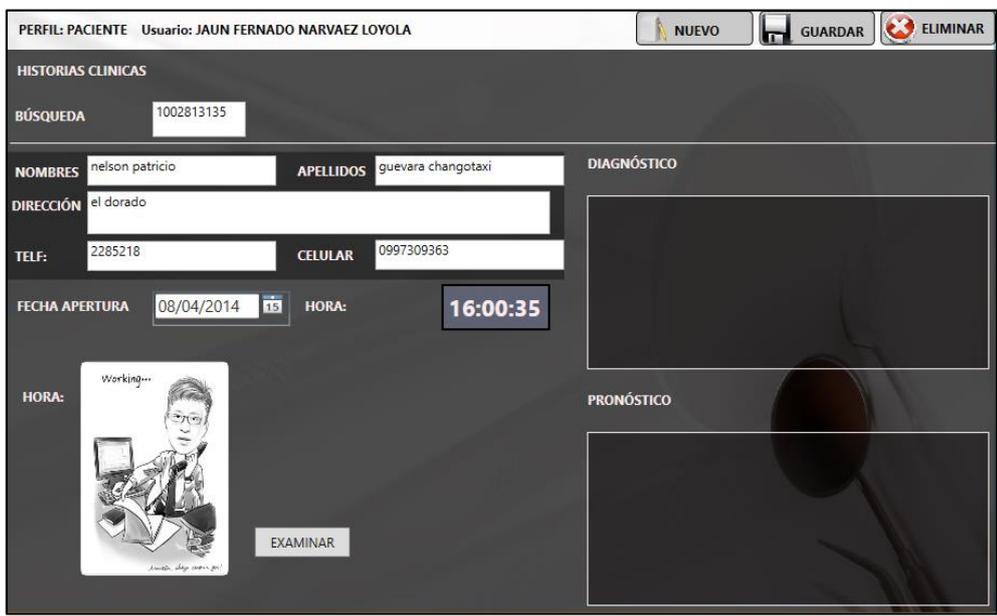
Fecha	Paciente	Diagnóstico
02/04/2014	RAMONA ELIANA	prueba
03/04/2014	guevara changotaxi nelson patricio	caries superior

NUEVO GUARDAR ELIMINAR

Figura 37. Buscar Historia Clínica

Muestra el listado de todas las Historias Clínicas almacenadas en el sistema

Nueva Historia Clínica



PERFIL: PACIENTE Usuario: JAUN FERNADO NARVAEZ LOYOLA

HISTORIAS CLINICAS

BÚSQUEDA 1002813135

NOMBRES nelson patricio APELLIDOS guevara changotaxi

DIRECCIÓN el dorado

TELF: 2285218 CELULAR 0997309363

FECHA APERTURA 08/04/2014 HORA: 16:00:35

HORA: Working... EXAMINAR

DIAGNÓSTICO

PRONÓSTICO

Figura 38. Interfaz Nueva Historia Clínica

Muestra los datos generales del paciente así como un diagnóstico y pronóstico que es ingresado por el Doctor

Buscar Nombre Historia Clínica de paciente para luego registrar el Tratamiento.

PERFIL PACIENTE Usuario: JAUN FERNADO NARVAEZ LOYOLA

HISTORIAS CLINICAS

BÚSQUEDA

Paciente	Médico	Fecha	Tratamiento	GrupoDental	Numeración
RAMONA ELIANA	CORTEZ PEDRO	02/04/2014		SUP-IZQUIERDO	18 - 11
RAMONA ELIANA	CORTEZ PEDRO	03/04/2014		SUP-IZQUIERDO	18 - 11

NUEVO GUARDAR ELIMINAR

Figura 39. Interfaz Lista de historias clínicas ingresadas

Muestra las Historias Clínicas para realizar el tratamiento respectivo

Interfaz Ingresar Tratamiento

PERFIL PACIENTE Usuario: JAUN FERNADO NARVAEZ LOYOLA

TRATAMIENTO AL PACIENTE:

BÚSQUEDA 1312597733 HISTORIAL CLÍNICO

HORA:

NOMBRES ELIANA APELLIDOS RAMONA

DIRECCIÓN CONOCOTO

TRATAMIENTO: ENDODONCIA

REFERENCIAS MÉDICAS Temperatura

TELF: 098765432 CELULAR 098765432

GRUPO DENTAL: SUP-IZQUIERDO

FECHA TRATAMIENTO Seleccione un día

BÚSQUEDA TURNO 2

ODONTOGRÁMA

MÉDICO CORTEZ PEDRO

OBSERVACIONES:

ESPECIALIDAD ODONTOLOGÍA GENERAL

CITA: Dr Ruben Arequipa

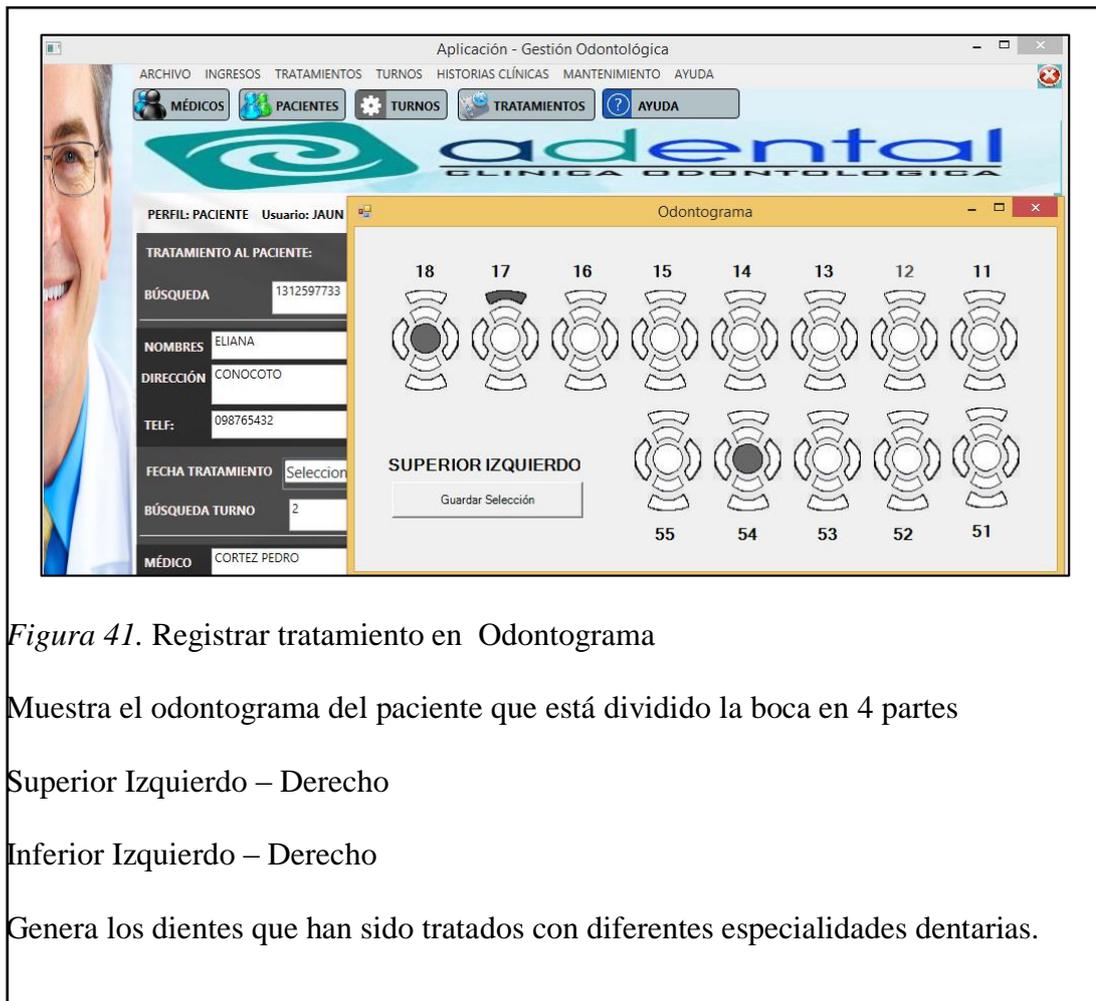
HORARIO Hora Inicio: 4:00:00 Hora Fin: 4:00:00

CARGAR OBSERVACIONES

Figura 40. Interfaz Ingresar Tratamiento

Muestra el tipo de tratamiento que va ser realizado por el doctor correspondiente

Interfaz Registro en Odontograma



Descripcion: Un odontograma es un esquema utilizado por los odontólogos que permite registrar información sobre la boca de una persona. En dicho gráfico, el profesional detalla qué cantidad de piezas dentales permanentes tiene el paciente, cuáles han sido restauradas y otros datos de importancia.

Pantalla Mantenimiento

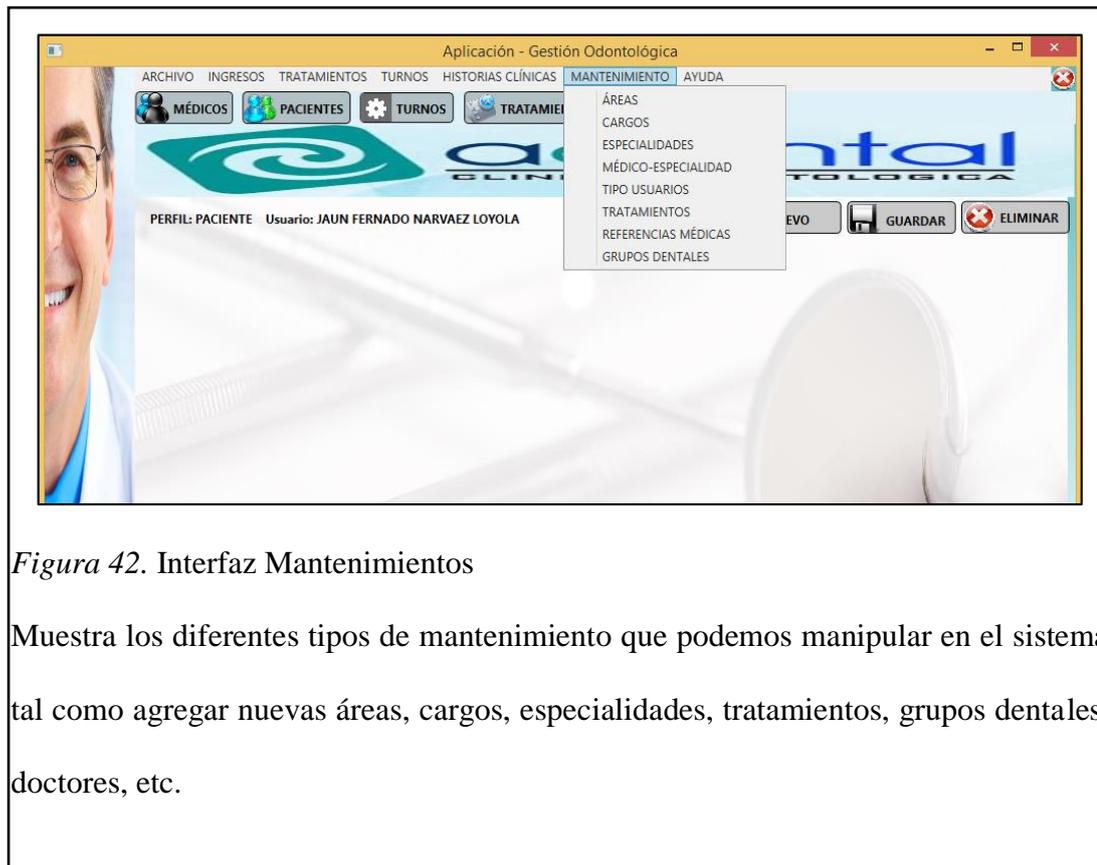


Figura 42. Interfaz Mantenimientos

Muestra los diferentes tipos de mantenimiento que podemos manipular en el sistema tal como agregar nuevas áreas, cargos, especialidades, tratamientos, grupos dentales, doctores, etc.

5.04 Pruebas

Previa la realización de las pruebas se definirá a continuación los parámetros de evaluación que permitan determinar el correcto cumplimiento de los requerimientos del sistema.

5.04.01 Casos de pruebas

Tabla 12.

Historial de Revisiones

Fecha	Descripción	Autor
18/03/2014	Inicio de Pruebas	Guevara Changotaxi Nelson Patricio
18/03/2014	Prueba de integridad de datos	Guevara Changotaxi Nelson Patricio
21/03/2014	Pruebas de sistema	Guevara Changotaxi Nelson Patricio
25/03/2014	Pruebas de interfaz de usuarios	Guevara Changotaxi Nelson Patricio
28/03/2014	Pruebas de desempeño	Guevara Changotaxi Nelson Patricio

Plan de pruebas maestro

Propósito

Este documento describe el plan para probar las funcionalidades y características del sistema *"GESTION DE ATENCION DE HISTORIAS CLÍNICAS, TOMA DE TURNOS PARA ADENTAL CLÍNICA ODONTOLÓGICA DE LA CIUDAD DE QUITO"*. Este documento está basado sobre los siguientes objetivos:

- Identificar que la información existente del proyecto y los componentes de software sean probados.
- Listar los requerimientos recomendados de prueba (de alto nivel).
- Recomendar y describir las estrategias a ser empleadas.

- Identificar los recursos requeridos y estimar los esfuerzos de las pruebas.
- Listar los elementos a entregar de las actividades de pruebas.

Alcances

Este plan de pruebas aplica para la integración y las pruebas de sistema que serán conducidos en el lanzamiento de la versión 1.0 del sistema *"GESTION DE ATENCION DE HISTORIAS CLÍNICAS, TOMA DE TURNOS PARA ADENTAL CLÍNICA ODONTOLÓGICA DE LA CIUDAD DE QUITO"*

Se asume que pruebas unitarias previas han debido proveer de pruebas de caja negra totales a través de una extensiva cobertura del código fuente y pruebas de todas las interfaces de los módulos.

Requerimientos de pruebas

La lista que prosigue este párrafo identifica aquellos elementos (requerimientos funcionales, no funcionales) que han sido identificados como objetivos de las pruebas. Esta lista representa el qué será probado. Los detalles de cada prueba serán determinados posteriormente mientras los casos de prueba sean identificados y los scripts sean desarrollados.

Pruebas de integridad de datos y BD

- Verificar el acceso a la Base de Datos llamada Odontologico_BD
- Verificar el acceso simultáneo en la lectura de registro de las distintas tablas.
- Verificar el bloqueo realizado durante actualizaciones de registros de las tablas transaccionales
- Verificar la correcta obtención de data actualizada.

Pruebas del sistema

- Verificar CU01 – Asignar Turno
- Verificar CU02 – Registrar Pago Consulta
- Verificar CU03 – Crear Historia Clínica
- Verificar CU04 – Ingresar Diagnostico
- Verificar CU05 – Registrar Tratamientos

Pruebas de desempeño

- Verificar el tiempo de respuesta para entregar el turno oportunamente.
- Verificar el tiempo de respuesta para registrar el pago de la consulta
- Verificar el tiempo de respuesta para registrar una Historia Clínica
- Verificar el tiempo de respuesta para consultar el tipo de diagnostico
- Verificar el tiempo de respuesta para registrar los tipos de tratamientos

Estrategia de pruebas

La estrategia de pruebas presenta el alcance recomendado para la prueba de aplicaciones de software. La sección previa a los requerimientos de pruebas describen qué será probado; ésta describirá cómo será probado.

Tipos de pruebas

Las consideraciones principales para la estrategia de pruebas son las técnicas a usarse y los criterios para determinar si la prueba fue completada.

Además de las consideraciones provistas para cada prueba mencionada, las pruebas deberían ser únicamente ejecutadas usando bases de datos conocidas y controladas en entornos seguros.

Pruebas de integridad de datos y BD

La base de datos y los procesos de bases de datos deberían ser probadas en sistemas separados. Estos sistemas deberían ser probados sin la aplicación *GESTION DE*

ATENCIÓN DE HISTORIAS CLÍNICAS, TOMA DE TURNOS PARA ADENTAL CLÍNICA ODONTOLÓGICA DE LA CIUDAD DE QUITO. Revisión exhaustiva sobre el gestor de base de datos a usarse necesita ser realizada para identificar las herramientas y técnicas que puedan existir para soportar las pruebas a realizarse.

Objetivo

Asegurar que los métodos de acceso y los procesos funcionen apropiadamente y sin corrupción de datos

Técnicas

Invocar cada método de acceso a la BD, intentando con datos válidos e inválidos.

Inspeccionar la base de datos para asegurar que la data ha sido poblada como se esperaba, que todos los eventos ocurran apropiadamente, o revisar la data retornada para asegurar que la data correcta fue obtenida (por las razones correctas).

Criterio de cumplimiento

Todos los métodos de acceso a la base de datos y procesos funcionan como fueron diseñados y sin corrupción de datos.

Pruebas del sistema

Las pruebas sobre la aplicación deberían enfocarse en requerimientos que puedan ser asociados directamente a casos de uso (o funciones de negocio), y reglas del negocio.

Las metas de estas pruebas son verificar la aceptación, el procesamiento y obtención de data apropiada, así como la apropiada implementación de reglas del negocio.

Objetivo

Asegurar la navegación apropiada en la aplicación; el correcto ingreso de datos, procesamiento y obtención.



Técnicas

Ejecutar cada CU, cada flujo de CU o función, usando data válida e inválida, para verificar:

- a) Que los resultados ocurran cuando la data sea válida.
- b) Que se muestren apropiados mensajes de error o alerta cuando data inválida sea empleada.
- c) Cada regla de negocio es apropiadamente aplicada.

Criterio de cumplimiento

Todas las pruebas planificadas fueron ejecutadas

Todos los defectos de pruebas han sido manejados.

Pruebas de la interfaz de usuario (IU)

Verifica la interacción del usuario con el software. La meta de las pruebas de IU es asegurar que la interfaz de usuario provea al usuario el acceso apropiado para acceder y navegar por las funciones de la aplicación. Además, las pruebas IU asegura que los objetivos dentro de la IU funcionen como se esperaba y conforme a los estándares de la compañía.

Objetivo

Verificar: a) la navegación por la aplicación refleje propiamente las funciones y requerimientos de negocio; b) los objetos de ventanas y sus características, como menús medidas posición, estado y foco sea conforme a los estándares.

Técnicas

Crear modificar las pruebas para cada ventana para verificar apropiadamente la navegación y los estados de los objetos para cada ventana y objeto de la aplicación.

Criterio de cumplimiento

Cada ventana fue verificada exitosamente para comparar si se sigue el estándar o no.

Pruebas de desempeño

Realizar las pruebas que miden los tiempos de respuesta, las tasas de transacción y otros requerimientos sensibles al tiempo. La meta de las pruebas de desempeño es verificar y validar que los requerimientos de desempeño han sido alcanzados. Este tipo de pruebas es ejecutado muchas veces, y cada ejecución emplea una carga subrepticia (background load) en el sistema.

Objetivo

Validar el tiempo de respuesta para transacciones diseñadas o funciones de negocio bajo las siguientes condiciones: a) volumen normal anticipado, b) volumen de caso mal anticipado.

Técnicas

Usar scripts de prueba desarrollados por pruebas de modelo de negocio (pruebas de sistema).

Modificar archivos de datos (para incrementar el número de transacciones) o modificar los scripts para incrementar el número de iteraciones en que cada transacción ocurre.

Lo scripts deben correr en una sola máquina (en el mejor de los casos simular un usuario único, una única transacción) y ser repetido en múltiples clientes (virtuales o actuales).

Criterio de cumplimiento

Una transacción / un único usuario. El cumplimiento exitoso de estas pruebas, es cuando no se encuentran fallas en los tiempos esperados o requerido (en cada transacción).

Múltiples transacciones / múltiples usuarios. El cumplimiento exitoso de estas pruebas, es cuando no se encuentran fallas en los tiempos aceptables.

Pruebas de carga

Las pruebas de carga miden las situaciones en las que el sistema se somete a variaciones en su carga de trabajo para evaluar la habilidad del sistema para continuar funcionando adecuadamente, más allá de la carga de trabajo esperada. Adicionalmente, las pruebas evalúan las características de desempeño (tiempos de respuestas, tasas de transacción y otros problemas sensibles a tiempos).

Objetivo

Verificar el tiempo de respuesta del sistema para transacciones diseñada o casos de negocio bajo condiciones de carga de trabajo variada.

5.04.02 Pruebas de Caso de Uso

Caso de Uso: CU_01 Asignación de Turnos
Caso de Prueba: CU_01 Asignación de Turnos
Actor: Auxiliar
Propósito: Generar Turno
Pre Condiciones: La auxiliar de admisión esta logeado en el sistema

Escenario: CP-01_E01 Comprobar la correcta generación de turnos

Tabla 13.

Sec.	Actividad	Clase de Equivalencia	Resultado esperado
1	La auxiliar encargado de admisión selecciona Buscar Historia Clínica	Valida	Muestra datos de la historia clínica
2	Selecciona la especialidad del medico	Valida	ok
3	Selecciona Buscar Medico	Valida	Muestra datos del medico
4	Solicita "Grabar"	Valida	Turno Generado Exitosamente

Escenario: CP-01_E02: Comprobar que el sistema genere el mensaje de error al no encontrar la Historia Clínica

Tabla 14.

Sec.	Actividad	Clase de Equivalencia	Resultado esperado
1	La auxiliar encargado de admisión selecciona Buscar Historia Clínica	Valida	Muestra Historia Clínica No Encontrado
2	Selecciona la especialidad del medico	Valida	ok
	Selecciona Buscar Medico	Valida	Muestra datos del medico
	Solicita "Grabar"	Valida	Muestra Turno Generado Exitosamente

Escenario: CP-01_E03: Comprobar que el sistema genere el mensaje de error al no encontrar al Medico

Tabla 15.

Sec.	Actividad	Clase de Equivalencia	Resultado esperado
1	La auxiliar encargado de admisión selecciona Buscar Medico	Valida	Muestra MSG: Medico No Encontrado

Escenario: CP-01_E04: Comprobar que el sistema genere el mensaje de error al no encontrar al Medico pero si la historia clínica

Tabla 16.

Sec.	Actividad	Clase de Equivalencia	Resultado esperado
1	La auxiliar encargado de admisión selecciona Buscar Historia Clínica	Valida	Muestra datos de la historia clínica
2	La auxiliar encargado de admisión selecciona Buscar Medico	Valida	Muestra Medico No Encontrado

Caso de Uso:	CU_02 Registrar pago consulta medica
Caso de Prueba:	CU_02 Registrar pago consulta medica
Actor:	Auxiliar
Propósito:	Comprobar emisión de factura
Pre Condiciones:	La auxiliar de admisión esta logeado en el sistema

Escenario: CP-02_E02 Comprobar la correcta generación comprobante de pago

Tabla 17.

Sec.	Actividad	Clase de Equivalencia	Resultado esperado
1	La auxiliar encargada de admisión selecciona Buscar Paciente	Valida	Muestra datos del paciente
2	Selecciona la especialidad del medico	Valida	ok
4	Solicita "Grabar"	Valida	Comprobante de pago generado exitosamente

Escenario: CP-02_E03: Comprobar que el sistema genere el mensaje de error al no encontrar la Historia Clínica

Tabla 18.

Sec.	Actividad	Clase de Equivalencia	Resultado esperado
1	La auxiliar encargado de admisión selecciona Buscar Historia Clínica	Valida	Muestra Historia Clínica No Encontrado
2	Selecciona la especialidad del medico	Valida	ok
	Selecciona Buscar Medico	Valida	Muestra datos del medico
	Solicita "Grabar"	Valida	Muestra Turno Generado Exitosamente

Escenario: CP-021_E04: Comprobar que el sistema genere el mensaje de error al no encontrar al Medico

Tabla 19.

Sec.	Actividad	Clase de Equivalencia	Resultado esperado
1	La auxiliar encargado de admisión selecciona Buscar Medico	Valida	Muestra MSG: Medico No Encontrado

Escenario: CP-02_E05: Comprobar que el sistema genere el mensaje de error al no encontrar al Medico pero si la historia clínica

Tabla 20.

Sec.	Actividad	Clase de Equivalencia	Resultado esperado
1	La auxiliar encargado de admisión selecciona Buscar Historia Clínica	Valida	Muestra datos de la historia clínica
2	La auxiliar encargado de admisión selecciona Buscar Medico	Valida	Muestra Medico No Encontrado

Caso de Uso: CU_03 Crear Historia Clínica

Caso de Prueba: CU_03 Crear Historia Clínica

Actor: Doctor

Propósito: Registrar Historia Clínica

Pre Condiciones: El encargado de admisión esta logeado en el sistema

Escenario: CP-03_E03 Comprobar el correcto registro de una Historia Clínica

Tabla 21.

Sec.	Actividad	Clase de Equivalencia	Resultado esperado
1	La auxiliar encargada de admisión ingresa los datos del paciente	Valida	Muestra datos del paciente
2	La auxiliar encargado admisión ingresa los datos de contacto	Valida	ok
3	La auxiliar encargado admisión ingresa los datos de ubicación		ok
4	Solicita "Grabar"	Valida	Muestra Historia Clínica Registrada Exitosamente

Escenario: CP-03_E03: Comprobar que el sistema genere el mensaje de error al no encontrar datos de paciente

Tabla 22.

Sec.	Actividad	Clase de Equivalencia	Resultado esperado
1	La auxiliar encargado de admisión ingresa los datos del paciente	Valida	Muestra completar los datos correctamente
2	Selecciona la especialidad del medico	Valida	Ok
3	Selecciona Buscar Medico	Valida	Muestra datos del medico
4	Solicita "Grabar"	Valida	Muestra registro de historia clínica Generado Exitosamente

Escenario: CP-03_E03: Comprobar que el sistema genere el mensaje de error al no encontrar datos de contacto

Tabla 23.

Sec.	Actividad	Clase de Equivalencia	Resultado esperado
1	La auxiliar encargado de admisión ingresa los datos de contacto	Valida	Muestra completar los datos correctamente

Escenario: CP-03_E03 Comprobar que el sistema genere el mensaje de error al no encontrar datos de ubicación

Tabla 24.

Sec.	Actividad	Clase de Equivalencia	Resultado esperado
1	La auxiliar encargado de El encargado de admisión ingresa los datos de ubicación	Valida	Muestra completar los datos correctamente

Escenario: CP-03_E03 Comprobar que el sistema genere el mensaje de error al tener datos de paciente pero no de contacto

Tabla 25.

Sec.	Actividad	Clase de Equivalencia	Resultado esperado
1	La auxiliar encargado de admisión ingresa los datos del paciente	Valida	Muestra completar los datos correctamente
2	La auxiliar encargado de admisión ingresa los datos de contacto	Valida	Muestra completar los datos correctamente

Escenario: CP-03_E03 Comprobar que el sistema genere el mensaje de error al tener datos de contacto pero no de ubicación

Tabla 26.

Sec.	Actividad	Clase de Equivalencia	Resultado esperado
1	La auxiliar encargado admisión ingresa los datos de contacto	Valida	Muestra completar los datos correctamente
2	La auxiliar encargado de admisión ingresa los datos de ubicación	Valida	Muestra completar los datos correctamente

Caso de Uso: CU_04 Ingresar Diagnostico
Caso de Prueba: CU_04 Ingresar Diagnostico
Actor: Doctor
Propósito: Registrar Diagnostico
Pre Condiciones: El encargado de admisión esta logeado en el sistema

Escenario: CP-04_E04 Comprobar el correcto registro del Diagnostico

Tabla 27.

Sec.	Actividad	Clase de Equivalencia	Resultado esperado
1	El medico ingresa los datos del paciente	Valida	Muestra datos del paciente
2	Solicita "Grabar"	Valida	Muestra el diagnostico Registrado Exitosamente



Escenario: CP-04_E04: Comprobar que el sistema genere el mensaje de error al no encontrar datos de paciente

Tabla 28.

Sec.	Actividad	Clase de Equivalencia	Resultado esperado
1	El medico ingresa los datos del paciente	Valida	Muestra completar los datos correctamente
	Solicita "Grabar"	Valida	Muestra el diagnostico guardado Exitosamente

Capítulo VI: Aspectos Administrativos

6.01 Recursos

Capital humano

Para la realización del proyecto han sido necesarias las siguientes personas.

- Equipo de trabajo: Guevara Changotaxi Nelson Patricio
- Asesores: Ing. Johnny Coronel, Ing. Richard Mafla, Ing. Diana Teran.
- Usuarios: Administrador y personal de Clínica Odontológica "ADENTAL".

Recursos Materiales

Los materiales necesarios para la ejecución del proyecto son los siguientes.

- Equipos: Computador Portátil.
- Bibliografía: Tutoriales y manuales de programación C# .NET.
- Copias: Impresión del documento de la realización del proyecto.
- Servicios: Internet Inalámbrico, energía eléctrica.
- Transporte: Pago de pasajes para traslado al ITSCO.

Recursos Económicos

El costo total del proyecto es de \$2100,00 dólares, tomando como referencia, las facturas de los equipos y recursos necesarios para la ejecución del mismo.



6.02 Presupuesto

Ingresos

Aporte personal.....2100

EGRESOS VALOR

Elaboración del proyecto 500

Material de escritorio..... 200

Copias..... 200

Adquisición de equipos..... 550

Gastos administrativos..... 200

Transporte..... 150

Imprevistos.....300

TOTAL **2100**

6.03 Cronograma

El siguiente cronograma corresponde a la programación de las actividades básicas del proyecto y la distribución del tiempo estimado para su cumplimiento.

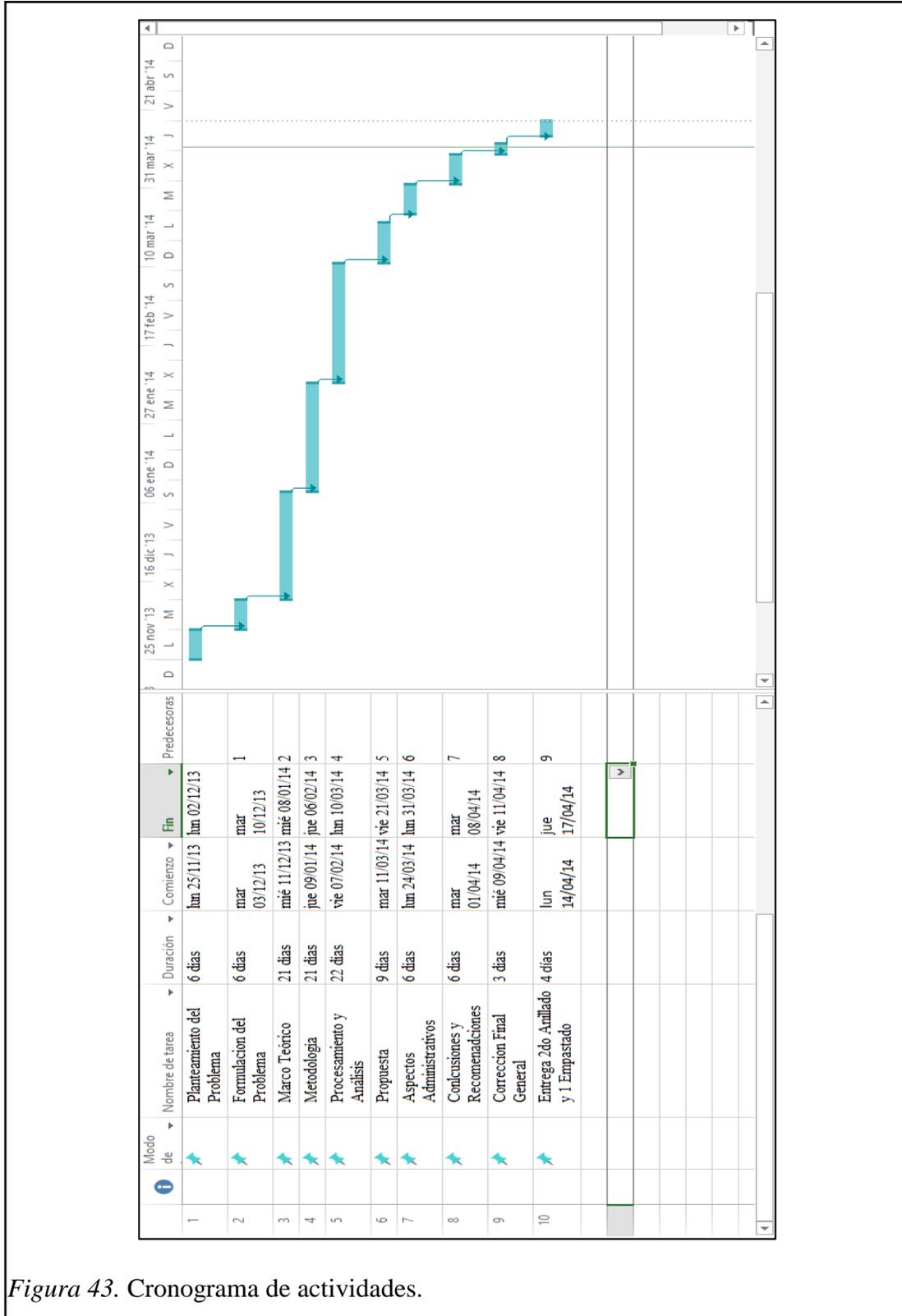


Figura 43. Cronograma de actividades.

Capítulo VII: Conclusiones y Recomendaciones

7.01 Conclusiones

Al finalizar este trabajo de investigación se han podido abstraer las siguientes principales conclusiones:

- Es importante realizar un debido proceso de recolección y análisis de los requerimientos, ya que el sistema ya tiene una base con el *"GESTION DE ATENCION DE HISTORIAS CLÍNICAS, TOMA DE TURNOS PARA ADENTAL CLÍNICA ODONTOLÓGICA DE LA CIUDAD DE QUITO"* que se usa desde inicios de las atenciones con computadoras. Al realizar un adecuado proceso de ingeniería de software facilita y garantiza en gran parte el éxito del desarrollo del proyecto.
- El sistema de información para la gestión de turnos, disminuiría los costos evaluados en factor tiempo y costo operativos ya que no se tendría que trabajar con teleoperadoras. Ya que el internet es un nuevo canal para asignar las citas médicas y esto disminuiría el flujo de llamadas de parte de los pacientes para solicitar sus citas, con esto los empleados encargados de esta función podrían lograr un mejor desempeño hasta que definitivamente ya no habría este canal por llamadas telefónicas.
- Satisfacer todos los requerimientos funcionales y no funcionales para el sistema de citas médicas.
- El sistema de información al ser diseñado con grandes ventajas frente a las aplicaciones de escritorio de este tipo. Estas ventajas le dan un valor agregado al producto y a su vez permite a los centros asistenciales innovar en la prestación de sus servicios, ofreciendo mejoras en infraestructura y personal médico por especialidad que mantengan un record de calidad de atención.



7.02 Recomendaciones

- Se recomienda que en nuevas versiones del sistema se maneje información mas detallada de los responsables del paciente y se admita más de una responsable por paciente.
- Se recomienda implementar una solución de inteligencia de negocios que permita realizar un análisis de la información que facilite una mejor toma de decisiones estratégicas para la clínica.
- Se recomienda que la base de datos esté alojada en un servidor exclusivo para no tener problemas de rendimiento a futuro.
- Se recomienda que se capacite al personal en el uso de nuevas tecnologías de la información para que puedan implementar nuevos módulos al sistema y mantenerlos.



ANEXOS

A.01. Matriz de análisis del impacto de los objetivos

Factibilidad (Alta.Media.Baja) (4 - 2 - 1)	Impacto en genero (Alta.Media.Baja) (4 - 2 - 1)	Impacto Ambiental (Alta.Media.Baja) (4 - 2 - 1)	Relevancia (Alta.Media.Baja) (4 - 2 - 1)	Sostenibilidad (Alta.Media.Baja) (4 - 2 - 1)	Total
<ul style="list-style-type: none"> -Cuenta con financiamiento propio -Es aceptable y conveniente para los pacientes -Existe tecnología adecuada para su realización -Los beneficios son mayores que los costos -Pérdida de tiempo de espera 	<ul style="list-style-type: none"> -Fortalece la aplicación de los derechos de los pacientes -Incrementa el valor educativo de los empleados -Incrementa los ingresos familiares -Aumenta la confianza y autoestima de los empleados 	<ul style="list-style-type: none"> -Favorece la educación ambiental -Mejora el entorno social. -Contribuye a proteger el entorno físico. -Mejora el entorno cultural. -Protege el uso de los recursos. -Enseña a reciclar. 	<ul style="list-style-type: none"> -Es una prioridad sentida por los pacientes. -Los beneficios son deseados por los pacientes. -Beneficia a los empleados y pacientes. -Responde las expectativas de la Clínica Odontológica. -Satisface la expectativa del personal administrativo. 	<ul style="list-style-type: none"> -Se puede conseguir financiamiento a futuro. -Se fortalece la organización de la Clínica Odontológica. -Favorece la participación de los empleados y pacientes. -Disponibilidad de información a futuro. -Es un proceso innovador 	<p>96 puntos</p> <p>20-40 Baja</p> <p>40-60 Media Baja</p> <p>60-80 Media Alta</p> <p>80-100 Alta</p> <p>Total</p>
<p>20 Puntos</p>	<p>16 Puntos</p>	<p>24 Puntos</p>	<p>18 Puntos</p>	<p>18 Puntos</p>	<p>Total</p>

Agilidad al asignar turnos y el llenado de las Historias Clínicas



Bibliografía

<http://vilela.pe/blogdental/tag/importancia-odontograma>

http://www.bvs.sld.cu/revistas/mgi/vol20_2_04/mgi05204.htm

<http://www.centrodental31deagosto.com/2012/07/diagno%CC%81stico-en-odontologi%CC%81a/>

http://es.wikipedia.org/wiki/Diagrama_de_secuencia

http://es.wikipedia.org/wiki/Diagrama_de_clases

http://es.wikipedia.org/wiki/Caso_de_uso

<http://es.scribd.com/doc/11032774/Modelamiento-en-Rational-Rose-Migrarlo-a-SQL-SERVER>

http://www.ibm.com/developerworks/ssa/rational/library/08/0129_ali_gangulli/

http://books.google.com.ec/books?id=ExK0AQRjPk4C&pg=PR14&lpg=PR14&dq=modelamientos+de+base+de+datos+sql+server+2008+r2&source=bl&ots=v75XpXS KIE&sig=U1D_HlxqdmM6JOztEiVs8zippSM&hl=es&sa=X&ei=ddpOU5HsAYaE0AGfIIDAAQ&ved=0CF8Q6AEwCQ#v=onepage&q=modelamientos%20de%20base%20de%20datos%20sql%20server%202008%20r2&f=false

http://www.clubdigital.mx/assets/files/course/237/526a910d7604bManual_Visual_Studio_2010_.pdf

<http://webtelematica.wordpress.com/2010/02/16/tutorial-desde-0-crear-base-de-datos-en-sql-server-20052008-usando-sql-server-management-studio-express/>

<http://www.devtroce.com/2010/07/12/como-programar-en-n-capas-con-c-parte-1/>

Manual de Usuario del Sistema

Ingreso al sistema de gestión de historias clínicas de “ADENTAL” Clínica Odontológica.

Para ingresar al sistema debemos hacer doble clic en el icono del sistema ya previamente instalado

Nos aparecerá la pantalla que nos pide digitar Usuario y Clave



Figura 43. Pantalla inicial de ingreso de usuario

A esta opción accede desde el menú principal *Sistema*, y dentro de éste seleccionando *Usuarios y Clave*.

Permite realizar el mantenimiento de los distintos Usuarios que tendrán acceso al sistema y Personas que el sistema manejará.

Si ingresa incorrecto el nombre de usuario y clave indicara que son nombres no validos

Configuración de Tipo de Usuarios que van a tener acceso al sistema



Figura 44. Ingreso a configurar usuarios

Presenta como podemos acceder a la configuración del tipo de usuario.

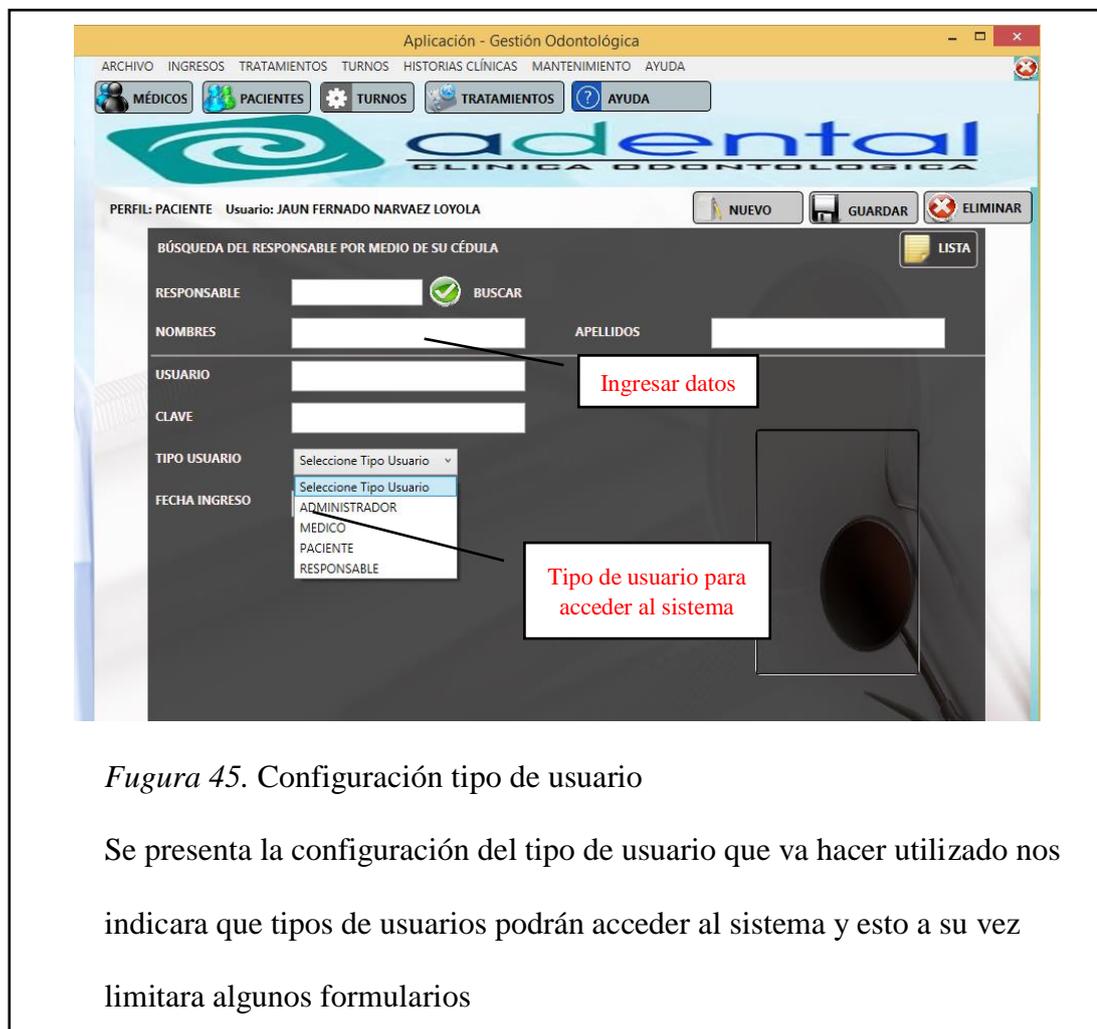


Figura 45. Configuración tipo de usuario

Se presenta la configuración del tipo de usuario que va hacer utilizado nos indicara que tipos de usuarios podrán acceder al sistema y esto a su vez limitara algunos formularios

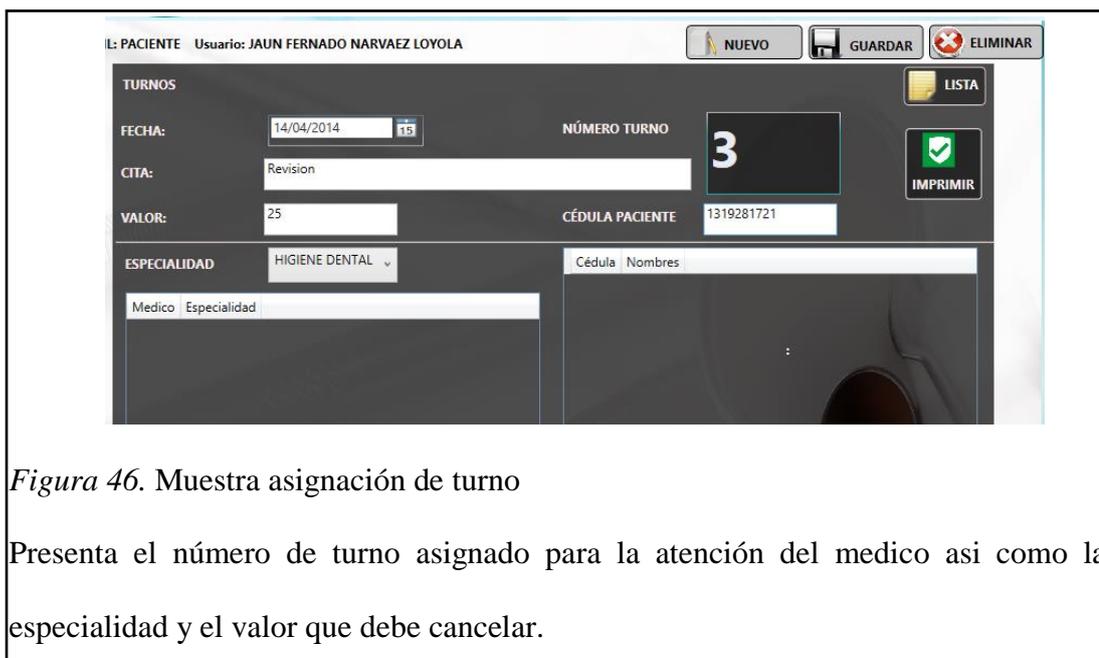
Registro de asignación de un Turno

Para registrar un turno se deben seguirse los siguientes pasos:

- Hacemos clic en el siguiente botón



- Ubicar al paciente en la pantalla de búsqueda
- Clic sobre el número de Historia Clínica que en este caso es el número de cedula.



IL: PACIENTE Usuario: JAUN FERNADO NARVAEZ LOYOLA

NUEVO GUARDAR ELIMINAR

TURNOS LISTA

FECHA: 14/04/2014 NÚMERO TURNO 3

CITA: Revisión IMPRIMIR

VALOR: 25 CÉDULA PACIENTE 1319281721

ESPECIALIDAD HIGIENE DENTAL

Cédula Nombres

Medico	Especialidad

Figura 46. Muestra asignación de turno

Presenta el número de turno asignado para la atención del medico así como la especialidad y el valor que debe cancelar.

Administración de Médicos

Para registrar los médicos se debe hacer lo siguiente:

- Clic en el botón



- Muestra la siguiente pantalla que gestiona el registro, actualización y habilitación de los médicos
- Muestra la hora de ingreso y salida que se le asigna a cada medico.

Busqueda de médicos



BUSQUEDA DEL RESPONSABLE POR MEDIO DE SU CEDULA LISTA

CÉDULA: 050263139

PRIMER NOMBRE: SEGUNDO SEGUNDO NOMBRE: RUBEN

PRIMER APELLIDO: AREQUIPA SEGUNDO APELLIDO: TOAQUIZA

TELÉFONO: 022464722

CELULAR: 0986573763

GÉNERO: MASCULINO FEMENINO

RELOJ: 14 : 00

HORA INGRESO: 09:00

HORA SALIDA: 14:00

TURNOS MÁXIMOS: 5

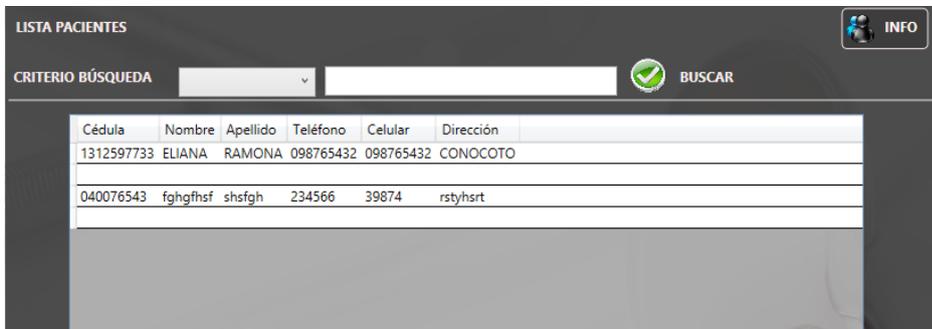
Figura 47. Se presenta el registro de un médico nuevo indicando el número de turnos (pacientes) máximos para atender y la hora de entrada y salida

Administración de Pacientes

- Hacemos clic en el siguiente botón



- Podemos ver que tenemos una lista de los pacientes agregados



LISTA PACIENTES INFO

CRITERIO BÚSQUEDA: BUSCAR

Cédula	Nombre	Apellido	Teléfono	Celular	Dirección
1312597733	ELIANA	RAMONA	098765432	098765432	CONOCOTO
040076543	fgghfsf	shsfgh	234566	39874	rstyhsrt

Figura 48. Permite ver el listado de pacientes ya agregados así evitarnos ingresar otro paciente con el mismo número de cédula aunque el sistema no lo permite.

Registro de un nuevo Paciente

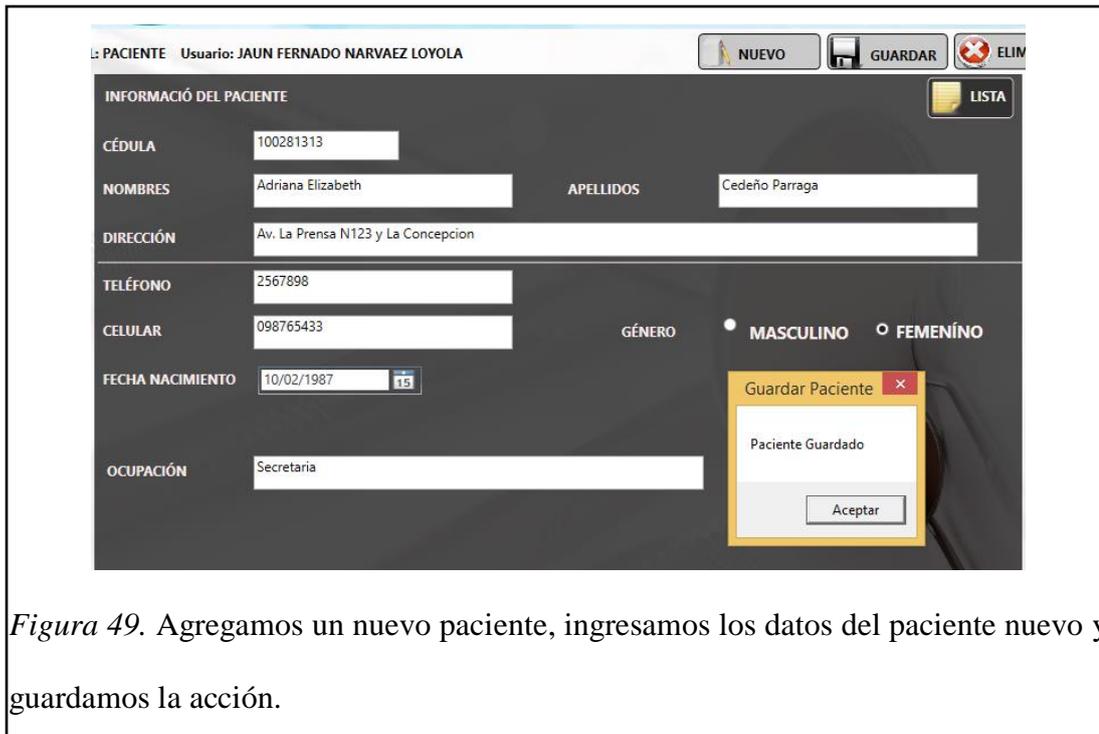


Figura 49. Agregamos un nuevo paciente, ingresamos los datos del paciente nuevo y guardamos la acción.

Administración de Historia Clínica

- Damos clic en el siguiente menú para acceder a la historia clínica



Figura 50. Muestra cómo acceder al formulario Historias Clínicas

A continuación aparecerá la pantalla de registro de nueva Historia Clínica

Figura 51. Muestra los datos que son ingresados en la Historia Clínica así como se encuentra el registro del diagnóstico y el pronóstico del paciente.

Administrar Tratamientos

- Hacemos clic en el siguiente botón para poder acceder



- Muestra la pantalla principal de tratamientos
- Para acceder al tratamiento respectivo debemos cargar la historia clínica para ver la información del paciente y seguir con el tratamiento

Visualización de Historia Clínica para luego acceder al Tratamiento



The screenshot shows a web-based form for patient history. The form is titled "TRATAMIENTO AL PACIENTE:" and includes a search bar with the number "1312597733" and a button labeled "HISTORIAL CLÍNICO". The form is divided into several sections: "NOMBRES" (ELIANA) and "APELLIDOS" (RAMONA); "DIRECCIÓN" (CONOCOTO); "TELF:" (098765432) and "CELULAR" (098765432); "FECHA TRATAMIENTO" (14/04/2014) with a calendar icon; "BÚSQUEDA TURNO" (1); "MÉDICO" (CORTEZ PEDRO); "ESPECIALID" (ODONTOLOGIA GENERAL); "CITA:" (revisión); and "HORARIO" (Hora Inico: 8:00:00 Hora Fin: 8:00:00). On the right side, there are dropdown menus for "TRATAMIENTO:" (PERIODONCIA), "REFERENCIAS MÉDICAS" (Ninguna), and "GRUPO DENTAL:" (SUP-IZQUIERDO). There is a green checkmark icon and a button labeled "ODONTOGRÁMA". Below this, there is a section for "OBSERVACIONES:" with a large empty text area and a button labeled "CARGAR OBSERVACIONES".

Figura 52. Muestra los datos del paciente previo cargar la historia clínica donde nos indicara el tratamiento que se le está realizando.

Odontograma

El odontograma es aquella representación gráfica de lo que encontramos en la boca de un paciente, principalmente en las piezas dentales, las cuales están representadas en sus cinco caras (las proximales, la anterior, la posterior y la de masticación)

En este documento de carácter médico – legal se grafica lo que ya está hecho como lo que se tiene por hacer. Tiene una nomenclatura internacional que permite ser interpretado y entendido por cualquier profesional odontólogo.

Dentro de esta nomenclatura está que todo lo que tiene por hacerse un paciente en los dientes se pinta de color rojo y lo que ya está hecho se pinta de color azul.

El odontograma nos sirve principalmente a nosotros los dentistas para poder llevar un registro exacto de lo que hemos hecho y lo que ya encontramos en un paciente que acude a nuestra consulta, pero el correcto registro de este documento va más allá de esto.



Muchas veces es realizado de una manera poco seria por parte del dentista, olvidándonos que este documento puede servir incluso para poder identificar a una persona que haya perecido en algún accidente y que no pueda ser identificado por su rostro, huellas digitales o señas particulares.

Diagnostico

Definición: Es el procedimiento por el cual se identifica una enfermedad dental o bucal Para llegar a ello se necesitan dos circunstancias interdependientes:

- 1- Conocimiento y preparación del profesional que nos trate.
- 2- Disponer de los medios diagnósticos adecuados que nos ayuden a confirmar dicha enfermedad gracias a la correcta interpretación de las pruebas realizadas.

En odontología se han extraído piezas dentales sanas, se han realizado obturaciones que precisaban endodoncias, así como endodoncias que no eran necesarias, se han colocado restauraciones protésicas con raíces incluidas... ; en fin, multitud de malas actuaciones por no llegar a un diagnóstico correcto.

A veces damos poca importancia a la odontología, nos basamos en si un tratamiento vale más o menos dinero, sin pedir pruebas diagnósticas. No valoramos la importancia que puede tener una enfermedad dental o bucal y su repercusión en el resto del organismo.

Pronostico

Definicion: El pronóstico es la previsión del surgimiento, el carácter del desarrollo y el término de la enfermedad, basada en el conocimiento de las regularidades del curso de los procesos patológicos. El pronóstico se refiere a los resultados de una enfermedad y la frecuencia con que se espera que ocurran. Es una preocupación inherente a la clínica. En la práctica médica es constante el ejercicio de la predicción del curso futuro de una enfermedad, de sus resultados y de su frecuencia. Esta



previsión puede ser importante para la decisión de tratar a un paciente y cómo tratarlo, reducir su ansiedad, así como promover determinadas conductas en las personas que atendemos.

Además de ser el pronóstico algo indispensable para un tratamiento exitoso, si es correcto, aumenta la autoridad del médico y la confianza del enfermo en este. Lo primero que interesa al médico y al paciente es si es o no mortal la enfermedad, y si es mortal, cuánto tiempo va a vivir el enfermo. Lo segundo es, si se curará o no por completo, si la afección va a dejar secuelas o incapacidades, qué tiempo va a durar, cómo va a transcurrir la enfermedad al principio y en lo posterior, si hay recidivas o remisiones; y si es una enfermedad crónica, qué limitaciones va a producir en la vida del paciente, cuáles son las complicaciones de la enfermedad, y cuáles son los efectos de los tratamientos existentes en el curso de ella.

Todos estos aspectos deben ser de dominio del médico para optimizar el tratamiento y su relación con el paciente y la familia.

Registro en el Odontograma

Muestra la representación gráfica de lo que encontramos en la boca de un paciente principalmente las piezas dentales

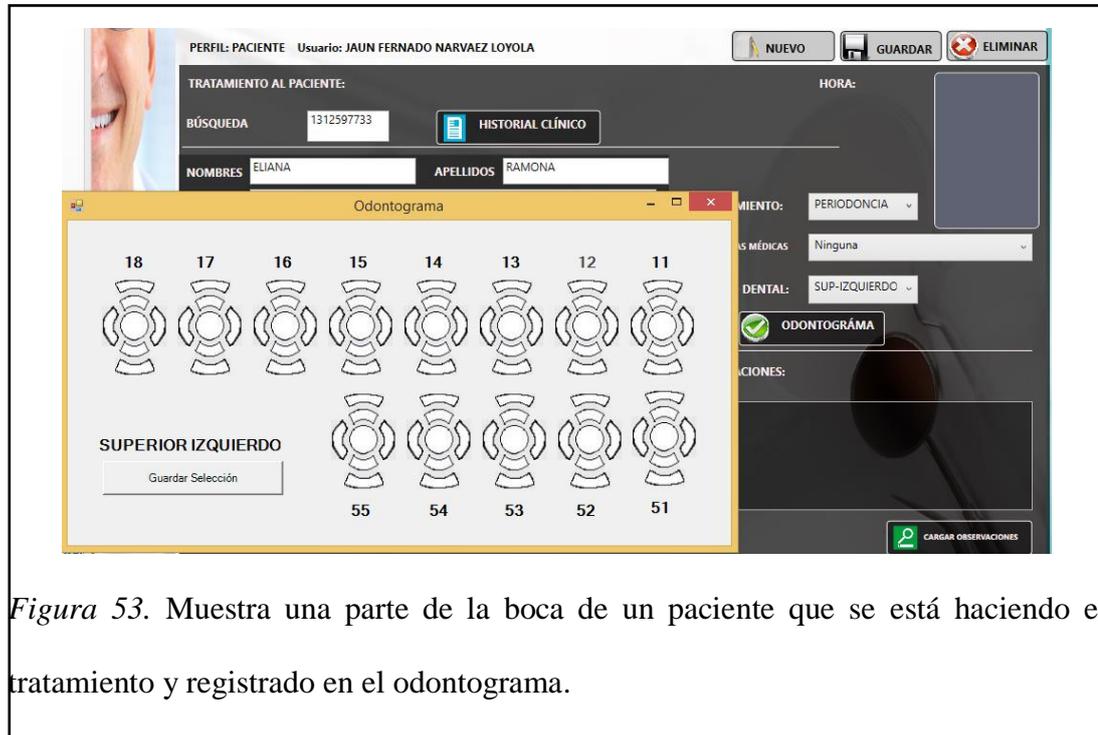


Figura 53. Muestra una parte de la boca de un paciente que se está haciendo el tratamiento y registrado en el odontograma.

Administración de Mantenimiento

- Para acceder al mantenimiento del sistema ingresamos al siguiente menú
- Tenemos varias opciones como
 - Areas. Se encarga de agregar otras áreas más en la clínica
 - Cargos. Se puede agregar cargos y de personal
 - Especialidades. Registra los tipos de tratamientos que existen
 - Tipo Usuarios.
 - Tratamientos

Pantalla Mantenimientos

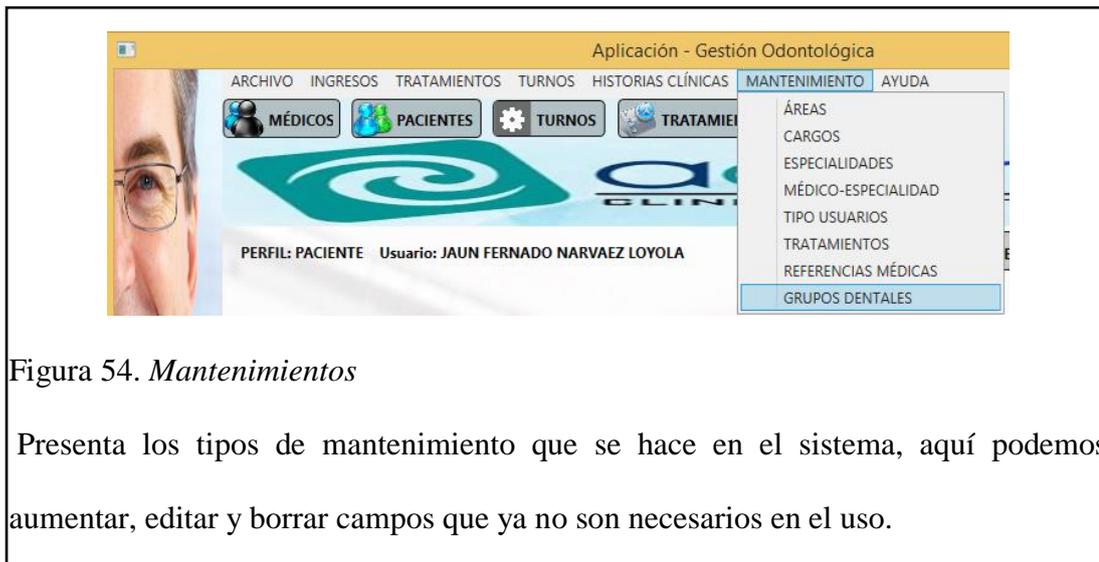


Figura 54. *Mantenimientos*

Presenta los tipos de mantenimiento que se hace en el sistema, aquí podemos aumentar, editar y borrar campos que ya no son necesarios en el uso.

Diccionario de Datos

1. Introducción

En el siguiente diccionario de datos contiene las características lógicas y puntuales de los datos, aquí se encuentra la lista de todos los elementos que forman parte del flujo de datos de todo el sistema.

Tabla 29.

Tabla Pacientes

Tabla	Campo	Tipo Dato	Longitud	Llave primaria	Tabla Foranea
Tbl_Paciente	Id_Pciente	Int		PK	
	Cedula	Varchar	12		
	Apellido	Varchar	140		
	Nombre	Varchar	140		
	FechaNacimie nto	Datetime			
	Direccion	Text			
	Fono	Varchar	15		
	Celular	Varchar	15		
	Ocupacion	Varchar	100		
	Sexo	Char	1		
	Estado	Char	1		

Tabla 30.
 Tabla Medicos

Tabla	Campo	Tipo Dato	Longitud	Llave primaria	Tabla Foranea
Tbl_Medico	Id_medico	Int		PK	
	Med_PrimerNom	Varchar	60		
	Med_SegundoNom	Varchar	60		
	Med_ApellidoMat	Varchar	60		
	Med_ApellidoPater	Varchar	70		
	Med_Fono	Varchar	15		
	Med_Celular	Varchar	15		
	Med_Horaentrada	Datetime			
	Med_HoraSalida	Datetime			
	Med_TurnosMax	Int			
	Med_Sexo	Char	1		
	Med_Cedula	Varchar	12		
	Estado	Char	1		

Tabla 31.
 Tabla Historia Clinica

Tabla	Campo	Tipo Dato	Longitud	Llave primaria	Tabla Foranea
Tbl_Historia Clinica	Id_Historial	Int		PK	
	His_Fecha	Datetime			
	His_Hora	Varchar	10		
	Id_Turno	int			
	His_Diagnostico	varchar	100		
	His_Pronostico	Varchar	100		
	Id_Paciente	int			
	Estado	char	1		
	His_Foto	text			

Tabla 32.

Tabla Usuarios

Tabla	Campo	Tipo Dato	Longitud	Llave primaria	Tabla Foranea
Tbl_Usuarios	Id_Usuarios	Int		PK	
	Nom_Usuario	varchar	20		
	Usu_Clave	Varchar	40		
	Usu_Estado	Char	1		
	Usu_Fecha_Ingreso	Datetime			
	Id_TipoUsuarios	Int			
	Id_Responsable	Int			

Tabla 33.

Tabla Area

Tabla	Campo	Tipo Dato	Longitud	Llave primaria	Tabla Foranea
Tbl_Area	Id_Area	Int		PK	
	Descripcion	varchar	70		
	Estado	Char	1		

Tabla 34.

Tabla Cargo

Tabla	Campo	Tipo Dato	Longitud	Llave primaria	Tabla Foranea
Tbl_Cargo	Id_Cargo	Int		PK	
	Descripcion	varchar	70		
	Estado	Char	1		

Tabla 35.

Tabla Especialidad

Tabla	Campo	Tipo Dato	Longitud	Llave primaria	Tabla Foranea
Tbl_Especialidad	Id_Especialidad	Int		PK	
	Descripcion	varchar	70		
	Estado	Char	1		

Tabla 36.

Tabla GrupoDiente

Tabla	Campo	Tipo Dato	Longitud	Llave primaria	Tabla Foranea
tbl_Grupodiente	Id_Grupo	Int		PK	
	Gd_Nombre	Text			
	Gd_Numeracion	Varchar	100		
	Estado	char	1		

Tabla 37.

Tabla Horario

Tabla	Campo	Tipo Dato	Longitud	Llave primaria	Tabla Foranea
Tbl_Horario	Id_Horario	int			
	Hor_HR_Inicio	datetime		PK	
	Hor_Hr_Fin	datetime			
	Hor_Lunes	tinyint			
	Hor_Martes	tinyint			
	Hor_Miercoles	tinyint			
	Hor_Jueves	tinyint			
	Hor_Viernes	tinyint			
	Hor_Estado	char	1		

Tabla 38.

Tabla Responsable

Tabla	Campo	Tipo Dato	Longitud	Llave primaria	Tabla Foranea
Tbl_Responsable	Id_Responsable	Int		PK	
	Res_Nombre	Varchar	70		
	Res_Apellido	Varchar	60		
	Res_Titulo	Varchar	100		
	Res_Fono	Varchar	15		
	Res_Celular	Varchar	14		
	Res_Estado	Char	1		
	Id_Area	Int			
	Id_Cargo	Int			
	Res_Cedula	varchar	12		



Código Fuente

Usuario

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using Acceso_Datos;
using System.Data.Linq;

namespace Logica_Nogocios
{
    public class Usuario_Logica
    {
        public static odontologicoDataContext dc = new
odontologicoDataContext();
        public static List<Tbl_Usuarios> ObtenerUsuario()
        {
            var lista = dc.Tbl_Usuarios.Where(c => c.Usu_Estado == 'A');
            return lista.ToList();
        }

        public static List<Tbl_Usuarios> ObtenerUsuarioXTipoUsuario(int Tipo)
        {
            var lista = dc.Tbl_Usuarios.Where(c => c.Usu_Estado == 'A' &
c.Id_TipoUsuarios.Equals(Tipo));
            return lista.ToList();
        }
        public static Tbl_Usuarios ObtenerUsuarios_IdInfo(int id)
        {
            var lista = dc.Tbl_Usuarios.FirstOrDefault(c => c.Usu_Estado ==
'A' & c.Id_Usuarios.Equals(id));
            return lista;
        }

        public static bool AutenticarUsuario(string Nombre, Byte[] clave)
        {
            try
            {
                string clavestring = ASCIIEncoding.ASCII.GetString(clave);
                var lista = dc.Tbl_Usuarios.Any(u => u.Usu_Estado == 'A'
&
u.Nom_Usuario.Equals(Nombre)
&
u.Usu_Clave.Equals(clavestring));
                return lista;
            }
            catch (Exception ex)
            {
                throw new ArgumentException(ex.Message);
            }
        }

        public static Tbl_Usuarios UsuarioActual(string Nombre, Byte[] clave)
        {
            try
            {
                string clavestring = ASCIIEncoding.ASCII.GetString(clave);
                var lista = dc.Tbl_Usuarios.FirstOrDefault(u => u.Usu_Estado
== 'A'
```



```

        &
u.Nom_Usuario.Equals(Nombre)
        &
u.Usu_Clave.Equals(clavestring));
    return lista;
}
catch (Exception)
{
    return null;
}
}

private static int Id_Codigo()
{
    try
    {
        return dc.Tbl_Usuarios.Max(c => c.Id_Usuarios + 1);
    }
    catch (Exception)
    {
        return 1;
    }
}
public static void Save(Tbl_Usuarios UsuarioInfo)
{
    try
    {
        Validacion(UsuarioInfo);
        dc.ExecuteCommand("Insert into Tbl_Usuarios(Id_Usuarios,
Nom_Usuario, Usu_Clave, Usu_Estado, Usu_Fecha_Ingreso,
Id_TipoUsuarios,Id_Responsable"
+ ") values({0},{1},{2},{3},{4},{5},{6})", new object[]
{
        UsuarioInfo.Id_Usuarios = Id_Codigo(),
        UsuarioInfo.Nom_Usuario,
        UsuarioInfo.Usu_Clave,
        UsuarioInfo.Usu_Estado = 'A',
        UsuarioInfo.Usu_Fecha_Ingreso,
        UsuarioInfo.Id_TipoUsuarios,
        UsuarioInfo.Id_Responsable
    });
    }
    catch (Exception ex)
    {
        throw new ArgumentException("Usuario no Guardado\n" +
ex.Message);
    }
}

public static void Modify(Tbl_Usuarios UsuarioInfo)
{
    try
    {

```



```
dc.ExecuteNonQuery("Update Tbl_Usuarios set Nom_Usuario={0},
Usu_Clave={1}, Usu_Estado={2}, Usu_Fecha_Ingreso={3},
Id_TipoUsuarios={4},Id_Responsable={5} where Id_Usuarios={6}", new object[]
{
    UsuarioInfo.Nom_Usuario,
    UsuarioInfo.Usu_Clave,
    UsuarioInfo.Usu_Estado = 'A',
    UsuarioInfo.Usu_Fecha_Ingreso,
    UsuarioInfo.Id_TipoUsuarios,
    UsuarioInfo.Id_Responsable,
    UsuarioInfo.Id_Usuarios
});
}
catch (Exception ex)
{
    throw new ArgumentException("Usuario Modificado" +
ex.Message);
}

}

public static void Delete(Tbl_Usuarios UsuarioInfo)
{
    try
    {
        dc.ExecuteNonQuery("Update Tbl_Usuarios set Usu_Estado={0}
where Id_Usuarios={1}", new object[]
{
            UsuarioInfo.Usu_Estado = 'E',
            UsuarioInfo.Id_Usuarios
        });
    }
    catch (Exception ex)
    {
        throw new ArgumentException("Usuario Eliminado" + ex.Message);
    }
}

public static void Validacion(Tbl_Usuarios UsuarioInfo)
{
    string sms = "";
    if (UsuarioInfo.Id_Responsable == 0)
    {
        sms += "Ingresar el Responsable\n";
    }
    if (UsuarioInfo.Id_TipoUsuarios == 0)
    {
        sms += "Ingresar el Tipo Usuario\n";
    }

    if (UsuarioInfo.Nom_Usuario.Length <= 12)
    {
```



```
        sms += "Ingresar máximo 12 caracteres en el Nombre del  
Usuario\n";  
    }  
    if (UsuarioInfo.Usu_Clave.Length <= 8)  
    {  
        sms += "Ingresar máximo 8 caracteres en la clave del  
Usuario\n";  
    }  
  
    if (sms != "")  
    {  
        throw new ArgumentException(sms);  
    }  
    }  
    }  
}
```

Turno

```
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Text;  
using Acceso_Datos;  
using System.Data.Linq;  
  
namespace Logica_Nogocios  
{  
    public class Turno_Logica  
    {  
        public static odontologicoDataContext dc = new  
odontologicoDataContext();  
        public static List<Tbl_Turno> ObtenerTurnos()  
        {  
            var lista = dc.Tbl_Turno.Where(c => c.Estado == 'A');  
            return lista.ToList();  
        }  
  
        public static List<Tbl_Turno> ObtenerTurno_XPaciente(int Pac)  
        {  
            var lista = dc.Tbl_Turno.Where(c => c.Estado == 'A' &  
c.Id_Paciente.Equals(Pac));  
            return lista.ToList();  
        }  
  
        public static Tbl_Turno ObtenerTurno_XTurnoNumero(int Numero)  
        {  
            var lista = dc.Tbl_Turno.FirstOrDefault(c => c.Estado == 'A' &  
c.Tur_Numero.Equals(Numero));  
            return lista;  
        }  
  
        public static List<Tbl_Turno> ObtenerTurno_XResponsable(int Res)  
        {  
            var lista = dc.Tbl_Turno.Where(c => c.Estado == 'A' &  
c.Id_Responsable.Equals(Res));  
            return lista.ToList();  
        }  
  
        public static List<Tbl_Turno> ObtenerTurno_XMedEspecialista(int  
MedEsp)  
        {
```



```
        var lista = dc.Tbl_Turno.Where(c => c.Estado == 'A' &
c.Id_Medico_Espe.Equals(MedEsp));
        return lista.ToList();
    }
    public static Tbl_Turno ObtenerTurno_IdInfo(int id)
    {
        var lista = dc.Tbl_Turno.FirstOrDefault(c => c.Estado == 'A' &
c.Id_Turno.Equals(id));
        return lista;
    }

private static int Id_Codigo()
{
    try
    {
        return dc.Tbl_Turno.Max(c => c.Id_Turno + 1);
    }
    catch (Exception)
    {
        return 1;
    }
}
public static void Save(Tbl_Turno TurnoInfo)
{
    try
    {
        Validacion(TurnoInfo);
        dc.ExecuteCommand("Insert into Tbl_Turno(Id_Turno, Tur_Fecha,
Tur_Cita, Tur_Valor, Tur_Numero, Id_Medico_Espe," +
        "Id_Responsable, Id_Paciente, Estado,Id_Horarios)" +
        "values({0},{1},{2},{3},{4},{5},{6},{7},{8},{9})", new
object[]
        {
            TurnoInfo.Id_Turno = Id_Codigo(),
            TurnoInfo.Tur_Fecha,
            TurnoInfo.Tur_Cita,
            TurnoInfo.Tur_Valor,
            TurnoInfo.Tur_Numero,
            TurnoInfo.Id_Medico_Espe,
            TurnoInfo.Id_Responsable,
            TurnoInfo.Id_Paciente,
            TurnoInfo.Estado = 'A',
            TurnoInfo.Id_Horarios
        }
    );
    }
    catch (Exception ex)
    {
        throw new ArgumentException("Turno no Guardado\n" +
ex.Message);
    }
}

public static void Modify(Tbl_Turno TurnoInfo)
{
    try
```



```
{
    dc.ExecuteCommand("Update Tbl_Turno set Tur_Fecha={0},
Tur_Cita={1}, Tur_Valor={2}, Tur_Numero={3}, Id_Medico_Espe={4}," +
        "Id_Responsable={5}, Id_Paciente={6}, Estado={7},
Id_Horarios={8} where Id_Turno={9}", new object[]
    {
        TurnoInfo.Tur_Fecha,
        TurnoInfo.Tur_Cita,
        TurnoInfo.Tur_Valor,
        TurnoInfo.Tur_Numero,
        TurnoInfo.Id_Medico_Espe,
        TurnoInfo.Id_Responsable,
        TurnoInfo.Id_Paciente,
        TurnoInfo.Estado = 'A',
        TurnoInfo.Id_Horarios,
        TurnoInfo.Id_Turno
    }
);
}
catch (Exception ex)
{
    throw new ArgumentException("Turno Modificados" + ex.Message);
}

public static void Delete(Tbl_Turno TurnoInfo)
{
    try
    {
        dc.ExecuteCommand("Update Tbl_Turno set Estado={0} where
Id_Turno={1}", new object[]
        {
            TurnoInfo.Estado = 'E',
            TurnoInfo.Id_Turno
        }
    );
    }
    catch (Exception ex)
    {
        throw new ArgumentException("Turno Eliminado" + ex.Message);
    }
}

private static void Validacion(Tbl_Turno TurnoInfo)
{
    string sms = "";
    if (TurnoInfo.Tur_Fecha == null)
    {
        sms += "Ingresar la Fecha de Turno\n";
    }
    else
    {
        if (TurnoInfo.Tur_Fecha < DateTime.Now.Date)
        {
            sms += "La fecha debe ser mayor a la actual\n";
        }
    }
}
```

```
    }  
  }  
  if (TurnoInfo.Tur_Cita == "")  
  {  
    sms += "Ingresar la descripción de la cita\n";  
  }  
  if (TurnoInfo.Id_Medico_Espe == 0)  
  {  
    sms += "Ingresar el médico especialista\n";  
  }  
  if (TurnoInfo.Id_Paciente == 0)  
  {  
    sms += "Ingresar el Paciente\n";  
  }  
  if (TurnoInfo.Id_Horarios == 0)  
  {  
    sms += "Ingresar la Hora\n";  
  }  
  if (TurnoInfo.Tur_Valor == 0)  
  {  
    sms += "Ingresar la Valor\n";  
  }  
  if (sms != "")  
  {  
    throw new ArgumentException(sms);  
  }  
}  
}
```

Pacientes

```
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Text;  
using Acceso_Datos;  
using System.Data.Linq;  
  
namespace Logica_Nogocios  
{  
    public class Pacientes_Logica  
    {  
        public static odontologicoDataContext dc = new  
odontologicoDataContext();  
        public static List<Tbl_Paciente> ObtenerPacientes()  
        {  
            var lista = dc.Tbl_Paciente.Where(c => c.Estado == 'A');  
            return lista.ToList();  
        }  
  
        public static List<Tbl_Paciente> ObtenerPaciente_XCedula(string ced)  
        {  
            var lista = dc.Tbl_Paciente.Where(c => c.Estado == 'A' &  
c.Cedula.Equals(ced));  
            return lista.ToList();  
        }  
  
        public static List<Tbl_Paciente> ObtenerPaciente_XCedula(int ced)  
        {
```



```
        var lista = dc.Tbl_Paciente.Where(c => c.Estado == 'A' &
c.Cedula.Equals(ced));
        return lista.ToList();
    }

    public static bool existePaciente(string cedula)
    {
        var existe = dc.Tbl_Paciente.Any(p => p.Estado == 'A' &
p.Cedula.Equals(cedula));
        return existe;
    }
    public static Tbl_Paciente ObtenerPaciente_XCedulaInfo(string ced)
    {
        var lista = dc.Tbl_Paciente.FirstOrDefault(c => c.Estado == 'A' &
c.Cedula.Equals(ced));
        return lista;
    }
    public static List<Tbl_Paciente> ObtenerPaciente_XApellido(string
Apellido)
    {
        var lista = dc.Tbl_Paciente.Where(c => c.Estado == 'A' &
c.Apellido.Equals(Apellido));
        return lista.ToList();
    }

    public static Tbl_Turno ObtenerPaciente_IdInfo(int id)
    {
        var lista = dc.Tbl_Turno.FirstOrDefault(c => c.Estado == 'A' &
c.Id_Paciente.Equals(id));
        return lista;
    }

    private static int Id_Codigo()
    {
        try
        {
            return dc.Tbl_Paciente.Max(c => c.Id_Paciente + 1);
        }
        catch (Exception)
        {
            return 1;
        }
    }
    public static void Save(Tbl_Paciente PacienteInfo)
    {
        try
        {
            Validacion(PacienteInfo);
            dc.ExecuteCommand("Insert into Tbl_Paciente(Id_Paciente,
Cedula, Apellido, Nombre, FechaNacimiento, Direccion," +
"Fono, Celular, Ocupacion, Sexo, Estado)" +
"values({0},{1},{2},{3},{4},{5},{6},{7},{8},{9},{10})",
new object[]
            {
                PacienteInfo.Id_Paciente = Id_Codigo(),
                PacienteInfo.Cedula,
                PacienteInfo.Apellido,
                PacienteInfo.Nombre,
                PacienteInfo.FechaNacimiento,
```



```
        PacienteInfo.Direccion,
        PacienteInfo.Fono,
        PacienteInfo.Celular,
        PacienteInfo.Ocupacion,
        PacienteInfo.Sexo,
        PacienteInfo.Estado = 'A'
    }
    );
}
}
catch (Exception ex)
{
    throw new ArgumentException("Paciente no Guardado\n" +
ex.Message);
}

}

public static void Modify(Tbl_Paciente PacienteInfo)
{
    try
    {
        dc.ExecuteCommand("Update Tbl_Paciente set Cedula={0},
Apellido={1}, Nombre={2}, FechaNacimiento={3}, Direccion={4}," +
"Fono={5}, Celular={6}, Ocupacion={7}, Sexo={8},
Estado={9} where Id_Paciente={10}", new object[]
{
        PacienteInfo.Cedula,
        PacienteInfo.Apellido,
        PacienteInfo.Nombre,
        PacienteInfo.FechaNacimiento,
        PacienteInfo.Direccion,
        PacienteInfo.Fono,
        PacienteInfo.Celular,
        PacienteInfo.Ocupacion,
        PacienteInfo.Sexo,
        PacienteInfo.Estado = 'A',
        PacienteInfo.Id_Paciente
    }
    );
}
catch (Exception ex)
{
    throw new ArgumentException("Paciente Modificado" +
ex.Message);
}

}

public static void Delete(Tbl_Paciente PacienteInfo)
{
    try
    {
        dc.ExecuteCommand("Update Tbl_Paciente Estado={0} where
Id_Paciente={1}", new object[]
{

```



```
                PacienteInfo.Estado = 'E',
                PacienteInfo.Id_Paciente
            }
        );
    }
    catch (Exception ex)
    {
        throw new ArgumentException("Paciente Eliminado" +
ex.Message);
    }
}

private static void Validacion(Tbl_Paciente PacienteInfo)
{
    string sms = "";
    if (PacienteInfo.Apellido == "")
    {
        sms += "Ingresar Apellido\n";
    }
    if (PacienteInfo.Cedula == "")
    {
        sms += "Ingresar Cédula'\n";
    }
    if (PacienteInfo.Direccion == "")
    {
        sms += "Ingresar Dirección del Paciente\n";
    }
    if (PacienteInfo.FechaNacimiento == null)
    {
        sms += "Ingresar Fecha de Nacimiento del Paciente\n";
    }
    if (PacienteInfo.Nombre == "")
    {
        sms += "Ingresar Nombre de Nacimiento del Paciente\n";
    }
    if (PacienteInfo.Nombre == "")
    {
        sms += "Ingresar Nombre de Nacimiento del Paciente\n";
    }
    if (sms != "")
    {
        throw new ArgumentException(sms);
    }
}
}
}
```

Medico

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using Acceso_Datos;
using System.Data.Linq;
```



```
namespace Logica_Nogocios
{
    public class Medico_Logica
    {
        public static odontologicoDataContext dc = new
odontologicoDataContext();
        public static List<Tbl_Medico> ObtenerMedicos()
        {
            var lista = dc.Tbl_Medico.Where(c => c.Estado == 'A');
            return lista.ToList();
        }

        public static List<Tbl_Medico> ObtenerMedico_XCedula(string ced)
        {
            var lista = dc.Tbl_Medico.Where(c => c.Estado == 'A' &
c.Med_Cedula.Equals(ced));
            return lista.ToList();
        }
        public static Tbl_Medico ObtenerMedico_IdInfo(int id)
        {
            var lista = dc.Tbl_Medico.FirstOrDefault(c => c.Estado == 'A' &
c.Id_Medico.Equals(id));
            return lista;
        }
        public static bool Existecedula(string cedula)
        {
            var existe = dc.Tbl_Medico.Any(a => a.Estado == 'A' &
a.Med_Cedula.Equals(cedula));
            return existe;
        }

        public static List<Tbl_Medico> ObtenerMedico_XApellido(string ape)
        {
            var lista = dc.Tbl_Medico.Where(c => c.Estado == 'A' &
c.Med_ApellidoPater.Equals(ape));
            return lista.ToList();
        }

        private static int Id_Codigo()
        {
            try
            {
                return dc.Tbl_Medico.Max(c => c.Id_Medico + 1);
            }
            catch (Exception)
            {
                return 1;
            }
        }
        public static void Save(Tbl_Medico MedicoInfo)
        {
            try
            {
                Validacion(MedicoInfo);
                dc.ExecuteCommand("Insert into Tbl_Medico(Id_Medico,
Med_PrimerNom, Med_SegundoNom, Med_ApellidoMat, Med_ApellidoPater, Med_Fono,"
+
                "Med_Celular, Med_HoraEntrada, Med_HoraSalida,
Med_TurnosMax, Med_Sexo, Med_Cedula, Estado)" +
```



```
"values({0},{1},{2},{3},{4},{5},{6},{7},{8},{9},{10},{11},{12})", new object[]
    {
        MedicoInfo.Id_Medico = Id_Codigo(),
        MedicoInfo.Med_PrimerNom,
        MedicoInfo.Med_SegundoNom,
        MedicoInfo.Med_ApellidoMat,
        MedicoInfo.Med_ApellidoPater,
        MedicoInfo.Med_Fono,
        MedicoInfo.Med_Celular,
        MedicoInfo.Med_HoraEntrada,
        MedicoInfo.Med_HoraSalida,
        MedicoInfo.Med_TurnosMax,
        MedicoInfo.Med_Sexo,
        MedicoInfo.Med_Cedula,
        MedicoInfo.Estado = 'A'
    }
    );
}
catch (Exception ex)
{
    throw new ArgumentException("Médico Guardado" + ex.Message);
}

}

public static void Modify(Tbl_Medico MedicoInfo)
{
    try
    {
        dc.ExecuteCommand("Update Tbl_Medico set Med_PrimerNom={0},
Med_SegundoNom={1}, Med_ApellidoMat={2}, Med_ApellidoPater={3}, Med_Fono={4},"
+
        "Med_Celular={5}, Med_HoraEntrada={6}, Med_HoraSalida={7},
Med_TurnosMax={8}, Med_Sexo={9}, Med_Cedula={10}, Estado={11} where
Id_Medico={12}", new object[]
        {
            MedicoInfo.Med_PrimerNom,
            MedicoInfo.Med_SegundoNom,
            MedicoInfo.Med_ApellidoMat,
            MedicoInfo.Med_ApellidoPater,
            MedicoInfo.Med_Fono,
            MedicoInfo.Med_Celular,
            MedicoInfo.Med_HoraEntrada,
            MedicoInfo.Med_HoraSalida,
            MedicoInfo.Med_TurnosMax,
            MedicoInfo.Med_Sexo,
            MedicoInfo.Med_Cedula,
            MedicoInfo.Estado = 'A',
            MedicoInfo.Id_Medico
        }
    );
}
catch (Exception ex)
{
    throw new ArgumentException("Médico Modificado" + ex.Message);
}
}
```



```
    }

    public static void Delete(Tbl_Medico MedicoInfo)
    {
        try
        {
            dc.ExecuteCommand("Update Tbl_Medico set Estado={0} where
Id_Medico={1}", new object[]
            {
                MedicoInfo.Estado = 'E',
                MedicoInfo.Id_Medico
            }
            );
        }
        catch (Exception ex)
        {
            throw new ArgumentException("Médico Eliminado" + ex.Message);
        }
    }

    private static void Validacion(Tbl_Medico MedicoInfo)
    {
        string sms = "";
        if (MedicoInfo.Med_ApellidoPater == "")
        {
            sms += "Ingresar el Apellido Paterno\n";
        }
        if (MedicoInfo.Med_Cedula == "")
        {
            sms += "Ingresar Cédula del Médico\n";
        }
        if (MedicoInfo.Med_Celular == "")
        {
            sms += "Ingresar Celular del Médico\n";
        }
        if (MedicoInfo.Med_Fono == "")
        {
            sms += "Ingresar Teléfono del Médico\n";
        }
        if (MedicoInfo.Med_HoraEntrada == null)
        {
            sms += "Ingresar Hora de entada del Médico\n";
        }
        if (MedicoInfo.Med_HoraSalida == null)
        {
            sms += "Ingresar Hora de salida del Médico\n";
        }
        if (MedicoInfo.Med_PrimerNom == "")
        {
            sms += "Ingresar Primer Nombre del Médico\n";
        }
        if (MedicoInfo.Med_TurnosMax == 0)
        {
            sms += "Ingresar Turnos Máximos del Médico\n";
        }
        if (sms != "")
        {
            throw new ArgumentException(sms);
        }
    }
}
```



```
    }  
  }  
}
```

Medico Especialista

```
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Text;  
using Acceso_Datos;  
using System.Data.Linq;  
  
namespace Logica_Nogocios  
{  
    public class MedicoEspecialista_Logica  
    {  
        public static odontologicoDataContext dc = new  
odontologicoDataContext();  
  
        public static List<Tbl_Medico_Especialista>  
ObtenerMedico_XEspecialidad(int Especialidad)  
        {  
            var lista = dc.Tbl_Medico_Especialista.Where(c =>  
c.Id_Especialidad.Equals(Especialidad));  
            return lista.ToList();  
        }  
        public static List<Tbl_Medico_Especialista> ObtenerMedico_LISTA()  
        {  
            var lista = dc.Tbl_Medico_Especialista.Where(c => c.Mes_Estado ==  
'A');  
            return lista.ToList();  
        }  
  
        public static Tbl_Medico_Especialista  
ObteneMedicoEspecialidad_IdInfo(int id)  
        {  
            var lista = dc.Tbl_Medico_Especialista.FirstOrDefault(c =>  
c.Mes_Estado == 'A' & c.Id_Medico_Espe.Equals(id));  
            return lista;  
        }  
  
        private static int Id_Codigo()  
        {  
            try  
            {  
                return dc.Tbl_Medico_Especialista.Max(c => c.Id_Medico_Espe +  
1);  
            }  
            catch (Exception)  
            {  

```



```
        return 1;
    }
}
public static void Save(Tbl_Medico_Especialista
MedicoEspecialidadInfo)
{
    try
    {
        Validaciones(MedicoEspecialidadInfo);
        dc.ExecuteCommand("Insert into
Tbl_Medico_Especialista(Id_Medico_Espe, Mes_Estado, Id_Medico,
Id_Especialidad) values({0},{1},{2},{3})", new object[]
{
            MedicoEspecialidadInfo.Id_Medico_Espe =
Id_Codigo(),
            MedicoEspecialidadInfo.Mes_Estado = 'A',
            MedicoEspecialidadInfo.Id_Medico,
            MedicoEspecialidadInfo.Id_Especialidad,
        });
    }
    catch (Exception ex)
    {
        throw new ArgumentException("Datos Guardado" + ex.Message);
    }
}

public static void Modify(Tbl_Medico_Especialista
MedicoEspecialidadInfo)
{
    try
    {
        dc.ExecuteCommand("Update Tbl_Medico_Especialista set
Mes_Estado={0}, Id_Medico={1}, Id_Especialidad={2} where Id_Medico_Espe={4}",
new object[]
{
            MedicoEspecialidadInfo.Mes_Estado = 'A',
            MedicoEspecialidadInfo.Id_Medico,
            MedicoEspecialidadInfo.Id_Especialidad,
            MedicoEspecialidadInfo.Id_Medico_Espe
        });
    }
    catch (Exception ex)
    {
        throw new ArgumentException("Datos Modificados" + ex.Message);
    }
}

public static void Delete(Tbl_Medico_Especialista
MedicoEspecialidadInfo)
{
    try
    {
```



```
        dc.ExecuteCommand("Update Tbl_Medico_Especialista set
Mes_Estado={0} where Id_Medico_Espe={1}", new object[]
        {
            MedicoEspecialidadInfo.Mes_Estado = 'E',
            MedicoEspecialidadInfo.Id_Medico_Espe
        }
    );
}
catch (Exception ex)
{
    throw new ArgumentException("Datos Eliminados" + ex.Message);
}

}

private static void Validaciones(Tbl_Medico_Especialista Info)
{
    string sms = "";
    if (Info.Id_Especialidad == 0)
    {
        sms += "Ingresae la Especialidad\n";
    }
    if (Info.Id_Medico == 0)
    {
        sms += "Ingresar el Médico\n";
    }
    if (sms != "")
    {
        throw new ArgumentException(sms);
    }
}
}
```

Horario

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using Acceso_Datos;
using System.Data.Linq;

namespace Logica_Nogocios
{
    public class Horario_Logica
    {
        public static odontologicoDataContext dc = new
odontologicoDataContext();

        public static List<Tbl_Horario> ObtenerHorrios()
        {
            var lista = dc.Tbl_Horario.Where(c => c.Hor_Estado == 'A');
            return lista.ToList();
        }
    }
}
```



```
public static Tbl_Horario ObtenerHorariosXIdInfo(int id)
{
    var lista = dc.Tbl_Horario.FirstOrDefault(c => c.Hor_Estado == 'A'
& c.Id_Horario.Equals(id));
    return lista;
}

private static int Id_Codigo()
{
    try
    {
        return dc.Tbl_Horario.Max(c => c.Id_Horario + 1);
    }
    catch (Exception)
    {
        return 1;
    }
}

public static void Save(Tbl_Horario HorarioInfo)
{
    try
    {
        dc.ExecuteCommand("Insert into Tbl_Horario(Id_Horario,
Hor_HR_Inicio, Hor_Hr_Fin, Hor_Lunes, Hor_Martes, Hor_Miercoles," +
"Hor_Jueves, Hor_Viernes, Hor_Estado)
values({0},{1},{2},{3},{4},{5},{6},{7},{8})", new object[]
{
            HorarioInfo.Id_Horario = Id_Codigo(),
            HorarioInfo.Hor_HR_Inicio,
            HorarioInfo.Hor_Hr_Fin,
            HorarioInfo.Hor_Lunes,
            HorarioInfo.Hor_Martes,
            HorarioInfo.Hor_Miercoles,
            HorarioInfo.Hor_Jueves,
            HorarioInfo.Hor_Viernes,
            HorarioInfo.Hor_Estado = 'A'
        });
    }
    catch (Exception ex)
    {
        throw new ArgumentException("Datos Guardado" + ex.Message);
    }
}

public static void Modify(Tbl_Horario HorarioInfo)
{
    try
    {
        dc.ExecuteCommand("Update Tbl_Horario set Hor_HR_Inicio={0},
Hor_Hr_Fin={1}, Hor_Lunes={2}, Hor_Martes={3}, Hor_Miercoles={4}," +
"Hor_Jueves={5}, Hor_Viernes={6}, Hor_Estado={7} where
Id_Horario={8}", new object[]
{
            HorarioInfo.Hor_HR_Inicio,
            HorarioInfo.Hor_Hr_Fin,
            HorarioInfo.Hor_Lunes,
```



```
        HorarioInfo.Hor_Martes,
        HorarioInfo.Hor_Miercoles,
        HorarioInfo.Hor_Jueves,
        HorarioInfo.Hor_Viernes,
        HorarioInfo.Hor_Estado,
        HorarioInfo.Id_Horario
    }
    );
}
}
catch (Exception ex)
{
    throw new ArgumentException("Datos Modificados" + ex.Message);
}
}
public static void Delete(Tbl_Horario HorarioInfo)
{
    try
    {
        dc.ExecuteCommand("Update Tbl_Horario set Hor_Estado={0} where
Id_Horario={1}", new object[]
        {
            HorarioInfo.Hor_Estado = 'E',
            HorarioInfo.Id_Horario
        }
    );
    }
    catch (Exception ex)
    {
        throw new ArgumentException("Datos Eliminados" + ex.Message);
    }
}
public static void HorarioUtilizado(Tbl_Horario HorarioInfo)
{
    try
    {
        dc.ExecuteCommand("Update Tbl_Horario set Hor_Estado={0} where
Id_Horario={1}", new object[]
        {
            HorarioInfo.Hor_Estado = '0',
            HorarioInfo.Id_Horario
        }
    );
    }
    catch (Exception ex)
    {
        throw new ArgumentException("Datos Eliminados" + ex.Message);
    }
}
}
}
```

Responsible

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
```



```
using Acceso_Datos;
using System.Data.Linq;

namespace Logica_Nogocios
{
    public class Responsable_Logica
    {
        public static odontologicoDataContext dc = new
odontologicoDataContext();
        public static List<Tbl_responsable> ObtenerResponsable()
        {
            var lista = dc.Tbl_responsable.Where(c => c.Res_Estado == 'A');
            return lista.ToList();
        }

        public static List<Tbl_responsable>
ObtenerResponsable_XApellido(string Responsable)
        {
            var lista = dc.Tbl_responsable.Where(c => c.Res_Estado == 'A' &
c.Id_Responsable.Equals(Responsable));
            return lista.ToList();
        }

        public static List<Tbl_responsable> ObtenerResponsable_XArea(int Area)
        {
            var lista = dc.Tbl_responsable.Where(c => c.Res_Estado == 'A' &
c.Id_Area.Equals(Area));
            return lista.ToList();
        }

        public static List<Tbl_responsable> ObtenerResponsable_XCargo(int
Cargo)
        {
            var lista = dc.Tbl_responsable.Where(c => c.Res_Estado == 'A' &
c.Id_Cargo.Equals(Cargo));
            return lista.ToList();
        }

        public static Tbl_responsable ObtenerTurno_IdInfo(int id)
        {
            var lista = dc.Tbl_responsable.FirstOrDefault(c => c.Res_Estado ==
'A' & c.Id_Responsable.Equals(id));
            return lista;
        }

        public static Tbl_responsable ObtenerTurno_cedulaInfo(string ced)
        {
            var lista = dc.Tbl_responsable.FirstOrDefault(c => c.Res_Estado ==
'A' & c.Res_Cedula.EndsWith(ced));
            return lista;
        }

        private static int Id_Codigo()
        {
            try
            {
                return dc.Tbl_responsable.Max(c => c.Id_Responsable + 1);
            }
            catch (Exception)
            {
                return 1;
            }
        }
    }
}
```



```
    }
}
public static void Save(Tbl_responsable Responsablefo)
{
    try
    {
        dc.ExecuteCommand("Insert into Tbl_responsable(Id_Responsable,
Res_Nombre, Res_Apellido, Res_Titulo, Res_Fono, Res_Celular," +
"Res_Estado, Id_Area, Id_Cargo, Res_Cedula)" +
"values({0},{1},{2},{3},{4},{5},{6},{7},{8},{9})", new
object[]
        {
            Responsablefo.Id_Responsable = Id_Codigo(),
            Responsablefo.Res_Nombre,
            Responsablefo.Res_Apellido,
            Responsablefo.Res_Titulo,
            Responsablefo.Res_Fono,
            Responsablefo.Res_Celular,
            Responsablefo.Res_Estado = 'A',
            Responsablefo.Id_Area,
            Responsablefo.Id_Cargo,
            Responsablefo.Res_Cedula
        }
    );
    }
    catch (Exception ex)
    {
        throw new ArgumentException("Responsable Guardado" +
ex.Message);
    }
}

public static void Modify(Tbl_responsable Responsablefo)
{
    try
    {
        dc.ExecuteCommand("Update Tbl_responsable set Res_Nombre={0},
Res_Apellido={1}, Res_Titulo={2}, Res_Fono={3}, Res_Celular={4}," +
"Res_Estado={5}, Id_Area={6}, Id_Cargo={7},Res_Cedula={8}
where Id_Responsable={9}", new object[]
        {
            Responsablefo.Res_Nombre,
            Responsablefo.Res_Apellido,
            Responsablefo.Res_Titulo,
            Responsablefo.Res_Fono,
            Responsablefo.Res_Celular,
            Responsablefo.Res_Estado = 'A',
            Responsablefo.Id_Area,
            Responsablefo.Id_Cargo,
            Responsablefo.Res_Cedula,
            Responsablefo.Id_Responsable
        }
    );
    }
    catch (Exception ex)
    {

```



```
        throw new ArgumentException("Responsable Modificado" +
ex.Message);
    }

    }

    public static void Delete(Tbl_responsable Responsablefo)
    {
        try
        {
            dc.ExecuteCommand("Update Tbl_responsable set Res_Estado={0}
where Id_Responsable={1}", new object[]
            {
                Responsablefo.Res_Estado = 'E',
                Responsablefo.Id_Responsable
            }
        );
        }
        catch (Exception ex)
        {
            throw new ArgumentException("Responsable Eliminado" +
ex.Message);
        }
    }
}
}
```

Script de la base de datos.

```
USE [Odontologico_BD]
GO
/***** Object: Table [dbo].[tbl_GrupoDiente] Script Date: 04/11/2014 16:53:32 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[tbl_GrupoDiente](
    [Id_Grupo] [int] NOT NULL,
    [Gd_Nombre] [text] NULL,
    [Gd_Numeracion] [varchar](100) NULL,
    [Estado] [char](1) NULL,
    CONSTRAINT [PK_tbl_GrupoDiente] PRIMARY KEY CLUSTERED
(
    [Id_Grupo] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
```

```

) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO
SET ANSI_PADDING OFF
GO
INSERT [dbo].[tbl_GrupoDiente] ([Id_Grupo], [Gd_Nombre], [Gd_Numeracion], [Estado]) VALUES
(1, N'SUP-IZQUIERDO', N'18 - 11', N'A')
INSERT [dbo].[tbl_GrupoDiente] ([Id_Grupo], [Gd_Nombre], [Gd_Numeracion], [Estado]) VALUES
(2, N'SUP-DERECHO', N'28 - 21', N'A')
INSERT [dbo].[tbl_GrupoDiente] ([Id_Grupo], [Gd_Nombre], [Gd_Numeracion], [Estado]) VALUES
(3, N'INF-IZQUIERDO', N'48 - 41', N'A')
INSERT [dbo].[tbl_GrupoDiente] ([Id_Grupo], [Gd_Nombre], [Gd_Numeracion], [Estado]) VALUES
(4, N'INF-DERECHO', N'38 - 31', N'A')
/***** Object: Table [dbo].[Tbl_Especialidad]   Script Date: 04/11/2014 16:53:32 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[Tbl_Especialidad](
    [Id_Especialidad] [int] NOT NULL,
    [Descripcion] [varchar](70) NULL,
    [Estado] [char](1) NULL,
PRIMARY KEY CLUSTERED
(
    [Id_Especialidad] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
SET ANSI_PADDING OFF
GO
INSERT [dbo].[Tbl_Especialidad] ([Id_Especialidad], [Descripcion], [Estado]) VALUES (1,
N'ODONTOLOGÍA GENERAL', N'A')
INSERT [dbo].[Tbl_Especialidad] ([Id_Especialidad], [Descripcion], [Estado]) VALUES (2,
N'PERIODONCIA', N'A')
INSERT [dbo].[Tbl_Especialidad] ([Id_Especialidad], [Descripcion], [Estado]) VALUES (3,
N'ENDODONCIA', N'A')
INSERT [dbo].[Tbl_Especialidad] ([Id_Especialidad], [Descripcion], [Estado]) VALUES (4,
N'ORTODONCIA', N'A')
INSERT [dbo].[Tbl_Especialidad] ([Id_Especialidad], [Descripcion], [Estado]) VALUES (5,
N'PROTESIS DENTAL', N'A')
INSERT [dbo].[Tbl_Especialidad] ([Id_Especialidad], [Descripcion], [Estado]) VALUES (6,
N'IMPLANTOLOGIA', N'A')
INSERT [dbo].[Tbl_Especialidad] ([Id_Especialidad], [Descripcion], [Estado]) VALUES (7,
N'ODONTOLOGÍA PEDIÁTRICA', N'A')

GESTION DE HISTORIAS CLÍNICAS Y CONTROL DE CITAS MÉDICAS PARA ADENTAL CLÍNICA
ODONTOLÓGICA DE LA CIUDAD DE QUITO

```

```

INSERT [dbo].[Tbl_Especialidad] ([Id_Especialidad], [Descripcion], [Estado]) VALUES (8,
N'ODONTOGERIATRIA', N'A')
INSERT [dbo].[Tbl_Especialidad] ([Id_Especialidad], [Descripcion], [Estado]) VALUES (9,
N'HIGIENE DENTAL', N'A')
INSERT [dbo].[Tbl_Especialidad] ([Id_Especialidad], [Descripcion], [Estado]) VALUES (10,
N'ESTETICA DENTAL', N'A')
INSERT [dbo].[Tbl_Especialidad] ([Id_Especialidad], [Descripcion], [Estado]) VALUES (11,
N'BLANQUEAMIENTO DENTAL', N'A')
INSERT [dbo].[Tbl_Especialidad] ([Id_Especialidad], [Descripcion], [Estado]) VALUES (12,
N'PROSTODONCIA', N'A')
INSERT [dbo].[Tbl_Especialidad] ([Id_Especialidad], [Descripcion], [Estado]) VALUES (13,
N'ODONTOLOGIA PREVENTIVA', N'A')
INSERT [dbo].[Tbl_Especialidad] ([Id_Especialidad], [Descripcion], [Estado]) VALUES (14,
N'ODONTOLOGIA RESTAURADORA', N'A')
INSERT [dbo].[Tbl_Especialidad] ([Id_Especialidad], [Descripcion], [Estado]) VALUES (15,
N'CIRUGIA BUCOMAXILOFACIAL', N'A')
INSERT [dbo].[Tbl_Especialidad] ([Id_Especialidad], [Descripcion], [Estado]) VALUES (16,
N'RADIOLOGIA DENTAL', N'A')
INSERT [dbo].[Tbl_Especialidad] ([Id_Especialidad], [Descripcion], [Estado]) VALUES (17,
N'REHABILITACION ORAL', N'A')
INSERT [dbo].[Tbl_Especialidad] ([Id_Especialidad], [Descripcion], [Estado]) VALUES (18,
N'REHABILITACION OCLUSAL', N'E')
/***** Object: Table [dbo].[Tbl_Cargo]   Script Date: 04/11/2014 16:53:32 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[Tbl_Cargo](
    [Id_Cargo] [int] NOT NULL,
    [Descripcion] [varchar](70) NULL,
    [Estado] [char](1) NULL,
    PRIMARY KEY CLUSTERED
(
    [Id_Cargo] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
SET ANSI_PADDING OFF
GO
INSERT [dbo].[Tbl_Cargo] ([Id_Cargo], [Descripcion], [Estado]) VALUES (1, N'DOCTOR', N'A')
INSERT [dbo].[Tbl_Cargo] ([Id_Cargo], [Descripcion], [Estado]) VALUES (2, N'CIRUJANO', N'A')
/***** Object: Table [dbo].[Tbl_Area]   Script Date: 04/11/2014 16:53:32 *****/
SET ANSI_NULLS ON

```

```

GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[Tbl_Area](
    [Id_Area] [int] NOT NULL,
    [Descripcion] [varchar](70) NULL,
    [Estado] [char](1) NULL,
PRIMARY KEY CLUSTERED
(
    [Id_Area] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
SET ANSI_PADDING OFF
GO
INSERT [dbo].[Tbl_Area] ([Id_Area], [Descripcion], [Estado]) VALUES (1, N'GENERAL', N'A)
INSERT [dbo].[Tbl_Area] ([Id_Area], [Descripcion], [Estado]) VALUES (2, N'ODONTOLÓGICO',
N'A)
INSERT [dbo].[Tbl_Area] ([Id_Area], [Descripcion], [Estado]) VALUES (3, N'PAGOS', N'E)
/***** Object: Table [dbo].[Tbl_Medico]   Script Date: 04/11/2014 16:53:32 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[Tbl_Medico](
    [Id_Medico] [int] NOT NULL,
    [Med_PrimerNom] [varchar](60) NOT NULL,
    [Med_SegundoNom] [varchar](60) NOT NULL,
    [Med_ApellidoMat] [varchar](60) NOT NULL,
    [Med_ApellidoPater] [varchar](70) NOT NULL,
    [Med_Fono] [varchar](15) NULL,
    [Med_Celular] [varchar](15) NULL,
    [Med_HoraEntrada] [datetime] NULL,
    [Med_HoraSalida] [datetime] NULL,
    [Med_TurnosMax] [int] NULL,
    [Med_Sexo] [char](1) NULL,
    [Med_Cedula] [varchar](12) NOT NULL,
    [Estado] [char](1) NULL,
PRIMARY KEY CLUSTERED
(
    [Id_Medico] ASC

```

```

)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
SET ANSI_PADDING OFF
GO
INSERT [dbo].[Tbl_Medico] ([Id_Medico], [Med_PrimerNom], [Med_SegundoNom],
[Med_ApellidoMat], [Med_ApellidoPater], [Med_Fono], [Med_Celular], [Med_HoraEntrada],
[Med_HoraSalida], [Med_TurnosMax], [Med_Sexo], [Med_Cedula], [Estado]) VALUES (1,
N'JORGE', N'ALVERTO', N'MANTILLA', N'MANTILLA', N'022545678', N'095443211',
CAST(0x0000A2FB008C1360 AS DateTime), CAST(0x0000A2FB010FE960 AS DateTime), 6,
N'M', N'1734562234', N'A')
INSERT [dbo].[Tbl_Medico] ([Id_Medico], [Med_PrimerNom], [Med_SegundoNom],
[Med_ApellidoMat], [Med_ApellidoPater], [Med_Fono], [Med_Celular], [Med_HoraEntrada],
[Med_HoraSalida], [Med_TurnosMax], [Med_Sexo], [Med_Cedula], [Estado]) VALUES (2,
N'PEDRO', N'MIGUEL', N'QUIÑONEZ', N'CORTEZ', N'023454676', N'092764533',
CAST(0x0000A2FB008C1360 AS DateTime), CAST(0x0000A2FB010FE960 AS DateTime), 3,
N'M', N'1718946401', N'A')
/***** Object: Table [dbo].[Tbl_Horario] Script Date: 04/11/2014 16:53:32 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[Tbl_Horario](
    [Id_Horario] [int] NOT NULL,
    [Hor_HR_Inicio] [datetime] NULL,
    [Hor_Hr_Fin] [datetime] NULL,
    [Hor_Lunes] [tinyint] NULL,
    [Hor_Martes] [tinyint] NULL,
    [Hor_Miercoles] [tinyint] NULL,
    [Hor_Jueves] [tinyint] NULL,
    [Hor_Viernes] [tinyint] NULL,
    [Hor_Estado] [char](1) NULL,
PRIMARY KEY CLUSTERED
(
    [Id_Horario] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
SET ANSI_PADDING OFF
GO
INSERT [dbo].[Tbl_Horario] ([Id_Horario], [Hor_HR_Inicio], [Hor_Hr_Fin], [Hor_Lunes],
[Hor_Martes], [Hor_Miercoles], [Hor_Jueves], [Hor_Viernes], [Hor_Estado]) VALUES (1,

```

GESTION DE HISTORIAS CLÍNICAS Y CONTROL DE CITAS MÉDICAS PARA ADENTAL CLÍNICA ODONTOLÓGICA DE LA CIUDAD DE QUITO

```
CAST(0x0000A2FF0083D600 AS DateTime), CAST(0x0000A2FF00A4CB80 AS DateTime), 0, 0, 0,
0, 1, N'O')
```

```
INSERT [dbo].[Tbl_Horario] ([Id_Horario], [Hor_HR_Inicio], [Hor_Hr_Fin], [Hor_Lunes],
[Hor_Martes], [Hor_Miercoles], [Hor_Jueves], [Hor_Viernes], [Hor_Estado]) VALUES (2,
CAST(0x0000A3070083D600 AS DateTime), CAST(0x0000A30700C5C100 AS DateTime), 0, 0, 0,
0, 0, N'O')
```

```
/****** Object: Table [dbo].[Tbl_Referencias] Script Date: 04/11/2014 16:53:32 *****/
```

```
SET ANSI_NULLS ON
```

```
GO
```

```
SET QUOTED_IDENTIFIER ON
```

```
GO
```

```
SET ANSI_PADDING ON
```

```
GO
```

```
CREATE TABLE [dbo].[Tbl_Referencias](
    [Id_Referencia] [int] NOT NULL,
    [Ref_Nombre] [varchar](50) NULL,
    [Estado] [char](1) NULL,
    CONSTRAINT [PK_Tbl_Referencias] PRIMARY KEY CLUSTERED
```

```
(
```

```
    [Id_Referencia] ASC
```

```
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
```

```
) ON [PRIMARY]
```

```
GO
```

```
SET ANSI_PADDING OFF
```

```
GO
```

```
INSERT [dbo].[Tbl_Referencias] ([Id_Referencia], [Ref_Nombre], [Estado]) VALUES (1,
N'Ninguna', N'A')
```

```
INSERT [dbo].[Tbl_Referencias] ([Id_Referencia], [Ref_Nombre], [Estado]) VALUES (2,
N'Temperatura', N'A')
```

```
INSERT [dbo].[Tbl_Referencias] ([Id_Referencia], [Ref_Nombre], [Estado]) VALUES (3,
N'Hepatitis', N'A')
```

```
/****** Object: Table [dbo].[Tbl_Paciente] Script Date: 04/11/2014 16:53:32 *****/
```

```
SET ANSI_NULLS ON
```

```
GO
```

```
SET QUOTED_IDENTIFIER ON
```

```
GO
```

```
SET ANSI_PADDING ON
```

```
GO
```

```
CREATE TABLE [dbo].[Tbl_Paciente](
    [Id_Paciente] [int] NOT NULL,
    [Cedula] [varchar](12) NOT NULL,
    [Apellido] [varchar](140) NOT NULL,
    [Nombre] [varchar](140) NOT NULL,
    [FechaNacimiento] [datetime] NULL,
    [Direccion] [text] NULL,
```

```

[Fono] [varchar](15) NULL,
[Celular] [varchar](15) NULL,
[Ocupacion] [varchar](100) NULL,
[Sexo] [char](1) NULL,
[Estado] [char](1) NULL,
PRIMARY KEY CLUSTERED
(
    [Id_Paciente] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO
SET ANSI_PADDING OFF
GO
INSERT [dbo].[Tbl_Paciente] ([Id_Paciente], [Cedula], [Apellido], [Nombre], [FechaNacimiento],
[Direccion], [Fono], [Celular], [Ocupacion], [Sexo], [Estado]) VALUES (1, N'1312597733',
N'RAMONA', N'ELIANA', CAST(0x0000A2FB00000000 AS DateTime), N'CONOCOTO',
N'098765432', N'098765432', N'SECRETARIA 32', N'F', N'A)
/***** Object: Table [dbo].[Tbl_TratamientoInfo] Script Date: 04/11/2014 16:53:32 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[Tbl_TratamientoInfo](
    [Id_Tratamiento] [int] NOT NULL,
    [Nombre] [varchar](100) NULL,
    [Estado] [char](1) NULL,
    [Descripcion] [text] NULL,
CONSTRAINT [PK_Tbl_TratamientoInfo] PRIMARY KEY CLUSTERED
(
    [Id_Tratamiento] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO
SET ANSI_PADDING OFF
GO
INSERT [dbo].[Tbl_TratamientoInfo] ([Id_Tratamiento], [Nombre], [Estado], [Descripcion])
VALUES (1, N'ODONTOLOGÍA GENERAL', N'A', NULL)
INSERT [dbo].[Tbl_TratamientoInfo] ([Id_Tratamiento], [Nombre], [Estado], [Descripcion])
VALUES (2, N'PERIODONCIA', N'A', NULL)
INSERT [dbo].[Tbl_TratamientoInfo] ([Id_Tratamiento], [Nombre], [Estado], [Descripcion])
VALUES (3, N'ENDODONCIA', N'A', NULL)

```



```
INSERT [dbo].[Tbl_TratamientoInfo] ([Id_Tratamiento], [Nombre], [Estado], [Descripcion])
VALUES (4, N'ORTODONCIA', N'A', NULL)
INSERT [dbo].[Tbl_TratamientoInfo] ([Id_Tratamiento], [Nombre], [Estado], [Descripcion])
VALUES (5, N'PROTESIS DENTAL', N'A', NULL)
INSERT [dbo].[Tbl_TratamientoInfo] ([Id_Tratamiento], [Nombre], [Estado], [Descripcion])
VALUES (6, N'IMPLANTOLOGIA', N'A', NULL)
INSERT [dbo].[Tbl_TratamientoInfo] ([Id_Tratamiento], [Nombre], [Estado], [Descripcion])
VALUES (7, N'ODONTOPEDIATRIA', N'A', NULL)
INSERT [dbo].[Tbl_TratamientoInfo] ([Id_Tratamiento], [Nombre], [Estado], [Descripcion])
VALUES (8, N'ODONTOGERIATRIA', N'A', NULL)
INSERT [dbo].[Tbl_TratamientoInfo] ([Id_Tratamiento], [Nombre], [Estado], [Descripcion])
VALUES (9, N'HIGIENE DENTAL', N'A', NULL)
INSERT [dbo].[Tbl_TratamientoInfo] ([Id_Tratamiento], [Nombre], [Estado], [Descripcion])
VALUES (10, N'ESTETICA DENTAL', N'A', NULL)
INSERT [dbo].[Tbl_TratamientoInfo] ([Id_Tratamiento], [Nombre], [Estado], [Descripcion])
VALUES (11, N'BLANQUEAMIENTO DENTAL', N'A', NULL)
INSERT [dbo].[Tbl_TratamientoInfo] ([Id_Tratamiento], [Nombre], [Estado], [Descripcion])
VALUES (12, N'PROSTODONCIA', N'A', NULL)
INSERT [dbo].[Tbl_TratamientoInfo] ([Id_Tratamiento], [Nombre], [Estado], [Descripcion])
VALUES (13, N'ODONTOLOGIA PREVENTIVA', N'A', NULL)
INSERT [dbo].[Tbl_TratamientoInfo] ([Id_Tratamiento], [Nombre], [Estado], [Descripcion])
VALUES (14, N'ODONTOLOGIA RESTAURADORA', N'A', NULL)
INSERT [dbo].[Tbl_TratamientoInfo] ([Id_Tratamiento], [Nombre], [Estado], [Descripcion])
VALUES (15, N'CIRUGIA BUCOMAXILOFACIAL', N'A', NULL)
INSERT [dbo].[Tbl_TratamientoInfo] ([Id_Tratamiento], [Nombre], [Estado], [Descripcion])
VALUES (16, N'RADIOLOGIA DENTAL', N'A', NULL)
INSERT [dbo].[Tbl_TratamientoInfo] ([Id_Tratamiento], [Nombre], [Estado], [Descripcion])
VALUES (17, N'REHABILITACION ORAL', N'A', NULL)
INSERT [dbo].[Tbl_TratamientoInfo] ([Id_Tratamiento], [Nombre], [Estado], [Descripcion])
VALUES (18, N'REHABILITACION OCLUSAL', N'A', NULL)
/***** Object: Table [dbo].[Tbl_TipoUsuarios] Script Date: 04/11/2014 16:53:32 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[Tbl_TipoUsuarios](
    [Id_TipoUsuarios] [int] NOT NULL,
    [Descripcion] [varchar](70) NULL,
    [Estado] [char](1) NULL,
    PRIMARY KEY CLUSTERED
(
    [Id_TipoUsuarios] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
```

```

) ON [PRIMARY]
GO
SET ANSI_PADDING OFF
GO
INSERT [dbo].[Tbl_TipoUsuarios] ([Id_TipoUsuarios], [Descripcion], [Estado]) VALUES (1,
N'ADMINISTRADOR', N'A')
INSERT [dbo].[Tbl_TipoUsuarios] ([Id_TipoUsuarios], [Descripcion], [Estado]) VALUES (2,
N'MEDICO', N'A')
INSERT [dbo].[Tbl_TipoUsuarios] ([Id_TipoUsuarios], [Descripcion], [Estado]) VALUES (3,
N'PACIENTE', N'A')
INSERT [dbo].[Tbl_TipoUsuarios] ([Id_TipoUsuarios], [Descripcion], [Estado]) VALUES (4,
N'RESPONSABLE', N'A')
INSERT [dbo].[Tbl_TipoUsuarios] ([Id_TipoUsuarios], [Descripcion], [Estado]) VALUES (5,
N'Prueba', N'E')
/***** Object: Table [dbo].[tbl_temp]  Script Date: 04/11/2014 16:53:32 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[tbl_temp](
    [id] [int] NOT NULL,
    [Nombre] [varchar](50) NULL,
    [Pieza] [varchar](50) NULL,
    [estado] [int] NULL,
    CONSTRAINT [PK_tbl_temp] PRIMARY KEY CLUSTERED
(
    [id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
SET ANSI_PADDING OFF
GO
INSERT [dbo].[tbl_temp] ([id], [Nombre], [Pieza], [estado]) VALUES (1, N'LunaCentral',
N'Pieza_18', 2)
INSERT [dbo].[tbl_temp] ([id], [Nombre], [Pieza], [estado]) VALUES (2, N'EstadoLunaCentral',
N'Pieza_17', 2)
INSERT [dbo].[tbl_temp] ([id], [Nombre], [Pieza], [estado]) VALUES (3, N'LunaCentral',
N'Pieza_21', 2)
INSERT [dbo].[tbl_temp] ([id], [Nombre], [Pieza], [estado]) VALUES (4, N'LunaCentral',
N'Pieza_22', 2)
INSERT [dbo].[tbl_temp] ([id], [Nombre], [Pieza], [estado]) VALUES (5, N'EstadoLunaCentral',
N'Pieza_62', 2)
/***** Object: Table [dbo].[Tbl_responsable]  Script Date: 04/11/2014 16:53:32 *****/

```

```
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[Tbl_responsable](
    [Id_Responsable] [int] NOT NULL,
    [Res_Nombre] [varchar](70) NOT NULL,
    [Res_Apellido] [varchar](60) NOT NULL,
    [Res_Titulo] [varchar](100) NULL,
    [Res_Fono] [varchar](15) NULL,
    [Res_Celular] [varchar](15) NULL,
    [Res_Estado] [char](1) NULL,
    [Id_Area] [int] NOT NULL,
    [Id_Cargo] [int] NOT NULL,
    [Res_Cedula] [varchar](12) NULL,
PRIMARY KEY CLUSTERED
(
    [Id_Responsable] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
SET ANSI_PADDING OFF
GO
INSERT [dbo].[Tbl_responsable] ([Id_Responsable], [Res_Nombre], [Res_Apellido], [Res_Titulo],
[Res_Fono], [Res_Celular], [Res_Estado], [Id_Area], [Id_Cargo], [Res_Cedula]) VALUES (1,
N'JAUN FERNADO', N'NARVAEZ LOYOLA', N'MEDICO GENERAL', N'022434665',
N'098233400', N'A', 1, 1, N'1715456412')
/***** Object: Table [dbo].[Tbl_Medico_Especialista]  Script Date: 04/11/2014 16:53:32 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[Tbl_Medico_Especialista](
    [Id_Medico_Espe] [int] NOT NULL,
    [Mes_Estado] [char](1) NULL,
    [Id_Medico] [int] NOT NULL,
    [Id_Especialidad] [int] NOT NULL,
PRIMARY KEY CLUSTERED
(
    [Id_Medico_Espe] ASC
```



```
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
SET ANSI_PADDING OFF
GO
INSERT [dbo].[Tbl_Medico_Especialista] ([Id_Medico_Espe], [Mes_Estado], [Id_Medico],
[Id_Especialidad]) VALUES (1, N'A', 2, 1)
INSERT [dbo].[Tbl_Medico_Especialista] ([Id_Medico_Espe], [Mes_Estado], [Id_Medico],
[Id_Especialidad]) VALUES (2, N'A', 1, 10)
/***** Object: Table [dbo].[Tbl_HistoriaClinica] Script Date: 04/11/2014 16:53:32 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[Tbl_HistoriaClinica](
    [Id_Historial] [int] NOT NULL,
    [His_Fecha] [datetime] NULL,
    [His_Hora] [varchar](10) NULL,
    [Id_Turno] [int] NULL,
    [His_Diagnostico] [varchar](100) NULL,
    [His_Pronostico] [varchar](100) NULL,
    [Id_Paciente] [int] NULL,
    [Estado] [char](1) NULL,
    [His_Foto] [text] NULL,
    CONSTRAINT [PK_Tbl_HistoriaClinica] PRIMARY KEY CLUSTERED
(
    [Id_Historial] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO
SET ANSI_PADDING OFF
GO
INSERT [dbo].[Tbl_HistoriaClinica] ([Id_Historial], [His_Fecha], [His_Hora], [Id_Turno],
[His_Diagnostico], [His_Pronostico], [Id_Paciente], [Estado], [His_Foto]) VALUES (2,
CAST(0x0000A30100000000 AS DateTime), N'13:54:19', 0, N'prueba', N'prueba', 1, N'A',
N'C:\Users\adrian\Desktop\LOGO.jpg')
INSERT [dbo].[Tbl_HistoriaClinica] ([Id_Historial], [His_Fecha], [His_Hora], [Id_Turno],
[His_Diagnostico], [His_Pronostico], [Id_Paciente], [Estado], [His_Foto]) VALUES (4,
CAST(0x0000A30200000000 AS DateTime), N'16:37:53', 0, N'alksdjflaksjfd', N'asdjflkajsdfilkjas', 1,
N'A', N'C:\Users\adrian\Desktop\LOGO.jpg')
/***** Object: Table [dbo].[Tbl_Usuarios] Script Date: 04/11/2014 16:53:32 *****/
SET ANSI_NULLS ON
```

```

GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[Tbl_Usuarios](
    [Id_Usuarios] [int] NOT NULL,
    [Nom_Usuario] [varchar](40) NOT NULL,
    [Usu_Clave] [varchar](20) NOT NULL,
    [Usu_Estado] [char](1) NULL,
    [Usu_Fecha_Ingreso] [datetime] NULL,
    [Id_TipoUsuarios] [int] NOT NULL,
    [Id_Responsable] [int] NOT NULL,
    CONSTRAINT [PK__Tbl_Usua__AEF904297F60ED59] PRIMARY KEY CLUSTERED
(
    [Id_Usuarios] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
SET ANSI_PADDING OFF
GO
INSERT [dbo].[Tbl_Usuarios] ([Id_Usuarios], [Nom_Usuario], [Usu_Clave], [Usu_Estado],
[Usu_Fecha_Ingreso], [Id_TipoUsuarios], [Id_Responsable]) VALUES (1, N'JFERNANDO',
N'12341234', N'A', CAST(0x0000A2F500000000 AS DateTime), 3, 1)
/***** Object: Table [dbo].[Tbl_Turno]  Script Date: 04/11/2014 16:53:32 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[Tbl_Turno](
    [Id_Turno] [int] NOT NULL,
    [Tur_Fecha] [datetime] NULL,
    [Tur_Cita] [varchar](150) NULL,
    [Tur_Valor] [decimal](18, 2) NULL,
    [Tur_Numero] [int] NULL,
    [Id_Medico_Espe] [int] NOT NULL,
    [Id_Responsable] [int] NOT NULL,
    [Id_Paciente] [int] NOT NULL,
    [Estado] [char](1) NULL,
    [Id_Horarios] [int] NULL,
    CONSTRAINT [PK__Tbl_Turn__5CF9003F25869641] PRIMARY KEY CLUSTERED
(
    [Id_Turno] ASC

```



```
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
SET ANSI_PADDING OFF
GO
INSERT [dbo].[Tbl_Turno] ([Id_Turno], [Tur_Fecha], [Tur_Cita], [Tur_Valor], [Tur_Numero],
[Id_Medico_Espe], [Id_Responsable], [Id_Paciente], [Estado], [Id_Horarios]) VALUES (1,
CAST(0x0000A2FF00000000 AS DateTime), N'rrrr', CAST(44.00 AS Decimal(18, 2)), 1, 1, 1, 1, N'A',
1)
INSERT [dbo].[Tbl_Turno] ([Id_Turno], [Tur_Fecha], [Tur_Cita], [Tur_Valor], [Tur_Numero],
[Id_Medico_Espe], [Id_Responsable], [Id_Paciente], [Estado], [Id_Horarios]) VALUES (2,
CAST(0x0000A30700000000 AS DateTime), N'sadfasdf', CAST(0.00 AS Decimal(18, 2)), 2, 1, 1, 1,
N'A', 2)
/***** Object: Table [dbo].[Tbl_Tratamientos] Script Date: 04/11/2014 16:53:32 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[Tbl_Tratamientos](
    [Id_HistorialTratamiento] [int] NOT NULL,
    [Id_Historia] [int] NULL,
    [Tra_Fecha] [datetime] NULL,
    [Id_Tratamiento] [int] NULL,
    [Tra_Observacion] [text] NULL,
    [Estado] [char](1) NULL,
    [Id_Referencia] [int] NULL,
    [FormasPago] [varchar](50) NULL,
    [id_Turno] [int] NOT NULL,
    [Id_Grupo] [int] NULL,
    CONSTRAINT [PK_Tbl_Tratamientos] PRIMARY KEY CLUSTERED
(
    [Id_HistorialTratamiento] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO
SET ANSI_PADDING OFF
GO
INSERT [dbo].[Tbl_Tratamientos] ([Id_HistorialTratamiento], [Id_Historia], [Tra_Fecha],
[Id_Tratamiento], [Tra_Observacion], [Estado], [Id_Referencia], [FormasPago], [id_Turno],
[Id_Grupo]) VALUES (1, 2, CAST(0x0000A30100E90F71 AS DateTime), 1, N'', N'A', 1, N'Efectivo',
1, 1)
```



```
INSERT [dbo].[Tbl_Tratamientos] ([Id_HistorialTratamiento], [Id_Historia], [Tra_Fecha],  
[Id_Tratamiento], [Tra_Observacion], [Estado], [Id_Referencia], [FormasPago], [id_Turno],  
[Id_Grupo]) VALUES (2, 2, CAST(0x0000A30200EF1B44 AS DateTime), 3, N'[Pieza_18  
LunaCentral 2]: E
```

```
ndodoncia', N'A', 1, N'Efectivo', 1, 1)
```

```
INSERT [dbo].[Tbl_Tratamientos] ([Id_HistorialTratamiento], [Id_Historia], [Tra_Fecha],  
[Id_Tratamiento], [Tra_Observacion], [Estado], [Id_Referencia], [FormasPago], [id_Turno],  
[Id_Grupo]) VALUES (3, 4, CAST(0x0000A3020112524D AS DateTime), 2, N'[Pieza_24  
LunaCentral 2]:
```

```
[Pieza_63 EstadoLunaCentral 2]:
```

```
[Pieza_64 EstadoLunaCentral 2]:
```

```
[Pieza_18 LunaCentral 2]:
```

```
[Pieza_17 EstadoLunaSuperior1 2]:
```

```
[Pieza_21 LunaCentral 2]:
```

```
[Pieza_22 LunaSuperior1 2]:
```

```
', N'A', 1, N'Efectivo', 1, 2)
```

```
/****** Object: StoredProcedure [dbo].[Sp_Turnos] Script Date: 04/11/2014 16:53:45 *****/
```

```
SET ANSI_NULLS ON
```

```
GO
```

```
SET QUOTED_IDENTIFIER ON
```

```
GO
```

```
create procedure [dbo].[Sp_Turnos]
```

```
(
```

```
    @pNumeroTurno int
```

```
)
```

```
as
```

```
begin
```

```
select
```

```
P.Cedula as CedulaPaciente,
```

```
P.Apellido + ' ' + p.Nombre as nombrePaciente,
```

```
T.Tur_Cita,
```

```
CONVERT(char(10),T.Tur_Fecha, 101)as Fecha_Turno,
```

```
T.Tur_Numero,
```

```
M.Med_ApellidoPater + ' ' + M.Med_PrimerNom as NomMedico,
```

```
E.Descripcion as Especialidad,
```

```
CONVERT(char(10),H.Hor_HR_Inicio, 108)as HotaInicio,
```

```
CONVERT(char(10),H.Hor_Hr_Fin, 108)as HotaInicio
```

```
from tbl_Turno T
```

```
inner join Tbl_Paciente P
```

```
on T.Id_Paciente = P.Id_Paciente
```



```
inner join Tbl_Horario H
on T.Id_Horarios = H.Id_Horario
inner join Tbl_Medico_Especialista ME
on T.Id_Medico_Espe = T.Id_Medico_Espe
inner join Tbl_Medico M
on M.Id_Medico = ME.Id_Medico
inner join Tbl_Especialidad E
on E.Id_Especialidad = ME.Id_Especialidad
```

```
where T.Tur_Numero = @pNumeroTurno
```

```
end
```

```
GO
```

```
/****** Object: ForeignKey [FK_Tbl_HistoriaClinica_Tbl_Paciente] Script Date: 04/11/2014
16:53:32 *****/
```

```
ALTER TABLE [dbo].[Tbl_HistoriaClinica] WITH CHECK ADD CONSTRAINT
[FK_Tbl_HistoriaClinica_Tbl_Paciente] FOREIGN KEY([Id_Paciente])
REFERENCES [dbo].[Tbl_Paciente] ([Id_Paciente])
```

```
GO
```

```
ALTER TABLE [dbo].[Tbl_HistoriaClinica] CHECK CONSTRAINT
[FK_Tbl_HistoriaClinica_Tbl_Paciente]
```

```
GO
```

```
/****** Object: ForeignKey [FK__Tbl_Medic__Id_Es__2B3F6F97] Script Date: 04/11/2014
16:53:32 *****/
```

```
ALTER TABLE [dbo].[Tbl_Medico_Especialista] WITH CHECK ADD FOREIGN
KEY([Id_Especialidad])
REFERENCES [dbo].[Tbl_Especialidad] ([Id_Especialidad])
```

```
GO
```

```
/****** Object: ForeignKey [FK__Tbl_Medic__Id_Me__2C3393D0] Script Date: 04/11/2014
16:53:32 *****/
```

```
ALTER TABLE [dbo].[Tbl_Medico_Especialista] WITH CHECK ADD FOREIGN
KEY([Id_Medico])
REFERENCES [dbo].[Tbl_Medico] ([Id_Medico])
```

```
GO
```

```
/****** Object: ForeignKey [FK__Tbl_respo__Id_Ar__286302EC] Script Date: 04/11/2014
16:53:32 *****/
```

```
ALTER TABLE [dbo].[Tbl_responsable] WITH CHECK ADD FOREIGN KEY([Id_Area])
REFERENCES [dbo].[Tbl_Area] ([Id_Area])
```

```
GO
```

```
/****** Object: ForeignKey [FK__Tbl_respo__Id_Ca__29572725] Script Date: 04/11/2014
16:53:32 *****/
```

```
ALTER TABLE [dbo].[Tbl_responsable] WITH CHECK ADD FOREIGN KEY([Id_Cargo])
REFERENCES [dbo].[Tbl_Cargo] ([Id_Cargo])
```

```
GO
```

```
/****** Object: ForeignKey [FK_Tbl_Tratamientos_tbl_grupoDiente] Script Date: 04/11/2014
16:53:32 *****/
```



```
ALTER TABLE [dbo].[Tbl_Tratamientos] WITH CHECK ADD CONSTRAINT
[FK_Tbl_Tratamientos_tbl_GrupoDiente] FOREIGN KEY([Id_Grupo])
REFERENCES [dbo].[tbl_GrupoDiente] ([Id_Grupo])
GO
ALTER TABLE [dbo].[Tbl_Tratamientos] CHECK CONSTRAINT
[FK_Tbl_Tratamientos_tbl_GrupoDiente]
GO
/***** Object: ForeignKey [FK_Tbl_Tratamientos_Tbl_HistoriaClinica] Script Date: 04/11/2014
16:53:32 *****/
ALTER TABLE [dbo].[Tbl_Tratamientos] WITH CHECK ADD CONSTRAINT
[FK_Tbl_Tratamientos_Tbl_HistoriaClinica] FOREIGN KEY([Id_Historia])
REFERENCES [dbo].[Tbl_HistoriaClinica] ([Id_Historial])
GO
ALTER TABLE [dbo].[Tbl_Tratamientos] CHECK CONSTRAINT
[FK_Tbl_Tratamientos_Tbl_HistoriaClinica]
GO
/***** Object: ForeignKey [FK_Tbl_Tratamientos_Tbl_Referencias] Script Date: 04/11/2014
16:53:32 *****/
ALTER TABLE [dbo].[Tbl_Tratamientos] WITH CHECK ADD CONSTRAINT
[FK_Tbl_Tratamientos_Tbl_Referencias] FOREIGN KEY([Id_Referencia])
REFERENCES [dbo].[Tbl_Referencias] ([Id_Referencia])
GO
ALTER TABLE [dbo].[Tbl_Tratamientos] CHECK CONSTRAINT
[FK_Tbl_Tratamientos_Tbl_Referencias]
GO
/***** Object: ForeignKey [FK_Tbl_Tratamientos_Tbl_TratamientoInfo] Script Date: 04/11/2014
16:53:32 *****/
ALTER TABLE [dbo].[Tbl_Tratamientos] WITH CHECK ADD CONSTRAINT
[FK_Tbl_Tratamientos_Tbl_TratamientoInfo] FOREIGN KEY([Id_Tratamiento])
REFERENCES [dbo].[Tbl_TratamientoInfo] ([Id_Tratamiento])
GO
ALTER TABLE [dbo].[Tbl_Tratamientos] CHECK CONSTRAINT
[FK_Tbl_Tratamientos_Tbl_TratamientoInfo]
GO
/***** Object: ForeignKey [FK_Tbl_Tratamientos_Tbl_Turno] Script Date: 04/11/2014 16:53:32
*****/
ALTER TABLE [dbo].[Tbl_Tratamientos] WITH CHECK ADD CONSTRAINT
[FK_Tbl_Tratamientos_Tbl_Turno] FOREIGN KEY([id_Turno])
REFERENCES [dbo].[Tbl_Turno] ([Id_Turno])
GO
ALTER TABLE [dbo].[Tbl_Tratamientos] CHECK CONSTRAINT
[FK_Tbl_Tratamientos_Tbl_Turno]
GO
/***** Object: ForeignKey [FK__Tbl_Turno__Id_Me__2F10007B] Script Date: 04/11/2014
16:53:32 *****/
```



```
ALTER TABLE [dbo].[Tbl_Turno] WITH CHECK ADD CONSTRAINT  
[FK__Tbl_Turno__Id_Me__2F10007B] FOREIGN KEY([Id_Medico_Espe])  
REFERENCES [dbo].[Tbl_Medico_Especialista] ([Id_Medico_Espe])
```

GO

```
ALTER TABLE [dbo].[Tbl_Turno] CHECK CONSTRAINT [FK__Tbl_Turno__Id_Me__2F10007B]  
GO
```

```
/****** Object: ForeignKey [FK__Tbl_Turno__Id_Pa__2A4B4B5E] Script Date: 04/11/2014  
16:53:32 *****/
```

```
ALTER TABLE [dbo].[Tbl_Turno] WITH CHECK ADD CONSTRAINT  
[FK__Tbl_Turno__Id_Pa__2A4B4B5E] FOREIGN KEY([Id_Paciente])  
REFERENCES [dbo].[Tbl_Paciente] ([Id_Paciente])
```

GO

```
ALTER TABLE [dbo].[Tbl_Turno] CHECK CONSTRAINT [FK__Tbl_Turno__Id_Pa__2A4B4B5E]  
GO
```

```
/****** Object: ForeignKey [FK__Tbl_Turno__Id_Re__2D27B809] Script Date: 04/11/2014  
16:53:32 *****/
```

```
ALTER TABLE [dbo].[Tbl_Turno] WITH CHECK ADD CONSTRAINT  
[FK__Tbl_Turno__Id_Re__2D27B809] FOREIGN KEY([Id_Responsable])  
REFERENCES [dbo].[Tbl_responsable] ([Id_Responsable])
```

GO

```
ALTER TABLE [dbo].[Tbl_Turno] CHECK CONSTRAINT [FK__Tbl_Turno__Id_Re__2D27B809]  
GO
```

```
/****** Object: ForeignKey [FK__Tbl_Turno_Tbl_Horario] Script Date: 04/11/2014 16:53:32  
*****/
```

```
ALTER TABLE [dbo].[Tbl_Turno] WITH CHECK ADD CONSTRAINT  
[FK__Tbl_Turno_Tbl_Horario] FOREIGN KEY([Id_Horarios])  
REFERENCES [dbo].[Tbl_Horario] ([Id_Horario])
```

GO

```
ALTER TABLE [dbo].[Tbl_Turno] CHECK CONSTRAINT [FK__Tbl_Turno_Tbl_Horario]
```

GO

```
/****** Object: ForeignKey [FK__Tbl_Usuar__Id_Re__2E1BDC42] Script Date: 04/11/2014  
16:53:32 *****/
```

```
ALTER TABLE [dbo].[Tbl_Usuarios] WITH CHECK ADD CONSTRAINT  
[FK__Tbl_Usuar__Id_Re__2E1BDC42] FOREIGN KEY([Id_Responsable])  
REFERENCES [dbo].[Tbl_responsable] ([Id_Responsable])
```

GO

```
ALTER TABLE [dbo].[Tbl_Usuarios] CHECK CONSTRAINT  
[FK__Tbl_Usuar__Id_Re__2E1BDC42]
```

GO

```
/****** Object: ForeignKey [FK__Tbl_Usuar__Id_Ti__276EDEB3] Script Date: 04/11/2014  
16:53:32 *****/
```

```
ALTER TABLE [dbo].[Tbl_Usuarios] WITH CHECK ADD CONSTRAINT  
[FK__Tbl_Usuar__Id_Ti__276EDEB3] FOREIGN KEY([Id_TipoUsuarios])  
REFERENCES [dbo].[Tbl_TipoUsuarios] ([Id_TipoUsuarios])
```

GO



```
ALTER          TABLE          [dbo].[Tbl_Usuarios]          CHECK          CONSTRAINT  
[FK__Tbl_Usuar__Id_Ti__276EDEB3]  
GO
```



MANUAL TECNICO DEL SISTEMA

MANUALES DE INSTALACION DE SOFTWARES QUE SE NECESITA PARA LA CREACION DEL SISTEMA

INSTALACION DE SQL SERVER 2008 R2

Antes de la instalación de sql server 2008 R2 debemos descargar de la página oficial de Microsoft.

Requisitos de hardware son los siguientes:

Al menos 2,0 GB de espacio disponible en disco duro

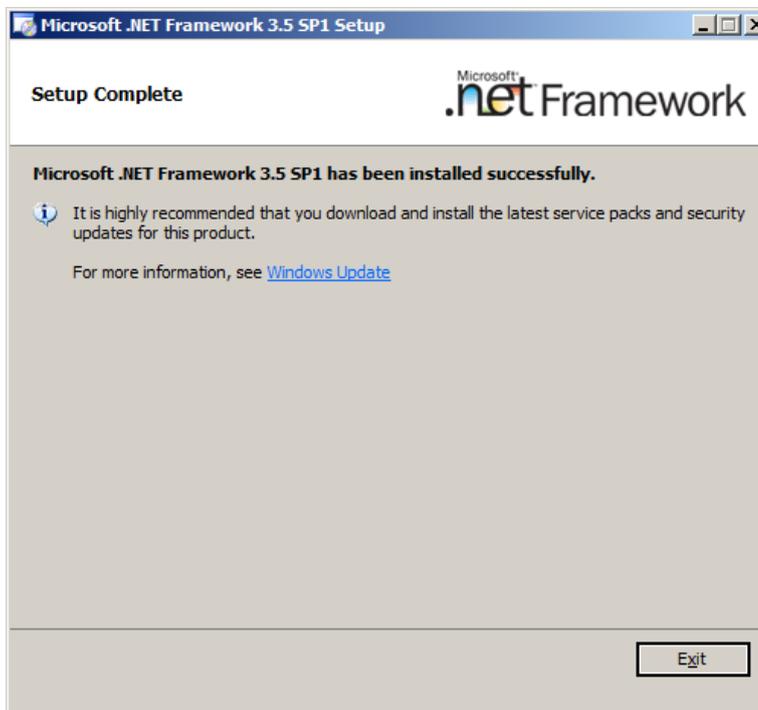
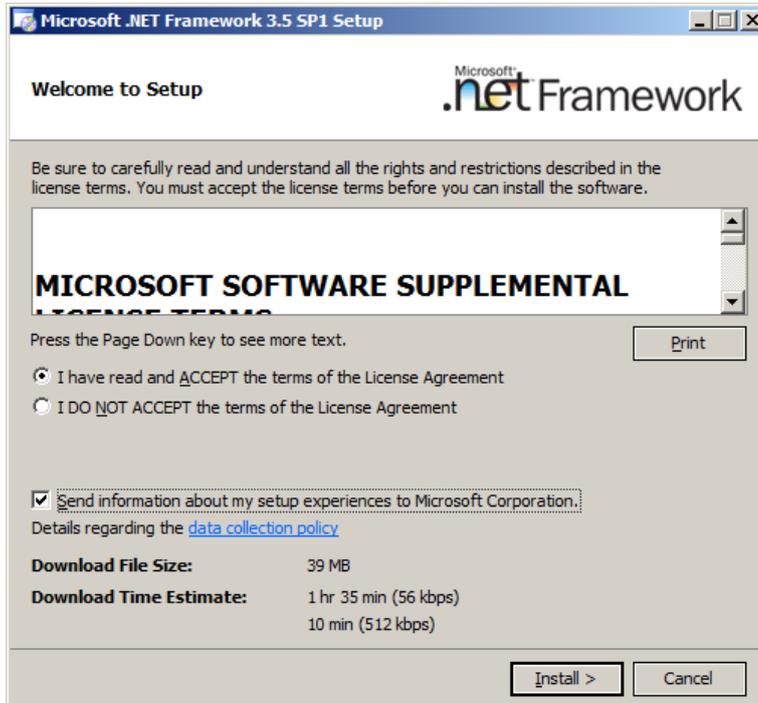
Sistemas operativos compatibles: Windows Server 2008; Windows Server 2003 Service Pack 2; Windows Server 2003 Small Business Server R2; Windows Vista; Windows XP Service Pack 2

Para conocer los requisitos de memoria del procesador y, por favor haga clic aquí.

Ejecutar SQL Server 2008 R2 configuración. En primer lugar, la configuración de una comprobación para determinar se instalacion los requisitos de software. Si no es así, se le pedirá que instale.

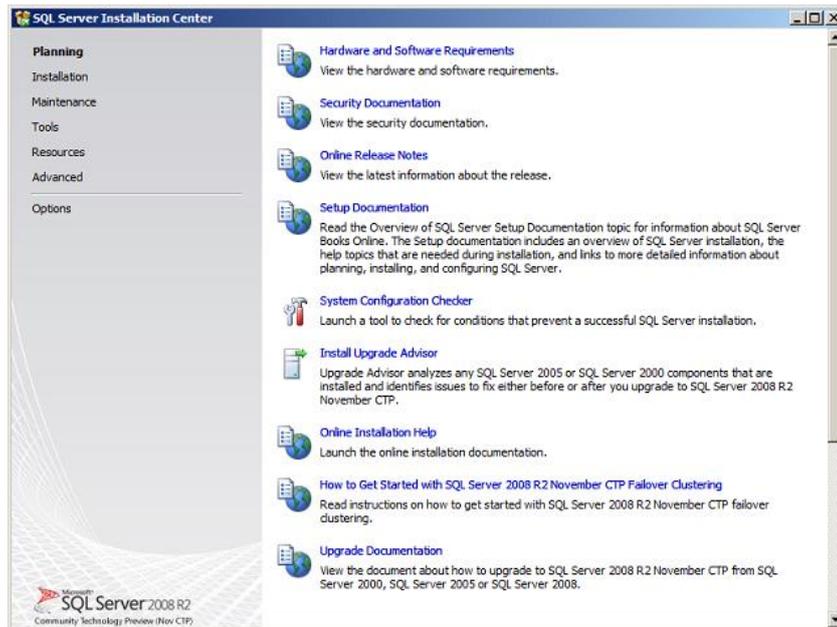


Como se mencionó anteriormente, SQL Server 2008 R2 requiere Windows Installer 4.5 y Net Framework 3.5 SP1 para ser instalado. Haga clic en el botón Aceptar para instalar. Net Framework



Después de instalar la revisión, el Centro de instalación se iniciará

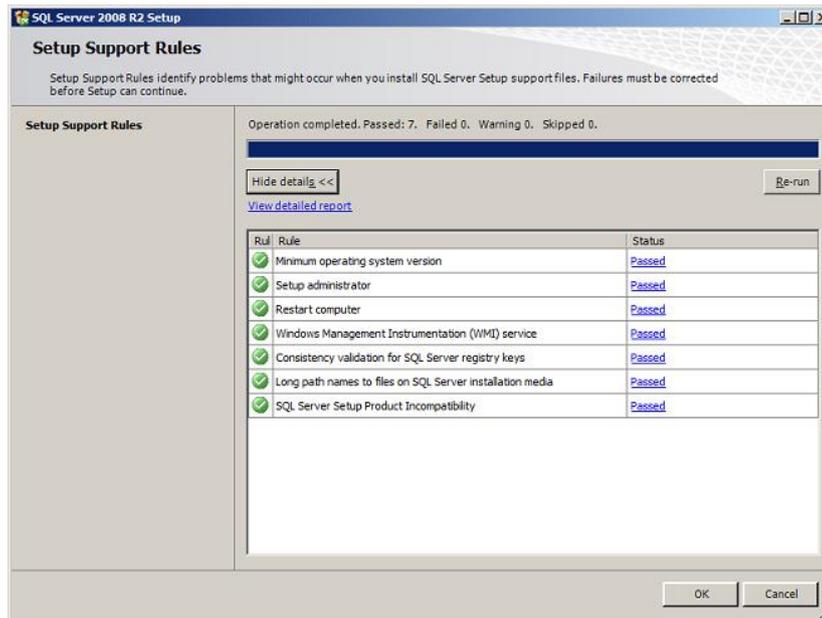




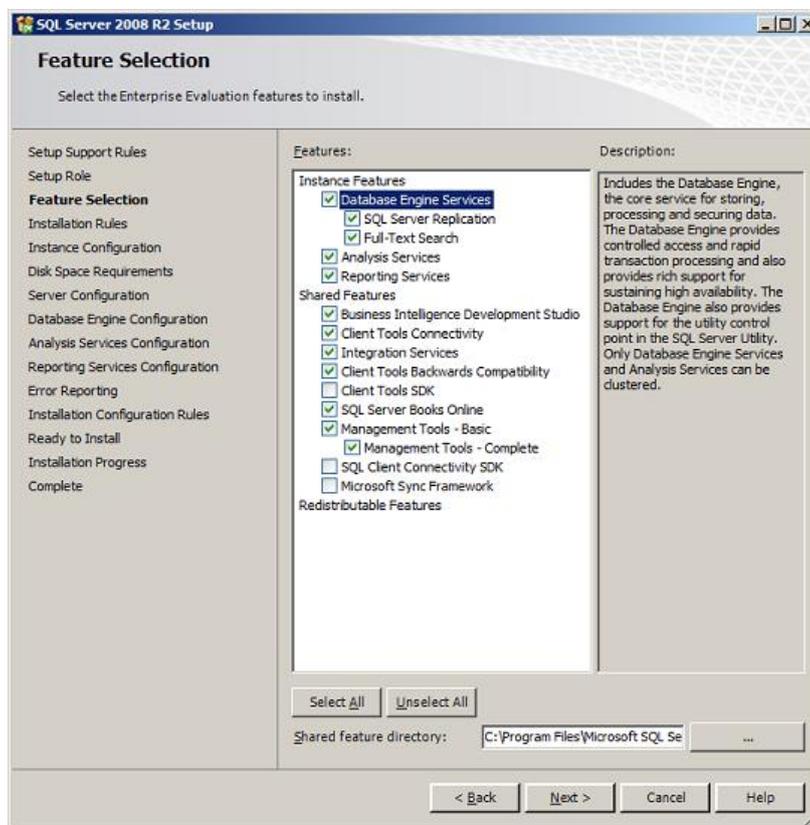
En la pantalla anterior, hacer un clic en el hipervínculo "Instalación" en la parte izquierda de la pantalla.



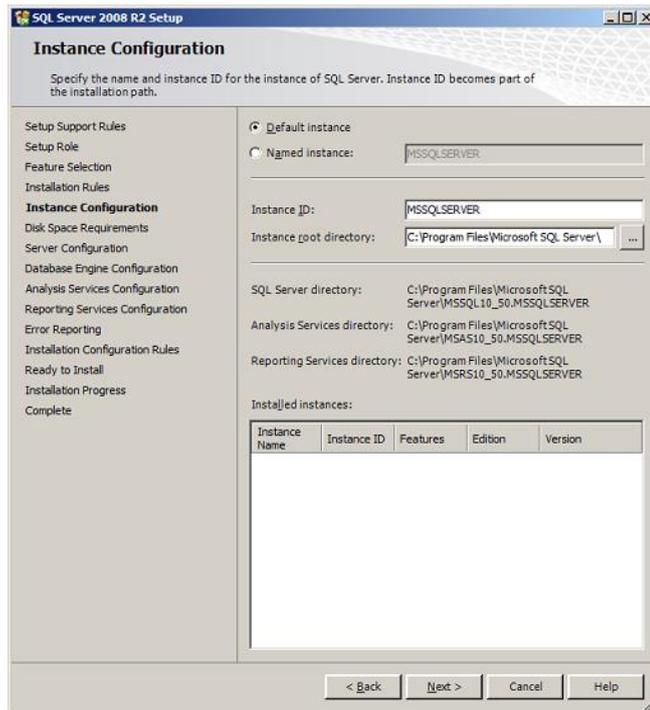
Esta pantalla muestra los tipos de instalación disponibles. Vamos a seleccionar el primer tipo "instalación independiente de Nueva SQL Server o agregar características a una instalación existente".



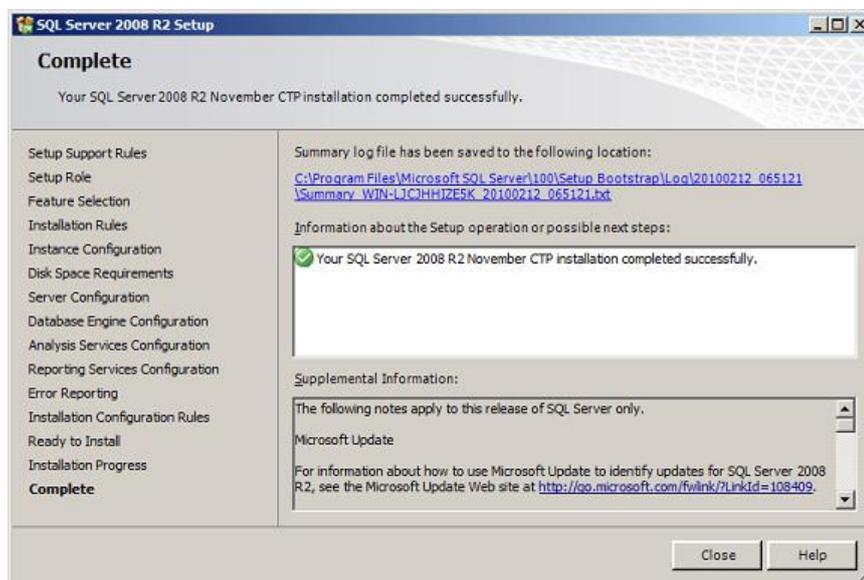
On the Feature Selection page, select the features for your installation.



En la página Configuración de instancia, seleccione una instancia predeterminada o con nombre para su instalación, y seleccione de directorio en el que desea instalar De características que usted ha elegido.



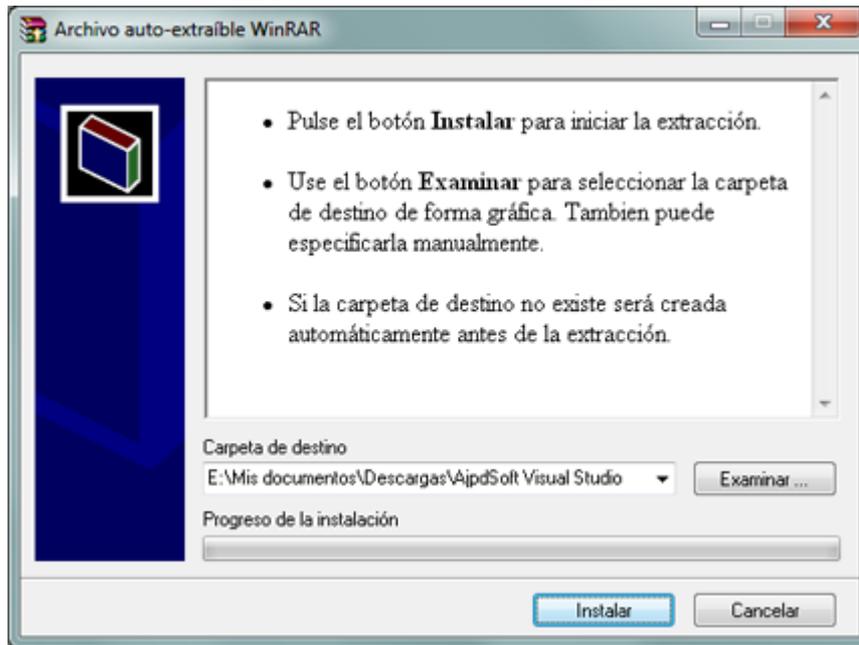
En la página Finalización del Asistente para la instalación de Microsoft SQL Server, puede ver el registro de resumen de configuración, haga clic en el enlace que aparece en esta página. Para salir del asistente de instalación de SQL Server, haga clic en Cerrar.



INSTALACION DE VISUAL STUDIO 2010

Después de haber adquirido el software empezamos la instalación haciendo doble clic en "setup.exe"

Seleccionaremos la carpeta de destino para la descompresión de los ficheros y pulsaremos "Instalar":

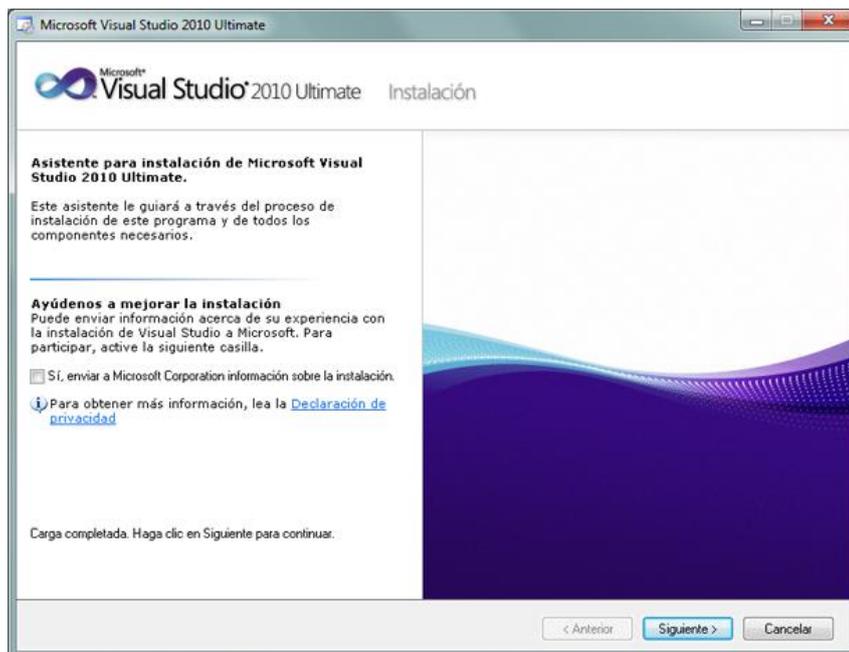


Se iniciará la descompresión de los ficheros descargados en el fichero .iso con la instalación de Microsoft Visual Studio 2010

Se iniciará el asistente para la instalación de Microsoft Visual Studio 2010 (versión de evaluación). Pulsaremos en "Instalar Microsoft Visual Studio 2010":



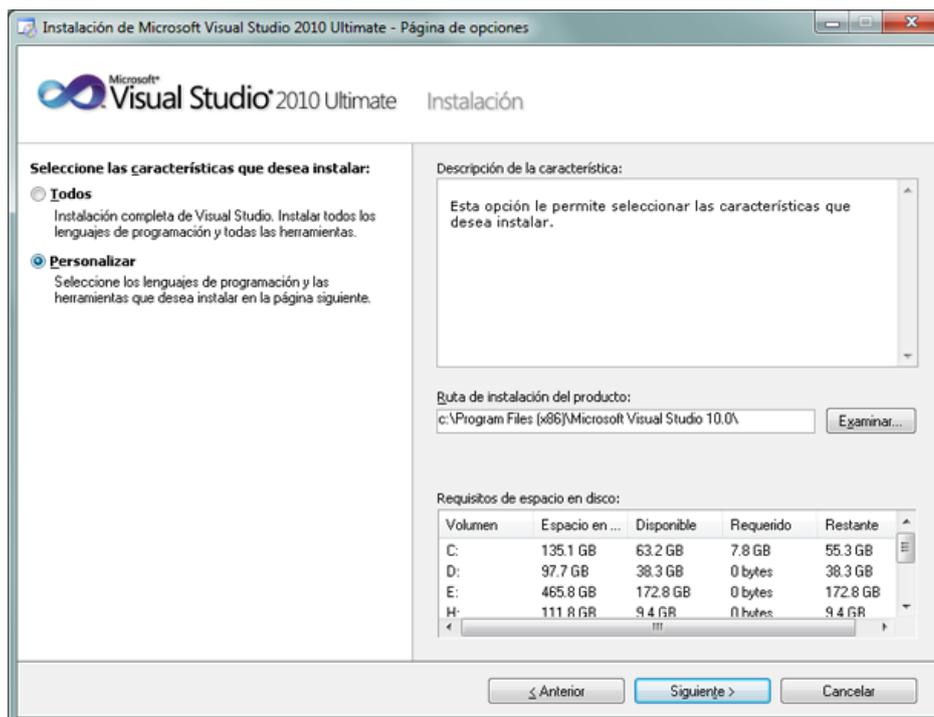
En el primer paso de la instalación de Microsoft Visual Studio 2010 Ultimate, desmarcaremos "Sí, enviar a Microsoft Corporation información sobre la instalación" (si no queremos enviar esta información) y pulsaremos "Siguiente":



Leeremos los términos de licencia del software de Microsoft Visual Studio 2010 Ultimate (edición de prueba). Si estamos de acuerdo marcaremos "He leído los términos de la licencia y los acepto". Pulsaremos "Siguiente" para continuar:



A continuación marcaremos "Personalizar" para seleccionar los lenguajes de programación y las herramientas que se quieran instalar. En "Ruta de instalación del producto" indicaremos la unidad y carpeta de destino de la instalación:



Seleccionaremos los lenguajes a instalar:

Visual Basic.

Visual C++.

Visual C#.

Visual F#.

Seleccionaremos también las características a instalar:

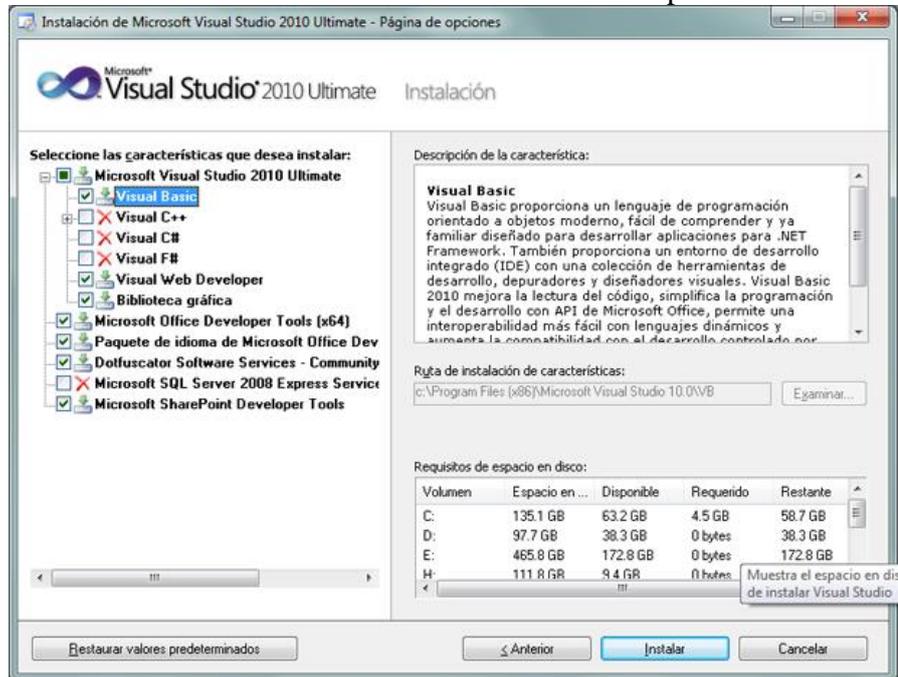
Microsoft Office Developer Tools.

Dotfuscator Software Services.

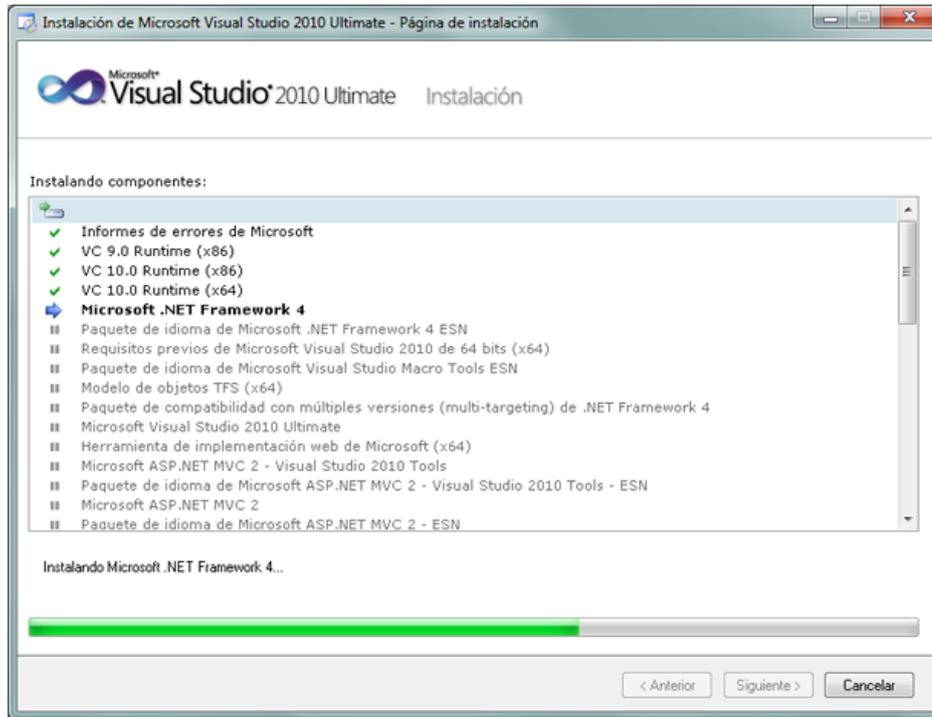
Microsoft SQL Server 2008 Express.

Microsoft SharePoint Developer Tools.

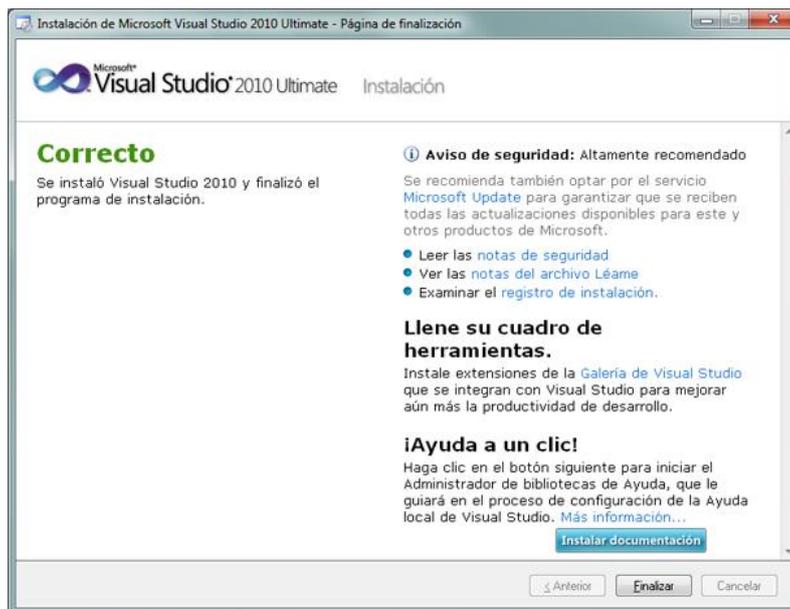
Una vez seleccionadas las características a instalar pulsaremos en el botón "Instalar":



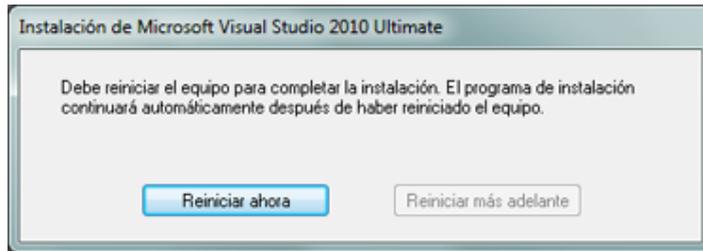
Se iniciará la instalación de Microsoft Visual Studio 2010 Ultimate: informe de errores de Microsoft, VC 9.0 Runtime, VC 10.0 Runtime, Microsoft .Net Framework 4, Microsoft Visual Studio 2010 Ultimate, Microsoft ASP .Net, etc.:



Tras la instalación, el asistente para instalar Visual Studio mostrará la siguiente ventana, con el texto "Correcto. Se instaló Visual Studio 2010 y finalizó el programa de instalación". Pulsaremos "Finalizar":

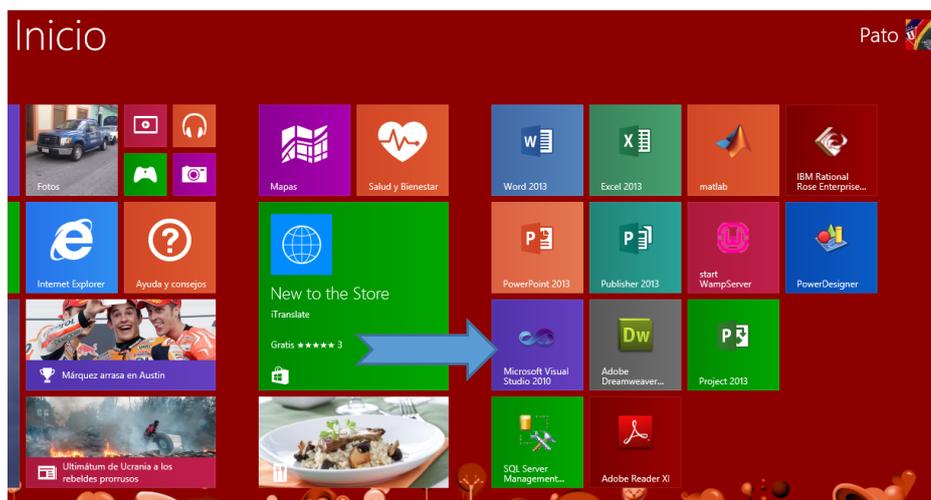


Tras el proceso de instalación puede que pida reiniciar el equipo:

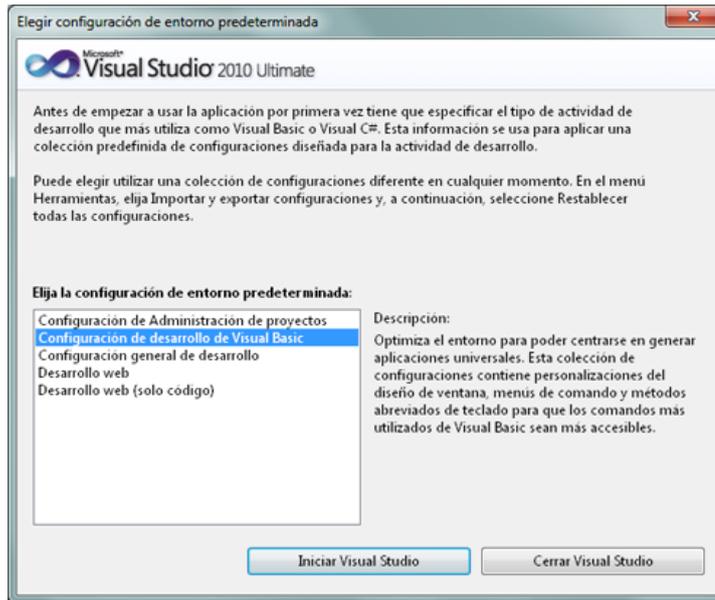


Con el texto: "Debe reiniciar el equipo para completar la instalación. El programa de instalación continuará automáticamente después de haber reiniciado el equipo".

Desde el botón "Inicio" - "Todos los programas" podremos ver el grupo de programas "Microsoft Visual Studio 2010":



Tras iniciar por primera vez Microsoft Visual Studio 2010 Ultimate, seleccionaremos la configuración de entorno predeterminada, por ejemplo "Configuración de desarrollo de Visual Basic":



Y ya podremos disponer de este entorno de programación:

