

ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACION TECNOLÓGICA

SISTEMA PARA ADMINISTRACIÓN DENTAL DENTALSYS

PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE TECNÓLOGO EN ANÁLISIS DE SISTEMAS INFORMÁTICOS

VÍCTOR GONZALO AGUIRRE GUACHAMBOZA

DIRECTOR: ING. LUZMARINA VINTIMILLA

Quito, Junio del 2006

Eliminado: ABRIL

DECLARACIÓN

YO, Aguirre Guachamboza Víctor Gonzalo, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedo mi derecho de propiedad intelectual correspondiente a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

Eliminado: ¶

VICTOR GONZALO AGUIRTRE GUACHAMBOZA

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Víctor Gonzalo Aguirre Guachamboza bajo mi supervisión.

|

Ing. Luz marina Vintimilla Msc.

AGRADECIMIENTOS

Agradezco a Dios por que me ha bendecido con la vida y me concedió, la oportunidad de estudiar.

Por su Apoyo desinteresado al Ing. Daniel Manangón, Coordinador de la carrera de Análisis de Sistemas Informáticos.

A la Ing. Veintimilla , Directora del presente trabajo de investigación, por su valiosa dedicación, orientación y colaboración, al haber compartido conmigo sus nutridos conocimientos.

A la Escuela Politécnica Nacional en especial a la Escuela de Formación Tecnológica que en forma generosa e incondicional colaboro en mi formación como profesional .

A mi esposa y a nuestros hijos que con su amor y ternura supieron llenarme de fortaleza y perseverancia

A mi familia por la paciencia, confianza y apoyo que me han ofrecido día a día.

VICTOR GONZALO AGUIRRE GUACHAMBOZA

DEDICATORIA

A ti Señor Jesús que con tu palabra guiaste mi vida por el camino del bien y la verdad .

A mis Padres que con su amor y sacrificio alimentaron mi espíritu y fortalecieron mi mente, de manera especial a mi madre que esta en el cielo por haber luchado incansablemente por sus hijos

A mi esposa por ayudarme y comprenderme en los momentos mas difíciles, a nuestros hijos que con su inocencia y dulzura alimentan en mí esa fuerza y motivación que me a permitido llegar a culminar mi carrera, gracias por ser tan lindos.

A mis Hermanos quienes me brindaron su ayuda y respaldo incondicional con sus oraciones y estuvieron en los buenos y malos momentos de mi vida.

Víctor Aguirre

INDICE

Eliminado: ¶

CAPITULO I

Eliminado: ¶
¶

CONTENIDO

1.1 Antecedentes.....	
1.2 Planteamiento del problema.....	
1.3 Misión.....	
1.4 Formulación y Sistematización.....	
1.4.1 Formulación.....	
1.4.2 Sistematización.....	
1.5 Objetivos de la Investigación.....	
1.5.1 Objetivo General.....	
1.5.2 Objetivos Específicos.....	
1.6 Justificación del Proyecto.....	
1.6.1 Justificación Metodológica.....	
1.6.2 Justificación Práctica.....	
1.7 Aspectos Metodológicos.....	

Eliminado: 3

Eliminado: 3

CAPITULO II

MARCO TEÓRICO

2.1 Paradigma.....	
2.2 Paradigma Espiral Incremental.....	
2.3 Metodología.....	
2.3.1 Metodología OMT.....	
2.3.1.1 Procesos de la Metodología OMT.....	
2.3.1.1.1 Análisis.....	
2.3.1.1.2 Diseño del Sistema.....	
2.3.1.1.3 Diseño de Objetos.....	

Eliminado: ¶

Eliminado: 2.3.1.1
Introducción.....
.....¶

Eliminado: 2

Eliminado: ¶

Eliminado: 2

Eliminado: 2

Eliminado: 2

2.3.1.1.4 Implementación.....	Eliminado: 2
2.4 Modelos de Objetos.....	
2.4.1 Modelo Estático.....	
2.4.2 Modelo Dinámico.....	
2.4.3 Modelo Funcional.....	
2.5 Lenguaje Unificado del Modelado(UML).....	
2.5.1 Diagramas UML.....	Eliminado: 2.5.1 Introducción.....¶
2.5.2 Diagramas de caso de uso.....	Eliminado: 2
2.5.2.1 Tipos de Relaciones en los casos de uso.....	Eliminado: 2.5.3 Descripción de Actores.....¶
2.5.3 Diagramas de Clases.....	Eliminado: 4
2.5.3.1 Relación entre clases.....	Eliminado: 4
2.5.3.1.1 Asociación.....	Eliminado: 5
2.5.3.1.2 Agregación.....	Eliminado: 5
2.5.3.1.3 Generalización.....	Eliminado:
2.5.4 Diagrama de Secuencias.....	Eliminado: 5
2.5.5 Diagrama de Colaboración.....	Eliminado: 5
2.5.6 Diagrama de Actividades.....	Eliminado: 5
2.5.7 Diagrama de Estados.....	Eliminado: 2.5.5.2 Diccionario de clases.....¶ 2.5.6 Diagrama de Interacción.....¶
2.5.8 Implementación del Sistema.....	Eliminado:
2.5.9 Pruebas.....	Eliminado: 6.1
2.5.9.1 Prueba funcional.....	Eliminado: 6

CAPITULO III

CONSTRUCCIÓN	
3.1 Crystal Report 8.0.....	Eliminado: 5.7
3.1.1 Plataforma.....	Eliminado: 5.8
3.2 Visual Basic.....	Eliminado: 5.9
3.3 Base de datos Access.....	Eliminado: 5.10
3.4 Rational Rose 2000.....	Eliminado: ¶ CONSTRUCCION
3.5 Cliente Servidor.....	Eliminado: .
3.5.1 Características del Cliente Servidor.....	Eliminado: ..
3.5.1 Ventajas de Cliente Servidor.....	
3.5.2 Desventajas de Cliente Servidor.....	

CAPITULO IV

Eliminado: 3.6 Prueba Funcional.....
.....¶
3.7 Mantenimiento.....
.....

Eliminado: ¶

CONCLUSIONES Y RECOMENDACIONES

4.1 Conclusiones.....
4.2 Recomendaciones.....
4.3 Bibliografía.....

CAPITULO V

Eliminado: ¶

Con formato

MANUAL DE USUARIO

5.1 Introducción.....
5.2 Requerimientos del sistema.....
5.3 Instalación del sistema.....
5.4 Perfiles de usuario.....
5.5 Acceso al sistema.....
5.6 Menú Consultas.....
5.8 Glosario.....

MANUAL TÉCNICO

Eliminado: ¶
¶

MANUAL DE USUARIO

MANUAL DE INSTALACIÓN

ANÁLISIS DE COSTOS

ANEXOS

CAPITULO I

1.1 ANTECEDENTES

La Clínica Dental "" ubicada en la ciudad de Ambato, sector Ingahurco Bajo, es una institución de carácter privado cuya finalidad es servir a toda persona que vive dentro y fuera del sector brindando un servicio eficiente.

Se proyecta como una institución sólida con principios y valores éticos de honestidad, integridad, y transparencia, comprometidos con la ciudadanía, brindando un servicio apoyado en un equipo profesional y con una infraestructura muy adecuada.

1.2 PLANTEAMIENTO DEL PROBLEMA

La clínica dental es una entidad privada cuya finalidad es la de prestar atención a la población gracias al respaldo de la gente esta se encuentra en un proceso de crecimiento en el cual se ha notado un incremento en el numero de pacientes atendidos, en horario de 09h:00 a 20H.00, con un promedio de 10 a 30 pacientes diarios.

La clínica dental cuenta con varios servicios como: curaciones, calzas, extracciones, entre otros. El funcionamiento y administración de esta clínica dental se lo realiza en forma manual por lo cual este proceso está sujeto a errores y mal manejo de información fundamentada en los siguientes aspectos:

- El registro de datos se lo realiza manualmente en carpetas envinchadas lo cual ocasiona problemas al momento de emitir reportes porque en ciertas ocasiones los datos que se anotaron son ilegibles.
- Se lleva un registro manual de los pacientes atendidos diariamente por lo que el tiempo en localizar una ficha toma un gran esfuerzo y tiempo.
- No se realiza control de inventarios semanales, mensuales o trimestrales para obtener existencias y faltantes de gastos.

Con formato: Numeración y viñetas

- El incremento poblacional y el número de usuarios de la clínica dental se deben al excelente trato y atención de calidad, lo cual conlleva a que la administración del laboratorio sea aun más problemática en el futuro principalmente en el aspecto contable.

Si la clínica dental permanece con estos problemas podría perder competitividad frente a otras entidades similares que hacen uso de la tecnología existente.

Ante estas situaciones se concluye que los problemas antes descritos exige a la clínica la implementación de un sistema computarizado para optimizar el recurso humano y tiempo, disminuir errores de en el manejo de la información y mejorar la presentación de informes de la clínica.

1.3 VISION

La clínica dental quiere ser una entidad que provee a los pacientes todos los servicios necesarios para en el área de la salud para satisfacer las necesidades de la población

Eliminado:

Con formato: Numeración y viñetas

1.4 FORMULACIÓN Y SISTEMATIZACIÓN

1.4.1 Formulación

Frente a esta realidad, ¿Cómo administrar la información en forma eficiente para otorgar un mejor servicio al público en general, brindando una atención personalizada y dinámica en la clínica dental?

1.4.2 Sistematización

- ¿Cómo mejorar la manipulación de datos generados por atención a los pacientes de la clínica?
- ¿Cómo facilitar la administración de los servicios que brindan la clínica dental?
- ¿Qué hacer para mantener y emitir facturaciones de los servicios prestados?
- ¿Cómo facilitar la atención al cliente?
- ¿Cómo proporcionar seguridad a la información generada por la clínica dental?

Con formato: Numeración y viñetas

1.5 OBJETIVOS DE LA INVESTIGACIÓN

1.5.1 Objetivo General

Desarrollar e implementar un sistema informático que ayude a mejorar la administración de los servicios que ofrece la Clínica Dental.

1.5.2 Objetivos Específicos

- Crear un módulo que permita registrar, guardar y actualizar la información de los pacientes.
- Crear un módulo que permita administrar los servicios que brinda la Clínica Dental.
- Crear un módulo que permita registrar las facturaciones de cada servicio al paciente.
- Crear un modulo para emitir citas medicas.
- Establecer en el sistema perfiles de usuarios y passwords.
- Crear los módulos para generar los reportes.

Con formato: Numeración y viñetas

1.6 JUSTIFICACIÓN DEL PROYECTO

Eliminado: ¶

¶
¶
¶

1.6.1 Justificación Metodológica

El sistema de Administración dental permitirá el control de datos proporcionados por los pacientes de esta manera tanto el personal medico y los pacientes se verán beneficiados ya que la información estará actualizada y estará a disposición de sus administradores.

En este proyecto se emplea la metodología OMT, por las siguientes razones:

- Pone énfasis en la importancia del modelo para lograr una abstracción, en la cuál el análisis está enfocado al mundo real.
- Promueve la reutilización de código, la reducción de posteriores errores y el futuro y fácil mantenimiento de los sistemas.
- Satisface las necesidades actuales de desarrollo de software complejo y distribuido.
- Permite elaborar sistemas robustos, eficaces, y fiables con un presupuesto económico.

1.6.2 Justificación Práctica

El cambio que experimenta la clínica dental en relación al numero de paciente exige y a despertado en su directivo el interés por contar con un sistema informático para el manejo y funcionamiento de la clínica Dental .

El uso del sistema informático optimizará las actividades que se realizan en el laboratorio, facilitando a los pacientes, personal médico y administrativo la utilización de los recursos y servicios de la misma.

La clínica dental . brindará apoyo para:

- Optimizar la información que se genera.
- Aligerar las consultas y atención de pacientes.
- La firmeza y seguridad de la información.
- Evitará la acumulación innecesaria de documentos.
- Proporcionará reportes de manera ágil y oportuna.

1.7 ASPECTOS METODOLÓGICOS

Con formato: Numeración y viñetas

<u>PARADIGMA</u> <u>ESPIRAL</u> <u>ORIENTADO A OBJETOS</u>	<u>METODOLOGÍA OMT</u>	<u>DIAGRAMAS UML</u>	
<u>Análisis</u>	<u>Modelo Estático</u>	<u>Descripción de Actores</u> <u>Diagrama de Casos de Uso</u> <u>Descripción de Casos de Uso</u> <u>Diagrama de Clases</u> <u>Diccionario de Clases</u>	
<u>Diseño</u>	<u>Modelo Dinámico</u>	<u>Diagrama de Interacción</u>	<u>Diagrama de secuencia</u> <u>Diagrama de colaboración</u>
	<u>Modelo Funcional</u>	<u>Diagrama de actividades</u>	
<u>Construcción</u>	<u>Herramientas de Software</u>	<u>Rational Rose 2000</u> <u>Crystal Report 8.0</u> <u>DB Access</u> <u>Visual Basic</u>	
<u>Pruebas</u>	<u>Prueba Funcional</u>	<u>Descripción de la Prueba</u> <u>Procedimiento de la Prueba</u>	
<u>Mantenimiento</u>	<u>La documentación del desarrollo y seguimiento de los modelos facilita el posterior mantenimiento.</u>		

MARCO TEÓRICO

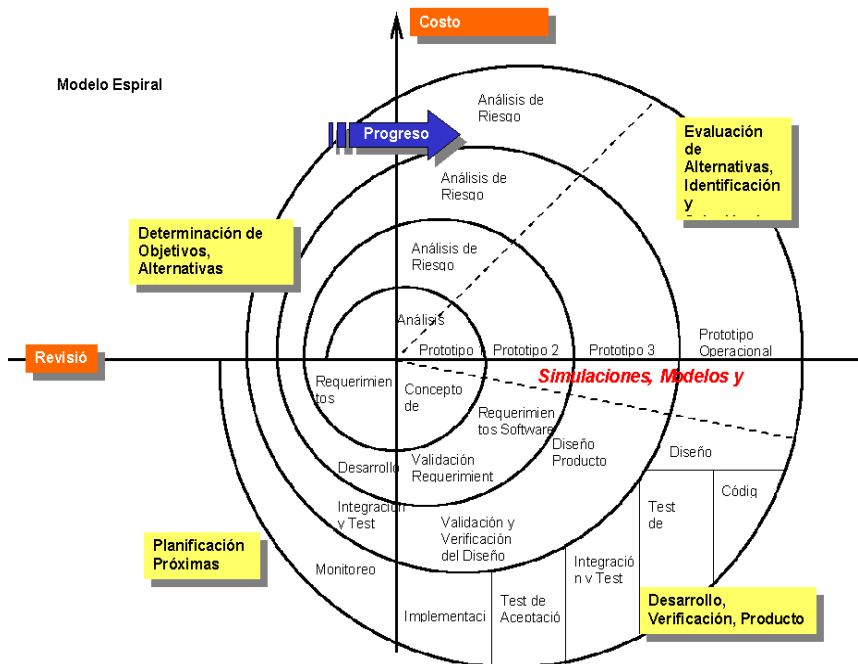
2.1 PARADIGMA

Un paradigma es una técnica, un modelo o un conjunto de herramientas para representar la solución de problemas específicos.

2.2 PARADIGMA ESPIRAL INCREMENTAL

“El modelo espiral para la ingeniería de software ha sido desarrollado para cubrir las mejores características tanto del ciclo de vida clásico, como de la creación de prototipos, añadiendo al mismo tiempo un nuevo elemento: el análisis de riesgo. El modelo representado mediante la espiral de la figura, define cuatro actividades principales”.

MODELO DE FASES DEL CICLO DE VIDA (ESPIRAL)



El modelo espiral es el esfuerzo de desarrollo iterativo (que se repite); tan pronto como uno completa un esfuerzo de desarrollo, otro comienza.

1. **Planificación:** determinación de objetivos, alternativas y restricciones.
2. **Análisis de riesgo:** análisis de alternativas e identificación / resolución de riesgos.
3. **Ingeniería:** desarrollo del producto del "siguiente nivel",
4. **Evaluación del cliente:** Valorización de los resultados de la ingeniería.

Este modelo nos muestra el labor que se va realizando cada vez dando una curva de la espiral; en la primera curva alrededor de la espiral, analizamos la situación y determinamos que los mayores riesgos son la interfaz del usuario.

Después de un cuidadoso análisis de las formas alternativas de direccionar esto, determinamos que el mejor curso de acción es construir un prototipo el cual lo probamos y el cliente es quien nos da su aprobación de si está o no yendo su sistema de acuerdo a sus requerimientos

En la segunda vuelta alrededor de la espiral notamos que pueden ir apareciendo nuevos requerimientos por parte del cliente; entonces comienza la tercera vuelta.

2.3 **METODOLOGÍA**

Es una colección de técnicas basadas sobre una filosofía común o una forma de trabajo, que se establece en conjunto sobre una plataforma llamada Ciclo de Vida del Desarrollo de Sistemas o simplemente Ciclo de Vida.

“La metodología corresponde a un conjunto de modelos, lenguajes y otras herramientas que nos facilitan la representación de los datos de cada fase del proceso (análisis, diseño, construcción, etc.) junto con las reglas que permiten el paso de una fase a la siguiente”.

Para el desarrollo de este sistema en el cual los procesos están definidos claramente utilizaremos la Metodología OMT de **Rumbaugh**.

Eliminado: ¶

2.3.1 METODOLOGIA OMT

Eliminado: ¶

¶
¶
¶

Eliminado: ¶

La metodología OMT (Object Modeling Technique) fue creada por James Rumbaugh y Michael Blaha en 1991, mientras James dirigía un equipo de investigación de los laboratorios General Electric.

Existen muchas aproximaciones de desarrollo de software que utilizan modelos orientado a objetos, pero que no tienen todos los soportes para desarrollo de aplicaciones de base de datos. Algunas aproximaciones carecen de suficientes abstracciones y tienen un bajo relacionamiento para detalles de implementación.

Otros métodos de programación orientados ponen un escaso énfasis en la estructura de datos y constantes, que son muy importantes para aplicaciones de base de datos.

OMT es una de las metodologías de análisis y diseño orientadas a objetos, más maduras y eficientes que existen en la actualidad. Esto facilita su evolución para acoplarse a todas las necesidades actuales y futuras de la ingeniería de software. Pone énfasis en la importancia del modelo y uso de modelo para lograr una abstracción, en el cual el análisis esta enfocado en el mundo real para un nivel de diseño, también pone detalles particulares para modelado de recursos de la computadora. Esta Tecnología puede ser aplicada en varios aspectos de implementación incluyendo archivos, base de datos relacionales, base de datos orientados a objetos. OMT esta construido alrededor de descripciones de estructura de datos, constantes, sistemas para procesos de transacciones.

2.3.1.1 PROCESO DE LA METODOLOGÍA OMT

Los pasos para desarrollar diseño OMT son:

- Conceptualización

El desarrollo empieza con el análisis de la empresa o negocio, o de cómo los usuarios conciben el sistema y formulan sus requerimientos.

La Conceptualización es a menudo por la re-ingeniería de procesos de la empresa, es una observación crítica de los procesos de la empresa, y su impacto económico.

Eliminado: conceptualización

En esta etapa se debe tener en cuenta las siguientes preguntas:

- ¿Cuál es la aplicación ?
- ¿Que problemas tendrán que ser resueltos ?
- ¿Dónde será usado el sistema ?
- ¿Cuándo será requerido el sistema ?
- ¿ Para qué es necesario el sistema ?

2.3.1.1.1 ANÁLISIS

Los requerimientos formados durante la conceptualización son revisados analizados para la construcción del modelo real. La meta del análisis es especificar las necesidades que deben ser satisfechas.

Durante el análisis se construye un modelo de dominio de la aplicación sin tener en cuenta la implementación. Deberá incluir aquella información que sea significativa desde el punto de vista del mundo real, presentando el aspecto externo del sistema.

Se establece la definición del problema.

Se construye un modelo de objetos.

Se desarrolla un modelo dinámico.

Se construye un modelo funcional.

Se verifican, integran y retinan los tres modelos.

2.3.1.1.2 DISEÑO DEL SISTEMA

El diseño tiene un alto nivel estratégico y decisión para resolver los problemas. Los problemas grandes se deben ver desde el punto de análisis y diseño, este sistema se divide en subsistemas, a su vez este subsistema puede ser dividido en otros subsistemas de manera que puedan ser manejados y cada componente pueda ser comprensible.

En esta etapa se deben crear estrategias, formular una arquitectura para el sistema y las políticas que deben guiarla además un detalle del diseño.

Debiendo

tener en cuenta los siguientes aspectos:

- Divisar una arquitectura
- Elegir una implementación para un control externo
- Si se usa base de datos elegir el paradigma de administración de base de datos
- Determinar oportunidades para el rehusó.
- Elegir estrategia para interacción de datos.
- Elegir una forma de identificar los objetos.
- Detallar el diseño.

Durante el diseño del sistema se debe hacer un cuadro de estrategias y decisiones arquitecturales, tener una idea más precisa de clases y métodos individuales. Adicionalmente se puede mejorar el modelo de diseño para mejorar la implementación.

Se debe considerar los siguientes pasos:

- Uso de transformaciones para simplificar y optimizar el modelo de objetos desde el análisis.
- Elaborar un modelo de objeto
- Elaborar un modelo funcional
- Evaluar la calidad del diseño del modelo
- Implementación

El diseño es trasladado a un lenguaje de programación actual y código de base de datos. Este paso puede ser aplicado y considerado durante el análisis y diseño, para subir el performance del sistema.

2.3.1.1.3 DISEÑO DE OBJETOS

El diseñador de objetos construye un modelo de diseño basándose en el modelo de análisis, pero incorporando detalles de implementación. El diseño de objetos se centra en las estructuras de datos y algoritmos que son necesarios para implementar cada clase. OMT describe la forma en que el diseño puede ser implementado en distintos lenguajes (orientados y no orientados a objetos, bases de datos, etc.).

En el diseño de objetos se toman las decisiones necesarias para construir un sistema sin descender a los detalles particulares de un lenguaje o sistema de base de datos.

- Obtención de las operaciones para el modelo de objetos a partir de los demás modelos.
- Diseño de algoritmos para la implementación de las operaciones.
- Optimización de las vías de acceso de datos.
- Implementar el control del software completando la aproximación seleccionada durante el diseño del sistema.
- Ajuste de la estructura de clases para incrementar la herencia.
- Diseño de la implementación de las asociaciones.
- Se determina la representación exacta de los atributos que son objetos.
- Empaquetamiento de las clases y asociaciones en módulos.

2.3.1.1.4 IMPLEMENTACION

Las clases de objetos y relaciones desarrolladas durante el análisis de objetos se traducen finalmente a una implementación concreta. Durante la fase de implementación es importante tener en cuenta los principios de la ingeniería del software de forma que la correspondencia con el diseño sea directa y el sistema implementado sea flexible y extensible

Durante la implementación se codifican, tanto las estructuras en el dominio de la solución. La base que la sustenta es el diseño de objetos.

El código puede ser una simple transición de las decisiones de diseño a las características propias de un lenguaje.

En esta parte se discuten los detalles específicos de la implementación de un sistema utilizando lenguajes orientados a objetos, lenguajes no orientados a objetos y sistemas de administración de bases de datos.

Para implementar un diseño orientado a objetos es utilizado un lenguaje orientado a objetos ya que estos lenguajes ofrecen distintos grados de apoyo para los conceptos orientados a objetos.

2.4 MODELOS DE OBJETOS

2.4.1 MODELO ESTÁTICO

Es el modelo más importante, describe la estructura de los objetos de un sistema-identidad, relacionados con otros objetos, atributos y operaciones, el modelo de objetos proporciona el entorno esencial en el cual se pueden situar el modelo dinámico y el modelo funcional.

Identificar los objetos y las clases

Preparar un diccionario de datos.

Identificar las asociaciones entre objetos.

Identificar atributos de objetos y enlaces.

Organizar y simplificar las clases de objetos empleando la herencia

Verificación de las vías de acceso necesarias para llevar a cabo las probables consultas.

Realizar las interacciones necesarias para el refinamiento del modelo.

Agrupar las clases en módulos.

2.4.2 MODELO DINÁMICO

Representa los aspectos temporales de comportamiento “de control” del sistema, mediante la secuencia de operaciones en el tiempo.

Se preparan escenarios de secuencias típicas de interacción.

Se identifican sucesos que actúen entre objetos.

Se prepara un seguimiento de sucesos para cada escenario.

Se construye diagramas de estado

Se compara los sucesos intercambiados entre objetos para verificar la congruencia.

2.4.3 MODELO FUNCIONAL

Representa los aspectos transformacionales “de función” del sistema, mediante la transformación de valores de los datos. Se representa mediante un diagrama de flujo.

Identificación de los valores de entrada y salida.

Construcción de diagramas de flujo de datos que muestren las dependencias funcionales.

Descripción de las funciones.

Identificación de las restricciones.

Especificación de los criterios de optimización.

2.5 LENGUAJE UNIFICADO MODELADO(UML)

El UML (Unified Modeling Lenguaje o lenguaje unificado de modelado) es un lenguaje gráfico para la especificación, visualización, construcción y documentación de piezas de información usadas o producidas durante el proceso de desarrollo de software. A estas piezas de información se les conoce como artefactos.

El Lenguaje Unificado de Modelado (UML) es una técnica para la especificación de sistemas en todas sus fases. Esta ha sido desarrollado por los más importantes autores en materia de Análisis y Diseño de Sistemas y ha sido usada con éxito en sistemas hechos para toda clase de industrias alrededor del mundo: Salud, Bancos, Comunicaciones, Aeronáutica, Finanzas, etc.

Sin lugar a dudas OOAD (Object Oriented Analysis and Design), implementado con UML (Unified Modeling Language), es la metodología más avanzada en la actualidad. Esta metodología introduce los Casos de Uso, una poderosa herramienta para reducir los riesgos en la definición de requerimientos de sistemas nuevos. Los Casos de Uso sirven como columna vertebral del proceso de desarrollo de aplicaciones y tienen como objetivo garantizar que los resultados se ajusten completamente a las expectativas de los usuarios finales.

UML sirve para hacer modelos que permitan:

- Visualizar como es un sistema o como queremos que sea.
- Especificar la estructura y/o comportamiento de un sistema.
- Hacer una plantilla que guíe la construcción de los sistemas
- Documentar las decisiones que hemos tomado

El modelado sirve no solamente para los grandes sistemas; aún en aplicaciones de pequeño tamaño se obtienen beneficios de modelar, sin embargo, es un hecho que entre mas grande y más complejo es el sistema, el modelado juega un papel más importante. Esto se debe a una razón simple:

“Hacemos modelos de sistemas complejos porque no podemos entenderlos en su totalidad”

Con formato

Con formato

Hay límites para el entendimiento de la complejidad. A través del modelado reducimos el ámbito del problema de estudio al enfocar solo un aspecto a la vez.

UML puede ser usado extensivamente en: Recopilación de requerimientos, Análisis de aplicaciones, Diseño de sistemas, en pruebas, en implementación, en reingeniería y prácticamente en cualquier actividad de desarrollo que sea susceptible de ser modelada.

Cabe aclarar que aunque UML es orientado a objetos preferentemente, es útil en cualquier modelo tecnológico ya que es independiente de lenguajes de programación o tecnología determinada.

2.5.1 DIAGRAMAS UML

Los elementos de UML se muestran mediante diagramas que presentan múltiples vistas del sistema, ese conjunto de vistas son conocidos como *modelos*.

UML presenta varios diagramas donde cada uno representa un aspecto del sistema. De ahí que varios investigadores según sus criterios y puntos de vista mencionan qué diagramas emplear en el desarrollo de los sistemas de información; sin mencionar cuáles son los diagramas más adecuados en las distintas etapas de desarrollo del Proceso Unificado, viendo esta necesidad, se propone un conjunto de diagramas necesarios para cada etapa según la complejidad del Sistema de Información a solucionar.

Dado un sistema a desarrollar no es necesario emplear todos los diagramas; para sistemas sencillos un diagrama de clases junto con un par de diagramas de actividades e interacción sería suficiente, asimismo si los sistemas son complejos requieren de la utilización de más diagramas, debido a que requieren de etapas incrementales e iterativas(ciclos de desarrollo) en el análisis, diseño e implementación, por ello es que el conjunto actividades deberá especificar la etapa de desarrollo.

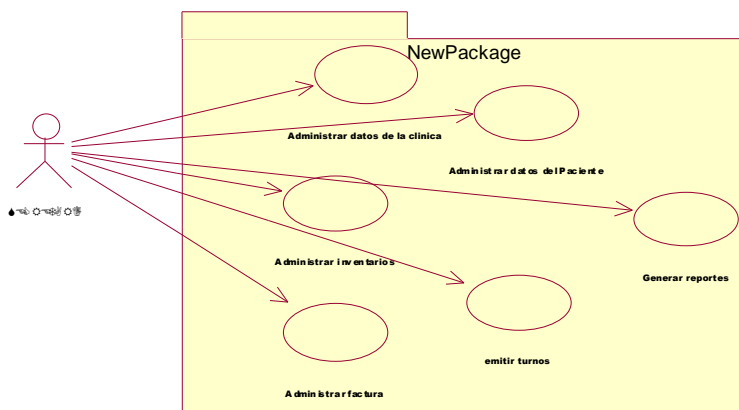
2.5.2 DIAGRAMA DE CASOS DE USO

Casos de uso es una técnica para capturar información de cómo un sistema o negocio trabaja, o de cómo se desea que trabaje. No pertenece estrictamente al enfoque orientado a objetos, es una técnica para captura de requisitos.

Actores

- Principales: personas que usan el sistema.
- Secundarios: personas que mantienen o administran el sistema.
- Material externo: dispositivos materiales imprescindibles que forman parte del ámbito de la aplicación y deben ser utilizados.
- Otros sistemas: sistemas con el que el sistema interactúa.

La misma persona física puede interpretar varios papeles como actores distintos, el nombre del actor describe el papel desempeñado.



2.5.2.1 TIPOS DE RELACIONES EN LOS DIAGRAMAS CASO DE USO

- UML define cuatro tipos de relación en los Diagramas de Casos de Uso:

Con formato: Numeración y viñetas

- Comunicación

Eliminado: -

- Inclusión: una instancia del Caso de Uso origen incluye también el comportamiento descrito por el Caso de Uso destino. <<include>> reemplazó al denominado <<uses>>

Con formato: Numeración y viñetas

Eliminado: -

- Extensión: el Caso de Uso origen extiende el comportamiento del Caso de Uso destino <<extend>>

Eliminado: -

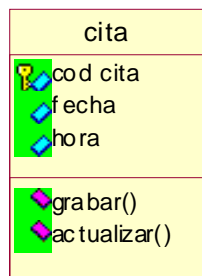
- Herencia: el Caso de Uso origen hereda la especificación del Caso de Uso destino y posiblemente la modifica y/o amplía.

Eliminado: -

2.5.3 DIAGRAMA DE CLASES

El diagrama de clases es el diagrama principal para el análisis y diseño. Un diagrama de clases presenta las clases del sistema con sus relaciones estructurales y de herencia. La definición de clase incluye definiciones para atributos y operaciones. El modelo de casos de uso aporta información para establecer las clases, objetos, atributos y operaciones. El mundo real puede ser visto desde abstracciones diferentes (subjetividad).

Cada clase se representa en un rectángulo con tres compartimientos:



Los atributos de una clase no deberían ser manipulables directamente por el resto de objetos. Por esta razón se crearon niveles de visibilidad para elementos que son:

(-) Privado.- Es el más fuerte. Esta parte es totalmente invisible (excepto para clases friends en terminología C++)

(#) Los atributos / operaciones protegidos están visibles para las clases friends y para las clases derivadas de la original.

(+) Los atributos / operaciones públicos son visibles a otras clases (cuando se trata de atributos se está transgrediendo el principio de encapsulación)

Relaciones entre clases

Los enlaces entre objetos pueden representarse entre las respectivas clases y sus formas de relación son:

Asociación y agregación (vista como un caso particular de asociación)

Generalización / Especialización

Las relaciones de Agregación y Generalización forman jerarquías de clases.

Eliminado: ¶

2.5.3.1 RELACION ENTRE CLASES

2.5.3.1.1 ASOCIACIÓN

La asociación expresa una conexión bidireccional entre objetos. Una asociación es una abstracción de la relación existente en los enlaces entre objetos. Puede determinarse por la especificación de multiplicidad (mínima... Máxima)



Eliminado:

Uno y sólo uno

0..1 cero o uno

m..n desde m hasta n (enteros naturales)

cero o muchos

0..* cero o muchos

1..* uno o muchos (al menos uno)

2.5.3.1.2 AGREGACIÓN

Eliminado: ¶
¶
¶

La agregación representa una relación parte de entre objetos. En UML se proporciona una escasa caracterización de la agregación. Esta relación puede ser caracterizada con precisión determinando las relaciones de comportamiento y estructura que existen entre el objeto agregado y cada uno de sus objetos componentes.

Eliminado: _

Una agregación se podría caracterizar según.

Puede el objeto parte comunicarse directamente con objetos externos al objeto agregado?

No => inclusiva

Si => no inclusiva

Puede cambiar la composición del objeto agregado?

Si => dinámica

No => estática

2.5.3.1.3 GENERALIZACIÓN

Diagrama de Clases y Diagrama de Objetos pertenecen a dos vistas complementarias del modelo. Un Diagrama de Clases muestra la abstracción de una parte del dominio. Un Diagrama de Objetos representa una situación concreta del dominio. Las clases abstractas no son instanciadas.

2.5.4 DIAGRAMA DE SECUENCIA

El Diagrama de Secuencia, representa la forma de cómo un cliente (actor) u objetos (clases) se comunican entre si en petición a un evento. Esto implica recorrer todo la secuencia de llamadas, de donde se obtienen las responsabilidades claramente.

Dicho diagrama puede ser obtenido de dos partes, desde el Diagrama Estático de clases o el de Casos de Uso (son diferentes).

Los componentes de un Diagrama de Secuencia son:

Eliminado: ¶

Un objeto o actor

Mensaje de un objeto a otro objeto

Mensaje de objeto a si mismo

Elementos

Objeto / actor

El rectángulo representa una instancia de un objeto en particular y la línea punteada representa las llamadas a métodos del objeto.

Mensaje a otro objeto

Se representa por una flecha entre un objeto y otro, representa la llamada de un método (operación) de un objeto en particular.

Mensaje al mismo objeto

No solo llamadas a métodos de objetos externos pueden realizarse, también es posible visualizar llamadas a métodos desde el mismo objeto en estudio

2.5.5 DIAGRAMA DE COLABORACIÓN

Son útiles en la fase exploratoria para identificar objetos.

La distribución de los objetos en el diagrama permite observar adecuadamente la interacción de un objeto respecto de los demás

La estructura estática viene dada por los enlaces, la dinámica por el envío de mensajes por los enlaces

2.5.6 DIAGRAMA DE ACTIVIDADES

El Diagrama de Actividad es una especialización del diagrama de Estado, organizado respecto de las acciones y usado para especificar:

- Un método
- Un caso de uso
- Un proceso de negocio

Con formato: Numeración y viñetas

Un estado de actividad representa una actividad: un paso en el flujo de trabajo o la ejecución de una operación. Un grafo de actividades describe grupos secuenciales y concurrentes de actividades. Los grafos de actividades se muestran en diagramas de actividades. Las actividades se enlazan por transiciones automáticas. Cuando una actividad termina se desencadena el paso a la siguiente actividad.

Un diagrama de actividades es provechoso para entender el comportamiento de alto nivel de la ejecución de un sistema, sin profundizar en los detalles internos de los mensajes. Los parámetros de entrada y salida de una acción se pueden mostrar usando las relaciones de flujo que conectan la acción y un estado de flujo de objeto.

2.5.7 DIAGRAMA DE ESTADOS

Muestra el conjunto de estados por los cuales pasa un objeto durante su vida en una aplicación, junto con los cambios que permiten pasar de un estado a otro. Los diagramas de Estado representan autómatas de estados finitos, desde el p.d.v de los estados y las transiciones. Son útiles sólo para los objetos con un comportamiento significativo. Cada objeto está en un estado en cierto instante. El estado está caracterizado parcialmente por los valores de algunos de los atributos del objeto. El estado en el que se encuentra u objeto determina su comportamiento. Cada objeto sigue el comportamiento descrito en el Diagrama de Estados asociado a su clase.

Los Diagramas de Estados y escenarios son complementarios, los Diagramas de Estados son autómatas jerárquicos que permiten expresar concurrencia, sincronización y jerarquía de objetos, son grafos dirigidos y deterministas. La transición entre estados es instantánea y se debe a la ocurrencia de un evento.

2.5.8 IMPLEMENTACIÓN DEL SISTEMA

Durante la implementación se codifican, tanto las estructuras en el dominio de la solución.

La base que la sustenta es el diseño de objetos.

El código puede ser una simple transición de las decisiones de diseño a las características propias de un lenguaje.

En esta parte se discuten los detalles específicos de la implementación de un sistema utilizando lenguajes orientados a objetos, lenguajes no orientados a objetos y sistemas de administración de bases de datos.

Para implementar un diseño orientado a objetos es utilizado un lenguaje orientado a objetos ya que estos lenguajes ofrecen distintos grados de apoyo para los conceptos orientados a objetos.

2.5.9 PRUEBAS

Es una actividad para determinar si el sistema está siendo construido correctamente. Tanto la implementación como las pruebas son dos etapas que están involucradas durante el análisis y diseño.

2.5.9.1 Prueba Funcional

El objetivo de estas pruebas es garantizar que el sistema ha sido desarrollado correctamente, sin errores de diseño y/o programación. Es conveniente que sean planteadas, al menos, tanto a nivel de cada módulo (aislado del resto), como de integración del sistema (según sea la naturaleza del proyecto en cuestión se podrán tener en cuenta pruebas adicionales, por ejemplo de rendimiento).

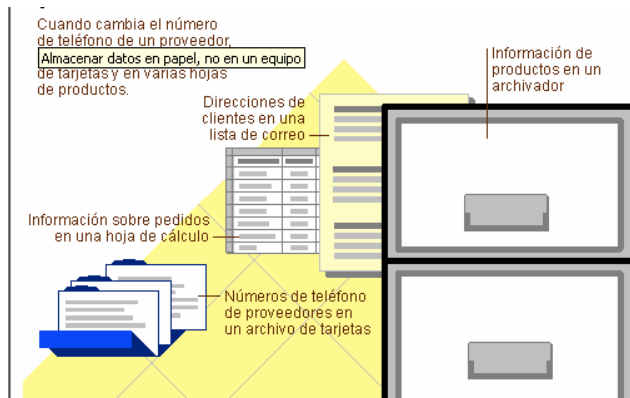
▼ **Eliminado:** Para este sistema que utilizara la prueba funcional

▼ **Eliminado: más**

La palabra "Basic" hace referencia al lenguaje BASIC (Beginners All-Purpose Symbolic Instruction Code), un lenguaje utilizado por más programadores que ningún otro lenguaje en la historia de la informática o computación. Visual Basic ha evolucionado a partir del lenguaje Basic original y ahora contiene centenares de instrucciones, funciones y palabras clave, muchas de las cuales están directamente relacionadas con la interfaz gráfica de Windows.

3.3 BASES DE DATOS ACCSESS

Una base de datos es una recopilación de información relativa a un asunto o propósito particular, como el seguimiento de pedidos de clientes o el mantenimiento de una colección de música. Si la base de datos no está almacenada en un equipo, o sólo están instaladas partes de la misma, puede que deba hacer un seguimiento de información procedente de varias fuentes en orden a coordinar y organizar la base de datos.



Por medio de Microsoft Access, puede administrar toda la información desde un único archivo de base de datos. Dentro del archivo, puede dividir los datos en contenedores de almacenamiento independientes denominados tablas; puede ver, agregar y actualizar datos de la tabla por medio de formularios en pantalla; puede asimismo buscar y recuperar sólo los datos que desee por medio de consultas; y puede analizar o imprimir datos con un diseño específico por medio de informes.

Puede permitir a otros usuarios ver, actualizar o analizar datos de la base de datos desde Internet o desde una intranet por medio de páginas de acceso a datos.

Almacene datos de una sola vez en una tabla, pero véalos desde varias ubicaciones. Cuando actualice los datos, éstos se actualizarán automáticamente en todos los lugares en los que aparezcan.

Id. de cliente	Nombre de compañía	Ciudad
BSBEV	B's Beverages	Londres
EASTC	Eastern Connection	Londres

Nombre de compañía	Ciudad	Id. de pedido	Fecha de entrega
B's Beverages	Londres	10943	05-Abr-96
Eastern Connection	Londres	10987	25-Abr-96

Un formulario, un informe, una consulta y una página de acceso a datos mostrando datos de la misma tabla

Id. de cliente: BSBEV

Nombre del contacto: Victoria Ashwort

Nombre de compañía: B's Beverages

Cliente: Around the Horn

Cliente: B's Beverages

Id. de pedido	Total
10288	479.00 \$

3.4 RATIONAL ROSE 2000.-

El navegador UML de Rational Rose nos permite establecer una trazabilidad real entre el modelo (análisis y diseño) y el código ejecutable. Facilita el desarrollo de un proceso cooperativo en el que todos los agentes tienen sus propias vistas de información (vista de Casos de Uso, vista Lógica, vista de Componentes y vista de Despliegue), pero utilizan un lenguaje común para comprender y comunicar la estructura y la funcionalidad del sistema en construcción.

3.5 CLIENTE SERVIDOR

El sistema se desarrollará con una arquitectura Cliente / Servidor que es un modelo para el desarrollo de sistemas de información en el que las transacciones se dividen en procesos independientes que cooperan entre sí para intercambiar información, servicios o recursos.

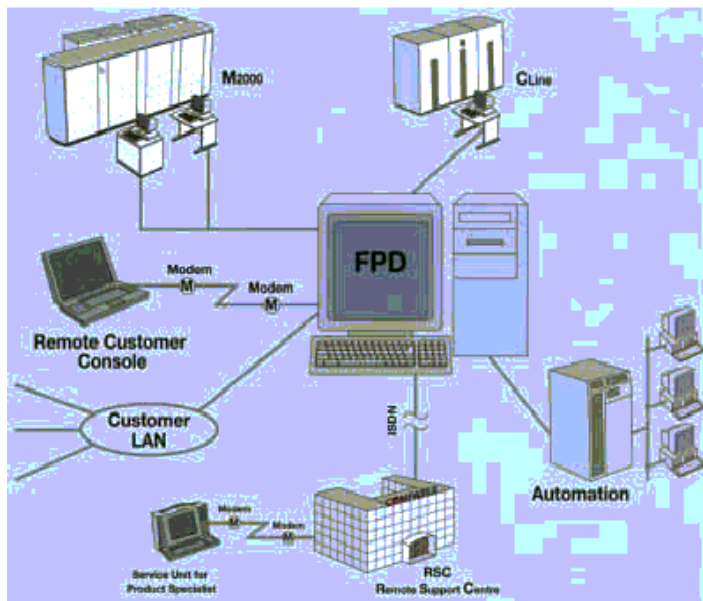
En este modelo las aplicaciones se dividen de forma que el servidor contiene la parte que debe ser compartida por varios usuarios, y en el cliente permanece solo lo particular de cada usuario, las ventajas que presenta este modelo es: aumento de productividad, mejora el rendimiento de la red, menores costos de operación.

3.5.1 CARACTERÍSTICAS DEL MODELO CLIENTE SERVIDOR

En el modelo CLIENTE/SERVIDOR podemos encontrar las siguientes características:

1. El Cliente y el Servidor pueden actuar como una sola entidad y también pueden actuar como entidades separadas, realizando actividades o tareas independientes.

Las funciones de Cliente y Servidor pueden estar en plataformas separadas, o en la misma plataforma.



Un servidor da servicio a múltiples clientes en forma concurrente.

4. Cada plataforma puede ser escalable independientemente. Los cambios realizados en las plataformas de los Clientes o de los Servidores, ya sean por actualización o por reemplazo tecnológico, se realizan de una manera transparente para el usuario final.

La interrelación entre el hardware y el software están basados en una infraestructura poderosa, de tal forma que el acceso a los recursos de la red no muestra la complejidad de los diferentes tipos de formatos de datos y de los protocolos.

Un sistema de servidores realiza múltiples funciones al mismo tiempo que presenta una imagen de un solo sistema a las estaciones Clientes. Esto se logra combinando los recursos de cómputo que se encuentran físicamente separados en un solo sistema lógico, proporcionando de esta manera el servicio más efectivo para el usuario final.

También es importante hacer notar que las funciones Cliente/Servidor pueden ser dinámicas. Ejemplo, un servidor puede convertirse en cliente cuando realiza la solicitud de servicios a otras plataformas dentro de la red.

Su capacidad para permitir integrar los equipos ya existentes en una organización, dentro de una arquitectura informática descentralizada y heterogénea .

7. Además se constituye como el nexo de unión mas adecuado para reconciliar los sistemas de información basados en mainframes o minicomputadores, con aquellos otros sustentados en entornos informáticos pequeños y estaciones de trabajo.

8. Designa un modelo de construcción de sistemas informáticos de carácter distribuido.

Su representación típica es un centro de trabajo (PC), en donde el usuario dispone de sus propias aplicaciones de oficina y sus propias bases de datos, sin dependencia directa del sistema central de información de la organización, al tiempo que puede acceder a los

recursos de este host central y otros sistemas de la organización ponen a su servicio.

En conclusión, Cliente/Servidor puede incluir múltiples plataformas, bases de datos, redes y sistemas operativos. Estos pueden ser de distintos proveedores, en arquitecturas propietarias y no propietarias y funcionando todos al mismo tiempo. Por lo tanto, su implantación involucra diferentes tipos de estándares: APPC, TCP/IP, OSI, NFS, DRDA corriendo sobre DOS, OS/2, Windows o PC UNIX, en TokenRing, Ethernet, FDDI o medio coaxial, sólo por mencionar algunas de las posibilidades.

3.5.1.1 VENTAJAS CLIENTE SERVIDOR

Costos. El enfoque cliente/servidor es económico, sobre todo cuando está unido al concepto de racionalización. Los costos de compra, arrendamiento y mantenimiento de macrocomputadoras centrales son tal elevados que los correspondientes a la compra de servidores, PCs y demás componentes para crear una red de área local parecen ridículos. Con frecuencia sucede que el costo de un sistema cliente/servidor completo es inferior al de la instalación de una computadora central para que pueda procesar una nueva aplicación.

Acceso a la información. Si bien el acceso a los datos es posible por otros medios, la arquitectura Cliente/Servidor constituye el ambiente ideal para facilitar el acceso a la información. El usuario no sólo puede tener un acceso transparente a toda la información que necesita, sino además está habilitado para procesarla como guste.

Ergonomía. Un buen sistema Cliente/Servidor no se concibe sin una interfaz gráfica de usuario y sin una transparencia total. De esta manera, el cliente puede trabajar en el ambiente que más le convenga sin preocuparse de la conversión de interfaces y protocolos. Se concentra en el trabajo que debe realizar más que en la tecnología.

Buena tecnología en el lugar adecuado. En teoría, un ambiente Cliente/Servidor puede conformarse de varias plataformas, sistemas operativos, SGBD, etc. De ahí que haya muchas opciones para cada aplicación; por ejemplo HP con UNIX y con estaciones de trabajo Macintosh para un servicio y un servidor Netware y estaciones de trabajo Windows para otro.

Modularidad. En un ambiente Cliente/Servidor, es factible agregar o eliminar estaciones de trabajo y servidores, puesto que el sistema puede ser más o menos fácil de volver a configurar. También es más sencillo implantar una aplicación nueva por medio de construir una extensión con un servidor y estaciones de trabajos nuevos.

3.5.1.2 DESVENTAJAS CLIENTE SERVIDOR

Incompatibilidad. El ambiente Cliente/Servidor supone que la época en que IBM tenía todo el mercado dominado ha concluido. Con el fin de esta etapa, se debe recurrir a varios proveedores. Todos sabemos lo que sucede en estos casos: cuando hay algún problema, el proveedor inicial lo remite a otro proveedor.

Si las especificaciones se ponen por escrito, no hay problema; pero en la práctica cotidiana, las incompatibilidades mayores o menores entre computadoras, sistemas operativos, lenguajes, protocolos, interfaces y programas de aplicación superan las expectativas. Cuanto más elevado es el número de proveedores, las incompatibilidades son mayores.

Capacitación. En casi todos los casos de implantación del modelo Cliente/Servidor, la principal dificultad es la capacitación de los usuarios. No se trata de sólo impartir cursos a los usuarios y a los ingenieros en computación, sino de cambiar toda una cultura, cual es más complicado y costoso. Es necesario redefinir todas las funciones computacionales; la visión de los sistemas de cómputo se debe cambiar por completo, la polivalencia debe ubicarse en el primer plano así como cambiar desarrollar nuevos expertos. El costo de una capacitación a fondo puede ser superior a la del conjunto del sistema. Sin embargo, se debemos considerar que la capacitación es una inversión a largo plazo.

Costos. Si bien el costo es uno de los principales factores que inclinan la balanza en favor de la arquitectura Cliente/Servidor, los inconvenientes antes mencionados conducen a reflexionar sobre la variedad de costos ocultos que conlleva: capacitación, solución de montones de pequeños problemas imprevistos, el tiempo perdido en reconciliarse con los proveedores, la reorganización y el desarrollo de aplicaciones que aún no se encuentran en el mercado.

La implantación del modelo Cliente/Servidor comprende varios elementos. En primer lugar, se debe contar con una arquitectura completa de telecomunicación; es decir, no basta tener un protocolo de comunicación común entre todos los sistemas llamados a cooperar, sino también se necesita toda una serie de funciones o aplicaciones de telecomunicación para retomar la terminología del modelo OSI de interconexión de sistemas abiertos.

En efecto, es necesario disponer de funciones como administración de archivos en red, subordinación de trabajos, mensajería, comunicación entre otros

CAPITULO IV

CONCLUSIONES Y RECOMENDACIONES

4.1 CONCLUSIONES

La metodología utilizada permitió conocer mas a fondo el modelo y los procesos en forma detallada del sistema, logrando de esta manera tener en claro el alcance y los requerimientos del mismo.

Eliminado: ¶
¶

Es importante definir las clases y objetos que intervienen en el desarrollo del sistema y su relación estructural entre cada uno de estos ya que con estos modelos podemos analizar la relación que tiene la estructura de una clase y los posibles cambios que pueden existir al momento de hacer variaciones en el diseño original.

Al utilizar diagramas UML obtenemos múltiples perspectivas del sistema, el modelo integra estas perspectivas de manera que podemos construir un sistema optimo y auto consistente.

La Ingeniería del Software se ha convertido en una disciplina, cuyo estudio es esencial para obtener una sólida base teórica y práctica que han permitido aplicar en el transcurso de todo desarrollo del sistema.

Eliminado: en la construcción del software de calidad que se está demandando hoy en día. ¶

EL cristal report 7.0 es una herramienta muy versátil para manejar subreportes, grupos, fórmulas y parámetros que en momento del diseño es muy necesario.

Con la elaboración del presente proyecto de titulación hemos puesto en práctica los conocimientos recopilados en las aulas de clase, así como también de la experiencia real en el desarrollo del sistema de información.

4.2 RECOMENDACIONES

En un tiempo prudencial (1 año), se realice un mantenimiento general del sistema con el propósito de evitar problemas referente a la información.

Se recomienda implementar un módulo de contabilidad para llevar un balance de pérdidas, ganancias y utilidades del la clínica dental.

Sacar copias de seguridad de la base de datos **cada cierto tiempo** dependiendo de la cantidad de información acumulada.

Desarrollar reportes mensuales y anuales de servicio prestados en la que muestre la información de manera gráfica con el propósito de mostrar al usuario datos confiables.

La persona que se va a encargar de la administración del sistema debe ser capacitado en su totalidad sobre el manejo del mismo, puesto que a la largo de este tiempo nos hemos podido dar cuenta que lleva tiempo a los usuarios el conocimiento y sobre todo el dominio del funcionamiento del sistema.

Implementar una página Web para que los clientes hagan sus reservaciones de citas por Internet.

4.3 BIBLIOGRAFÍA

TEXTOS

Rumbaugh, J y otros. (1999). "*Modelo y diseño Orientado a Objetos*". Metodología OMT. Santa Fé, Bogota: Prentice Hall.

Eliminado: ¶

Con formato: Numeración y viñetas

Bernal, Cesar A. (2000). "*Metodología de la Investigación*". Santa Fé, Bogota: Prentice Hall.

Con formato: Numeración y viñetas

JACOBSON Ivar, BOOCH Grady, RUMBAUGH James. (2000). "*El Proceso Unificado de desarrollo de software*"; Addison Wesley .

Con formato: Numeración y viñetas

Siler, Spotts. (1999). "*Edición Especial VISUAL BASIC 6*". Madrid, España: Prentice may.

Eliminado: ¶

Eliminado: Prentice Hall

Pressman, R. (4 Edición Año 1993). *Ingeniería de Software*. Madrid España: Mac Graw Hill.

Con formato: Numeración y viñetas

Pressman, Roger S. (1997). "*Ingeniería del Software. Un enfoque práctico*". McGraw-Hill. Mexico, D.F.

Eliminado: .

Eliminado: .

Ruble, D. "*Análisis y Diseño Practico Para Sistemas Cliente /Servidor Con Gui*". México: Prentice may.

Con formato: Numeración y viñetas

Eliminado: Hall

INTERNET

<http://www.vbsoftware.cl/manualvb.html>

Con formato: Numeración y viñetas

El Paradigma de la Orientación a Objetos

Eliminado:

Elementos de un modelo de objetos.

Eliminado:

<http://www.monografias.com/trabajos7/sql/sql.shtml>

Con formato: Numeración y viñetas

Eliminado:

Sentencias SQL en Visual Basic.

La ingeniería de software.

<http://www.angelfire.com/scifi/jzavalar/apuntes/IngSoftware.html#Cota1994>

Eliminado: -

Eliminado: software

Eliminado:

Eliminado: ¶

Arquitectura Cliente/Servidor .

http://dominiopublico.com/intranets/cliente_servidor.php

Eliminado: -

Eliminado:

Microsoft SQL Server – Características.

<http://www.microsoft.com/latam/sql/evaluation/features/default.asp>

Eliminado: -

Eliminado:

Microsoft Visual Basic..net.

<http://www.microsoft.com/catalog/display.asp?subid=45&site=730>

Eliminado: -

Eliminado: ¶

Eliminado:

SYBASE .

<http://www.uacam.mx/dgsc.nsf/pages/cadmpower>

Eliminado: -

Eliminado: ¶

Eliminado:

Ciclo de Vida del Software

http://alarcos.inf-cr.uclm.es/per/fgarcia/isoftware/doc/Tema3_1x1.pdf

Eliminado: -

Eliminado:

J. Rumbaugh. Rational Software Corporation, 1994,

[http:// www.rational.com](http://www.rational.com)

Eliminado: -

<http://www.inflexa.com/jsp/template.jsp?pag=uml2.htm&mnu=mnu-soluciones.jsp>

Eliminado: -

<http://www.monografias.com/trabajos16/lenguaje-modelado-unificado/lenguaje-modelado-unificado.shtml>

Eliminado: - .

Código de campo cambiado

Eliminado:

Eliminado: ¶

http://www.pcm.gob.pe/portal_ongei/publicaciones/cultura/Lib5038/indice.htm

Eliminado:

http://www.itlp.edu.mx/publica/revistas/revista_isc/anteriores/dic98/cliserv.html

Eliminado: -

Eliminado: -

http://www.elguille.info/colabora/puntoNET/jevergara_Multitier.htm

Eliminado: ¶

<http://www.microsoft.com/spanish/msdn/comunidad/mtj.net/voices/art140.asp>

¶

¶

¶

¶

▼

CAPITULO V

Eliminado: ¶

MANUAL TÉCNICO

Manual técnico

Identificación de Actores.....
Diagramas de caso de uso.....
Diagramas de caso de uso por actores.....
Diagramas de caso de uso por actividad.....
Diccionario de casos de uso.....
Diccionario del diagrama de casos de uso.....
Análisis Modelo Estático.....
Diagrama de clases.....
Diagrama de Objetos.....
Diccionario de clases.....
Diccionario del diagrama de clases.....
Diseño Modelo dinámico.....
Diagrama de interacción.....
Diagrama de Secuencia.....
Administración de Factura.....
Administración de citas.....
Diagrama de Colaboración.....
Administración de Factura.....
Administración de citas.....
Modelo funcional.....
Diagrama de actividades.....
Administración de Factura.....
Administración de citas.....
Construcción.....
Front End.....
Pruebas Funcionales.....

|

Eliminado: indice[]
1
1

ANÁLISIS

MODELO ESTATICO

Eliminado: ¶

Eliminado: ¶
Manual técnico¶

¶
Identificación de Actores.....

.....
¶
Diagramas de caso de uso.....

.....
Diagramas de caso de uso por actores.....

.....
Diagramas de caso de uso por actividad.....

.....
Diccionario de casos de uso.....

.....
Diccionario del diagrama de casos de uso.....

.....
Análisis Modelo Estático.....

.....
¶
Diagrama de clases.....

.....
¶
Diagrama de Objetos.....

.....
¶
Diccionario de clases.....

.....
¶
Diccionario del diagrama de clases.....

.....
¶
Diseño Modelo dinámico.....

.....
¶
Diagrama de interacción.....

.....
¶
Diagrama de Secuencia.....

.....
¶
Administración de Factura.....

.....
Administración de citas.....

.....
¶
Diagrama de Colaboración.....

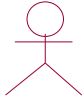
.....
¶
Administración de Factura.....

.....
Administración de citas.....

.....
¶
Modelo funcional.....

.....
¶
Diagrama de

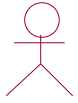
DESCRIPCIÓN DE ACTORES



Secretaria

Es la persona que se encarga de administrar los servicios de la clínica, como:

- Administrar datos de clínica dental
 - actualizar datos de la clínica
- Administra datos de pacientes
 - Ingresa datos del paciente
- Administrar Citas
 - Ingresa pedido de cita
- Administra facturas
 - Emite factura
- Genera reportes



Paciente

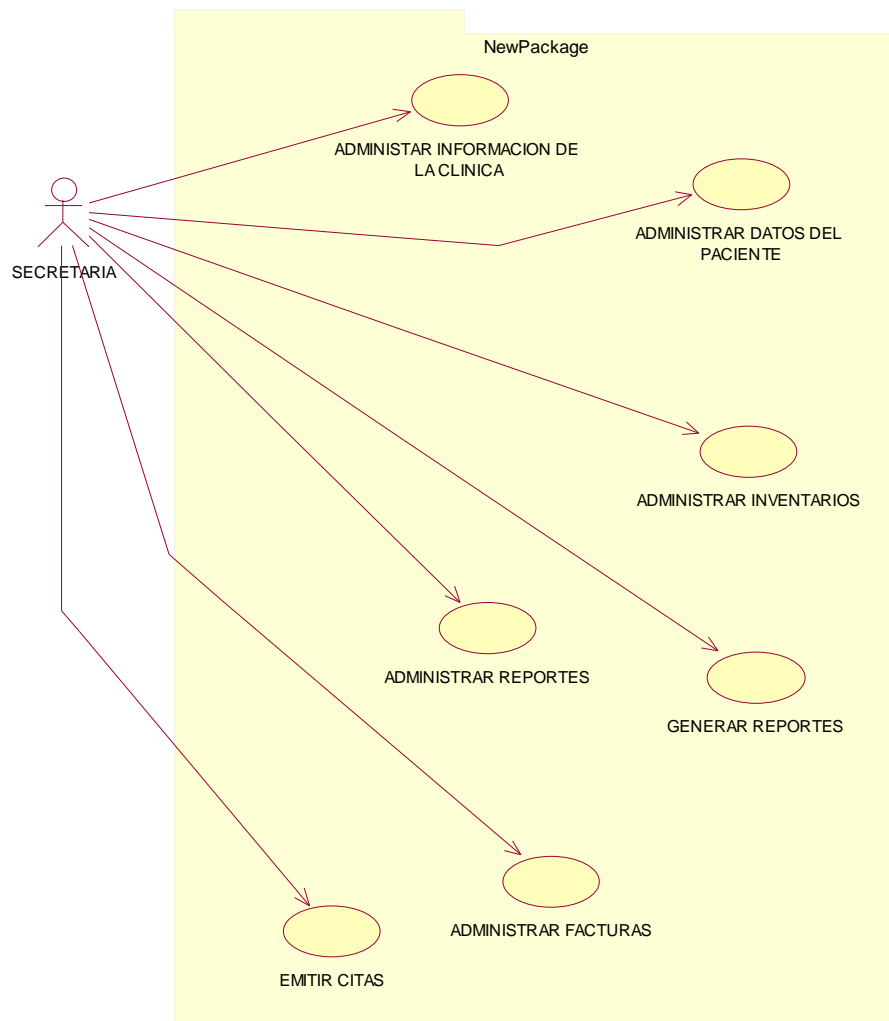
Es la persona que requiere la asistencias de la clínica y se encarga de facilitar los datos personales



Medico

El medico es el encargado de administrar los reportes generados así como también el que administra las historias clínicas

DIAGRAMAS DE CASOS DE USO POR ACTOR



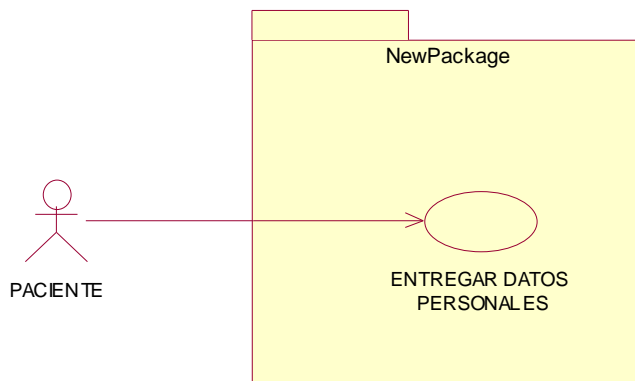
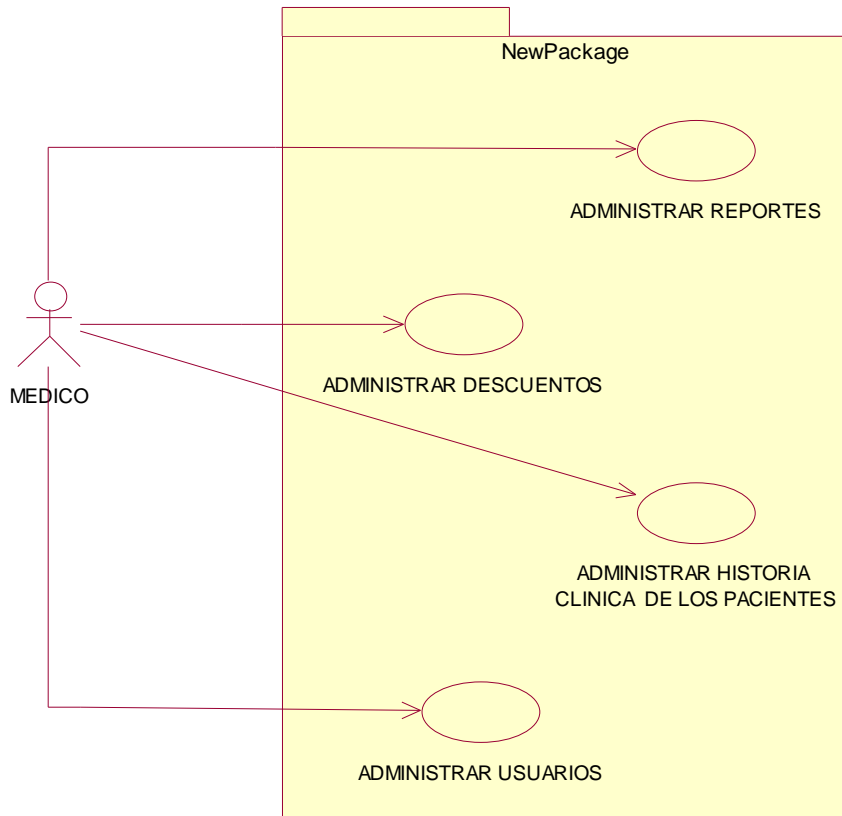
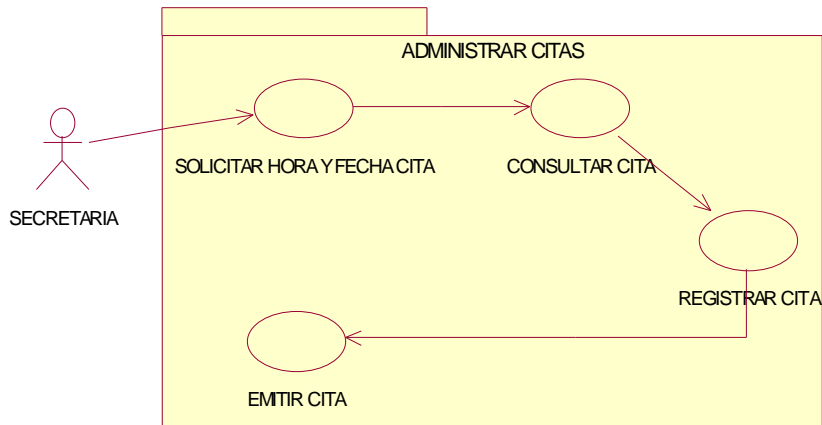
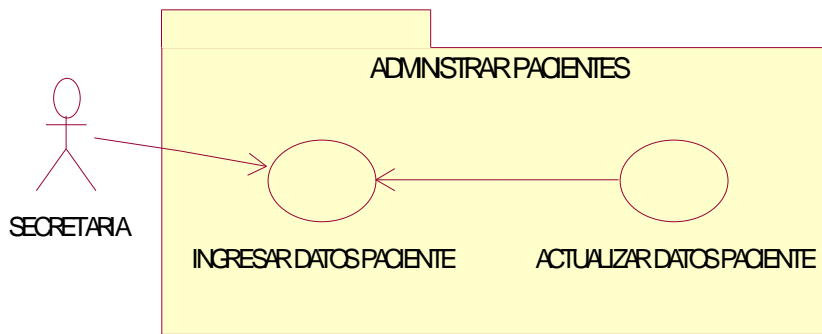
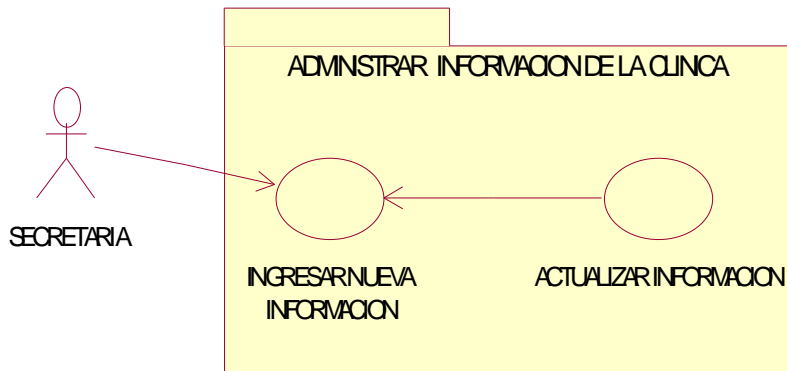
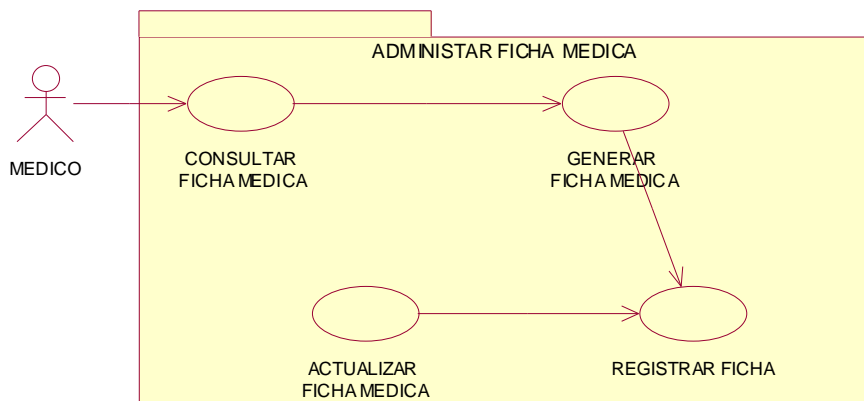
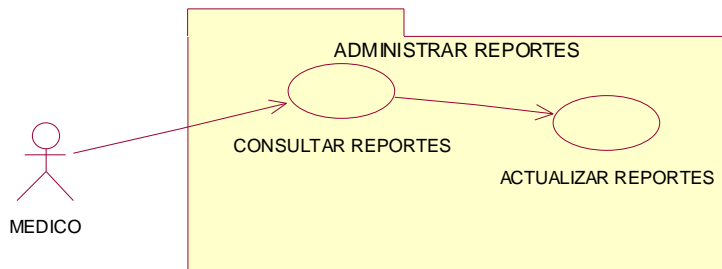
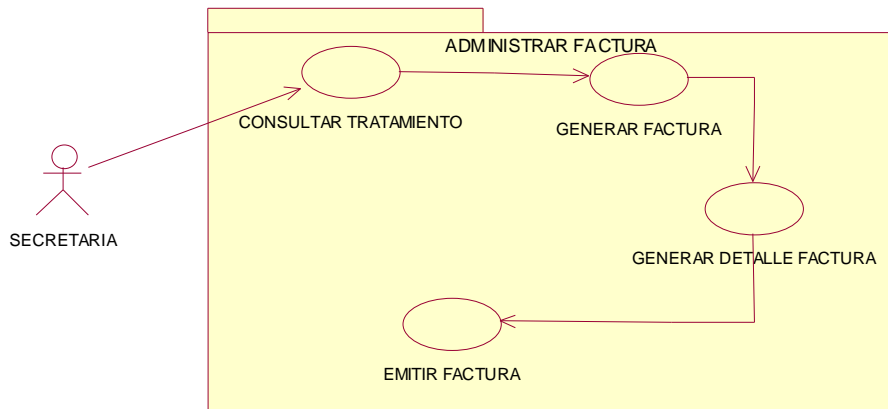
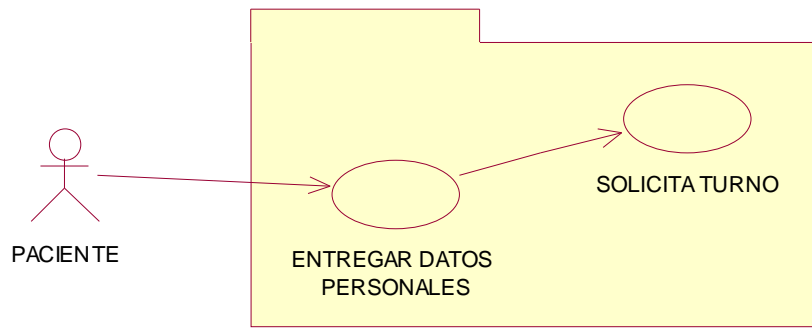


DIAGRAMA DE CASOS DE USO POR ACTIVIDADES







DICCIONARIO DE CASOS DE USO

DICCIONARIO DEL DIAGRAMA DE CASOS DE USO

CASO DE USO	DESCRIPCION
ADMINISTRAR DATOS CLINICA DENTAL	Administra los permisos y privilegios de cada uno de los usuarios de DentalSys
ACTUALIZAR DATOS DE USUARIOS	Ingresa los datos personales Un usuario nuevo y lo Actualiza
INGRESAR DATOS DEL PACIENTE	Ingresa los datos personales de cada paciente de la clínica dental
ACTUALIZAR LOS DATOS DEL PACIENTE	Ingresa y actualiza datos personales de un paciente nuevo de la clínica dental
SOLICITAR HORA FECHA CITA	Ingresa la hora y fecha exacta de la cita del paciente
CONSULTAR CITA	Se encarga de verificar si cita solicitada tiene disponibilidad
REGISTRAR CITA	Registra y actualiza la cita programada del paciente
EMITIR CITA	Emitir y actualiza las citas programadas
CONSULTAR TRATAMIENTO	Verifica el tipo de tratamiento realizado
CALCULAR COSTO	Realiza el calculo de costo según tratamiento
REGISTRAR FACTURA	Guarda y actualiza el registro de facturación
EMITIR FACTURA	Impresión de la factura para cada uno de los pacientes
CONSULTAR REPORTE	Consulta los reportes generados en la clínica
ACTUALIZAR REPORTE	Registra y actualiza los reportes generados
CONSULTAR FICHA MEDICA	Consulta la existencia del paciente

GENERAR FICHA MEDICA	Si el paciente es nuevo crea una ficha medica nueva
REGISTRAR FICHA MEDICA	registra la ficha medica del paciente
ACTUALIZAR FICHA MEDICA	Actualiza la ficha medica del paciente
ENTEGAR DATOS PERSONALES	Proveer todos los datos personales del paciente
SOLICITAR TURNO	solicita turno para atención medica odontológica

DIAGRAMA DE CLASES

DIAGRAMA DE CLASES

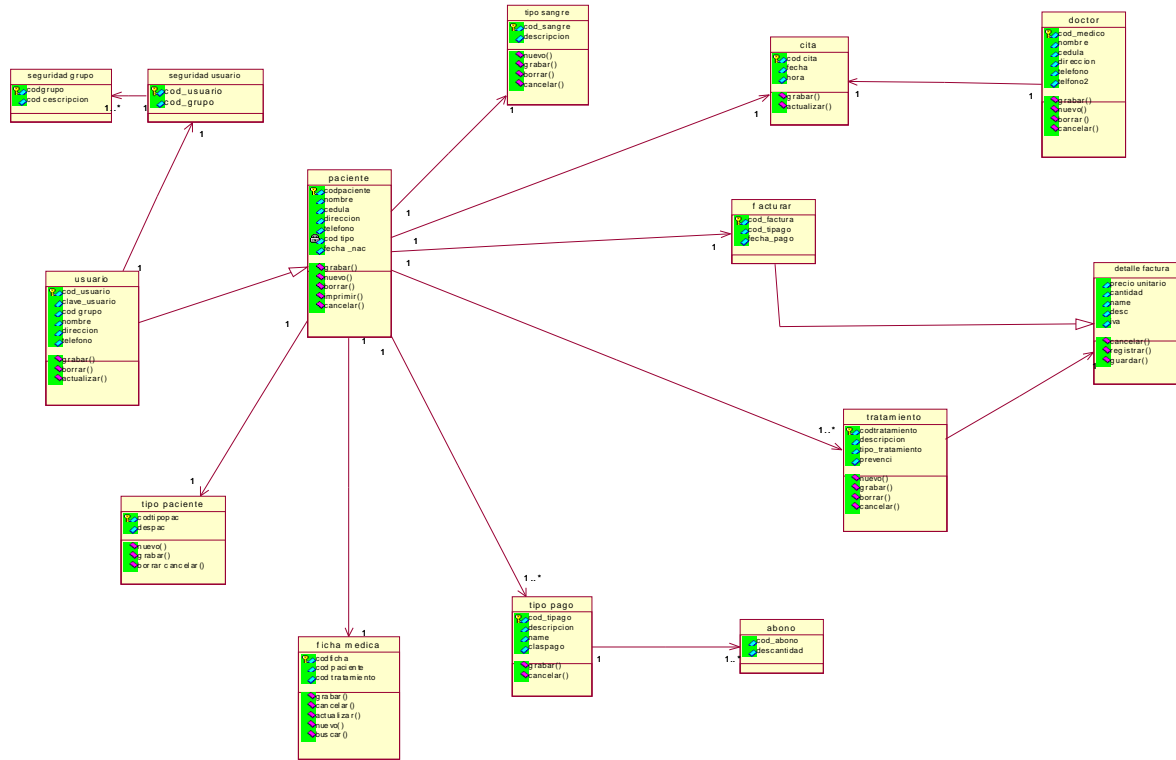
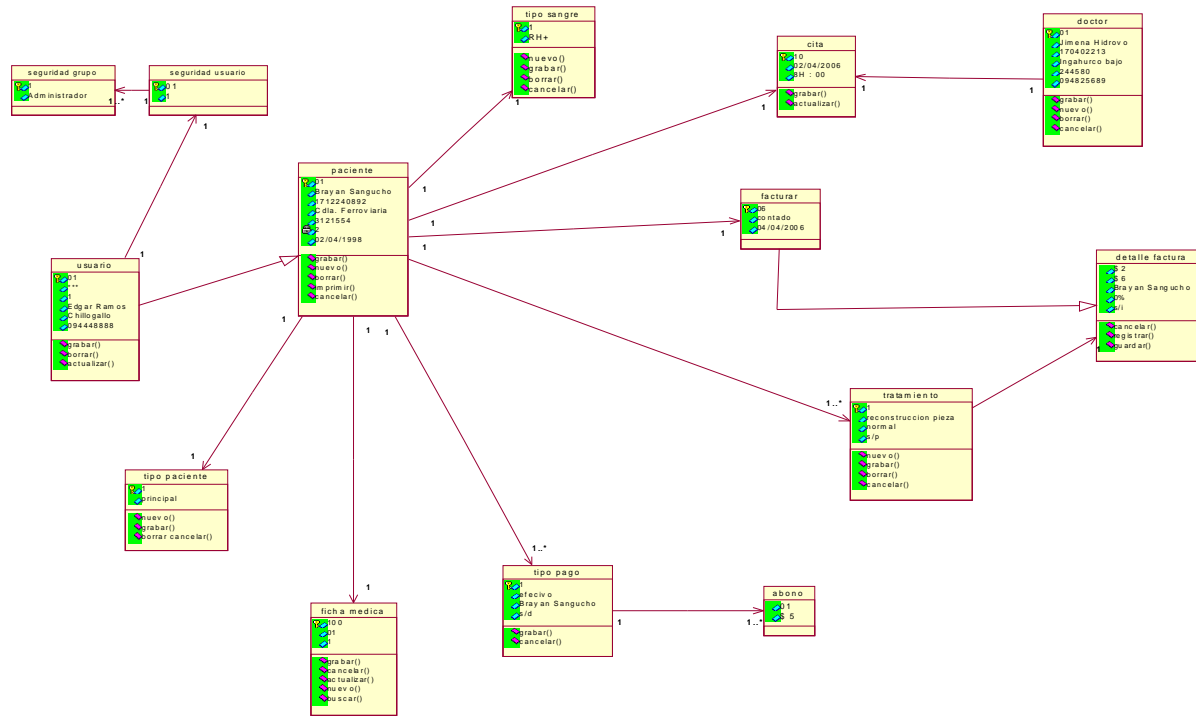


DIAGRAMA DE OBJETOS

DIAGRAMA DE OBJETOS



DICCIONARIO DE CLASES

DICCIONARIO DEL DIAGRAMA DE CLASES

MEDICO	Representa a las personas encargada de brindar atención profesional medica a lo pacientes.
PACIENTE	Representa a las personas que acuden al laboratorio clínico para realizarse algún tipo de tratamiento.
USUARIO	pertenece al grupo de secretarias administradores con los permisos propios para el uso del sistema.
FACTURA	Contiene los datos de la cabecera de una factura.
DET FACTURA	Contiene los datos de los tratamientos y el total de la factura.
COMPRA	Contiene los datos de la cabecera de un pedido de compra.
SEGURIDAD GRUPO	Contiene las seguridades de los grupos para asignar Usuarios
SEGURIDAD USUARIO	Contiene las seguridades de los usuarios encargados del manejo del sistema
TIPO SANGRE	Contiene el tipo e sangre de cada paciente
CITAS	Contiene las asignaciones de las citas para cada Paciente
TRATAMIENTO	registra el tipo de tratamiento ha realizarse en cada Paciente
FICHA MEDICA	Contiene los registros historiales de los pacientes de la Clínica dental
TIPO DE PAGO	Contiene la forma de pago que un cliente va ha realizar
ABONO	Registra los pagos secuenciales que el cliente va ha Realizar

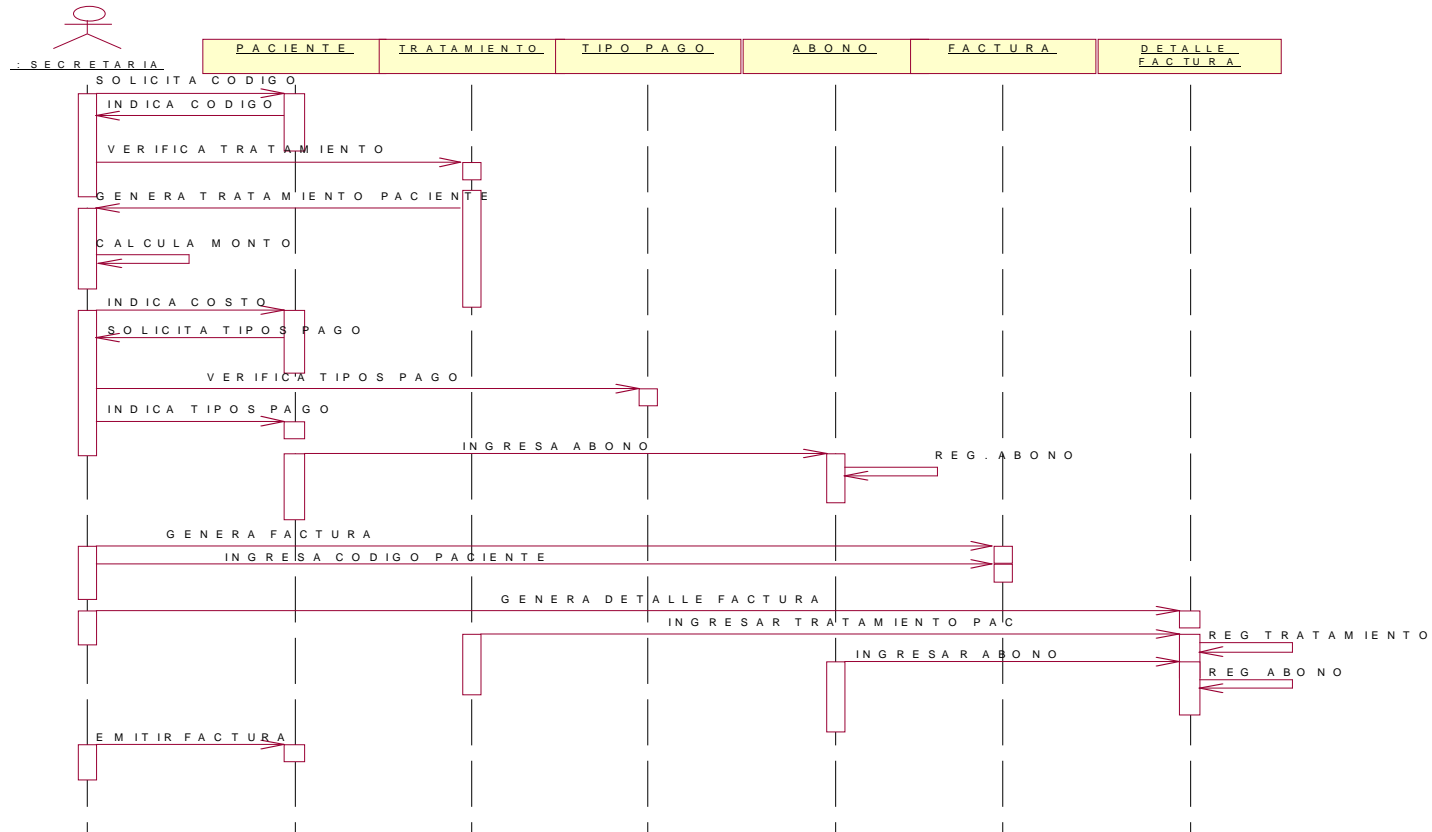
DISEÑO

Eliminado: ¶

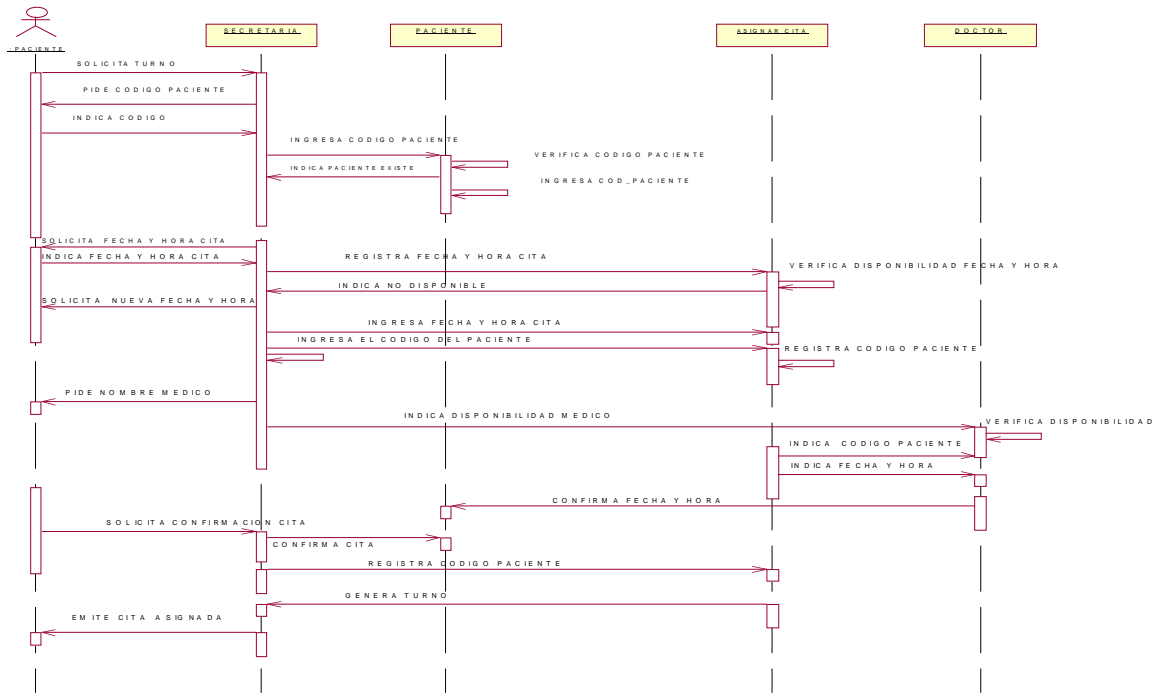
MODELO DINÁMICO

DIAGRAMAS DE SECUENCIA

ADMINISTRAR FACTURA

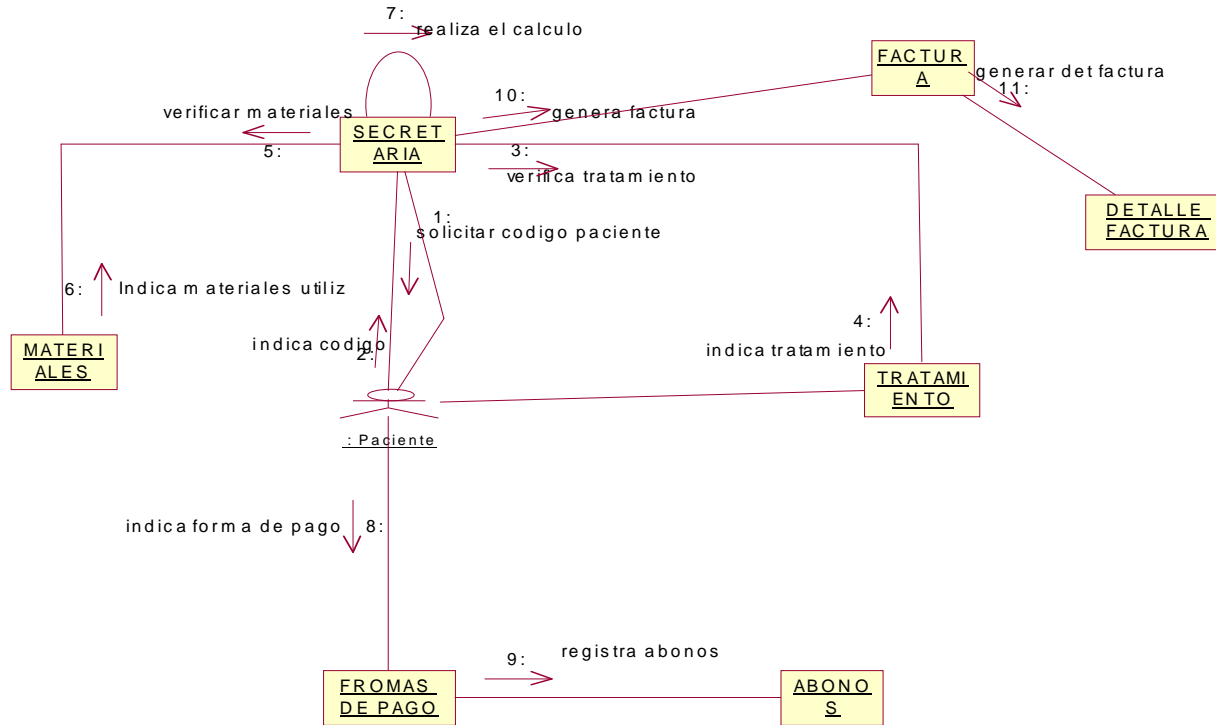


ADMINISTRAR CITA

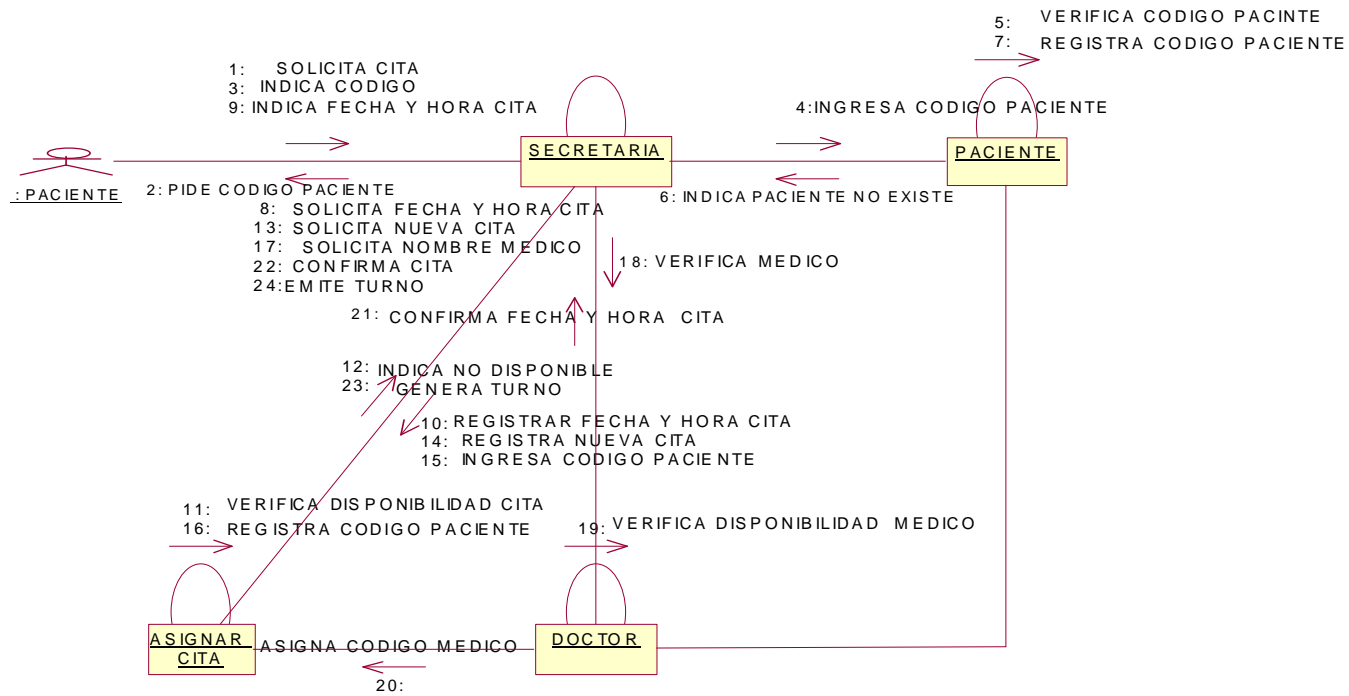


DIAGRAMAS DE COLABORACIÓN

ADMINISTRAR FACTURA



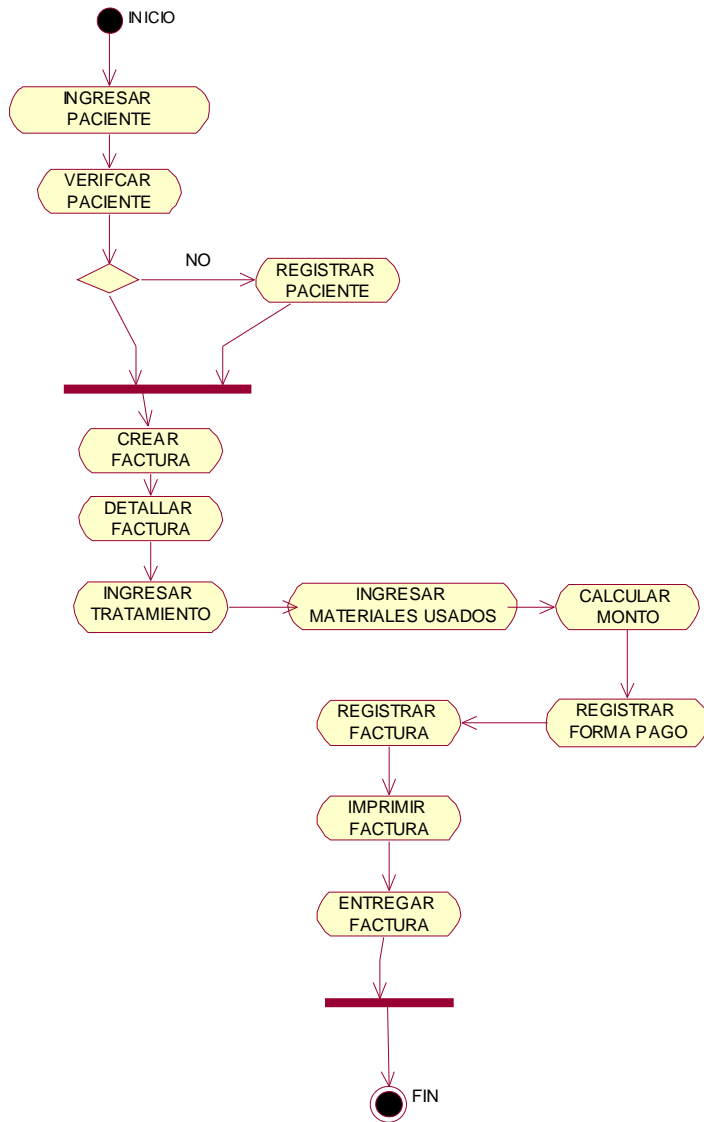
ADMINISTRAR CITA



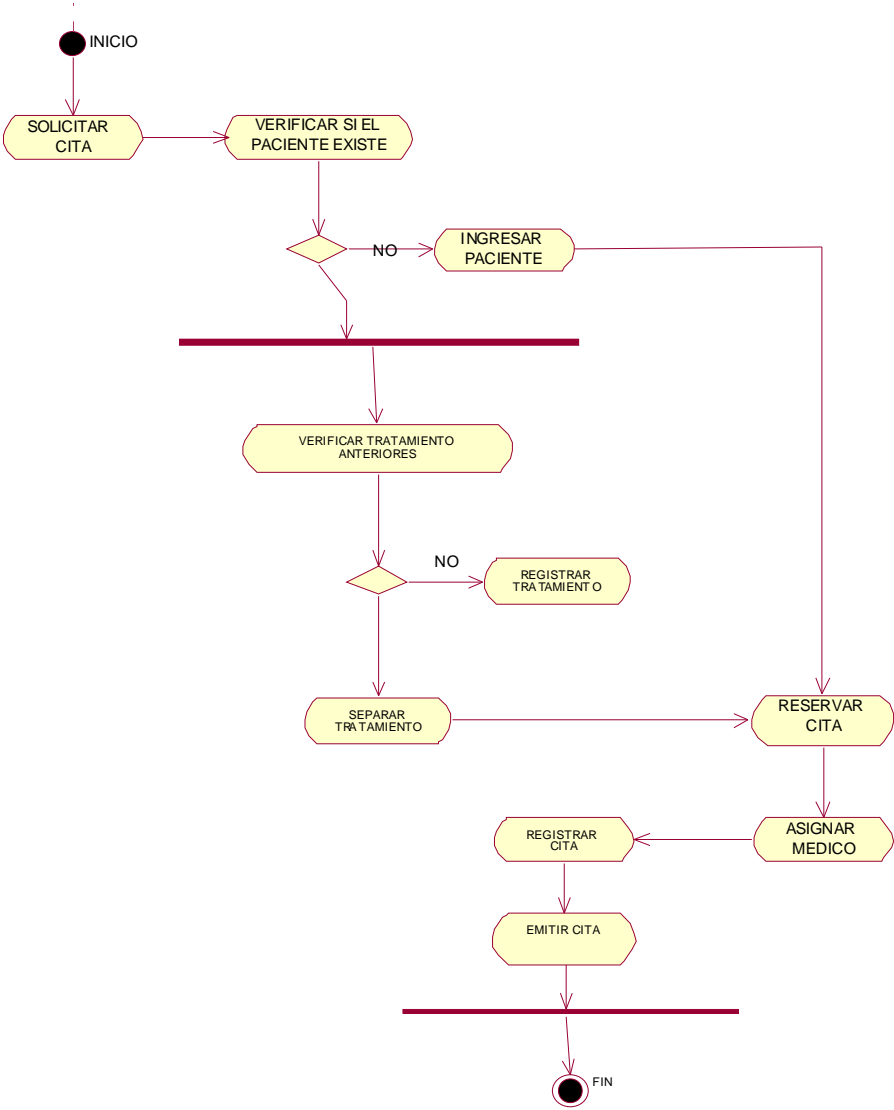
MODELO FUNCIONAL

DIAGRAMA DE ACTIVIDADES

ADMINISTRACIÓN DE FACTURA



ADMINISTRAR TURNO



CONSTRUCCIÓN

Código Emitir Factura

En la clase frmfactura de interfaz se hace las operaciones de la emisión de una factura al cliente por los servicios recibidos, esta Clase interactúa con la Clase Pacientes, las operaciones que se realiza en esta interfaz son imprimir factura actualizar

```
Dim numero_factura, Cliente
```

```
Private Sub Command1_Click()  
ABIERTO_abonos(3) = False  
Imp_Factura = False  
Unload Me  
End Sub
```

```
Private Sub Command2_Click()  
If Text23.Text <> frmabonos.Text2.Text Then  
MsgBox "No se puede Actualizar si ha Cambiado el Nombre", vbExclamation,  
Mensaje_Box  
Text23.SetFocus  
Exit Sub  
End If
```

```
If Text22.Text = "" Or Text22.Text = "No tiene" Or Len(Text22.Text) > 10 Then  
MsgBox "Ingrese el Numero de Cedula o Ruc del Cliente", vbExclamation,  
Mensaje_Box  
Text22.SetFocus  
Exit Sub  
End If
```

```
If Text24.Text = "" Or Text24.Text = "NINGUNA" Then  
MsgBox "Ingrese la dirección del Cliente", vbExclamation, Mensaje_Box  
Text24.SetFocus  
Exit Sub  
End If
```

```
If Len(Text25.Text) < 7 Then  
MsgBox "Ingrese un numero de telefonico de mal menos 7 digitos",  
vbExclamationApp, Mensaje_Box  
Text25.SetFocus  
End If
```

```
If MsgBox("Desea Actualizar la Información de " & frmabonos.Text2.Text,  
vbQuestion + vbYesNo, Mensaje_Box) = vbNo Then  
MsgBox "La información no ha sido Modificada", vbExclamation, Mensaje_Box  
Exit Sub
```

End If

```
act.ConnectionString = CONEXION
act.RecordSource = "select * from cliente where codcli=" &
frmabonos.Text1.Text & ""
act.Refresh
With act.Recordset
If .RecordCount > 0 Then
.MoveFirst
!dircli = Text24 & " "
!ruccli = Text22.Text
!telcli = "P593A02T" & Text25.Text
.Update
End If
End With
MsgBox "La informacion del cliente ha sido actualizada", vbExclamation,
Mensaje_Box
End Sub
```

```
Private Sub Command3_Click()
If frmabonos.Text47.Text = "si" Then
MsgBox "El Abono es con Tarjeta de Credito, debe facturarse", vbExclamation,
Mensaje_Box
Command3.Enabled = False
Exit Sub
End If
temp.ConnectionString = CONEXION
temp.RecordSource = "select * from id_factura where id_factura=" &
Label33.Caption & ""
temp.Refresh
With temp.Recordset
If .RecordCount > 0 Then
.MoveFirst
numero_factura = !id_factura - 1
!id_factura = numero_factura
.Update
End If
End With
Imp_Factura = False
ABIERTO_abonos(3) = False
Unload Me
```

End Sub

```
Private Sub Command5_Click()
Call Puntero(Me, 1)
Call Imprimir_Factura
'Call Log(3, "EMISION DE FACTURAS", "SE EMITE LA FACTURA N° " &
Label33.Caption)
```

```
Call Puntero(Me, 0)
End Sub
```

```
Public Function Imprimir_Factura()
On Error GoTo salto
temp.ConnectionString = CONEXION
temp.RecordSource = "select * from factura2"
temp.Refresh
```

```
If Text22.Text = "" Or Text22.Text = "No tiene" Or Text22.Text = "N/D" Then
MsgBox "Ingrese el # de Cedula o Ruc Valida y Actualize los Datos del
Cliente", vbExclamation, "Error"
Text22.SetFocus
Exit Function
End If
```

```
If Text25.Text = "" Or Text25.Text = "N/D" Then
MsgBox "Ingrese un numero telefonico", vbExclamation, Mensaje_Box
Text25.SetFocus
Exit Function
End If
```

```
If Text23.Text = "" Or Text23.Text = "FACTURA SERVICIOS" Then
MsgBox "Ingrese el nombre", vbExclamation, Mensaje_Box
Text23.SetFocus
Exit Function
End If
```

```
If Len(Text26.Text) = 0 Then
MsgBox "Ingrese el detalle de la factura", vbExclamation, Mensaje_Box
Text26.SetFocus
Exit Function
End If
```

```
If Text24.Text = "" Or Text24.Text = "NINGUNA" Or Text24.Text = "N/D" Then
MsgBox "Ingrese una Dirección Valida y Actualize los Datos del Cliente",
vbExclamation, Mensaje_Box
Text24.SetFocus
Exit Function
End If
Verifica_Factura:
Call Seleccion_Dao("SELECT VAL(ID_FACTURA) FROM FACTURA2 WHERE
ID_FACTURA=" & numero_factura & "'", c1, c2, c3, c4)
If Not Rs.EOF Then
```

```
Call Seleccion_Dao("SELECT VAL(ID_FACTURA) FROM FACTURA2 ORDER
BY VAL(ID_FACTURA) DESC", c1, c2, c3, c4)
numero_factura = c1 + 1
MsgBox "SE IMPRIMIRA LA FACTURA " & numero_factura, vbCritical,
Mensaje_Box
GoTo Verifica_Factura
End If
```

```
With temp.Recordset
.AddNew
!Id_abono = val(frmabonos.Text31.Text)
!id_cliente = frmabonos.Text1.Text
Cliente = frmabonos.Text1.Text
!id_factura = numero_factura
!ruc = Text22.Text
!telefono = Text25.Text
!Nombre = Text23.Text
!direccion = Text24.Text
!detalle = Text26.Text
!Valor = val(frmabonos.Text30.Text)
!fecha = Date
!Responsable = Responsable
!COUNTER = counterr
.Update
End With
```

```
temp.ConnectionString = CONEXION
temp.RecordSource = "select * from abono where id_abono=" &
val(frmabonos.Text31.Text)
temp.Refresh
With temp.Recordset
If .RecordCount > 0 Then
.MoveFirst
!id_factura = numero_factura
.Update
End If
End With
```

```
If frmabonos.Text49.Text = "si" Then
For h = 0 To 20
If Id_abonos(h) > 0 Then

temp.RecordSource = "select * from abono where id_abono=" &
Id_abonos(h)
temp.Refresh
With temp.Recordset
If .RecordCount > 0 Then
.MoveFirst
!id_factura = numero_factura
```

```
.Update  
End If  
End With
```

```
End If  
Next h  
End If
```

```
temp.ConnectionString = CONEXION  
temp.RecordSource = "select * from tarjeta where id_abono=" &  
val(frmabonos.Text31.Text)  
temp.Refresh
```

```
With temp.Recordset  
If .RecordCount > 0 Then  
.MoveFirst  
!detalle = Text26.Text  
.Update  
End If  
End With
```

```
temp.RecordSource = "select * from factura2 where id_factura=" &  
numero_factura & ""  
temp.Refresh
```

```
Call reporte
```

```
Frame8.Enabled = False
```

```
Command3.Enabled = False  
Command2.Enabled = False  
Command5.Enabled = False  
Command1.Enabled = True
```

```
Exit Function
```

```
salto:
```

```
Call Log(10, "ERROR EN LA FACTURACION", "SE PRODUCE UN ERROR AL  
GRABAR AL FACTURA " & Label33.Caption)
```

```
Resume Next
```

```
End Function
```

```
Public Function reporte()
```

```
On Error GoTo salir
```

```
Me.MousePointer = 11
```

```
temp.ConnectionString = CONEXION2
temp.RecordSource = "select * from rfactura"
temp.Refresh
```

```
With temp.Recordset
If .RecordCount > 0 Then
.MoveFirst
While Not .EOF
.Delete
.Update
.MoveNext
Wend
End If
.AddNew
!Id_abono = 1
!ruc = Text22.Text
!id_factura = numero_factura
!telefono = Text25.Text
!Nombre = Text23.Text
!direccion = Text24.Text
!detalle = Text26.Text
!Valor = val(frmabonos.Text30.Text)
a = ((" SON : " & UCase(Numero2Letra(Format(frmabonos.Text30.Text,
"0.00"), , 2, "Dolar", "Centavo", True, False))))
!Responsable = a
.Update
End With
```

```
Reporte10.Destination = crptToPrinter
Reporte10.Destination = crptToPrinter
MsgBox "Por favor prepare la Impresora de Facturacion", vbExclamation,
Mensaje_Box
Reporte10.Destination = crptToPrinter
Reporte10.ReportTitle = Format(Date, "long date")
Reporte10.Destination = crptToPrinter
Reporte10.ReportFileName = App.Path & "\Imprimir\Factura\rfactura.rpt"
Reporte10.Destination = crptToPrinter
Reporte10.DataFiles(0) = App.Path & "\datos\imprimir.mdb"
Reporte10.Destination = crptToPrinter
Reporte10.Action = 1
Imp_Factura = False
```

```
Me.MousePointer = 0
```

```
Exit Function
```



```
salir:  
Me.MousePointer = 0  
MsgBox Err.Description
```

```
End Function
```

```
Private Sub Form_Load()
```

```
On Error GoTo salir  
Cliente = frmabonos.Text1.Text  
Text23.Text = frmabonos.Text2.Text  
Text22.Text = frmabonos.Text27  
Text25.Text = frmabonos.Text29  
Text24.Text = frmabonos.Text28
```

```
Me.Top = 0  
Me.Left = 2700
```

```
temp.ConnectionString = CONEXION  
Call num_factura  
Call llena_default  
frmmanufactura.Show  
Text22.SetFocus  
frmabonos.Enabled = False  
If frmabonos.Text47.Text = "si" Then  
Command3.Enabled = False  
End If  
Command3.Enabled = False
```

```
If frmabonos.Text1.Text = "FACT" Then  
Command2.Enabled = False  
End If  
Exit Sub
```

```
salir:  
MsgBox Err.Description, vbCritical, Mensaje_Box  
MsgBox "Por favor escriba como ocurrio este error y comuniquese com  
Mvsystem's, gracias por su colaboración", vbExclamation, App.FileDescription  
& " " & App.Revision & Year(Date)  
End Sub
```

```
Private Sub Form_Unload(Cancel As Integer)
```

```
Variable = Label33.Caption
```

```
Unload frmabonos
```

```
frmabonos.Enabled = True
```

```
Mdiform10.Enabled = True
```

```
frmabonos.WindowState = 0
```

```
frmabonos.Show
```

```
For x = 0 To 20
```

```
  Id_abonos(x) = 0
```

```
Next x
```

```
frmabonos.Text49.Text = ""
```

```
If ABIERTO_abonos(6) = True Then
```

```
  NuevoCounter_1.ZOrder (1)
```

```
Elseif ABIERTO_abonos(7) = True Then
```

```
  NuevoCounter_2.ZOrder (1)
```

```
End If
```

```
frmabonos.Text1 = Cliente
```

```
frmabonos.Text1.SetFocus
```

```
frmabonos.Text3.SetFocus
```

```
End Sub
```

```
Private Sub Text22_Click()
```

```
  SendKeys "{Home}+{End}"
```

```
End Sub
```

```
Private Sub Text22_GotFocus()
```

```
  SendKeys "{Home}+{End}"
```

```
End Sub
```

```
Private Sub Text22_KeyPress(KeyAscii As Integer)
```

```
  If KeyAscii = 13 Then
```

```
    Text25.SetFocus
```

```
  Else
```

```
    KeyAscii = validar_enteros(KeyAscii)
```

```
  End If
```

```
End Sub
```

```
Private Sub Text22_LostFocus()
```

```
  a = Len(Text22.Text)
```

```
  If a <= 2 Then
```

```
    Text22.Text = frmabonos.Text27
```

```
  End If
```

```
End Sub
```

```
Private Sub Text23_Click()  
SendKeys "{Home}+{End}"  
End Sub
```

```
Private Sub Text23_GotFocus()  
SendKeys "{Home}+{End}"  
End Sub
```

```
Private Sub Text24_Click()  
SendKeys "{Home}+{End}"  
End Sub
```

```
Private Sub Text24_GotFocus()  
SendKeys "{Home}+{End}"  
End Sub
```

```
Private Sub Text25_Click()  
SendKeys "{Home}+{End}"  
End Sub
```

```
Private Sub Text25_GotFocus()  
SendKeys "{Home}+{End}"  
End Sub
```

```
Private Sub Text25_KeyPress(KeyAscii As Integer)  
If KeyAscii = 13 Then  
    Text23.SetFocus  
End If
```

```
End Sub
```

```
Private Sub Text25_LostFocus()  
a = Len(Text25.Text)  
If a <= 2 Then  
    Text25.Text = frmabonos.Text29  
End If  
End Sub
```

```
Private Sub Text24_KeyPress(KeyAscii As Integer)  
If KeyAscii = 13 Then  
    Text26.SetFocus  
Else  
    'KeyAscii = validar_letras(KeyAscii)  
End If  
End Sub
```

```
Private Sub Text24_LostFocus()  
a = Len(Text24.Text)  
If a <= 2 Then  
Text24.Text = frmabonos.Text28  
End If  
Text24.Text = UCase(Text24.Text)
```

```
End Sub
```

```
Public Function num_factura()  
If ABIERTO_abonos(3) = True Then  
Exit Function  
End If  
ABIERTO_abonos(3) = True  
On Error GoTo salir
```

```
'nuevo_numero:  
'temp.ConnectionString = CONEXION  
'temp.RecordSource = "select * from id_factura where counter=" & counterr &  
"'"  
  
'temp.Refresh  
'With temp.Recordset  
' If .RecordCount > 0 Then  
' If !id_factura = "N/D" Then  
' query = "select * from id_factura where counter='COUNTER #1"  
' temp.RecordSource = query  
' temp.Refresh  
' Command3.Enabled = False  
' End If  
' End If  
'End With
```

```
Call Seleccion_Dao("SELECT VAL(ID_FACTURA) FROM FACTURA2 ORDER  
BY VAL(ID_FACTURA) DESC", c1, c2, c3, c4)  
If IsNull(c1) Or c1 = "" Then  
numero_factura = InputBox("INGRESE EL NUMERO DE FACTURA A  
EMPEZAR", Mensaje_Box)  
Else  
numero_factura = c1 + 1  
End If  
'With temp.Recordset  
' If .RecordCount > 0 Then  
' .MoveFirst  
' numero_factura = !id_factura + 1  
' !id_factura = numero_factura  
' .Update  
' Else
```

```
' numero_factura = InputBox("Ingrese el Numero de Factura a Empezar",  
"Primer # de Factura para " & counterr, , 3500, 3500)  
' .AddNew  
' !id_factura = numero_factura  
' !COUNTER = counterr  
' .Update  
' End If  
' End With
```

```
Frame8.Visible = True  
Label33.Caption = numero_factura  
Exit Function  
salir:  
'GoTo nuevo_numero  
Call Log(10, "ERROR EN LA FACTURACION", "FACTURA # " & c1)  
End Function
```

```
Private Sub Text23_KeyPress(KeyAscii As Integer)  
If KeyAscii = 13 Then  
Text24.SetFocus  
Else  
KeyAscii = validar_letras(KeyAscii)  
End If  
End Sub
```

```
Private Sub Text23_LostFocus()  
a = Len(Text23.Text)  
If a <= 2 Then  
Text23.Text = frmabonos.Text2.Text  
End If  
Text23.Text = UCase(Text23.Text)  
End Sub
```

```
Private Sub Text26_Change()  
If Len(Text26.Text) > 0 Then  
Command5.Enabled = True  
Else  
Command5.Enabled = False  
End If  
End Sub
```

```
Private Sub Text26_Click()  
'SendKeys "{Home}+{End}"  
End Sub
```

```
Private Sub Text26_GotFocus()  
SendKeys "{Home}+{End}"  
End Sub
```

```
Private Sub Text26_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then
  If Command5.Enabled = True Then
    Command5.SetFocus
  End If
End If
End Sub
```

```
Public Function llen_a_default()
Text23.Text = frmabonos.Text50.Text
Text22.Text = frmabonos.Text27
Text25.Text = frmabonos.Text29
Text24.Text = frmabonos.Text28
```

```
End Function
```

```
Private Sub Text26_LostFocus()
Text26.Text = UCase(Text26.Text)
Text26.Text = Text26.Text & " "
End Sub
```

PRUEBAS FUNCIONALES

PRUEBAS FUNCIONALES

Las pruebas funcionales nos ayuda a determinar si el sistema está siendo construido correctamente. Tanto la implementación como las pruebas son dos etapas que están involucradas durante el análisis y diseño.

Objetivos

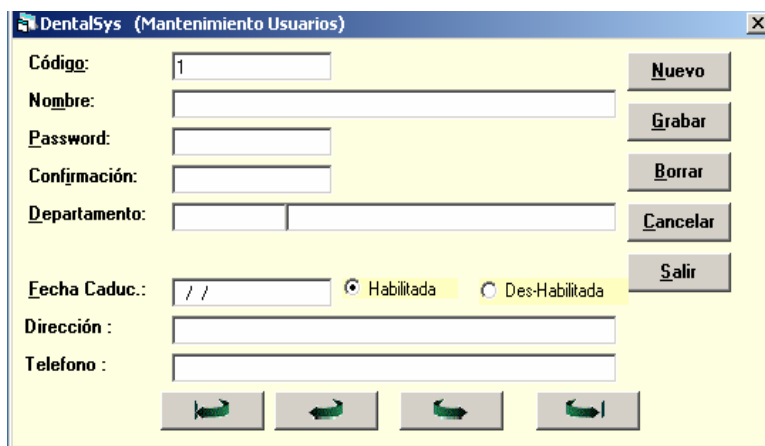
- Realizar ingresos de cada prueba
- Almacenar datos concisos en el sistema de acuerdo a las políticas de la empresa
- Verificar que los registros ingresados cumplan con las seguridades impuestas por el administrador.
- Ingresar datos de pacientes, usuarios , médicos
- Generar Reportes de la clínica dental

DESARROLLO

Mantenimiento Usuarios

1. creación de un usuario nuevo de la clínica dental

Seleccionamos el Menú General, ingresaremos al submenú administrado/ usuarios , desplegándonos la siguiente interfaz

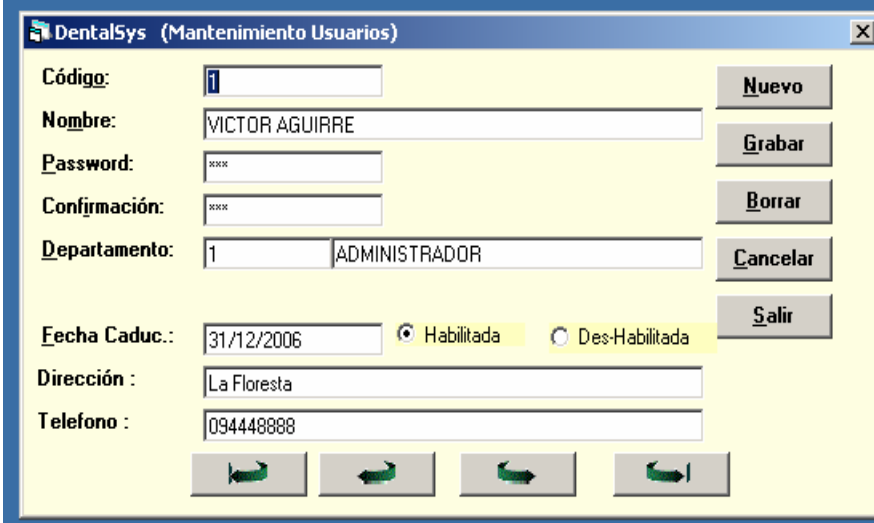


The screenshot shows a software window titled "DentalSys (Mantenimiento Usuarios)". The window contains a form with the following fields and controls:

- Código:** A text box containing the number "1".
- Nombre:** A text box.
- Password:** A text box.
- Confirmación:** A text box.
- Departamento:** A text box.
- Fecha Caduc.:** A date field with slashes (/ /) and two radio buttons: "Habilitada" (selected) and "Des-Habilitada".
- Dirección:** A text box.
- Telefono:** A text box.

On the right side of the form, there are five buttons: "Nuevo", "Grabar", "Borrar", "Cancelar", and "Salir". At the bottom of the form, there are four navigation buttons with green arrows pointing left and right.

Al ingresar a la pantalla realizaremos ingreso de nuevos usuarios , el código se genera automáticamente , procedemos a llenar el formulario .

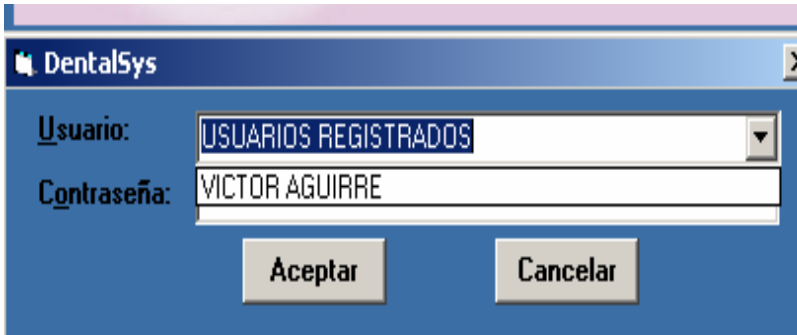


The screenshot shows a window titled "DentalSys (Mantenimiento Usuarios)". The form contains the following fields and controls:

- Código:** A text box containing the number "1".
- Nombre:** A text box containing "VICTOR AGUIRRE".
- Password:** A text box with masked characters "****".
- Confirmación:** A text box with masked characters "****".
- Departamento:** A text box containing "1" and a dropdown menu showing "ADMINISTRADOR".
- Fecha Caduc.:** A date picker showing "31/12/2006".
- Habilitación:** Two radio buttons: "Habilitada" (selected) and "Des-Habilitada".
- Dirección:** A text box containing "La Floresta".
- Telefono:** A text box containing "094448888".

On the right side of the form, there are five buttons: "Nuevo", "Grabar", "Borrar", "Cancelar", and "Salir". At the bottom of the form, there are four navigation buttons with green arrows pointing left and right.

2. Verificamos nuestro primer usuario creado saliendo del sistema e ingresando con este usuario

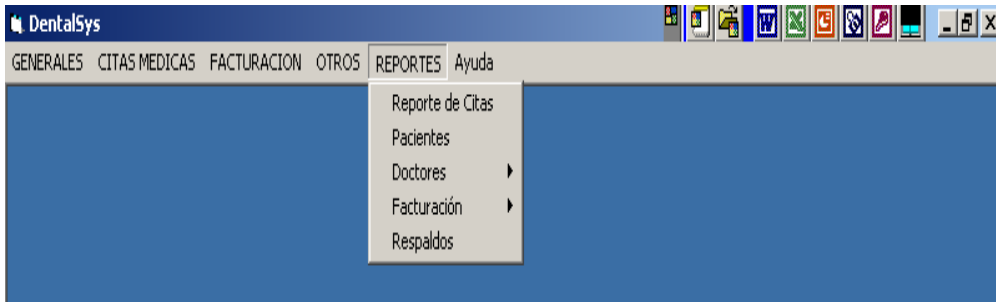


The screenshot shows a window titled "DentalSys" with a login form:

- Usuario:** A dropdown menu with "USUARIOS REGISTRADOS" selected.
- Contraseña:** A text box containing "VICTOR AGUIRRE".

At the bottom of the form, there are two buttons: "Aceptar" and "Cancelar".

3. Verificamos cuales son los permisos que tiene el administrador ya dentro del sistema con la siguiente salida .

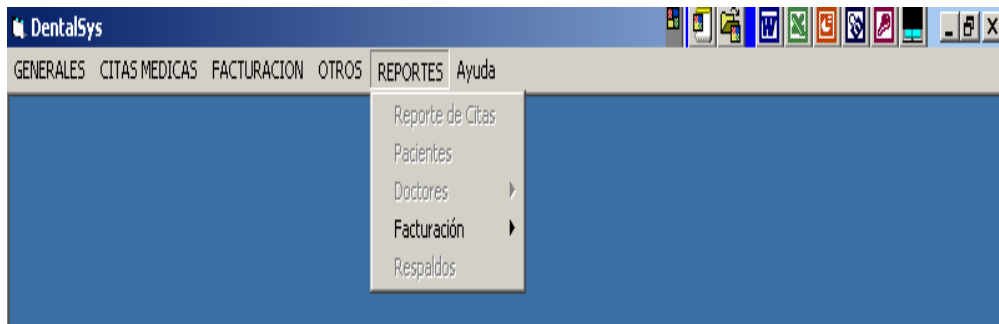


podemos observar que todos los menús y submenús se encuentran activados

4. Ahora crearemos otro usuario pero con menos privilegios que el primero

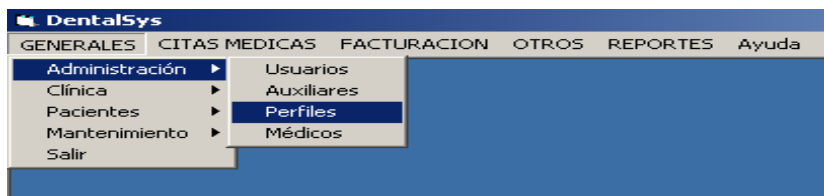
5. de igual manera ingresamos con este nuevo usuario para ver cuales son los efectos

6. ingresamos a la pantalla de menús

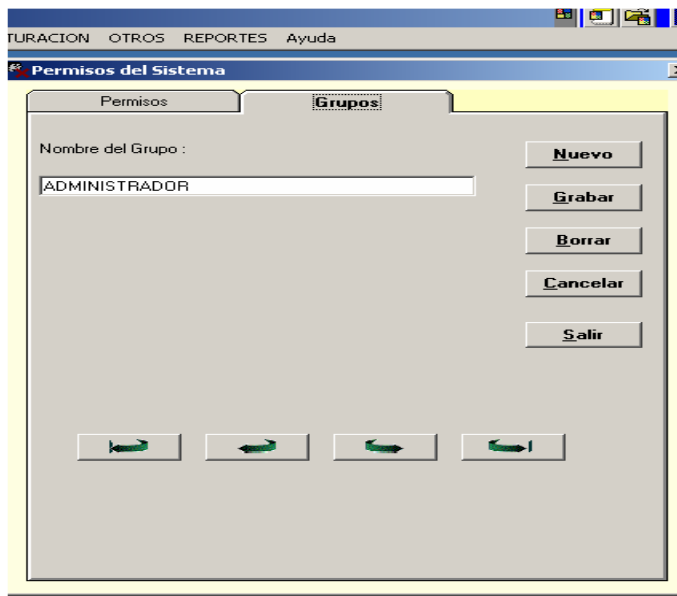


Podemos observar que solamente el submenú facturación se encuentra activado.

7. Realizaremos el proceso para administrar los permisos del sistema
8. Ingresamos por el Menú General / Administrador y escogemos la opción perfiles

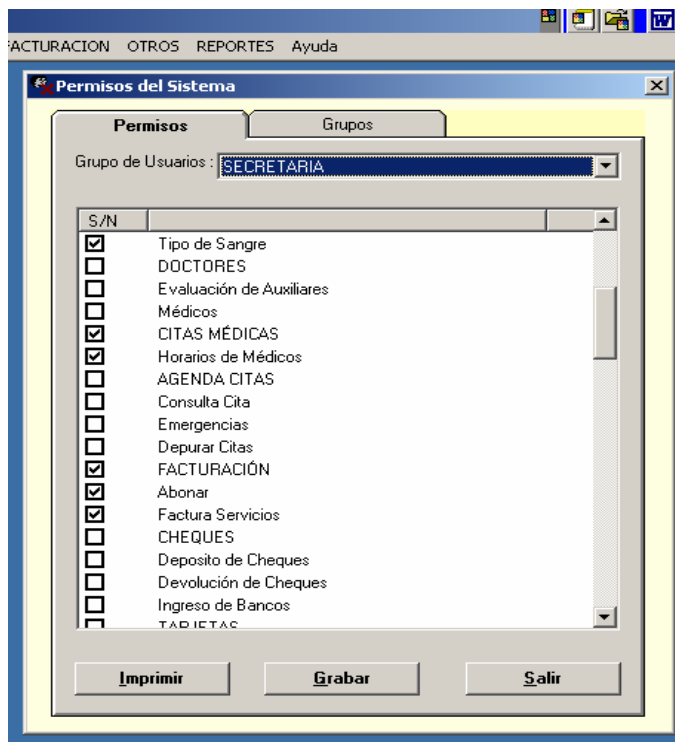


9. se activa la siguiente ventana.



10. en esta ventana realizamos el ingreso de nuevos grupo para los usuarios

11. en la siguiente ventana activamos todos los permisos que el grupo creado va ha tener



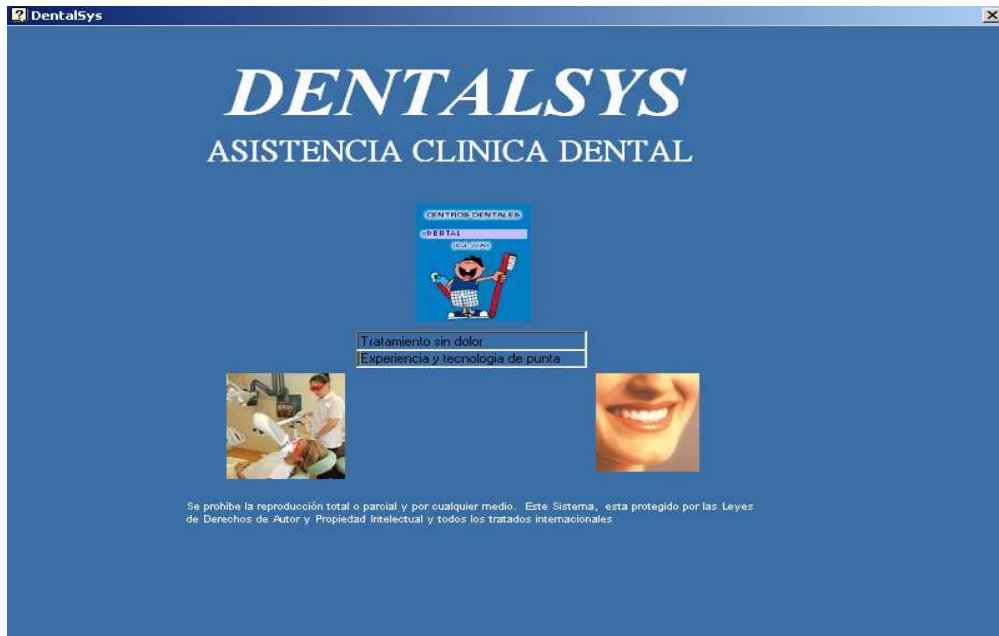
12. por ultimo guardamos los permisos

MANUAL DE USUARIO

MANUAL DE USUARIO SISTEMA DE ADMINISTRACIÓN DENTAL DENTASYS

Inicio de la Aplicación

Al iniciar el sistema nos presenta la siguiente pantalla.

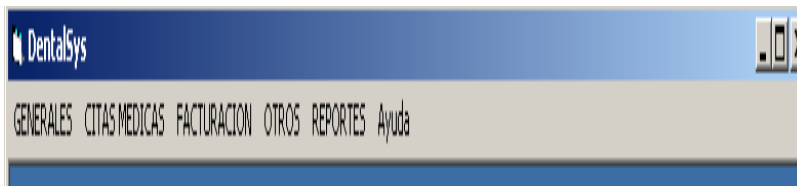


Inicio de sesión

En este formulario ingresamos el nombre de usuario y la contraseña asignada por el administrador del sistema., en este caso vamos a ingresar con el usuario administrador.

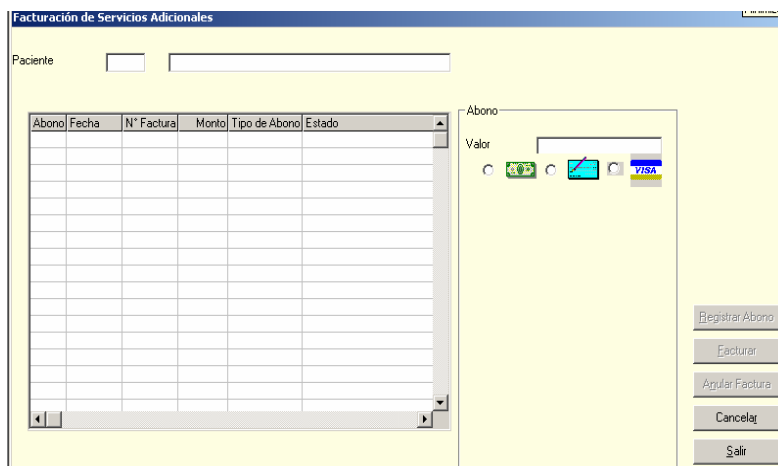


Al ingresar los datos correctos, se ingresa al sistema y nos presenta el menú principal

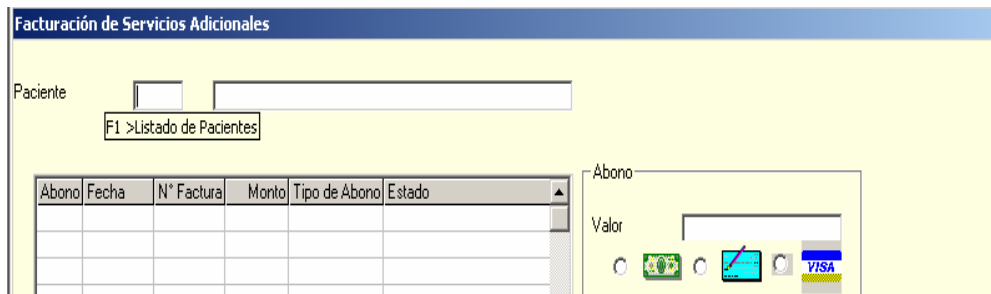


MÓDULOS DEL SISTEMA

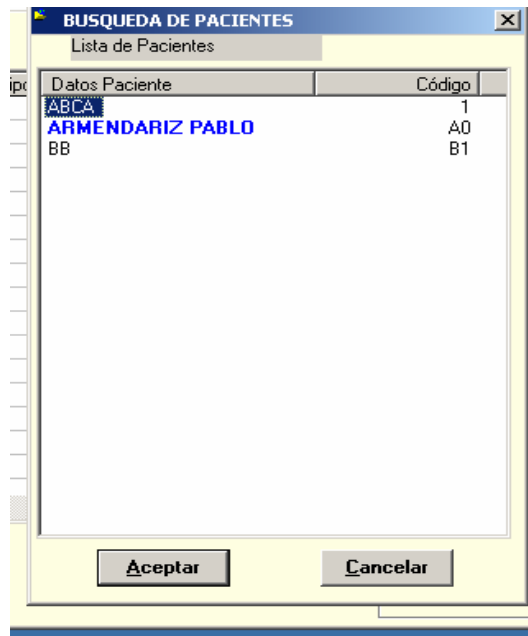
FACTURACIÓN DE SERVICIOS



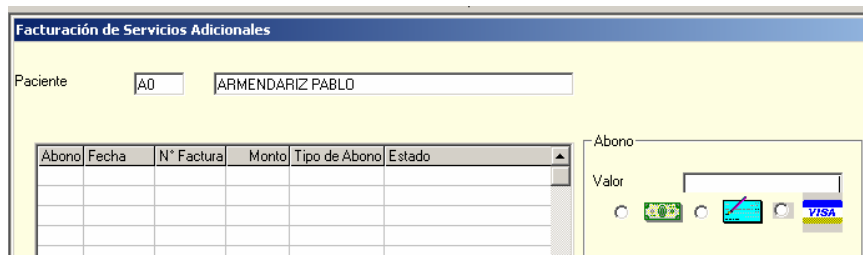
Con F1 consultamos la lista de pacientes



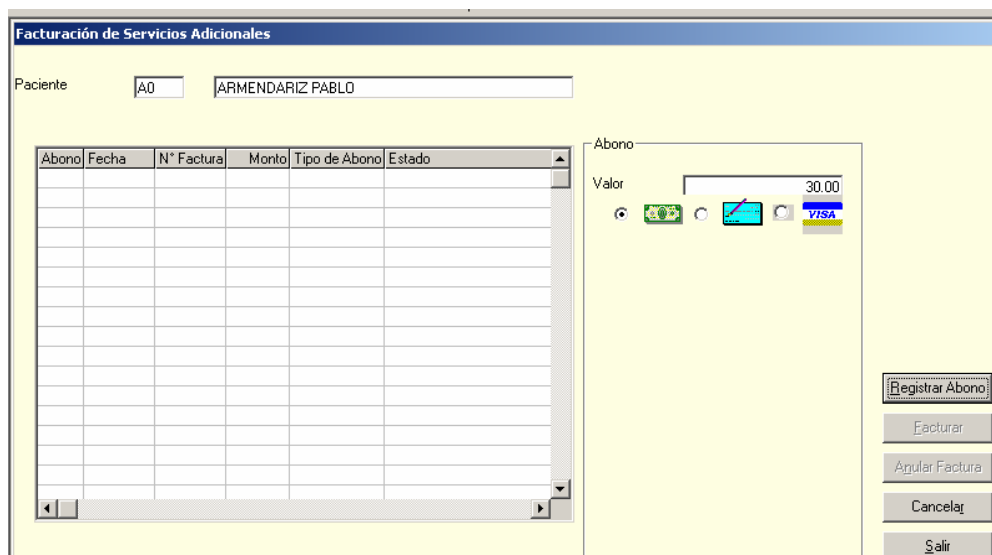
se activa la ventana de pacientes



en el formulario se registra el paciente señalado



se ingresa el valor de factura y se registra el abono



se despliega el siguiente mensaje y aceptamos

Factura

Factura # 102

Ruc 1745233656

Telefono 2801762

Nombre ARMENDARIZ PABLO

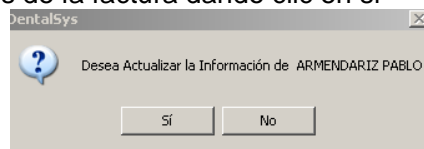
Direccion DESCONOCIDA

Detalle

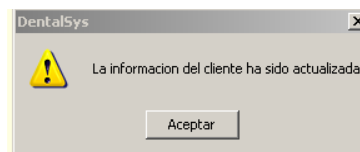
Sellamiento y Radiografia

Imprimir Factura Actualizar Cancelar Salir

actualizamos los datos de la factura dando clic en si



recibimos el siguiente mensaje y aceptamos



damos un clic en imprimir factura

Factura

Factura # 102

Ruc 1745233656

Telefono 2801762

Nombre ARMENDARIZ PABLO

Direccion DESCONOCIDA

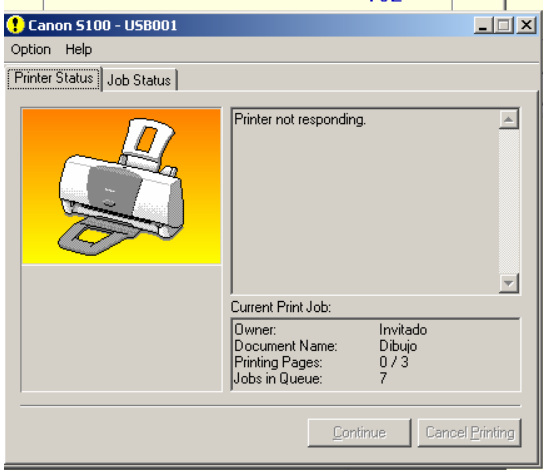
DentalSys

Por favor prepare la Impresora de Facturacion

Aceptar

Imprimir Factura Actualizar Cancelar Salir

finalizamos el proceso de facturación de servicios



MANUAL DE INSTALACIÓN


CONTENIDO

Manual de Instalación

<u>Inicio de Instalación.....</u>	<u>1</u>
<u>Copia de Archivos</u>	<u>1</u>
<u>Instalación del Sistema.....</u>	<u>1</u>
<u>Creación del Path de Instalación.....</u>	<u>2</u>
<u>Selección del Grupo.....</u>	<u>2</u>
<u>Copia de Archivos Necesarios instalación.....</u>	<u>3</u>
<u>Fin Instalación.....</u>	<u>4</u>

MANUAL DE INSTALACIÓN

1. Inicio de la instalación

En la Carpeta Package se encuentra el instalador (Setup), del sistema de administración dental "Dentalsys ", dando doble clic en  procedemos a instalar .

← - - - **Con formato:** Numeración y viñetas

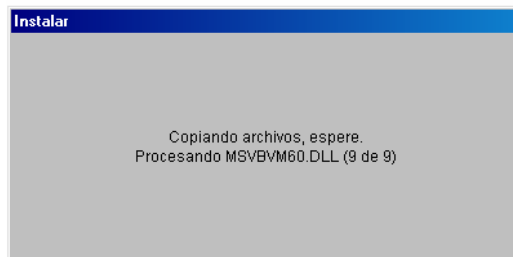


ingreso a la instalación del sistema

2. Copia de Archivos

En este momento se está copiando archivos necesarios para la instalación del sistema .

← - - - **Con formato:** Numeración y viñetas

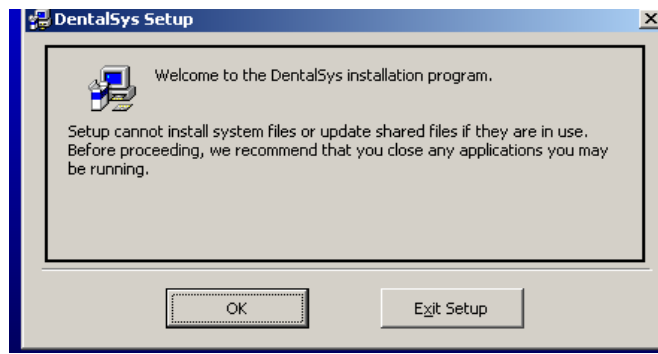


Copia de archivos del sistema

3. Instalación del Sistema

Empieza la instalación del paquete de archivos del sistema, seleccionamos el botón

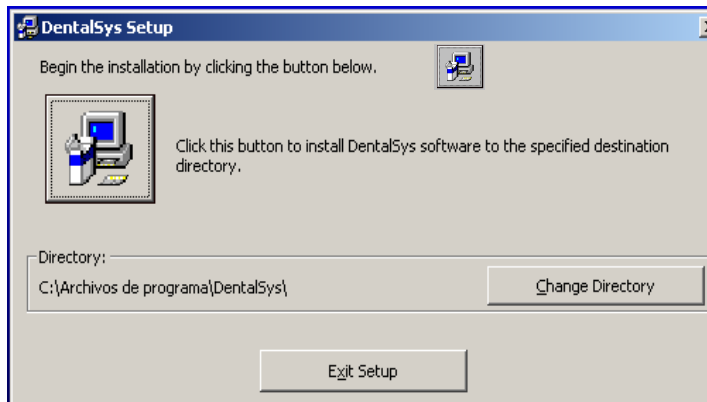
← - - - **Con formato:** Numeración y viñetas



Ingreso a la Instalación del Sistema)

4. Creación del Path de Instalación.


Esta pantalla nos indica la ruta donde será guardado el Sistema de Inventario y Facturación, o si usted desea crear otra dirección de click en

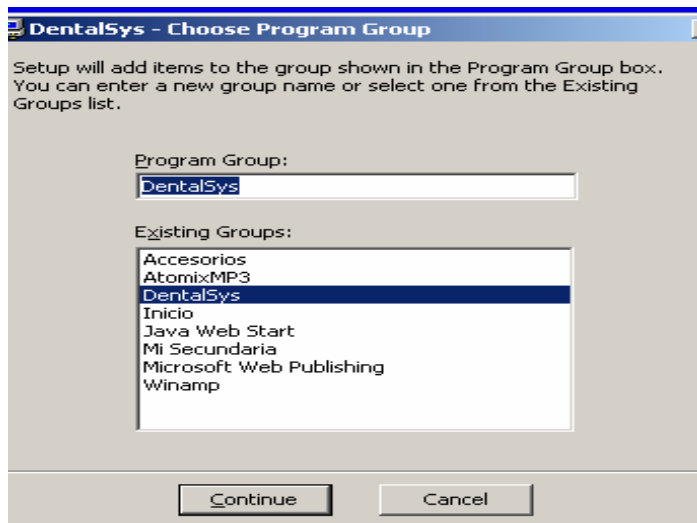


seleccionar la ruta a guardar. Ya elegida la ruta seleccionamos y proseguimos con la instalación

4. Selección del Grupo

← Con formato: Numeración y viñetas

En esta ocasión elegimos el grupo o damos un nombre del grupo al sistema que se está instalando y seleccionamos  para seguir la instalación

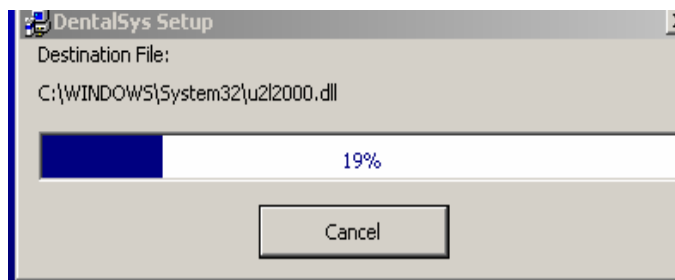


Selección del Grupo donde se instala el Sistema

6. Copia de Archivos Necesarios Para la Instalación

← - - - Con formato: Numeración y viñetas

Este es el proceso de copia de Archivos necesarios para la instalación del sistema .



Copia de archivos a su destino

7. Fin de la Instalación

← - - - Con formato: Numeración y viñetas

Al terminar la instalación nos indica que todo los archivos del sistema se han copiado e instalado satisfactoriamente

ANEXOS

GLOSARIO DE TERMINOS

Actor: es el rol que un usuario juega con respecto al sistema.

Asociación: Relación estructural que describe la conexión entre objetos.

Atributo: Propiedades comunes que describen las instancias

Calidad: Condición o requisito que se pone en un contrato.

Casos de Uso: Conjunto de acciones que un sistema lleva a cabo y que conduce a un resultado de interés para un actor.

Clases: Descripción de un conjunto de objetos que comparten los mismos atributos operaciones, relaciones y semántica.

Cliente / Servidor: Es un modelo para el desarrollo de sistemas de información.

Datos: Antecedente necesario para llegar al conocimiento exacto de una cosa.

Diagrama: Representación gráfica de un conjunto de elementos.

Diagrama de Secuencia: Diagrama de interacción que resalta la ordenación temporal de los mensajes.

Diagrama de Actividad: Diagrama que muestra el flujo de control entre actividades.

Diagrama de Casos de Uso: Diagrama que describe la funcionalidad del sistema.

Diagrama de Colaboración: Diagrama que describe la organización estructural de los objetos que envían y reciben mensajes.

Diagrama de Clases: Diagrama utilizado para describir la estructura estática del sistema.

Diccionario de Datos: Descripción de cada objeto y sus flujos.

Facturar: Comprender en ellas cada artículo, bulto u objeto.

Fase: Periodo comprendido entre dos puntos.

Generalización: relación en la cual los objetos del elemento hijo pueden sustituir a los elementos del objeto padre.

Inventario: Asiento de los bienes y demás cosas pertenecientes a una persona o comunidad, hecho con orden o distinción.

Línea de vida: Es el tiempo que se demora un proceso en realizar.

Mensaje: Comunicación entre objetos para que se lleve a cabo una actividad.

Método: Es una descripción esquemática de un conjunto de pasos que permite llegar a un objetivo

Metodología: Conjunto de métodos que se siguen en una investigación científica.

Modelo: Simplificación de la realidad, creada para comprender mejor el sistema que se está creando.

Objeto: Es algo que tiene entidad en un mundo real.

OMT: Técnicas de Modelado de Objetos es una de las metodologías de Análisis y Diseño Orientadas a Objetos de primera generación, mas maduras y eficientes que existen en la actualidad.

OO: Orientada a Objetos.

Paradigma: Son los esquemas en base de los cuales nos relacionamos con los usuarios, para realizar un control de manera consistente.

Proceso: transforma las entradas en salidas.

Proyecto: Es el cumplimiento de objetivos a través de la utilización eficiente de recursos.

Relación: Conexión semántica entre elementos.

Reportes: Producir una cosa algún beneficio o ventaja.

Rol: es la labor que realiza frente al sistema.

Sistema: Colección de subsistemas organizados para llevar a cabo un objetivo y descritos por un conjunto de elementos.

UML: Es un Lenguaje Unificado de Modelos para especificación, visualización y construcción de los artefactos de sistemas de software.

Usuario: Humano que interactúa con el sistema.

Eliminado: ¶

Manual técnico

Identificación de Actores.....	
Diagramas de caso de uso.....	
Diagramas de caso de uso por actores.....	
Diagramas de caso de uso por actividad.....	
Diccionario de casos de uso.....	
Diccionario del diagrama de casos de uso.....	
Análisis Modelo Estático.....	
Diagrama de clases.....	
Diagrama de Objetos.....	
Diccionario de clases.....	
Diccionario del diagrama de clases.....	
Diseño Modelo dinámico.....	
Diagrama de interacción.....	
Diagrama de Secuencia.....	
Administración de Factura.....	
Administración de citas.....	
Diagrama de Colaboración.....	
Administración de Factura.....	
Administración de citas.....	
Modelo funcional.....	
Diagrama de actividades.....	
Administración de Factura.....	
Administración de citas.....	
Construcción.....	
Front End.....	
Pruebas Funcionales.....	

ANEXOS

ANEXO 2

GLOSARIO DE TERMINOS

Actor: es el rol que un usuario juega con respecto al sistema.

Asociación: Relación estructural que describe la conexión entre objetos.

Atributo: Propiedades comunes que describen las instancias

Calidad: Condición o requisito que se pone en un contrato.

Casos de Uso: Conjunto de acciones que un sistema lleva a cabo y que conduce a un resultado de interés para un actor.

Clases: Descripción de un conjunto de objetos que comparten los mismos atributos operaciones, relaciones y semántica.

Cliente / Servidor: Es un modelo para el desarrollo de sistemas de información.

Datos: Antecedente necesario para llegar al conocimiento exacto de una cosa.

Diagrama: Representación gráfica de un conjunto de elementos.

Diagrama de Secuencia: Diagrama de interacción que resalta la ordenación temporal de los mensajes.

Diagrama de Actividad: Diagrama que muestra el flujo de control entre actividades.

Diagrama de Casos de Uso: Diagrama que describe la funcionalidad del sistema.

Diagrama de Colaboración: Diagrama que describe la organización estructural de los objetos que envían y reciben mensajes.

Diagrama de Clases: Diagrama utilizado para describir la estructura estática del sistema.

Diccionario de Datos: Descripción de cada objeto y sus flujos.

Facturar: Comprender en ellas cada artículo, bulto u objeto.

Fase: Periodo comprendido entre dos puntos.

Generalización: relación en la cual los objetos del elemento hijo pueden sustituir a los elementos del objeto padre.

Inventario: Asiento de los bienes y demás cosas pertenecientes a una persona o comunidad, hecho con orden o distinción.

Línea de vida: Es el tiempo que se demora un proceso en realizar.

Mensaje: Comunicación entre objetos para que se lleve a cabo una actividad.

Método: Es una descripción esquemática de un conjunto de pasos que permite llegar a un objetivo

Metodología: Conjunto de métodos que se siguen en una investigación científica.

Modelo: Simplificación de la realidad, creada para comprender mejor el sistema que se está creando.

Objeto: Es algo que tiene entidad en un mundo real.

OMT: Técnicas de Modelado de Objetos es una de las metodologías de Análisis y Diseño Orientadas a Objetos de primera generación, mas maduras y eficientes que existen en la actualidad.

OO: Orientada a Objetos.

Paradigma: Son los esquemas en base de los cuales nos relacionamos con los usuarios, para realizar un control de manera consistente.

Proceso: transforma las entradas en salidas.

Proyecto: Es el cumplimiento de objetivos a través de la utilización eficiente de recursos.

Relación: Conexión semántica entre elementos.

Reportes: Producir una cosa algún beneficio o ventaja.

Rol: es la labor que realiza frente al sistema.

Sistema: Colección de subsistemas organizados para llevar a cabo un objetivo y descritos por un conjunto de elementos.

UML: Es un Lenguaje Unificado de Modelos para especificación, visualización y construcción de los artefactos de sistemas de software.

Usuario: Humano que interactúa con el sistema.