



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

SISTEMA DE PUNTO DE VENTA PORTÁTIL DE SEGUROS

T E S I S

Que para obtener el Título de
INGENIERO EN COMPUTACIÓN

P r e s e n t a

DAVID CUAHUTENCOS ROJAS



DIRECTOR DE TESIS: ING. ORLANDO ZALDÍVAR ZAMORATEGUI

OCTUBRE 2009

Agradecimientos

Son tantas personas a las cuales debo parte de este triunfo, de lograr alcanzar mi culminación académica, la cual es uno de muchos objetivos que tengo en esta vida.

A mis padres, por darme la confianza y oportunidad de recibir este grado de educación, base para mi superación personal y profesional. En especial dedicado a mi PAPÁ, que en paz descanse y que desde el cielito me siga dando la fuerza para poder llegar hasta donde se pueda. Padre, siempre estas en mi mente, tus consejos siempre los tengo grabados y los seguiré al pie de la letra en tu honor. Madre, serás siempre mi inspiración para alcanzar mis metas, por estar conmigo en las buenas y en las malas, por enseñarme que todo esfuerzo es al final recompensa. Tu esfuerzo, se convirtió en tu triunfo y el mío, PAPÁS LOS AMO.

A mis hermanos, por su apoyo incondicional y por compartir las mismas metas, LOS QUIERO.

A mi pareja Noemi Vilchis, por su apoyo constante e incondicional para lograr concluir este proyecto, TE AMO.

A todos mis amigos pasados y presentes; pasados por ayudarme a crecer y madurar como persona y presentes por estar siempre conmigo apoyándome en todo las circunstancias posibles, también son parte de esta alegría, LOS RECUERDO.

A Luis Uribe y Veronica Chavez, dos amigos especiales, quienes fueron mi apoyo durante este agradable y difícil periodo, por ser MIS AMIGOS, por seguir soportándome y siendo parte de mi vida, LOS VALORO.

A mi director de tesis Ing. Orlando Zaldívar, por su dedicación y apoyo para concluir con este proyecto.

A la UNAM, por dejar ser mi segunda casa y por darme la facilidad de tener una superación profesional enriquecida por sus recintos. POR MI RAZA HABLARÁ MI ESPIRITÚ

A la Facultad de Ingeniería campus CU, por proporcionarme el conocimiento, las bases y herramientas, que fueron fundamentales actualmente en mi crecimiento profesional.

Un hombre de conocimiento elige un camino con corazón y lo sigue. Un hombre de conocimiento no tiene orgullo, se esfuerza, suda, resuella, y si uno lo mira es como cualquier hombre común, excepto que el desatino de su vida está bajo control.

Un hombre de conocimiento es un guerrero, que debe estar preparado para la batalla.

Índice

Contenido	Página
Capítulo 1	
Planteamiento general	3
1.1 Antecedentes	4
1.2 Definición del problema	5
1.3 Objetivo general	6
1.4 Objetivos específicos	8
1.5 Alcance	9
1.6 Descripción general	9
Capítulo 2	
Estructura y organización	11
2.1 Empresas aseguradoras	12
2.2 Inducción a la actividad aseguradora	13
2.3 Conceptos básicos y terminología de seguros	14
Capítulo 3	
Marco teórico	21
3.1 Ingeniería de software	22
3.2 Estrategia de selección de un modelo de ingeniería de software	25
3.3 Ciclo de vida del desarrollo de sistemas	26
3.4 Modelo de ciclo de vida clásico	28
3.5 Fase de análisis	28
3.5.1 Análisis de requerimientos	29
3.5.2 Análisis funcional	30
3.5.3 Análisis orgánico	30
3.6 Fase de diseño	30
3.6.1 Diseño arquitectónico	31
3.6.2 Diseño de datos	32
3.6.3 Diseño de la interfaz	33
3.6.4 Diseño de operación	34
3.6.5 Diseño de salida	34
3.7 Fase de codificación	35
3.7.1 Crear elementos del sistema para la gestión de datos	35
3.7.2 Preparación del entorno de desarrollo	35
3.7.3 Inicio de codificación	35
3.7.4 Generación de código	36
3.8 Fase de pruebas	36

3.9 Fase de mantenimiento	37
3.9.1 Formalización del plan de mantenimiento	38
Capítulo 4	
Desarrollo, modelo de ciclo de vida clásico	39
4.1 Fase de análisis	40
4.1.1 Definición de requerimientos	40
4.1.2 Requerimientos funcionales	40
4.1.3 Casos de uso	41
4.2 Fase de diseño	45
4.2.1 Diseño arquitectónico	45
4.2.2 Diseño de datos	46
4.2.3 Diseño de interfaz	64
4.2.4 Diseño de operación	73
4.2.5 Diseño de salida	82
4.3 Fase de codificación	83
4.3.1 Preparación del entorno de desarrollo	84
4.3.2 Estandarización de codificación	84
4.3.3 Codificación de interfaces visuales	87
4.4 Fase de pruebas	101
4.5 Fase de mantenimiento	103
Capítulo 5	
Resultados, impacto y conclusiones	105
5.1 Resultados	106
5.2 Impacto	112
5.3 Conclusiones	113
Anexos	115
Anexo 1 Instalación Oracle Server 9i	116
Anexo 2 Scripts	122
Anexo 3 Instalación Delphi	142
Anexo 4 Pasos para crear disco autoejecutable	146
Bibliografía	153

Capítulo 1

Planteamiento general

1.1 Antecedentes

El área de marketing de la empresa aseguradora propone la reestructuración de sus procesos de emisión de pólizas con el objetivo de disminuir el tiempo de operación y aumentar el número de ventas para la compañía.

Actualmente, un agente de ventas de seguros para llenar una cotización necesita realizar los siguientes pasos:

1. Utilizar una hoja de cálculo de Excel, en ésta se le proporcionan los factores para realizar los cálculos de las primas de acuerdo a las características de sus asegurados.
2. Una vez que el cliente acepta la prima de su cotización, el agente procede a llenar manualmente la documentación correspondiente, a través de cuestionarios prediseñados.
3. El agente recaba la documentación necesaria para cumplir con los requisitos de asegurabilidad y a través de mensajería envía la información a las oficinas centrales de la compañía de seguros.
4. El área de operaciones analiza toda la documentación e información recabada y realiza el dictamen correspondiente para aceptar el riesgo de asegurar al contratante-titular.
5. En caso de que dicha área detecte que la información no es legible o que la documentación es incompleta, se procede a hacer devolución de dicha documentación y de mencionar las fallas de captura del agente.
6. En caso de que se acepte al contratante, se le enviará a través de mensajería el contrato de la póliza para que sea firmada y aceptada por el contratante.
7. La información de los contratantes aceptados es administrada y cargada en bases de datos por el área de operaciones de la empresa aseguradora.

Como es posible detectar, el tiempo de respuesta para aceptar el riesgo de acuerdo a las características del contratante es muy grande y el margen de error es mayor en el llenado de información sobre los cuestionarios prediseñados.

Sobre estos factores se pretende iniciar un sistema que reduzca los márgenes de error y tiempo de captura de información, convirtiendo este nuevo sistema como una herramienta poderosa y como una estrategia para incrementar el desarrollo de la empresa aseguradora y ser competitivos dentro del mercado de seguros.

1.2 Definición del problema

Actualmente la empresa aseguradora tiene un proceso de emisión que, por sus características mencionadas en el punto anterior, invierte tiempo en la operación manual e incrementa el error humano en la recepción de información. Por tal motivo, muchas cotizaciones de seguros son fallidas en el dictamen para aceptarla como póliza, en consecuencia se pierden clientes y, por último, la empresa no satisface los requerimientos de los agentes como fuerza de ventas para que ésta tenga competitividad dentro del mercado de seguros.

Así, las herramientas que les proporciona la empresa aseguradora a su fuerza de ventas, *agentes*, no les permite tener la facilidad de incrementar el desarrollo de clientes. Estas herramientas son:

- Un archivo Excel como cotizador de seguros
- Papelería para registrar la información del contratante

Es evidente que por estas razones la empresa aseguradora pretenda desarrollar un sistema para su fuerza de ventas que contemple una aplicación amigable, fácil, rápida de usar y que le permita administrar su propia información, por cada agente y póliza vendida, así mismo este sistema deberá disminuir el tiempo de operación para hacer una cotización y automáticamente hacer la impresión del contrato para el cliente, con ello se le dará la confianza para aceptarlo y mantenerlo dentro de los contratantes de la empresa aseguradora.

A este nuevo sistema se le llamará *SISTEMA DE PUNTO DE VENTA PORTÁTIL DE SEGUROS*, ya que éste contará con todas las facilidades de instalación y de operación, proporcionándole al agente toda la confianza de operarlo con el fin de incrementar el número de ventas de seguros.

1.3 Objetivo general

Con el antecedente de la empresa aseguradora, hoy en día requiere del desarrollo de un sistema que cubra sus principales visiones como:

Facilitar el control de procesos operativos con rapidez y eficiencia

La mayoría de las actividades de una empresa están influenciadas por la capacidad de ésta para procesar transacciones con rapidez y eficiencia. Por lo que los sistemas basados en computadora pueden ser de ayuda para eliminar la necesidad de cálculos tediosos y comparaciones repetitivas.

El sistema agilizará de manera instantánea el procedimiento para cotizar un seguro de gastos médicos, ya que los procesos de cálculo serán de manera precisa sin la necesidad de hacer repetidas comparaciones.

El proceso de impresión instantánea de la documentación de las pólizas reflejará la información del contrato que el cliente aceptará de acuerdo a sus necesidades.

Aumentar la productividad

Un sistema automatizado puede ser de gran utilidad si lo que se necesita es un procesamiento acelerado para dar seguimiento en sus operaciones en un lapso de tiempo muy corto.

El sistema facilitará el cálculo de primas y reducirá el tiempo para realizar una cotización.

Aumentar calidad

La creciente competitividad del mercado crea la necesidad de mejores métodos para conseguir costos bajos y relacionarlos con la productividad individual y organizacional.

Debido a que el sistema manejará la información por cada agente, con ello se conseguirá una mayor calidad de envío de información a oficinas centrales de la empresa aseguradora.

Aumentar capacidad de desempeño

Las organizaciones almacenan grandes cantidades de datos, por eso, debe tenerse en cuenta dónde almacenarlos y cómo recuperarlos cuando se les necesita. Por lo que se busca tener manejadores de bases de datos confiables y con un performance de respuesta rápido, con ello también se mejora la velocidad de procesos, adicionalmente para lograr este objetivo se debe tener una estructura de datos entidad-relación bien diseñada para reducir redundancia de información. Así es como el sistema contará con una consistencia de información confiable y segura.

Tener mayor competitividad en la línea de negocio

Los sistemas de información computacionales son un arma estratégica, capaz de cambiar la forma en que la compañía compite en el mercado, en consecuencia estos sistemas mejoran la organización y le ayudan a ganar "ventaja competitiva". Con este módulo de emisión portátil de seguros se pretende incrementar el desempeño en ventas de seguros (*objetivo principal de la empresa*) y para cualquier agente, ya que este sistema tiene la facilidad y confiabilidad de manejo para garantizar a los clientes de que producto cubre con sus necesidades.

Reducir costos

El diseño de este sistema ayudará a disminuir los costos, ya que toma ventaja de las capacidades del cálculo automático y de la recuperación de datos que están incluidos en procedimientos de programas cuyo performance se activa en un 100%, es decir entre más automatizado sea el sistema en procesos de cálculo y de tareas repetitivas, mayor será la eficacia del sistema, en consecuencia se reducirán costos.

Tener mayor seguridad de información

El hecho de que los datos puedan ser guardados en una forma adecuada para su lectura, es necesario implementar esquemas de seguridad para salvaguardar la información, por lo que el sistema tendrá configuraciones de perfiles especiales para cada usuario encargado de cierta información, limitando a ciertas áreas o personal el acceso a toda la información. Adicionalmente, para cubrir este objetivo cada agente tiene un identificador personal en el que toda su información estará relacionada con ese único número y con ello se pueda llevar un mejor control y administración de los negocios de cada agente.

En conclusión, esta herramienta de punto de venta portátil de seguros como software cumplirá con las visiones de la empresa aseguradora y logrará los objetivos antes mencionados, ya que métodos y técnicas resolverán los requerimientos especiales de cada agente de ventas.

Esta propuesta va orientada a resolver las necesidades de empresas cuya línea de negocio es la venta de seguros. El sistema es la opción para cumplir con dichas necesidades.

El sistema es un emisor de pólizas de seguros cuyo objetivo es el de acelerar el proceso operativo de emisión.

1.4 Objetivos específicos

El sistema va a permitir alcanzar lo que cualquiera organización espera de un sistema automatizado.

Como primera instancia este sistema proporcionará:

Generar clientes: Como el cliente es lo más importante para la organización, los directivos buscan diferentes formas para conseguir nuevos clientes y mantener los que tienen. Con este sistema desarrollado especialmente para agentes de ventas de seguros, le proporcionarán todas las herramientas posibles para manejar los diferentes tipos de productos con una mayor rapidez, ofreciéndoles a sus clientes las mejores cotizaciones del mercado al momento, con ello, permitirá al agente agilizar el contrato, ya que el sistema generará la impresión de la póliza al instante y con sólo una conexión de Internet podrá hacer transferencia de su información a las oficinas centrales para que sea procesada y administrada por las áreas correspondientes.

Mejores precios: La ventaja en precios de cotizaciones y pólizas se observa continuamente en la actividad comercial, la estrategia eficaz de precios a menudo se alcanza al desarrollar sistemas de información que tienen control de cálculos de costos, lo que permite generar precios que pueden competir en el mercado. El sistema contará con interfaces para que al usuario pueda manejar tipos de planes, deducibles, sumas aseguradas y coaseguros para cada producto, esto es con el fin de generar primas netas accesibles y las más convenientes para cualquier tipo de persona. Para este propósito el sistema cuenta con un motor de cálculo poderoso y dinámico donde contiene los factores actuariales necesarios para hacer un cálculo preciso de primas, la exactitud del cálculo se refleja al instante y con ello es posible observar que el tiempo de elaboración de una póliza se reduce en un lapso de tiempo muy corto.

Servicios exclusivos: Generalmente cuando una compañía puede ofrecer servicios exclusivos, esto se convierte en una atracción para futuros clientes, por lo que es posible mejorar la competitividad en el mercado. Con este sistema los agentes podrán proporcionar sus servicios de ventas a domicilio y en cualquier parte de la República. Esto es en caso de que un cliente le sea imposible transportarse, así el agente tiene la oportunidad de generar una cotización en cualquier parte en que el cliente lo solicite y sea cómodo para éste, ya que el sistema tiene la finalidad de estar instalado en laptops portátiles, cuya preferencia es más cómoda por generar un menor espacio.

Productos diferentes: El sistema proporcionará al agente de seguros el producto más actual, basta con sólo una conexión a su oficina central vía Internet para cargar actualizaciones. Con ello se busca tener y generar clientes cuyas necesidades sean diferentes y poder cubrirlas.

Con este sistema se garantiza crecimiento, competitividad y productividad.

1.5 Alcance

Este sistema va a ser una herramienta poderosa para la venta de pólizas de seguros. Contará con su propio esquema de almacenamiento de información y de administración.

Para el almacenamiento operará con su propio manejador de base de datos, que cumpla con la mayoría de las reglas para administrar la información.

Para la instalación de la base de datos es necesario configurar desde Windows la herramienta administrativa de orígenes de bases de datos ODBC, una vez realizado se implanta el archivo que contiene el modelo entidad-relación.

Contará también con un esquema de seguridad propia para garantizar la confidencialidad de información del agente, en el sentido de que cualquier tipo de información es importante para quienes saben del significado de ésta.

Las interfaces visuales para el usuario serán completamente amigables para tener una facilidad de manejo, cada agente tendrá su propio perfil de seguridad personalizado, es decir, cada agente tendrá un código de usuario y contraseña para salvaguardar su información en forma confidencial, adicionalmente como ya se había comentado la información llevará relación a un único identificador para garantizar la consistencia de información para cada uno de los agentes.

1.6 Descripción general

El sistema tiene el objetivo de ofrecer a la fuerza productora, una herramienta más ágil y confiable que le permita ahorrar un sin número de trámites y tiempos de operación. Anteriormente, los pasos para tramitar una póliza era generar los documentos físicamente y éstos eran enviados a la oficina central vía telegramas cuyo servicio era proporcionado por Telégrafos de México y al igual la respuesta por parte de la oficina matriz era por la misma vía, con ello es transparente la ventaja que proporciona el sistema, ya que las respuestas para los clientes son rápidas y concisas, esto debido a las ventajas que hoy en día proporciona la tecnología web y las redes computacionales, el resultado es generar un cliente satisfecho de su contratación en un tiempo corto de operación.

Esta nueva herramienta da la oportunidad de ofrecer a sus clientes una manera sencilla y transparente la contratación de una póliza de seguros, debido a que es un sistema de emisión remota que opera en tiempo real. Es importante señalar que con esta nueva herramienta se puede cotizar, emitir, solicitar endosos e imprimir todos estos documentos, brindando al asegurado un servicio eficaz y ágil al momento de cerrar un negocio.

Mencionando otras de las principales ventajas competitivas, es la oportunidad que proporcionará el sistema al agente para ofrecer novedosas formas de pago y conductos de cobro, lo logra debido a que el sistema contará con una interfaz que genera diferentes modalidades de cobro, ya sea vía agente, vía tarjeta de crédito o cargo a cuenta y la forma de pago lo puede generar ya sea mensual, trimestral, semestral o anual, haciendo que el agente ofrezca la mejor comodidad de pago hacia su cliente.

Entre las principales funciones de este sistema están:

- Alta de contratantes (persona física o moral). El sistema cuenta con un submódulo de administración de contratantes, en el cual es posible visualizar todos los datos necesarios de la persona que está adquiriendo un seguro.
- Generar cotizaciones de Gastos Médicos Mayores y Múltiple Familiar.
- Emisión de pólizas e impresión de documentos oficiales al instante de terminar un contrato de seguro.
- Endosos, esto con el fin de realizar modificaciones a las condiciones iniciales de la póliza.
- Reimpresión de solicitud de endosos.
- Renovación automática.
- Actualización de tarifas.

Con estas características que proporcionará el sistema, el agente de ventas tendrá todas las ventajas para aumentar su productividad en el ramo de ventas de seguros proporcionando a su cliente la satisfacción de haber hecho una inversión confiable.

La confiabilidad será un factor importante que el sistema reflejará hacia los clientes.

Capítulo 2

Estructura y organización

El Instituto Mexicano Educativo de Seguros y Fianzas A. C., IMESFAC presenta una investigación preliminar sobre historia y conceptos para tratar de entender cuál es el rubro y objetivo de una empresa aseguradora, esto con el fin de que la empresa pueda practicar una nueva forma de tramitación electrónica de bienes y servicios con base en el rubro de ventas que ésta tiene, para una mejor competitividad en el mercado de seguros y especialmente en el ramo médico.

Este instituto se posiciona ante el mercado de la capacitación especializada en seguros y fianzas para ofrecer a las empresas un servicio integral, que puede ir desde la mera impartición de una plática hasta la detección de necesidades, diseño y conducción de cualquier programa de capacitación para el estudio especializado de seguros.

El propósito de esta investigación es fundamental para entender el mundo de los seguros, para conocer el origen, cómo se conforma el sector de aseguramiento y cómo funciona una compañía de seguros, utilizando la terminología apropiada.

Algunos de los puntos para este capítulo son:

- El papel que desempeñan las compañías de seguros en la economía de un país.
- Identificar las diferencias que existen entre las estructuras que puede adoptar una compañía de seguros y su funcionamiento.
- Definir y analizar los conceptos y principios básicos en materia de seguros.

2.1 Antecedentes de empresas aseguradoras

Hoy en día las instituciones de seguros son parte importante del desarrollo de los países por el impacto económico derivado de su operación.

Otorgan protección a la comunidad (asegurados, contratantes, beneficiarios, etc.) contra una pérdida económica ocasionada por algún riesgo, dando seguridad financiera a quien lo contrata.

Ante todos los peligros por los que se ve amenazado el hombre desde sus orígenes, ha tratado de encontrar soluciones, adoptando distintas actitudes, por ejemplo, transferir el peligro o riesgo al que se encuentra expuesto. Para llevar a cabo esta operación aparece el seguro.

El seguro responde a una necesidad de prevención para amortiguar o compensar las consecuencias económicas de acontecimientos con daños. Dado el carácter masivo que reviste la industria aseguradora, se ha convertido en una función de interés público.

El seguro surge como un tipo de financiamiento, mediante el cual no se traspa el peligro, sino parte de sus efectos financieros, disminuyendo así las pérdidas que pudiera sufrir el asegurado en sus bienes o persona de suceder un incidente no planeado que trae consecuencias económicas negativas.

2.2 Inducción a la actividad aseguradora

La base del negocio de seguros la constituye el riesgo y la probabilidad de que suceda, por lo que siempre estarán al tanto del análisis de las solicitudes y de las circunstancias en las que se encuentren, tanto personas como bienes a asegurar. La aseguradora clasifica al asegurado o al bien asegurable en una clase de riesgo.

Por ejemplo:

En el seguro de personas, un grupo de asegurados que presentan un riesgo similar como la muerte, un accidente o una enfermedad. En el caso del seguro de daños, un grupo de riesgos similares como la ruptura de cristales, el incendio, el robo de automóviles, etc.

Con ello cualquier aseguradora como institución financiera debe tener:

Control de operaciones financieras: Una operación financiera se lleva a cabo cuando se contrata un servicio o producto financiero.

Ofrecer un servicio: Proporciona las facilidades que ofrece la institución para realizar una operación, sólo participa como intermediario.

Ofrecer un producto: Son los diferentes instrumentos de que dispone una institución financiera, autorizados por la autoridad competente, para llevar a cabo su actividad financiera.

Manejo de personas físicas y morales: Se habla de usuarios con capacidad jurídica para contratar los servicios.

2.3 Conceptos básicos y terminología de seguros

Riesgo asegurable: Es la posible ocurrencia por azar de un acontecimiento que produce una necesidad económica y cuya aparición real o existencia se previene y garantiza en la póliza, obligando al asegurador a efectuar la indemnización que le corresponde.

Los tipos de riesgo son:

Tipo de riesgo	Ejemplo
Afecta a personas	La muerte prematura causada por un accidente, enfermedad o acto delictivo, ya que existen gastos que para la familia continúan, aun cuando el jefe de familia no exista.
Dañan a terceras personas	El ordenamiento jurídico, como la responsabilidad civil, que puede generarse por el daño a propiedad ajena o el daño involuntario a una persona.
Consecuenciales	La pérdida de ventas, utilidades o rentas a consecuencia de un riesgo que haya afectado a negocio o empresa.
Afecta a bienes	Actos delictivos como robo o asalto. Fenómenos de la naturaleza. Actos derivados de la impericia o negligencia.

Un riesgo en el área de la actividad aseguradora debe ser incierto o aleatorio, dado que debe haber una relativa incertidumbre, ya que si tenemos certeza de su existencia real, puede hacer desaparecer la aleatoriedad, principio básico del seguro.

Distribución de un riesgo: Consiste en el reparto o la dispersión de riesgos para obtener una compensación estadística. Igualando los riesgos que componen su cartera de bienes asegurados. Esta distribución se puede llevar a través de dos modos principales:

- **Coaseguro:** Consiste en compartir la cobertura de un mismo riesgo entre dos o más aseguradoras, esta partición también puede darse entre el asegurado y la aseguradora.
- **Reaseguro:** Es un instrumento técnico que consiste en la cesión de una parte de los riesgos y bienes asegurados por una institución de seguros, para conseguir la compensación estadística que necesita otra empresa llamada reaseguradora.

Valoración del riesgo: Para que un contrato de seguros se lleve a cabo es necesario que el contratante del seguro y/o asegurado tenga un interés asegurable.

Esto significa que debe haber una relación entre el que sufre el daño patrimonial por un evento determinado y el contratante del seguro y/o asegurado.

Suma asegurada: Es el valor del interés y sirve como medida para la reparación de daño y sólo representa el límite de responsabilidad del asegurador en el pago del siniestro.

Prima: Es el precio del seguro y debe pagarse en el domicilio del asegurador o en el lugar convenido entre las partes interesadas. Es el resultado del cálculo actuarial de acuerdo a las características del producto.

Contrato de seguro: El seguro es un contrato por el cual una de las partes (*el asegurador*) se obliga, mediante una prima que le abona la otra parte (*el asegurado*), a resarcir un daño o cumplir la prestación convenida si ocurre el evento previsto, como puede ser un accidente o un incendio, entre otras.

El contrato de seguro puede tener por objeto toda clase de riesgos, si existe interés asegurable o salvo alguna prohibición expresada en la ley.

El asegurador, por lo general, es una compañía de seguros organizada bajo la forma de sociedad anónima; pero también existen cooperativas y mutualidades de seguros e incluso un organismo oficial como la caja de ahorro y seguro, que realiza operaciones de este tipo. Así mismo, en algunos estados funcionan entidades aseguradoras oficiales.

El contrato de seguro es consensual, bilateral y aleatorio. Es consensual porque se perfecciona por el mero consentimiento de las partes y produce sus efectos desde que se ha realizado la convención; es bilateral puesto que origina derechos y obligaciones recíprocas entre asegurador y asegurado, es aleatorio porque se refiere a la indemnización de una pérdida o de un daño producido por un acontecimiento o un hecho incierto, pues no se sabe si se va a producir y en el caso contrario, como ocurre con la muerte, no se sabe cuándo ello ha de acontecer.

Póliza: La póliza de seguros es el documento que refleja las normas generales, particulares o especiales del contrato. Regulan las relaciones contractuales convenidas entre el asegurador y el asegurado

Las pólizas de seguros pueden ser de dos tipos:

A riesgo nombrado: Se especifican las eventualidades contra las que está asegurada la persona o el bien.

A todo riesgo: En donde se cubren todos los riesgos excepto los especificados en las exclusiones.

Dentro de la póliza podemos identificar diversas partes como son:

- Condiciones generales
- Condiciones particulares
- Condiciones especiales
- Endosos

Elemento	Descripción
Condiciones generales	<p>Las condiciones generales son principios básicos que regulan el contrato de seguros dependiendo del riesgo a asegurar.</p> <p>Las condiciones generales establecen normativas a:</p> <ul style="list-style-type: none">• Extensión y objeto del seguro• Riesgos excluidos con carácter general• Forma de liquidación de los siniestros• Pago de indemnizaciones• Cobro de recibos• Comunicaciones• Jurisdicción
Condiciones particulares	<p>Las condiciones particulares son aspectos concretos relativos al riesgo que se asegura, por ejemplo:</p> <ul style="list-style-type: none">• Nombre y domicilio de los contratantes• Designación del asegurado y beneficiarios• Tipo de aseguramiento (salud, vida, auto, etc.)• Naturaleza del riesgo asegurado• Designación de los objetos asegurados y su localización geográfica• Suma asegurada o alcance de la cobertura• Prima, recargos e impuestos• Vigencia de la póliza• Lugar y forma de pago• Vencimiento de las primas

Elemento	Descripción
Condiciones especiales	<p>Algunas normas especiales de la póliza como:</p> <ul style="list-style-type: none">• Deducibles• Coaseguros especiales• Supresión de exclusiones
Endosos	<p>Es el documento que altera o modifica los términos y condiciones del contrato de seguros, vía sistema de emisión, mientras el contrato se encuentre en vigor.</p> <p>Los endosos son de tres tipos:</p> <p>Endoso A: En este documento generalmente sirve para adicionar coberturas o ampliar la responsabilidad de la aseguradora. Genera movimiento de incremento en prima a cobrar.</p> <p>Endosos B: En este documento se corrigen los datos generales del asegurado o condiciones de aseguramiento. No genera alteración o cambio alguno de cifras.</p> <p>Endosos D: Este documento generalmente sirve para eliminar coberturas o disminuir responsabilidad de la aseguradora. Genera un movimiento de devolución de prima para el asegurado.</p>

Siniestro: Es la manifestación concreta del riesgo, es un acontecimiento que por originar daños concretos previstos en la póliza, obliga a la aseguradora a pagar la indemnización.

- Siniestro con pérdida total: El que ha afectado en totalidad el objeto asegurado o bien los daños que exceden de algún porcentaje establecido por la aseguradora para considerarlo como pérdida total.
- Siniestro con pérdida parcial: El que afecta sólo a una parte del objeto asegurado.
- Siniestro declarado: En el que el asegurado ya dio aviso a la aseguradora.
- Siniestro pendiente de pago: La compañía ha hecho procedente el pago de siniestro y se hacen los trámites para su liquidación.
- Siniestro pagado: Liquidación en su totalidad.

Indemnización: Es la compensación económica que la aseguradora debe pagar por el siniestro.

Su fin es conseguir una reposición económica en el patrimonio del asegurado afectado por un siniestro, a través de la reposición o sustitución del objeto dañado mediante el pago en económico del valor de los bienes lesionados. La indemnización tiene una doble limitación, no puede exceder:

1. Al capital asegurado en la póliza para el riesgo afectado por el siniestro.
2. Ni del precio del objeto dañado inmediatamente antes de que se produzca el accidente.

Tipos de indemnización.

- En efectivo: La indemnización puede darse en una sola exhibición o pagos periódicos según lo estipulado en el contrato o se presente el siniestro.
- En especie: La aseguradora restituye el bien material dañado.
- En servicios: Al ocurrir el evento amparado en el contrato el asegurado goza de la prestación de ciertos servicios que pueden o no ser descritos en el contrato.

Modalidades de indemnización

- **Proporción indemnizable:** Es la responsabilidad que asume la aseguradora, para indemnizar al asegurado en la misma proporción que existe entre la suma aseguradora y su valor real de los bienes al momento del siniestro.
- **A primer riesgo:** Se indemnizará la totalidad de la pérdida sin aplicar reglas de proporcionalidad aunque la suma asegurada sea la inferior al valor real de los bienes, estableciéndose como límite máximo a indemnizar la suma asegurada contratada.

Variables de indemnización

- El infraseguro o bajo seguro
- El sobreseguro
- El deducible
- El coaseguro

Con la investigación realizada en este capítulo se tiene ya un panorama general de los principios básicos, sobre los cuales se rige una empresa asegurada, así como la terminología usada para aplicar un seguro. Esto nos ayudará a comprender a usuarios y técnicos con los que se tendrá relación durante el desarrollo del proyecto.

Capítulo 3

Marco teórico

3.1 Ingeniería de software

El desarrollo de software hoy en día se ha convertido en un desafío constante, en un desarrollo evolutivo continuo y acelerado. Esto es producto a la gran influencia del software dentro del mundo de los negocios y de la vida cotidiana, desde entonces surgen propuestas y herramientas que proponen reducir tiempo y procesos, simplificando las tareas tediosas a una fácil y simple operación.

Así, las metodologías de desarrollo, han permitido que el software evolucionara provocando, que de acuerdo a las necesidades creadas, la aparición de una nueva disciplina: la ingeniería de software.

La ingeniería de software aparece a finales de los años sesenta y principios de los setenta, comenzando con las técnicas de programación estructurada, incorporándolas a las fases del ciclo de vida del software. La programación estructurada fue seguida por otros métodos estructurados de análisis y también métodos estructurados de diseño, existen también tecnologías orientadas a objetos. En un principio la programación era la tarea de oro de la ingeniería de software, pero ahora la ingeniería, el análisis y el diseño de la definición de los requisitos son más importantes, ya que se exige que el software sea eficaz y barato tanto en el desarrollo como en la compra. También se requiere una serie de características como fiabilidad, facilidad de mantenimiento y de uso, eficiencia, etc.

Los estándares de la ingeniería de software y la madurez de proceso, han caracterizado la industria del software como una disciplina madura. En un nivel más técnico, la ingeniería de software comienza con una serie de tareas que hacen modelos y que resultan en una especificación completa de requisitos y una representación comprensiva de diseño del software que será construido.

La ingeniería de software comprende un conjunto estructurado de procesos que representan los “paradigmas de la ingeniería de software”. Estos paradigmas se basan en la naturaleza del proyecto y de la aplicación, en los métodos y las herramientas que serán usados en el proyecto, los controles y los productos o servicios desarrollados [1].

[1] Pressman, R.S, *Ingeniería del software. Un enfoque práctico*, McGraw-Hill, 5ª Edición, España, 2002.

La ingeniería del software se podría definir como una nueva disciplina que define los estándares para el desarrollo de software basados en diversos indicadores, como: la corrección, la eficacia, la eficiencia, la fiabilidad, facilidad de comprensión, de uso y de mantenimiento, la interoperabilidad, la portabilidad, la reusabilidad, la robustez y funcionamiento eficiente que satisfaga las necesidades del usuario.

La ingeniería de software nos dará el enfoque sistemático del desarrollo, operación, mantenimiento y ciclo de vida del software. También ayudará a establecer el costo-beneficio dentro del desarrollo del producto.

A través de la ingeniería de software se buscan los siguientes objetivos:

- Establecer calidad en los productos dirigidos a los usuarios.
- Aumento de productividad.
- Control en el proceso del desarrollo de software.
- Bases para la construcción de software de alta calidad en una forma eficiente.
- Garantizar la producción y mantenimiento de los productos de software desarrollados en el plazo fijado y dentro del costo estimado.

Las ventajas que se observan al realizar un producto siguiendo un modelo de ciclo de vida dentro de la ingeniería de software son:

Capacidad: Las actividades de cualquier organización empresarial están influenciadas por la capacidad de ésta para procesar transacciones con rapidez y eficiencia. Los sistemas de información mejoran esta capacidad en tres formas:

- **Velocidad de procesamiento:** Los sistemas de cómputo eliminan tareas costosas en cuestión de tiempo y recursos humanos. Un sistema automatizado puede ser de gran utilidad si lo que se necesita es un procesamiento de información acelerado.
- **Volumen de actividades:** La implementación de un sistema deberá tener la capacidad de resistir el impacto ante la cantidad de actividades comerciales que deberá resolver en un tiempo factible para los usuarios.
- **Disponibilidad de información:** La mayoría de las organizaciones almacenan grandes cantidades de datos; por ende, debe tenerse en cuenta dónde almacenarlos y cómo recuperarlos cuando se necesita.

Todo sistema, en general, busca un manejo de información adecuada, manteniendo la recuperación rápida.

Costos: Cualquier sistema ayudará a la reducción de costos porque toma ventaja de las capacidades de cálculo automático y de recuperación de datos de manera efectiva, casi eliminando por completo actividades cuya ejecución es de manera manual.

Control: El hecho de que los datos puedan ser guardados en una forma adecuada para su lectura, se deben seguir planes de backups de la información de producción y brindar un respaldo ante cualquier contingencia. También el acceso de información puede estar controlado por un complejo esquema de contraseñas, limitando el acceso a ciertas áreas o personal; si está bien protegido, es difícil de acceder.

El desarrollo de un sistema mejora la exactitud y la consistencia de procesos laboriosos. Para lograrlo se requiere de un flujo lógico para cada proceso, para ello se debe crear y optimizar algoritmos sofisticados en la que cada paso se lleva a cabo de la misma manera, consistente y con exactitud.

Comunicación: La falta de comunicación es una fuente común de dificultades que afectan tanto a cliente como a empleados. Sin embargo, los sistemas de información bien desarrollados amplían la comunicación y facilitan la integración de funciones individuales.

Muchas empresas aumentan sus vías de comunicación por medio del desarrollo de redes para este fin, dichas vías abarcan todo el país y les permiten acelerar el flujo de información dentro de sus oficinas y otras instalaciones que no se encuentran en la misma localidad.

Una de las características más importantes de los sistemas de información para oficinas es la transmisión electrónica de información, como por ejemplo, los mensajes y los documentos. Dentro de estas interconexiones por redes, también es importante aplicar métodos de seguridad para que el envío de información hacia una matriz sea confiable y consistente.

Con frecuencia las actividades de las empresas abarcan varias áreas de la organización, la información que surge en un área se necesita en otra. Los sistemas de información ayudan a comunicar los detalles del diseño a los diferentes grupos, mantienen las especificaciones esenciales en un sitio de fácil acceso, calculan factores en el nivel de costos a partir de detalles proporcionados por otros grupos. Debe haber un flujo de información entre áreas involucradas sin perder la consistencia y veracidad de la información.

Competitividad: Los sistemas de información computacionales son un arma estratégica, capaz de cambiar la forma en que la compañía compite en el mercado. En consecuencia, estos sistemas mejoran la organización y la ayudan a ganar “ventaja competitiva”; sin embargo, si los competidores de la compañía tienen capacidades más avanzadas para el procesamiento de información, entonces los sistemas de información pueden convertirse en una “desventaja competitiva”.

Una organización puede ganar ventaja competitiva a través de sus sistemas de información de diferentes formas.

Como el cliente es lo más importante para una organización, los directivos buscan diferentes formas para conseguir nuevos clientes y mantener los que tienen. Para eso las empresas proporcionan:

- Mejores precios
- Servicios exclusivos
- Productos diferentes

La estrategia eficaz de precios a menudo se alcanza al desarrollar sistemas de información, por razones tales como reducción de costos y ganancia en la exactitud de cálculos.

Generalmente, cuando una compañía puede ofrecer servicios exclusivos y atraer clientes es posible que dicha empresa se vaya convirtiendo en líder ante los demás competidores.

En los negocios, los proveedores también tienen importancia estratégica. Una manera de utilizar los sistemas de información para favorecer arreglos con los proveedores es ofreciendo un mejor precio, disminuyendo los costos.

Los sistemas de información también forman la base de muchos productos y servicios nuevos. La generación de nuevos productos y servicios eficientes es posible porque los sistemas tienen la capacidad de establecerse a la medida de los clientes gracias al procesamiento de información confiable.

3.2 Estrategia de selección de un modelo de ingeniería de software

Como hemos mencionado antes, los sistemas de información sirven para diversas finalidades, que van desde el procesamiento de las transacciones de una empresa, hasta proveer de la información necesaria para decidir sobre asuntos que se presentan con frecuencia.

En un proyecto de sistema de información se deben tomar aspectos apropiados tanto de hardware o la tecnología de comunicaciones que se va a utilizar, el impacto del nuevo sistema sobre los empleados de la empresa y las características específicas que el sistema debe tener se pueden determinar de manera secuencial.

Adicionalmente, debemos tener en cuenta que la ingeniería de software nos proporcionará la detección, evaluación, construcción y asesoría de los sistemas de información y servicios, promoviendo además, una alta participación del cliente y usuario para lograr la más alta calidad, contribuyendo así al mejoramiento informático de las organizaciones, a través de la utilización de herramientas con tendencia tecnológica y estratégica para el desarrollo de la informática.

Por lo tanto, reiterando que todo proyecto de ingeniería de software tiene fines ligados a la obtención de un producto, proceso o servicio que es necesario generar a través de diversas actividades. Algunas de estas actividades pueden agruparse en fases porque globalmente contribuyen a obtener un producto intermedio, necesario para continuar hacia el producto final y facilitar la gestión del proyecto. Al conjunto de las fases empleadas se le denomina “ciclo de vida”.

3.3 Ciclo de vida del desarrollo de sistemas

El proceso de desarrollo de software requiere por un lado un conjunto de conceptos, una metodología y un lenguaje propio. A este proceso también se le llama el ciclo de vida del software que comprende fases como: concepción, elaboración, construcción y transición.

La concepción define el alcance del proyecto y desarrolla un caso de negocio. La elaboración define un plan del proyecto, especifica las características y fundamenta la arquitectura. La construcción crea el producto y la transición transfiere el producto a los usuarios.

Un modelo de ciclo de vida define el estado de las fases a través de las cuales se mueve un proyecto de desarrollo de software.

El ciclo de vida para el desarrollo de sistemas es el conjunto de actividades que los analistas, diseñadores y usuarios realizan para desarrollar e implantar un sistema de información.

Así, el modelo de ciclo de vida suministra una guía para los ingenieros de software con el fin de ordenar las diversas actividades técnicas en el proyecto, por otra parte permite un marco para la administración del desarrollo y el mantenimiento, en el sentido de estimar recursos, definir puntos de control intermedios, monitorear el avance, etc.

Antes de iniciar con cualquier proyecto es recomendable tomar en cuenta las siguientes actividades:

Definición de la solicitud

Antes de considerar cualquier investigación de sistemas, la solicitud de proyecto debe examinarse para determinar con precisión lo que el solicitante desea; ya que muchas solicitudes que provienen de empleados y usuarios no están formuladas de manera clara.

Estudio de factibilidad

En la investigación preliminar un punto importante es determinar que el sistema solicitado sea factible. Existen cinco aspectos relacionados con el estudio de factibilidad, que son realizados por lo general por analistas capacitados o directivos:

- **Factibilidad técnica:** Estudia si el trabajo para el proyecto, puede desarrollarse con el software y el personal existente, y si en caso de necesitar nueva tecnología, cuáles son las posibilidades de desarrollarla.
- **Factibilidad económica:** Investiga si los costos se justifican con los beneficios que se obtienen, y si se ha invertido demasiado para no crear el sistema, si se cree necesario.
- **Factibilidad operacional:** Investiga si será utilizado el sistema, si los usuarios usarán el sistema para obtener beneficios.
- **Factibilidad legal:** Se determina cualquier infracción, violación o responsabilidad legal en que se podría incurrir a nivel local, municipal, estatal o federal para el desarrollo del sistema.
- **Factibilidad de programa:** Se revisa las condiciones óptimas de implementación del sistema, se prepara un programa de operación y producción, el cual debe incrementar paulatinamente en función de las expectativas de la empresa, buscando dar experiencia y capacitación al personal encargado de las diversas operaciones para alcanzar el rendimiento máximo.

Una vez realizadas las actividades mencionadas se decide adoptar el modelo que cubra las necesidades del proyecto con métodos básicos que demostrará cumplir la ingeniería en el proyecto.

3.4 Modelo de ciclo de vida clásico

Según Pressman el modelo de vida clásico sugiere un enfoque sistemático, secuencial, para el desarrollo del software que comienza en un nivel de sistemas y progresa con el análisis, diseño, codificación, pruebas y mantenimiento. Éstas son las actividades a seguir para desarrollar un proyecto de ingeniería software:

- Análisis
- Diseño
- Codificación
- Pruebas
- Mantenimiento

Una forma gráfica de ver las etapas y su secuencia es como lo muestra la figura 1.

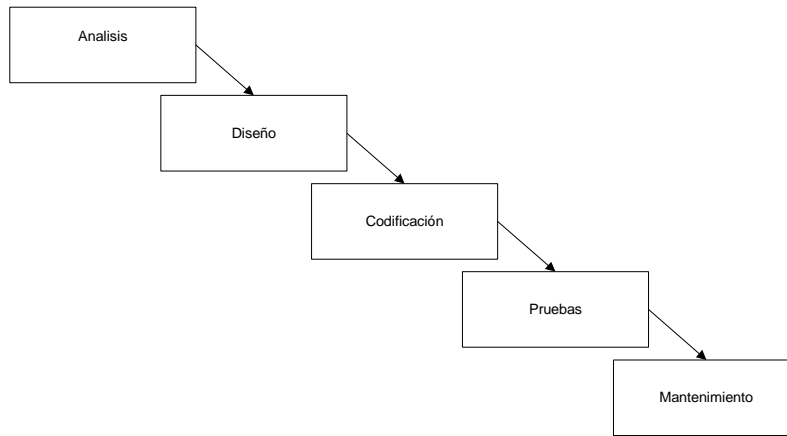


Figura 1. Ciclo de vida clásico

3.5 Fase de análisis

En esta fase se hace el estudio de las necesidades de información que debe satisfacer el sistema a desarrollar, elaborando una serie de especificaciones formales que describan la funcionalidad del mismo y que permitan abordar con garantías la fase de diseño.

El aspecto fundamental del análisis es comprender todas las facetas importantes de la parte de la empresa, en general el flujo del negocio que maneja y que se encuentra bajo estudio.

Es necesario realizar investigaciones detalladas que requieren el estudio de manuales, reportes, observación en condiciones reales de las actividades del trabajo, muestras de formas y documentos, también se realizan entrevistas con usuarios relacionados y de diferentes áreas para el fin de comprender el proceso en su totalidad. Reuniendo los

detalles, los analistas estudian los datos sobre requerimientos con la finalidad de identificar las características que debe tener el nuevo sistema.

La definición, análisis y validación de los requisitos a partir de la información facilitada se convierte en un catálogo detallado de requisitos, a partir del cual se puede comprobar que los productos generados en las actividades de modelado se ajustan a los requerimientos.

De esta actividad se descompone un conjunto de tareas con un orden que exige continuas realimentaciones entre sí y con otras tareas realizadas en paralelo. Lo que se tiene en un momento determinado es un catálogo de requisitos especificado en función de la progresión del proceso de análisis.

Se puede manejar el análisis en etapas como:

3.5.1 Análisis de requerimientos

Se trata de establecer el alcance, los objetivos y requisitos del sistema, examinando las posibles alternativas que podrían solucionar las necesidades del usuario y recomendar una de ellas. Al final de establecer los requerimientos se obtiene un documento llamado "Documento de requisitos del sistema".

La obtención de requisitos es la primera tarea y comienza con la obtención detallada de información mediante sesiones de trabajo con los usuarios, como ya se había mencionado.

Los requisitos, que sirven de base para establecer los niveles de servicios del sistema, hay que tener en cuenta, si existen, las posibles restricciones del entorno, tanto hardware como software, que puedan afectar al sistema de información. También se definen las prioridades que hay que asignar a los requisitos, considerando los criterios de los usuarios acerca de las funcionalidades a cubrir.

Los principales tipos de requisitos que se deben especificar son, por ejemplo:

- Funcionales
- Rendimiento
- Seguridad
- Implantación
- Disponibilidad

En el análisis de requerimientos se estudia la información capturada para detectar inconsistencias, ambigüedades, duplicidad o escasez de información, etc. También se analizan las prioridades establecidas por el usuario y se asocian los requisitos relacionados entre sí.

El análisis de los requisitos permite identificar funcionalidades o comportamientos comunes. Mediante sesiones de trabajo con los usuarios, se contrastan las conclusiones del análisis de la información recogida.

3.5.2 Análisis funcional

Una vez aceptados formalmente los requerimientos por ambas partes (equipo de desarrollo y usuario), se elabora un conjunto de especificaciones que describan la funcionalidad del sistema, estableciendo módulos en que se descompondrá, definición de datos e interfaces visuales para el usuario. Se planificarán las pruebas que deberá superar el sistema, se estimará la relación costo-beneficio para comprobar si interesa su construcción y se establecerán los plazos de entrega del sistema. Todo ello se recoge en dos documentos, denominados:

- Documento de especificación funcional
- Documento de especificación no funcional

Mediante esta tarea, los usuarios confirman que los requisitos especificados en el catálogo de requisitos, así como los casos de uso, son válidos, consistentes y completos.

3.5.3 Análisis orgánico

El objetivo de esta fase es obtener un conjunto de especificaciones que contemplarán los aspectos físicos del sistema, considerando las características tecnológicas del entorno específico en el que se implantará, y que constituirá el punto de partida para la construcción del sistema.

3.6 Fase de diseño

El diseño de un sistema de información produce los detalles que establecen la forma en la que el sistema cumplirá con los requerimientos identificados durante la fase de análisis. Los especialistas en sistemas se refieren, con frecuencia, a esta etapa como diseño lógico en contraste con la del desarrollo del software, a la que denominan diseño físico.

El diseño define el proceso de aplicar ciertas técnicas y principios con el propósito de definir un dispositivo, un proceso o un sistema, con suficientes detalles como para permitir su interpretación y realización física.

Es común que los diseñadores hagan una presentación de prototipos de interfaces visuales en la que los usuarios podrían visualizar una vez que el sistema sea terminado; esto se logra utilizando algunas de las herramientas que proveen los ambientes disponibles para el desarrollo de sistemas.

También es importante diseñar el diagrama de flujo del comportamiento de los datos, tanto los de entrada, como los que serán calculados y los que deben ser almacenados. Los diseñadores seleccionan las estructuras de archivo y los dispositivos de almacenamiento. Los procedimientos y algoritmos diseñados describirán cómo procesar los datos y producir salidas. Los documentos que contienen las especificaciones de diseño representan a éste mediante diagramas, tablas y símbolos especiales.

La información detallada del diseño se proporciona al equipo de programación para comenzar la fase de desarrollo de software.

En esta fase del diseño del sistema nos podemos apoyar en actividades como:

3.6.1 Diseño arquitectónico

El objetivo principal de la arquitectura del software es aportar elementos que ayuden a la toma de decisiones y, al mismo tiempo, proporcionar conceptos y un lenguaje común que permitan la comunicación entre los equipos que participen en un proyecto.

El diseño de aplicaciones modernas involucra la división de una aplicación en múltiples capas; la interfaz de usuario, la capa media de objetos de negocios y la capa de acceso a datos.

Capa de presentación

La capa de presentación provee a la aplicación una interface de usuario (IU). Aquí es donde la aplicación despliega, capta y envía información hacia la capa del negocio. Idealmente, la IU no desarrolla ningún procesamiento de negocios o reglas de validación de negocios.

Una de las mayores dificultades y factores importantes cuando desarrollamos aplicaciones cliente/servidor es mantener una separación completa entre la presentación, la lógica de negocio y los servicios de datos.

Capa de negocio

Toda aplicación tiene código para implementar reglas de negocios, procesos y relaciones entre objetos de entidad y abstractos.

Los objetos son una representación del mundo físico y la idea de usarlos, es modularizar esta capa para realizar funciones específicas de manera aislada y estar disponible para recibir y procesar peticiones. Un buen diseño de un objeto cliente, contiene todos los datos y rutinas necesarias para representarlo a través del negocio completo y puede ser usado a través de toda la aplicación.

Capa de servicios de datos

Muchas aplicaciones interactúan con datos y se almacenan en bases de datos. Existen funciones básicas que son comunes en todos los procesos, como:

- Crear datos
- Leer datos
- Actualizar datos
- Eliminar datos

3.6.2 Diseño de datos

Trasforma el modelo de dominio de la información, creado durante el análisis, en las estructuras de datos necesarios para implementar el software. Una vez que son identificadas las necesidades de información de cada uno de los procesos que conforman el sistema de información, se obtiene un modelo de datos que contemple todas las entidades, relaciones, atributos y reglas de negocio necesarias para dar respuesta a dichas necesidades.

El objetivo del diseño de una base de datos relacional es que sea flexible para el diseño de reportes y etiquetas, el cual consiste en una o más tablas de dos dimensiones a las que se llama relaciones. Los renglones de la tabla presentan los registros y las columnas contienen los atributos y para cada atributo una combinación de valores permitidos llamado dominio.

El modelo entidad-relación incluye los conceptos de entidad, relación y atributo identificador.

Entidad: Cualquier tipo de objeto o concepto sobre el que se recoge información: cosa, persona, concepto abstracto o suceso. Por ejemplo, coches, casas, empleados, clientes, empresas, oficios, diseños de productos, etc. Una entidad es un objeto distinguible de otros objetos por medio de su conjunto específico de atributos. Por otro lado, representa la estructura lógica global de la base de datos, la cual puede expresarse gráficamente por un diagrama entidad-relación.

Relación: Es una correspondencia o asociación entre dos o más entidades. Cada relación tiene un nombre que describe su función.

Los diagramas entidad-relación ayudan a que el analista comprenda las entidades y relaciones que comprenden el sistema organizacional.

Los cuatro tipos de diagramas de entidad-relación son:

- Relación uno a uno
- Relación uno a muchos
- Relación muchos a uno
- Relación muchos a muchos

Atributo: Es una característica de interés o un hecho sobre una entidad o sobre una relación. Los atributos representan las propiedades básicas de las entidades y de las relaciones. Toda la información extensiva es portada por los atributos.

Cada atributo tiene un conjunto de valores asociados denominado dominio. Éste define todos los valores posibles que puede tomar un atributo. Puede haber varios atributos definidos sobre un mismo dominio.

Identificador: Un identificador de una entidad es un atributo o conjunto de atributos que determina de modo único cada ocurrencia de esa entidad. Un identificador de una entidad debe cumplir dos condiciones:

1. No pueden existir dos ocurrencias de la entidad con el mismo valor del identificador.
2. Si se omite cualquier atributo del identificador, la condición anterior deja de cumplirse.

Toda entidad tiene al menos un identificador o varios identificadores alternativos.

3.6.3 Diseño de la interfaz

Describe cómo se comunica el software consigo mismo, con los sistemas que operan junto con él y con los operadores y usuarios que lo emplean. Se especifican las interfaces entre el sistema y el usuario: formatos de pantallas, diálogos e informes. El objetivo es realizar un análisis de los procesos del sistema de información en los que se requiere una interacción del usuario, con el fin de crear una interfaz que satisfaga todos los requisitos establecidos, teniendo en cuenta los diferentes perfiles a quienes va dirigido, acorde a sus necesidades, flexible, coherente, eficiente y sencillo de utilizar. Se identifican los distintos grupos de usuarios de acuerdo a las funciones que realizan: conocimientos, habilidades, características que poseen en el entorno donde trabajan. La identificación de los diferentes perfiles permite conocer mejor las necesidades y particularidades de cada uno de ellos.

Para cada proceso en línea, se especifica qué tipo de información requiere el usuario para completar su ejecución, realizando una descomposición en diálogos, que refleje la secuencia de operación de manera gráfica.

Finalmente, se define el formato y contenido de cada una de las interfaces de pantalla especificando su comportamiento dinámico. Como resultado de esta actividad se genera la especificación de interfaz de usuario, como producto que engloba los siguientes elementos:

- Interfaces generales
- Perfiles de usuario
- Descomposición funcional en diálogos
- Catálogo de controles y elementos de diseño de interfaz de pantalla
- Formatos individuales de interfaz de pantalla
- Modelo de navegación de interfaz de pantalla

- Formatos de impresión
- Prototipo de interfaz interactiva
- Prototipo de interfaz de impresión

3.6.4 Diseño de operación

Transforma elementos estructurales de la arquitectura del programa. La importancia del diseño de procedimientos se puede definir en una sola palabra: calidad. Es donde se fomenta la calidad del proyecto. El diseño de procedimientos es la única manera de materializar con precisión los requerimientos del cliente.

Es un proceso y un modelado a la vez. El proceso de diseño es un conjunto de pasos repetitivos que permiten al diseñador describir todos los aspectos del sistema a construir. A lo largo del diseño, se evalúa la calidad del desarrollo del proyecto con un conjunto de revisiones técnicas.

3.6.5 Diseño de salida

Se refiere a los resultados e informaciones generadas por el sistema, para la mayoría de los usuarios, la salida es la única razón para el desarrollo de un sistema y la base de evaluación de su utilidad. Sin embargo, cuando se realiza un sistema, como analista se debe realizar lo siguiente:

- Tipo de información a presentar. Se decide si la información será presentada en forma visual, verbal o impresa y seleccionar el medio de salida.
- Presentación de la información en un formato aceptable.
- Distribución de salida entre los posibles destinatarios.

3.7 Fase de codificación

Es la conversión del diseño de sistemas de información a instrucciones ejecutables por computadora, por lo que los algoritmos deben ser implementados para realizar procesos complejos tomando en cuenta el performance de la operación y de acuerdo al flujo de negocio establecido en los requerimientos.

Los desarrolladores son responsables de la documentación de los procedimientos codificados, ésta es esencial para probar el sistema y hacer el mantenimiento correctivo en el menor tiempo posible.

El objetivo de esta actividad es asegurar la disponibilidad de todos los medios y facilidades para que se pueda llevar a cabo la construcción del sistema de información. Entre estos medios, cabe destacar la preparación de los puestos de trabajo, equipos físicos y lógicos, gestores de bases de datos, bibliotecas de programas, herramientas de generación de código, bases de datos o ficheros de prueba, entre otros.

3.7.1 Crear elementos del sistema para la gestión de datos

Reservar el espacio de almacenamiento sobre los dispositivos físicos a emplear, tamaño de los bloques, opciones de almacenamiento de datos, etc.

Inicializar la base de datos, cargando los datos considerados necesarios en el espacio de almacenamiento previamente definido.

3.7.2 Preparación del entorno de desarrollo

Se prepara el entorno en el que se construirán los componentes del sistema de información, contemplando aspectos tales como:

- Bibliotecas o librerías a utilizar.
- Herramientas: Generadores de código, editores, compiladores, verificadores sintácticos, montadores de enlace, puestos de trabajo.
- Implementación de los procedimientos de operación y seguridad propios del entorno de desarrollo, de acuerdo a los requisitos de seguridad y operación.

3.7.3 Inicio de codificación

El objetivo es la codificación de los componentes del sistema de información, a partir de las especificaciones de construcción obtenidas en la fase de diseño del sistema de información, así como la construcción de los procedimientos de operación y seguridad establecidos para el mismo.

Para generar el código fuente se tienen en cuenta los estándares de nomenclatura, codificación y calidad utilizados por la organización. Con el fin de verificar que el código fuente especifica de forma correcta el componente, se realiza su ensamblaje o compilación, verificando y corrigiendo los errores sintácticos, y el enlace del código obtenido con las correspondientes bibliotecas.

3.7.4 Generación de código

Generación de procedimientos de operación y administración del sistema de información, así como los procedimientos de seguridad y control de acceso, necesarios para ejecutar el sistema una vez que se haya implantado y esté en producción. Para la generación de dichos procedimientos se tienen en cuenta, también, los estándares y normas de la organización identificados en el análisis y definición de requerimientos.

3.8 Fase de pruebas

Un sistema de pruebas implica la operación o aplicación del mismo a través de condiciones controladas y la consiguiente evaluación de la información.

Las condiciones controladas deben incluir tanto situaciones normales como anormales. El objetivo del módulo de pruebas es encontrar errores que contradigan el flujo del negocio.

Para la planeación de las pruebas se pueden aplicar al sistema distintos tipos que se explicarán a continuación:

Pruebas de caja negra: Las pruebas de caja negra no consideran la codificación dentro de sus parámetros a evaluar, es decir, no está basada en el conocimiento del diseño interno del programa. Estas pruebas se enfocan a los requerimientos establecidos y a la funcionalidad del sistema.

Pruebas de caja blanca: Al contrario de las pruebas de caja negra, éstas se basan en el conocimiento de la lógica interna del código del sistema. Las pruebas contemplan los distintos caminos que se pueden generar gracias a las estructuras condicionales, a los distintos estados del mismo, etc.

Pruebas de integración: Las pruebas de integración buscan probar la combinación de las distintas partes de la aplicación para determinar si funcionan correctamente en conjunto.

Esto es útil para ver cómo se comunica la aplicación con el servidor de base de datos.

Pruebas del sistema: Son similares a las pruebas de la caja negra, sólo que éstas buscan probar el sistema como un todo. Están basadas en los requerimientos generales y abarca todas las partes combinadas del sistema.

Haciendo uso de las pruebas explicadas anteriormente, se definieron tres rubros con base en los cuales se van a definir los puntos a evaluar. Estos rubros son los siguientes:

Pruebas de contenido: Estas pruebas como su nombre lo indica, buscan verificar que el contenido sea coherente y consistente a la vez. También se debe verificar que las palabras usadas para transmitir una idea al usuario sean las adecuadas y que la idea transmitida sea la misma.

Pruebas de funcionalidad: Este tipo de pruebas examina si el sistema cubre sus necesidades de funcionamiento, acorde a las especificaciones de diseño. En ella se debe verificar si el sistema lleva a cabo correctamente todas las funciones requeridas, se debe verificar la validación de los datos y se deben realizar pruebas de comportamiento ante distintos escenarios.

Estas pruebas deben estar enfocadas a tareas, a límites del sistema, a condiciones planeadas de error y de exploración. Para estas pruebas usamos los esquemas de pruebas de caja negra, ya que nos interesa saber si funciona o no, independientemente de la forma en que lo haga.

Pruebas de usabilidad: Las pruebas realizadas en este rubro tienen la finalidad de verificar qué tan fácil es de usar un sistema. Las pruebas de usabilidad deben promover el aprendizaje (qué tan fácil es para los usuarios realizar las tareas básicas en la primera vez que tienen contacto con el sistema), eficiencia (una vez que los usuarios han aprendido algo del sistema, qué tan rápido pueden llevar a cabo las tareas), manejo de errores (cuántos errores comete el usuario, qué tan graves son éstos y qué tan fácil es para el usuario recuperarse de ellos) y grado de satisfacción (qué tan satisfactorio es usar el sistema).

3.9 Fase de mantenimiento

El mantenimiento es una actividad para mejorar y optimizar el software implementado, uno de sus objetivos es corregir los defectos que se presentan durante la operación. Esto involucra cambios al software durante su uso, tanto como la adición de nuevas funcionalidades para mejorar la usabilidad y aplicabilidad del software.

La organización o equipo debe tener un mecanismo para documentar y rastrear defectos y deficiencias. Las personas involucradas en la fase de mantenimiento de software esperan trabajar en estos defectos conocidos, ubicarlos y preparar un nuevo lanzamiento del software, conocido como un lanzamiento de mantenimiento, el cual resolverá los temas pendientes.

Una vez que los responsables han analizado a detalle la funcionalidad del sistema, tendrán que valorar si la información disponible es suficiente para poder abordar en condiciones óptimas los diferentes tipos de mantenimiento que se pudieran presentar durante el ciclo de vida del software, asegurando que el sistema se incorpore al entorno de producción de manera completa, actualizada, consistente y precisa.

Los diferentes tipos de mantenimiento que debe resolver el equipo de trabajo son:

- **Perfectivo:** Acciones llevadas a cabo para mejorar la calidad interna del sistema en cualquiera de sus aspectos, como reestructuración del código, definición más clara del sistema y optimización del rendimiento y eficiencia. El resultado es un sistema con un performance mejorado hacia el usuario.
- **Preventivo:** Consiste en la modificación del producto software sin alterar las especificaciones del mismo, para mejorar las propiedades de mantenimiento del producto y facilitar así las futuras correcciones preventivas.
- **Adaptativo:** Modificaciones que afectan a los entornos en los que el sistema opera, por ejemplo, cambios de configuración del hardware, software de base, gestores de base de datos, comunicaciones. Dentro de este tipo de mantenimiento se busca la escalabilidad del sistema.
- **Correctivo:** Cambios precisos para corregir errores del producto software.

Finalmente, se recoge en el plan de mantenimiento toda la infraestructura necesaria para la gestión del futuro mantenimiento.

3.9.1 Formalización del plan de mantenimiento

Se establece formalmente el plan de mantenimiento para el sistema, una vez que haya sido aceptado y se incorpore al entorno de producción. Se fija el tipo de mantenimiento que se va a asumir para cada sistema de información, determinando los criterios de regulación necesarios para cada tipo de mantenimiento contemplado y reflejando los requisitos de formación esenciales, de manera que se pueda responder satisfactoriamente a las peticiones de mantenimiento. Se estiman los recursos humanos necesarios para el servicio de mantenimiento establecido, definiendo claramente sus perfiles, asignando responsabilidades y determinando las funciones que van a llevar a cabo, con el fin de garantizar la coordinación en la gestión del mantenimiento.

Capítulo 4

Desarrollo, modelo de ciclo de vida clásico

4.1 Fase de análisis

4.1.1 Definición de requerimientos

Realizar una herramienta con el poder de controlar y administrar emisiones de pólizas para agentes de ventas de seguros.

Esta herramienta debe ser portátil y de fácil instalación.

El sistema debe estar diseñado con el objetivo de ofrecer a la fuerza productora, *agentes de seguros*, una herramienta más ágil y confiable que les permita ahorrar un sin número de trámites y tiempos.

Este sistema debe ofrecer a sus clientes una manera sencilla y transparente para contratar pólizas de seguros.

Para lograrlo este sistema de emisión debe manejarse siempre de manera remota y exclusiva para cada agente.

Debe operar y permitir la emisión de pólizas en tiempo real al aceptar el contrato entre la aseguradora y el asegurado. Para lograrlo debe cotizar, emitir e imprimir todos estos documentos, brindando al asegurado un servicio eficaz y ágil al momento de cerrar un negocio.

Debe dar la oportunidad al agente de ofrecer diferentes formas de pago y conductos de cobro, haciendo que el agente tenga el tiempo y concentración necesario para dejar hacer lo que mejor sabe hacer que es vender.

4.1.2 Requerimientos funcionales

Como el sistema de punto de venta estará en manos de personas, cuya función es sólo realizar ventas de productos, el sistema en primera instancia deberá manejar la información de manera segura y responsable cumpliendo con los objetivos antes mencionados dentro del marco teórico.

Estas personas encargadas de realizar las ventas de productos frecuentemente no son técnicas, ante una nueva estrategia empresarial, se preguntan si los sistemas de información merecen su confianza, confianza que se ve mermada por cada fallo y sobre todo, cuando la inversión en defensa de los medios de trabajo no se traduce en ganancia. Lo ideal es que los sistemas no fallen. En consecuencia, aquí se busca conocer las necesidades de un agente de ventas para contrarrestar esa desconfianza ante lo nuevo de dicho sistema.

Éstos son los primeros requerimientos funcionales para los agentes:

- Usuarios personalizados por agente
- Alta de datos de un contratante (persona física o moral)
- Generar cotizaciones en productos de gastos médicos mayores
- Emitir pólizas
- Solicitar endosos B
- Cancelación de pólizas

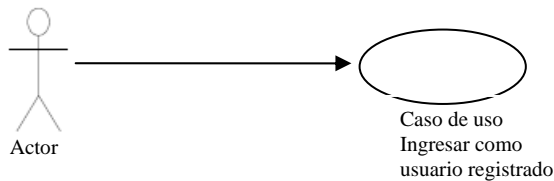
Dentro de estos requerimientos se tendrá que considerar la mayor parte de factores que puedan poner en juego la credibilidad y seguridad de la organización. El análisis, en consecuencia, nos permitirá tomar decisiones certeras y seguras para cumplir con todos los objetivos de la organización y adicionalmente la elección correcta de los componentes para el desarrollo de nuestro sistema.

Se deben elegir los elementos de trabajo adecuados para confrontar la complejidad del problema al que se enfrentará; complejidad en el sentido de que hay muchos elementos a considerar y que, si no se es riguroso, las conclusiones serán de poco fiar. Es por ello que se persigue un análisis consistente que no deje lugar a la improvisación, ni se dependa de la arbitrariedad del analista, para ello se apoyará en UML.

4.1.3 Casos de uso

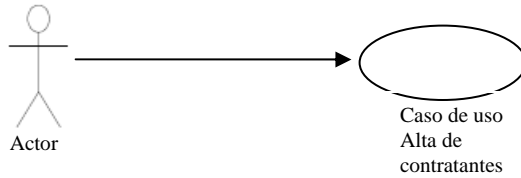
- Caso de uso: Ingresar como usuario registrado

Actor: Usuario



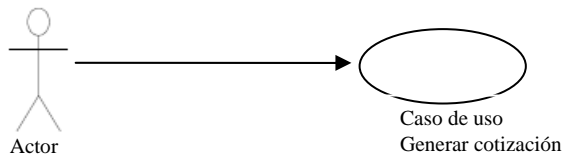
ACTOR		SISTEMA		
Paso	Acción	Paso	Acción	Excepción
1	Se activa sistema	2	Muestra interfaz de acceso	
3	Usuario ingresa claves de acceso	4	Valida información	
		5	Si es válido el usuario el sistema muestra interfaz principal	Si no es válido el usuario, se cierra el sistema después de tres intentos

- Caso de uso: Alta de contratantes
Actor: Usuario



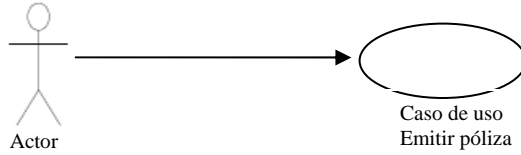
ACTOR		SISTEMA		
Paso	Acción	Paso	Acción	Excepción
1	Activa interfaz de alta de contratantes	2	Muestra interfaz de alta de personas	
3	Ingresa datos de persona: apellidos, nombre, fecha de nacimiento, dirección, etc.	4	Valida información de persona física o moral	
		5	Si la información es correcta se guarda en la base de datos	Si no es correcta la información, no guarda y muestra mensaje con el dato erróneo

- Caso de Uso: Generar cotización
Actor: Usuario



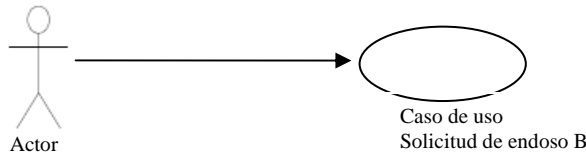
ACTOR		SISTEMA		
Paso	Acción	Paso	Acción	Excepción
1	Activa interfaz de nueva cotización	2	Muestra interfaz de nueva cotización	
3	Usuario ingresa un contratante dado de alta y escoge el producto a cotizar.	4	Valida información de persona física o moral, del tipo de producto e información ingresada para realizar el cálculo de prima	
		5	Si la información es correcta, guarda la cotización en la base de datos y puede hacer una impresión	Si no es correcta la información, muestra los mensajes con el dato erróneo

- Caso de uso: Emitir póliza
Actor: Usuario



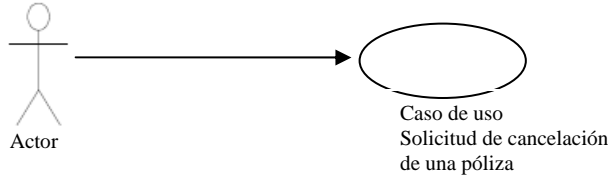
ACTOR		SISTEMA		
Paso	Acción	Paso	Acción	Excepción
1	Activa interfaz para abrir una cotización	2	Muestra interfaz de una cotización escogida	
3	Si el cliente acepta la cotización, el agente activa botón de emisión	4	Valida información de la cotización	
		5	Si la información es correcta calcula la prima final, guarda la emisión en la base de datos e imprime contrato de póliza	Si no es correcta la información, muestra los mensajes con el dato erróneo en la emisión

- Caso de uso: Solicitud de endoso B
Actor: Usuario



ACTOR		SISTEMA		
Paso	Acción	Paso	Acción	Excepción
1	Selecciona una póliza emitida	2	Muestra interfaz de una póliza emitida	
3	Selecciona opción para hacer un endoso B sobre la póliza seleccionada	4	Muestra interfaz de un endoso B	
5	Usuario ingresa información que va a ser modificada en la póliza	6	Valida información del endoso B	
		7	Si la información es correcta guarda el endoso en la base de datos e imprime endoso B	Si no es correcta la información, muestra los mensajes con el dato erróneo en el endoso

- Caso de uso: Solicitud de cancelación de una póliza
Actor: Usuario



ACTOR		SISTEMA		
Paso	Acción	Paso	Acción	Excepción
1	Selecciona una póliza emitida	2	Muestra interfaz de una póliza emitida	
3	Selecciona opción para hacer un endoso D de cancelación sobre la póliza seleccionada	4	Muestra interfaz de un endoso D de cancelación	
5	Usuario ingresa información del endoso D de cancelación	6	Valida información del endoso D de cancelación	
		7	Si la información es correcta guarda el endoso en la base de datos e imprime endoso D	Si no es correcta la información, muestra los mensajes con el dato erróneo en el endoso

La seguridad de la información recabada por los agentes en cada caso de uso dará la capacidad de resistencia para nuestro sistema de punto de venta en la que su comportamiento determinará el nivel de confianza ante los accidentes o acciones ilícitas o malintencionadas que comprometan la disponibilidad, autenticidad, integridad y confidencialidad de los datos almacenados y transmitidos.

4.2 Fase de diseño

4.2.1 Diseño arquitectónico

El diseño arquitectónico del sistema es en 3 capas, debido a que, esto es un enfoque a sistemas de tipo CASE o también llamados DESKTOP, como se muestra en la figura 2.

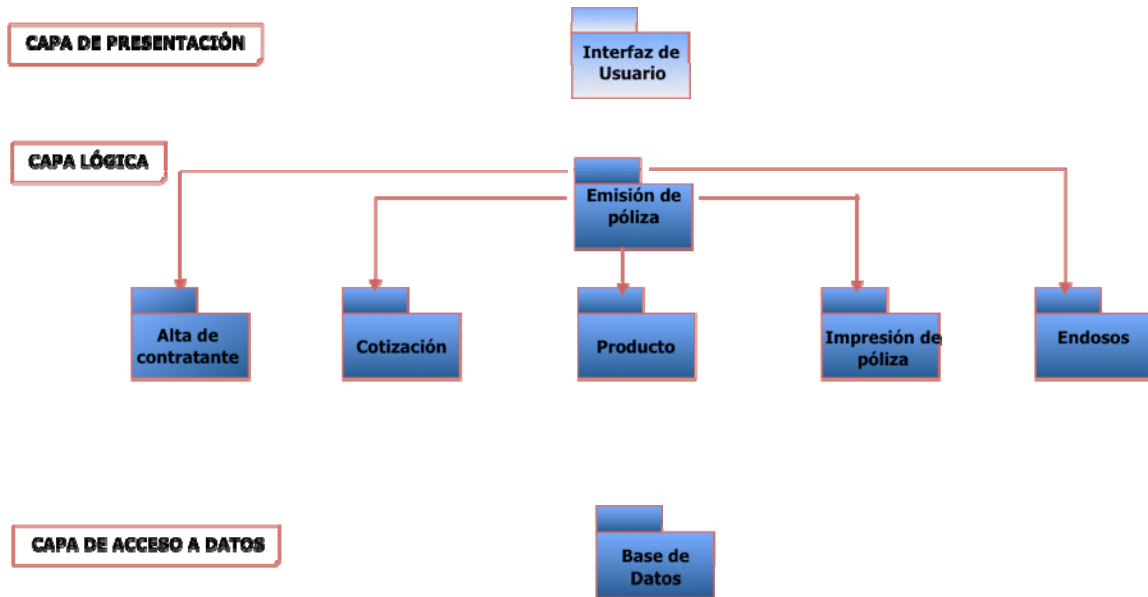


Figura 2. Diseño arquitectónico del sistema

4.2.2 Diseño de datos

Para iniciar con el diseño del modelo de base de datos, debe cumplir con la siguiente estructura de negocio:

- Reflejar la estructura del problema en el mundo real para los agentes de seguros.
- Ser capaz de representar todos los datos esperados, incluso con el paso del tiempo.
- Evitar el almacenamiento de información redundante.
- Proporcionar un acceso eficaz a los datos.
- Mantener la integridad de los datos a lo largo del tiempo.
- Tener claridad, coherencia y de facilidad de comprensión.

El objetivo de proteger la información es la misión de cualquier organización, teniendo en cuenta las diferentes dimensiones como:

- Disponibilidad: Debe de haber disposición de los servicios en cualquier momento que el cliente requiera y cada vez que sea necesario. La carencia de disponibilidad supone una interrupción del servicio. La disponibilidad afecta directamente a la productividad de las organizaciones. Haciendo referencia a que el incremento de productividad es un objetivo por parte de este sistema.
- Integridad: Deben de existir algoritmos que verifiquen la autenticidad de la información, ya que ante una mala validación, la información puede aparecer manipulada, corrupta o incompleta. La integridad puede afectar directamente al correcto desempeño de las funciones de una organización. Por lo tanto, para este sistema se deben tener los procedimientos adecuados que verifiquen la información al momento de registrarse.
- Confidencialidad: La información por cada agente de venta debe ser personal y resguardada por clave de emisor y clave de agente. En contra de la confidencialidad o secreto pueden darse fugas y filtraciones de información, así como accesos no autorizados a información de otros agentes. La confidencialidad es una propiedad de difícil recuperación, pudiendo minar la confianza de los demás en la organización que no es diligente en el mantenimiento del secreto y pudiendo suponer el incumplimiento de leyes y compromisos contractuales relativos a la custodia de los datos.
- Autenticidad: El agente de ventas debe ser quien se hace responsable de la información proporcionada, ya que las pólizas pasarán por una selección de riesgos por parte de la empresa con el fin verificar si existe incumplimientos o errores. Contra la autenticidad se dan suplantaciones y engaños que buscan realizar un fraude, es por ello que la autenticidad es la base para poder luchar contra el repudio y, como tal, fundamenta el comercio electrónico o la administración electrónica, permitiendo confiar sin papeles ni de presencia física.

A continuación se empieza a diseñar la estructura de datos para la información de tipo corporativo. Es decir, todas las características de una persona registrada en la base de datos.

Los campos subrayados dentro de las tablas son llaves primarias de la entidad.

La entidad principal de la información corporativa es PERSONA, sobre ella gira el entorno del sistema, ya que contiene los elementos para iniciar el cálculo de una póliza, ver figura 3.

PERSONA	
<u>ID_PERSONA</u>	<u>VARCHAR2(16)</u>
A_PATERNO	VARCHAR2(30)
A_MATERNO	VARCHAR2(30)
NOMBRE	VARCHAR2(30)
F_NACIMIENTO	DATE
CVE_GENERO	VARCHAR2(6)
CURP	VARCHAR2(30)
RAZON_SOCIAL	VARCHAR2(90)
ID_CONTACTO	VARCHAR2(16)
CVE_TIPO_PERSONA	VARCHAR2(6)
ID_DOMICILIO_PARTICULAR	VARCHAR2(16)
ID_DOMICILIO_OFNA	VARCHAR2(16)
RFC	VARCHAR2(15)
TEL_PARTICULAR	VARCHAR2(30)
TEL_OFNA_DIRECTO	VARCHAR2(30)
TEL_OFNA_CONMUTADOR	VARCHAR2(30)
EXT_OFNA	VARCHAR2(5)
TEL_MOVIL	VARCHAR2(30)
FAX_PARTICULAR	VARCHAR2(30)
FAX_OFNA	VARCHAR2(30)
TEL_RLOCALIZADOR	VARCHAR2(30)
PIN_RLOCALIZADOR	VARCHAR2(10)
EMAIL	VARCHAR2(30)
EPAGE_WWW	VARCHAR2(50)
CVE_NACIONALIDAD	VARCHAR2(6)
TX_PROFESION	VARCHAR2(100)
TX_GIROACTIVIDAD	VARCHAR2(100)
FH_CONSTITUCION	DATE

Figura 3. Tabla de atributos para la entidad PERSONA

Las entidades relacionadas con PERSONA son las siguientes:

La entidad UNIDAD_NEGOCIO corresponde a la característica del sistema que identifican a un flujo especial; el rubro de negocio en que la empresa aseguradora se desempeña, ver figura 4.

UNIDAD_NEGOCIO	
<u>ID_UNIDAD_NEGOCIO</u>	NUMBER(8)
UNIDAD_NEGOCIO	VARCHAR2(30)
ID_PERSONA	VARCHAR2(16)
ID_UNIDAD_PADRE	NUMBER(8)
CVE_TIPO_UNIDAD_NEGOCIO	VARCHAR2(6)

Figura 4. Tabla de atributos para la entidad UNIDAD_NEGOCIO

La entidad DOMICILIO identifica la localización de cualquier persona, colocando los datos de PAIS, ESTADO, MUNICIPIO y CODIGO_POSTAL en general representa la dirección en que está ubicada físicamente una persona, ver figura 5.

DOMICILIO	
<u>ID_DOMICILIO</u>	VARCHAR2(16)
ID_PAIS	NUMBER(4)
ID_ESTADO	NUMBER(3)
CODIGO_POSTAL	CHAR(5)
ID_USUARIO_CP	NUMBER(2)
COLONIA	VARCHAR2(50)
CALLE	VARCHAR2(50)
NUMERO_OFICIAL	VARCHAR2(5)
NUMERO_INTERIOR	VARCHAR2(5)
TX_REFERENCIA	VARCHAR2(2000)

PAIS	
<u>ID_PAIS</u>	NUMBER(4)
PAIS	VARCHAR2(50)

ESTADO	
<u>ID_PAIS</u>	NUMBER(4)
<u>ID_ESTADO</u>	NUMBER(3)
ESTADO	VARCHAR2(50)

MUNICIPIO	
<u>ID_PAIS</u>	NUMBER(4)
<u>ID_ESTADO</u>	NUMBER(3)
<u>ID_MUNICIPIO</u>	NUMBER(5)
MUNICIPIO	VARCHAR2(100)

CODIGO_POSTAL	
<u>ID_PAIS</u>	NUMBER(4)
<u>CODIGO_POSTAL</u>	CHAR(5)
<u>ID_USUARIO_CP</u>	NUMBER(2)
ID_ESTADO	NUMBER(3)
ID_MUNICIPIO	NUMBER(5)
USUARIO_CP	VARCHAR2(100)
PCT_IVA	NUMBER(8,4)
ZONA_TEV	NUMBER(3)
ZONA_SISMICA	VARCHAR2(5)
ZONA_HURACAN	NUMBER(3)
ZONA_AUTO	NUMBER(5)

Figura 5. Tabla de atributos para la entidad DOMICILIO

La relación de las entidades mencionadas forma el primer modelo para administrar todo tipo de persona en el sistema, el esquema tiene un tercer grado de normalización (3FN), ver figura 6.

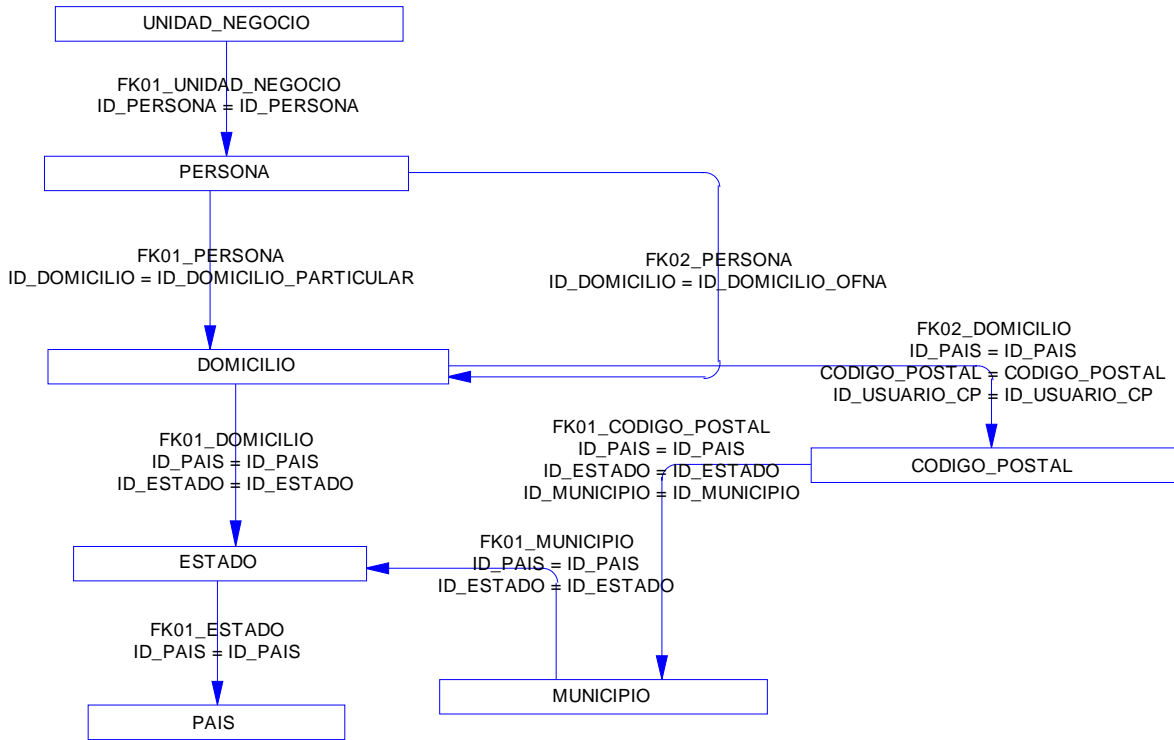


Figura 6. Diagrama entidad-relación para un tipo PERSONA

La entidad EMI_POLIZA registra información de emisión de una póliza, ver figura 7.

EMI_POLIZA	
<u>ID_UNIDAD_NEGOCIO</u>	NUMBER(8)
<u>ID_EMITOR</u>	VARCHAR2(8)
<u>ID_COTIZACION</u>	NUMBER(8)
<u>ID_RENOVACION</u>	NUMBER(2)
ANIO_OPERACION	NUMBER(2)
ID_CONTRATANTE	VARCHAR2(16)
ID_AGENTE	NUMBER(8)
F_INICIO_VIG	DATE
F_FIN_VIG	DATE
ID_PRODUCTO	NUMBER(6)
ID_VERSION	NUMBER(3)
B_RENOVABLE	CHAR(1)
B_RENOVADA	CHAR(1)
B_RECALCULO_ANUAL	CHAR(1)
FH_ULT_RECALCULO	DATE
CVE_TIPO_ADMON	VARCHAR2(6)
ID_MONEDA	NUMBER(3)
ID_CONDUCTO_COBRO	NUMBER(5)
ID_FORMA_PAGO	NUMBER(5)
CVE_NIVEL_COBRO	VARCHAR2(6)
ID_DESGLOSE_IMPORTE	NUMBER(12)
PCT_DESCTO_POL	NUMBER(8,4)
PCT_BON_COMIS	NUMBER(8,4)
PCT_RCGO_FPAGO	NUMBER(8,4)
PCT_IVA	NUMBER(8,4)
CVE_SIT_POLIZA	VARCHAR2(6)
CVE_MTVO_SIT_POLIZA	VARCHAR2(6)
NUM_POLIZA	VARCHAR2(16)
NUM_POLIZA_ANT	VARCHAR2(16)
ID_BITACORA	VARCHAR2(16)
B_DOM_CORRESP_CTE	CHAR(1)
ID_DOM_CORRESP	VARCHAR2(16)
CVE_ENVIO_RECIBO	VARCHAR2(6)
CVE_PER_ENVIO_RECIBO	VARCHAR2(6)
FH_CREACION	DATE
ID_USUARIO_CREACION	NUMBER(8)
FH_EMISION	DATE
ID_USUARIO_EMISION	NUMBER(8)
FH_CANCELACION	DATE
FH_SOL_AUTORIZACION	DATE
FH_SOL_PROCESADA	DATE
ID_USUARIO_PROCESO	NUMBER(8)
FH_ULT_MOD	DATE
TX_REFERENCIA	VARCHAR2(2000)
B_DERIVACION_AE	CHAR(1)
FH_DERIVACION_AE	DATE
CVE_DERIVACION_SC	VARCHAR2(6)
FH_DERIVACION_SC	DATE
FH_ULT_MOD_AE	DATE
CVE_MOD_AE	VARCHAR2(6)
TX_COND_COB_CLUSTER	LONG
COND_COB_NUM_CUENTA	VARCHAR2(50)
COND_COB_DATO1	VARCHAR2(100)
COND_COB_DATO2	VARCHAR2(100)
COND_COB_DATO3	VARCHAR2(100)
COND_COB_DATO4	VARCHAR2(100)
COND_COB_DATO5	VARCHAR2(100)
COND_COB_DATO6	VARCHAR2(100)
B_ERROR	CHAR(1)
B_AUTORIZAR	CHAR(1)
B_PRUEBA_PROD	CHAR(1)
B_PRUEBA_AGTE	CHAR(1)
B_POLSOSPECHOSA_LD	VARCHAR2(1)
B_DOCUMENTACION	VARCHAR2(1)

Figura 7. Tabla de atributos para la entidad EMI_POLIZA

La entidad EMI_ASEGURADO contiene la identificación de las personas aseguradas dentro de una póliza, ver figura 8.

EMI_ASEGURADO	
<u>ID_UNIDAD_NEGOCIO</u>	NUMBER(8)
<u>ID_EMITOR</u>	VARCHAR2(8)
<u>ID_COTIZACION</u>	NUMBER(8)
<u>ID_RENOVACION</u>	NUMBER(2)
<u>ID_PERSONA</u>	VARCHAR2(16)
NUM_SECUENCIA	NUMBER(3)

Figura 8. Tabla de atributos para la entidad EMI_ASEGURADO

La entidad EMI_CONTRATANTE referencia a la persona que acepta el contrato de una póliza, ver figura 9.

EMI_CONTRATANTE	
<u>ID_CONTRATANTE</u>	VARCHAR2(16)
<u>ID_PERSONA</u>	VARCHAR2(16)
<u>ID_AGRUPADOR</u>	VARCHAR2(16)
<u>FH_CREACION</u>	DATE
<u>ID_USUARIO_CREACION</u>	NUMBER(8)

Figura 9. Tabla de atributos para la entidad EMI_CONTRATANTE

La entidad EMI_AGENTE relaciona al agente de ventas, que emitió la póliza, ver figura 10.

EMI_AGENTE	
<u>ID_UNIDAD_NEGOCIO</u>	NUMBER(8)
<u>ID_EMITOR</u>	VARCHAR2(8)
<u>ID_COTIZACION</u>	NUMBER(8)
<u>ID_RENOVACION</u>	NUMBER(2)
<u>ID_AGENTE</u>	NUMBER(8)
PCT_PARTICIPACION	NUMBER(8,4)
IMP_COMIS_PN	NUMBER(16,2)
IMP_COMIS_FPAGO	NUMBER(16,2)
B_PRINCIPAL	CHAR(1)

Figura 10. Tabla de atributos para la entidad EMI_AGENTE

La entidad EMI_RECIBO describe información del número de pagos que tiene que realizar un contratante de acuerdo a la forma, ya sea anual, mensual, trimestral o semestral, ver figura 11.

EMI_RECIBO	
<u>NUM_RECIBO</u>	<u>VARCHAR2(18)</u>
ID_UNIDAD_NEGOCIO	NUMBER(8)
ID_EMITOR	VARCHAR2(8)
ID_COTIZACION	NUMBER(8)
ID_RENOVACION	NUMBER(2)
ANIO_OPERACION	NUMBER(2)
ID_RECIBO	NUMBER(10)
F_INICIO_COBERTURA	DATE
F_FIN_COBERTURA	DATE
F_LIM_PAGO	DATE
F_CONTABLE	DATE
ID_MONEDA	NUMBER(3)
SERIE	VARCHAR2(5)
FOLIO	NUMBER(10)
CVE_SIT_RECIBO	VARCHAR2(6)
NUM_RECIBO_SC	VARCHAR2(18)
IMP_COMIS_BON_PN	NUMBER(16,2)
IMP_PN_BON_SAI	NUMBER(16,2)
IMP_PN_BON_NAI	NUMBER(16,2)
IMP_DEREMIS_TOT_SAI	NUMBER(16,2)
IMP_DEREMIS_TOT_NAI	NUMBER(16,2)
IMP_RCGO_FPAGO_SAI	NUMBER(16,2)
IMP_RCGO_FPAGO_NAI	NUMBER(16,2)
IMP_COMIS_BON_FPAGO	NUMBER(16,2)
IMP_COMIS_BON_TOT	NUMBER(16,2)
IMP_IVA	NUMBER(16,2)
IMP_OTROS_IMPTOS_SAI	NUMBER(16,2)
IMP_OTROS_IMPTOS_NAI	NUMBER(16,2)
IMP_PT_SAI	NUMBER(16,2)
IMP_PT_NAI	NUMBER(16,2)
FH_ULT_MOD	DATE

Figura 11. Tabla de atributos para la entidad EMI_RECIBO

La entidad de EMI_RECIBO_DESGLOSADO agrupa los importes generados en cada cobertura contratada en la póliza; la sumatoria de estos importes arroja la prima total a pagar, ver figura 12.

EMI_RECIBO_DESGLOSADO	
<u>NUM_RECIBO</u>	<u>VARCHAR2(18)</u>
<u>ID_LINEA_NEGOCIO</u>	<u>NUMBER(3)</u>
<u>ID_INCISO</u>	<u>NUMBER(5)</u>
<u>ID_COBERTURA</u>	<u>NUMBER(8)</u>
ID_UNIDAD_NEGOCIO	NUMBER(8)
ID_EMITOR	VARCHAR2(8)
ID_COTIZACION	NUMBER(8)
ID_RENOVACION	NUMBER(2)
ID_RECIBO	NUMBER(10)
IMP_COMIS_BON_PN	NUMBER(16,2)
IMP_PN_BON_SAI	NUMBER(16,2)
IMP_PN_BON_NAI	NUMBER(16,2)
IMP_DEREMIS_TOT_SAI	NUMBER(16,2)
IMP_DEREMIS_TOT_NAI	NUMBER(16,2)
IMP_RCGO_FPAGO_SAI	NUMBER(16,2)
IMP_RCGO_FPAGO_NAI	NUMBER(16,2)
IMP_COMIS_BON_FPAGO	NUMBER(16,2)
IMP_COMIS_BON_TOT	NUMBER(16,2)
IMP_IVA	NUMBER(16,2)
IMP_OTROS_IMPTOS_SAI	NUMBER(16,2)
IMP_OTROS_IMPTOS_NAI	NUMBER(16,2)
IMP_PT_SAI	NUMBER(16,2)
IMP_PT_NAI	NUMBER(16,2)

Figura 12. Tabla de atributos para la entidad EMI_RECIBO_DESGLOSADO

La entidad EMI_COBERTURA contiene las características de cualquier tipo de cobertura disponible para una póliza de acuerdo al tipo de producto que se esté contratando, ver figura 13.

EMI_COBERTURA	
<u>ID_UNIDAD_NEGOCIO</u>	NUMBER(8)
<u>ID_EMITOR</u>	VARCHAR2(8)
<u>ID_COTIZACION</u>	NUMBER(8)
<u>ID_RENOVACION</u>	NUMBER(2)
<u>ID_LINEA_NEGOCIO</u>	NUMBER(3)
<u>ID_INCISO</u>	NUMBER(5)
<u>ID_COBERTURA</u>	NUMBER(8)
CVE_SIT_COBERTURA	VARCHAR2(6)
CVE_MTVO_SIT_COBERTURA	VARCHAR2(6)
F_INICIO_VIG	DATE
F_FIN_VIGENCIA	DATE
IMP_SUMAASEG_LIMITE	NUMBER(16,2)
IMP_SUMAASEG_ELEMENTO	NUMBER(16,2)
PCT_DESCTO_COB	NUMBER(8,4)
IMP_DEDUCIBLE_1SIN	NUMBER(16,2)
IMP_DEDUCIBLE_2SIN	NUMBER(16,2)
IMP_DEDUCIBLE_3SIN	NUMBER(16,2)
IMP_DEDUCIBLE_4SIN	NUMBER(16,2)
IMP_DEDUCIBLE_MIN	NUMBER(16,2)
IMP_DEDUCIBLE_MAX	NUMBER(16,2)
IMP_COASEGURO	NUMBER(16,2)
ID_DESGLOSE_IMPORTE	NUMBER(12)
TX_CLUSTER	LONG
TX_REFERENCIA	VARCHAR2(2000)
B_ERROR	CHAR(1)

Figura 13. Tabla de atributos de la entidad EMI_COBERTURA

La entidad EMI_COBERTURA_DATO indica los valores predeterminados para cada cobertura seleccionada por el agente al realizar una cotización de póliza, ver figura 14.

EMI_COBERTURA_DATO	
<u>ID_UNIDAD_NEGOCIO</u>	NUMBER(8)
<u>ID_EMITOR</u>	VARCHAR2(8)
<u>ID_COTIZACION</u>	NUMBER(8)
<u>ID_RENOVACION</u>	NUMBER(2)
<u>ID_LINEA_NEGOCIO</u>	NUMBER(3)
<u>ID_INCISO</u>	NUMBER(5)
<u>ID_COBERTURA</u>	NUMBER(8)
DATO	VARCHAR2(30)
VALOR	VARCHAR2(200)

Figura 14. Tabla de atributos de la entidad EMI_COBERTURA_DATO

La entidad EMI_DESGLOSEIMPORTE está relacionada con las características de todos los importes totales asociados a la prima total de una póliza, ver figura 15.

Adicionalmente, esta entidad representa la relación existente con otras identidades relacionadas con importes que constituyen la prima de una póliza de seguro.

EMI_DESGLOSEIMPORTE	
<u>ID_UNIDAD_NEGOCIO</u>	<u>NUMBER(8)</u>
<u>ID_EMITOR</u>	<u>VARCHAR2(8)</u>
<u>ID_COTIZACION</u>	<u>NUMBER(8)</u>
<u>ID_RENOVACION</u>	<u>NUMBER(2)</u>
<u>ID_DESGLOSE_IMPORTE</u>	<u>NUMBER(12)</u>
IMP_PN_REAL_SAI	NUMBER(16,2)
IMP_PN_REA_NAI	NUMBER(16,2)
IMP_DESCTO_POL_SAI	NUMBER(16,2)
IMP_DESCTO_POL_NAI	NUMBER(16,2)
IMP_DESCTO_VOL_SAI	NUMBER(16,2)
IMP_DESCTO_VOL_NAI	NUMBER(16,2)
IMP_DESCTO_BSIN_SAI	NUMBER(16,2)
IMP_DESCTO_BSIN_NAI	NUMBER(16,2)
IMP_DESCTO_LIN_SAI	NUMBER(16,2)
IMP_DESCTO_LIN_NAI	NUMBER(16,2)
IMP_DESCTO_COB_SAI	NUMBER(16,2)
IMP_DESCTO_COB_NAI	NUMBER(16,2)
IMP_DESCTO_OTROS_SAI	NUMBER(16,2)
IMP_DESCTO_OTROS_NAI	NUMBER(16,2)
IMP_DESCTO_TOT_SAI	NUMBER(16,2)
IMP_DESCTO_TOT_NAI	NUMBER(16,2)
IMP_PN_DESC_SAI	NUMBER(16,2)
IMP_PN_DESC_NAI	NUMBER(16,2)
IMP_COMIS_PN	NUMBER(16,2)
IMP_BON_COMIS_PN_SAI	NUMBER(16,2)
IMP_BON_COMIS_PN_NAI	NUMBER(16,2)
IMP_COMIS_BON_PN	NUMBER(16,2)
IMP_PN_BON_SAI	NUMBER(16,2)
IMP_PN_BON_NAI	NUMBER(16,2)
IMP_DEREMIS_PROD_SAI	NUMBER(16,2)
IMP_DEREMIS_PROD_NAI	NUMBER(16,2)
IMP_DEREMIS_LIN_SAI	NUMBER(16,2)
IMP_DEREMIS_LIN_NAI	NUMBER(16,2)
IMP_DEREMIS_TOT_SAI	NUMBER(16,2)
IMP_DEREMIS_TOT_NAI	NUMBER(16,2)
IMP_RCGO_FPAGO_SAI	NUMBER(16,2)
IMP_RCGO_FPAGO_NAI	NUMBER(16,2)
IMP_COMIS_FPAGO	NUMBER(16,2)
IMP_COMIS_TOT	NUMBER(16,2)
IMP_BON_COMIS_FPAGO_SAI	NUMBER(16,2)
IMP_BON_COMIS_FPAGO_NAI	NUMBER(16,2)
IMP_BON_COMIS_TOT_SAI	NUMBER(16,2)
IMP_BON_COMIS_TOT_NAI	NUMBER(16,2)
IMP_COMIS_BON_FPAGO	NUMBER(16,2)
IMP_COMIS_BON_TOT	NUMBER(16,2)
IMP_STOT_PN_SAI	NUMBER(16,2)
IMP_STOT_PN_NAI	NUMBER(16,2)
IMP_IVA	NUMBER(16,2)
IMP_OTROS_IMPTOS_SAI	NUMBER(16,2)
IMP_OTROS_IMPTOS_NAI	NUMBER(16,2)
IMP_PT_SAI	NUMBER(16,2)
IMP_PT_NAI	NUMBER(16,2)

Figura 15. Tabla de atributos para la entidad EMI_DESGLOSEIMPORTE

La entidad EMI_INCISO asocia el o los tipos de planes contratados a una póliza. A su vez estos tipos de planes están asociados a un desglose de importes a cobrar, ver figura 16.

EMI_INCISO	
<u>ID_UNIDAD_NEGOCIO</u>	NUMBER(8)
<u>ID_EMITOR</u>	VARCHAR2(8)
<u>ID_COTIZACION</u>	NUMBER(8)
<u>ID_RENOVACION</u>	NUMBER(2)
<u>ID_LINEA_NEGOCIO</u>	NUMBER(3)
<u>ID_INCISO</u>	NUMBER(5)
ID_DESGLOSE_IMPORTE	NUMBER(12)
ID_INCISO_POLIZA_ANT	NUMBER(5)
CVE_TIPO_UR	VARCHAR2(6)
ID_PAQUETE	NUMBER(3)
F_INICIO_VIG	DATE
F_FIN_VIG	DATE
NUM_UR	NUMBER(5)
PCT_DESCTO_B SIN	NUMBER(8,4)
PCT_DESCTO_OTROS	NUMBER(8,4)
CVE_SIT_INCISO	VARCHAR2(6)
CVE_MTVO_SIT_INCISO	VARCHAR2(6)
TX_REFERENCIA	VARCHAR2(2000)
B_ERROR	CHAR(1)
B_AUTORIZAR	CHAR(1)
B_OBLIGA_COT	CHAR(1)
B_OBLIGA_POL	CHAR(1)

Figura 16. Tabla de atributos para la entidad EMI_INCISO

La entidad EMI_LINEA corresponde a la característica principal del tipo de producto que está manejando en la contratación de una póliza, ver figura 17.

EMI_LINEA	
<u>ID_UNIDAD_NEGOCIO</u>	NUMBER(8)
<u>ID_EMITOR</u>	VARCHAR2(8)
<u>ID_COTIZACION</u>	NUMBER(8)
<u>ID_RENOVACION</u>	NUMBER(2)
<u>ID_LINEA_NEGOCIO</u>	NUMBER(3)
NUM_UR	NUMBER(5)
ID_DESGLOSE_IMPORTE	NUMBER(12)
PCT_DESCTO_VOL	NUMBER(8,4)
PCT_DESCTO_LIN	NUMBER(8,4)

Figura 17. Tabla de atributos para la entidad EMI_LINEA

El resultado de la asociación de las entidades mencionadas es el siguiente modelo, el esquema tiene un tercer grado de normalización (3FN), ver figura 18.

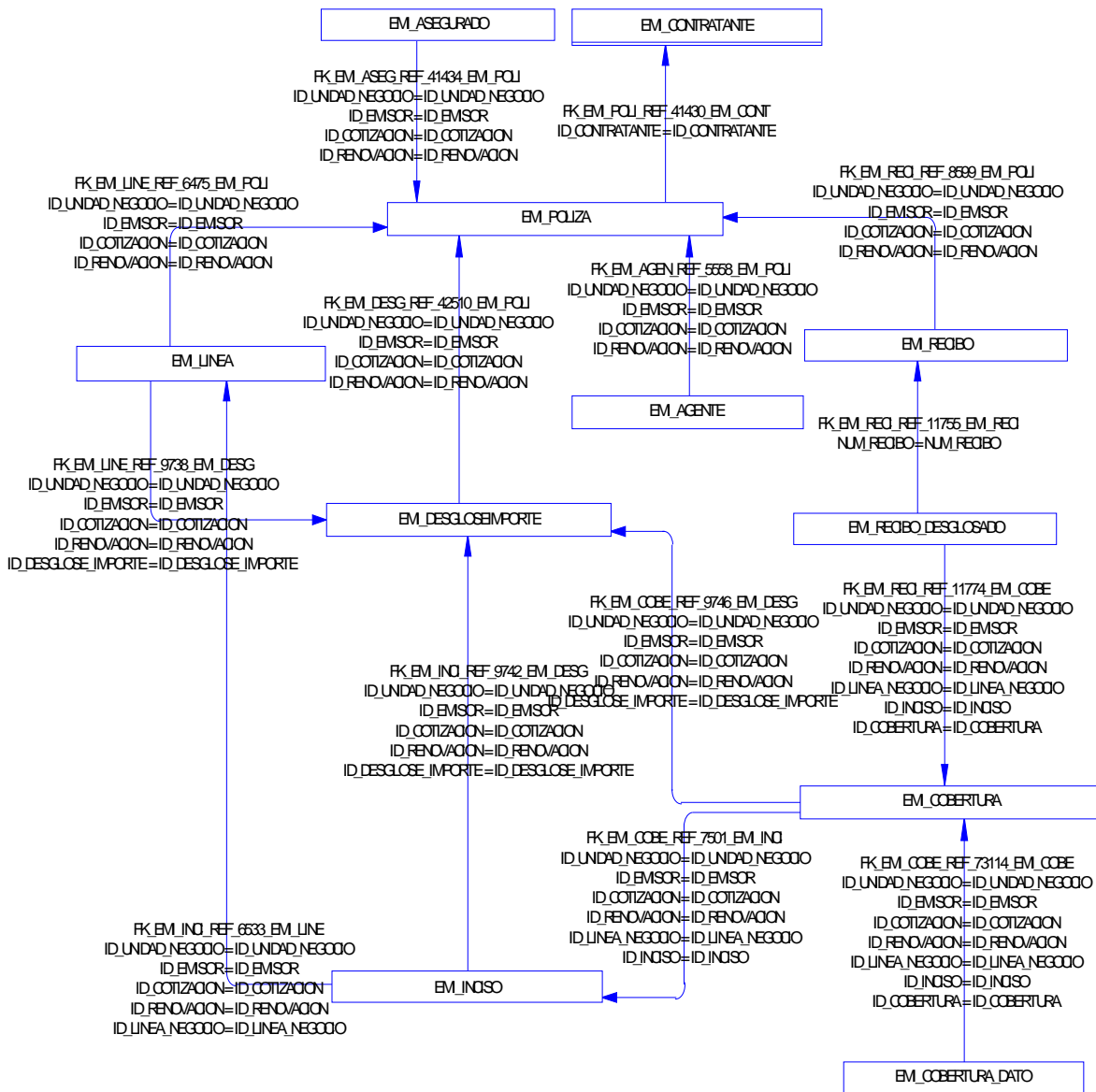


Figura 18. Diagrama entidad-relación para una entidad tipo PÓLIZA

La entidad SEG_USUARIO identificará las personas que están dadas de alta en el sistema por medio de claves de usuario y contraseñas para poder operar el sistema, ver figura 19.

SEG_USUARIO	
<u>ID_USUARIO</u>	NUMBER(8)
ID_PERSONA	VARCHAR2(16)
NOMBRE_USUARIO	VARCHAR2(15)
CONTRASENIA	VARCHAR2(15)
CVE_SIT_USUARIO	VARCHAR2(6)

Figura 19. Tabla de atributos para le entidad SEG_USUARIO

La entidad SEG_SIST_UN_USUARIO registra a usuarios asignados a un sistema a operar. Es decir se registra en el sistema un identificador (ID_SISTEMA) y se le asocia a cada usuario registrado para poder operar el sistema, ver figura 20.

SEG_SIST_UN_USUARIO	
<u>ID_SISTEMA</u>	NUMBER(8)
<u>ID_UNIDAD_NEGOCIO</u>	NUMBER(8)
<u>ID_USUARIO</u>	NUMBER(8)

Figura 20. Tabla de atributos para la entidad SEG_SIST_UN_USUARIO

La entidad SEG_SIST_UN es la asociación de un sistema a una unidad de negocio. El sistema estará asociado a la unidad de negocio dada de alta por la empresa aseguradora, ver figura 21.

SEG_SIST_UN	
<u>ID_SISTEMA</u>	NUMBER(8)
<u>ID_UNIDAD_NEGOCIO</u>	NUMBER(8)

Figura 21. Tabla de atributos para la entidad SEG_SIST_UN

La entidad SEG_SISTEMA es la identificación de las características del sistema dado de alta en la base de datos, ver figura 22.

SEG_SISTEMA	
<u>ID_SISTEMA</u>	NUMBER(8)
SISTEMA	VARCHAR2(30)
VERSION	VARCHAR2(5)
NOMBRE_EXE	VARCHAR2(50)
TX_DERECHOS	VARCHAR2(200)
TX_IMAGEN	LONG RAW
TX_REFERENCIA	VARCHAR2(200)

Figura 22. Tabla de atributos para la entidad SEG_SISTEMA

La entidad SEG_SISTEMA_OBJETO registra todos los objetos que componen el sistema, ver figura 23.

SEG_SISTEMA_OBJETO	
<u>ID_SISTEMA</u>	NUMBER(8)
<u>ID_OBJETO</u>	NUMBER(6)

Figura 23. Tabla de atributos para la entidad SEG_SISTEMA_OBJETO

La entidad SEG_OBJETO es el registro de todos los objetos PADRE, que hacen que el sistema tenga su razón de ser en cuanto a operación y funcionamiento, ver figura 24.

SEG_OBJETO	
<u>ID_OBJETO</u>	NUMBER(6)
OBJETO	VARCHAR2(30)
TX_DESCRIPCION	VARCHAR2(200)

Figura 24. Tabla de atributos para la entidad SEG_OBJETO

La entidad SEG_METODO es el registro de métodos asociados a objetos para controlar su acceso de ejecución, ver figura 24.

SEG_METODO	
<u>ID_OBJETO</u>	NUMBER(6)
<u>ID_METODO</u>	NUMBER(3)
METODO	VARCHAR2(30)
CVE_TIPO_METODO	VARCHAR2(6)

Figura 25. Tabla de atributos para la entidad SEG_METODO

La entidad SEG_USUARIO_PERFIL asocia un perfil a un usuario que a su vez tiene asignado un sistema en una unidad de negocio de la empresa, ver figura 26.

SEG_USUARIO_PERFIL	
<u>ID_UNIDAD_NEGOCIO</u>	NUMBER(8)
<u>ID_SISTEMA</u>	NUMBER(8)
<u>ID_USUARIO</u>	NUMBER(8)
<u>ID_PERFIL</u>	NUMBER(5)

Figura 26. Tabla de atributos para la entidad SEG_USUARIO_PERFIL

La entidad SEG_PERFIL es el catálogo de todos los tipos de perfil asociados al sistema, ver figura 27.

SEG_PERFIL	
<u>ID_SISTEMA</u>	NUMBER(8)
<u>ID_PERFIL</u>	NUMBER(5)
PERFIL	VARCHAR2(15)
TX_PERFIL	VARCHAR2(200)

Figura 27. Tabla de atributos para la entidad SEG_PERFIL

La entidad SEG_PERFIL_OBJETO es la relación de objetos del sistema asociados a un tipo de perfil. Es decir, de acuerdo al perfil se tendrá acceso o restricción de operar una funcionalidad del sistema, ver figura 28.

SEG_PERFIL_OBJETO	
<u>ID SISTEMA</u>	NUMBER(8)
<u>ID PERFIL</u>	NUMBER(5)
<u>ID OBJETO</u>	NUMBER(6)

Figura 28. Tabla de atributos para la entidad SEG_PERFIL_OBJETO

La entidad SEG_PERFIL_METODO también de acuerdo al perfil se accederá al objeto y a su vez estará ligado el tipo de método con permisos de ejecución, ver figura 29.

SEG_PERFIL_METODO	
<u>ID SISTEMA</u>	NUMBER(8)
<u>ID PERFIL</u>	NUMBER(5)
<u>ID OBJETO</u>	NUMBER(6)
<u>ID METODO</u>	NUMBER(3)

Figura 29. Tabla de atributos para la entidad SEG_PERFIL_METODO

El resultado de la asociación de todas las entidades mencionadas es el siguiente modelo, el esquema tiene un tercer grado de normalización (3FN), ver figura 30.

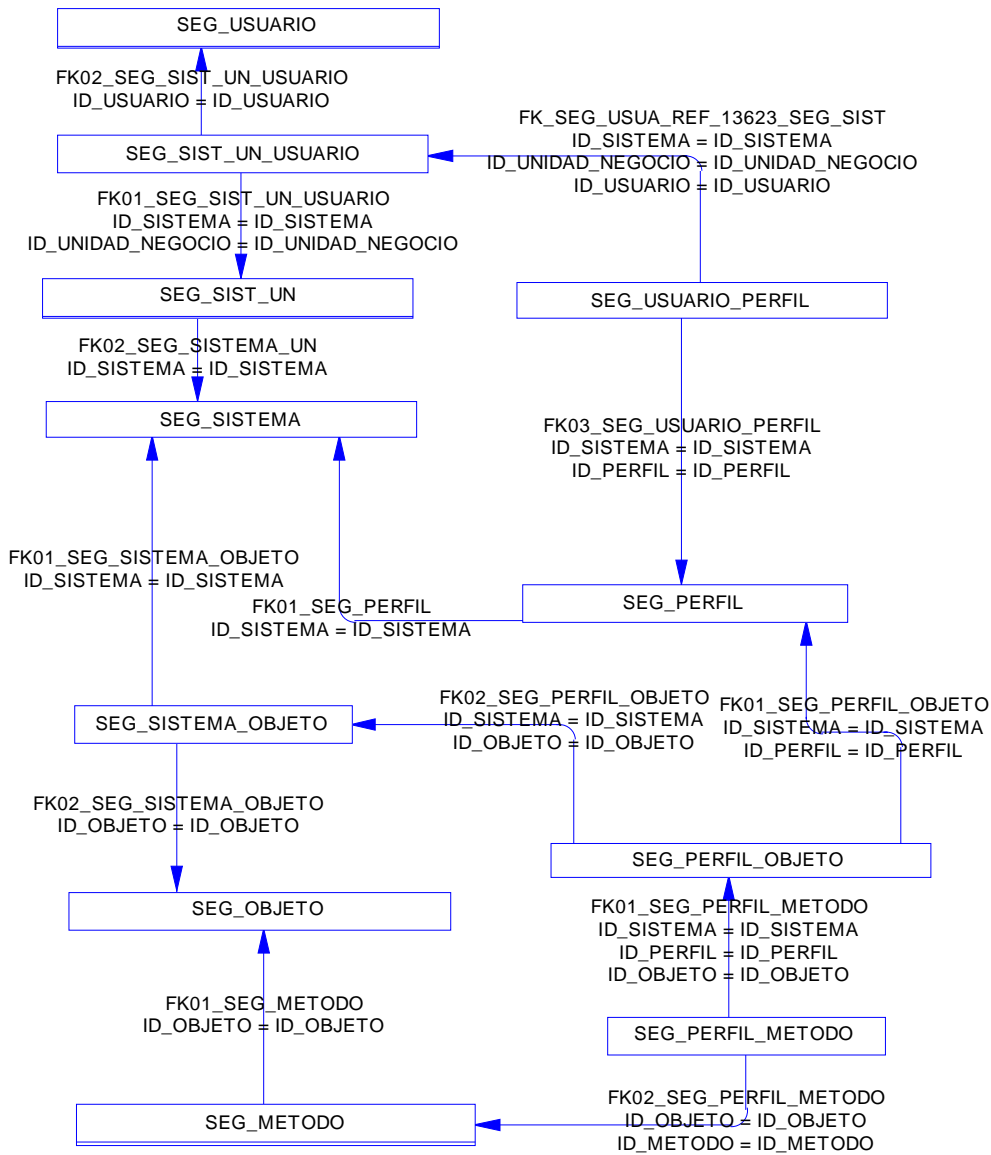


Figura 30. Diagrama entidad-relación para la entidad SEG_USUARIO

Una vez que se tiene el manejo de la información de seguridades, se procede al control de productos.

La entidad PRO_PRODUCTO permitirá llevar el registro de diferentes tipos de productos asignándoles un identificador único y describiendo sus diferentes características, ver figura 31.

PRO_PRODUCTO	
<u>ID_UNIDAD_NEGOCIO</u>	NUMBER(8)
<u>ID_PRODUCTO</u>	NUMBER(6)
PRODUCTO	VARCHAR2(30)
NOMBRE_CORTO_PROD	VARCHAR2(10)
B_RENUEVA	CHAR(1)
CVE_TIPO_PRODUCTO	VARCHAR2(6)

Figura 31. Tabla de atributos para la entidad PRO_PRODUCTO

La entidad PRO_PRODUCTO_VERSION, permite llevar el control de versiones para el producto, ver figura 32.

PRO_PRODUCTO_VERSION	
<u>ID_UNIDAD_NEGOCIO</u>	NUMBER(8)
<u>ID_PRODUCTO</u>	NUMBER(6)
<u>ID_VERSION</u>	NUMBER(3)
CVE_TIPO_PLAZO_VENTA	VARCHAR2(6)
PLAZO_MIN_VENTA	NUMBER(5)
PLAZO_MAX_VENTA	NUMBER(5)
PLAZO_DEF_VENTA	NUMBER(5)
DIAS_VALIDOS_COT	NUMBER(2)
B_RECALCULO_ANUAL	CHAR(1)
CVE_TIPO_RET_DIF	VARCHAR2(6)
NUM_DIAS_VENTA_RET	NUMBER(3)
NUM_DIAS_VENTA_DIF	NUMBER(3)
CVE_SIT_PRODUCTO	VARCHAR2(6)
TX_DESC_VERSION	VARCHAR2(2000)
ID_USUARIO_CREACION	NUMBER(8)
FH_CREACION	DATE
ID_USUARIO_ULT_MOD	NUMBER(8)
FH_ULT_MOD	DATE
ID_USUARIO_ACTIVACION	NUMBER(8)
FH_ACTIVACION	DATE
F_INI_VERSION	DATE
F_TER_VERSION	DATE
B_ULT_REG_DIA	CHAR(1)

Figura 32. Tabla de atributos para la entidad PRO_PRODUCTO_VERSION

La entidad PRO_LINEA_NEGOCIO es el catálogo de las líneas de negocio asociadas al producto. Una línea de negocio para la aseguradora es el tipo de rubro de venta: pólizas de gastos médicos, pólizas de vida, pólizas de inmuebles, pólizas educacionales, etc., ver figura 33.

PRO_LINEA_NEGOCIO	
<u>ID_UNIDAD_NEGOCIO</u>	NUMBER(8)
<u>ID_PRODUCTO</u>	NUMBER(6)
<u>ID_VERSION</u>	NUMBER(3)
<u>ID_LINEA_NEGOCIO</u>	NUMBER(3)
LINEA_NEGOCIO	VARCHAR2(30)
NOMBRE_CORTO_LN	VARCHAR2(10)
CVE_CONTABILIZACION	VARCHAR2(6)
NUM_SECUENCIA	NUMBER(3)
MIN_UR	NUMBER(3)
MAX_UR	NUMBER(3)
B_USA_CATEGORIA	CHAR(1)
ID_USUARIO_ULT_MOD	NUMBER(8)
FH_ULT_MOD	DATE

Figura 33. Tabla de atributos para la entidad PRO_LINEA_NEGOCIO

El resultado de la relación de las entidades mencionadas es el siguiente modelo, el esquema tiene un tercer grado de normalización (3FN), ver figura 34.

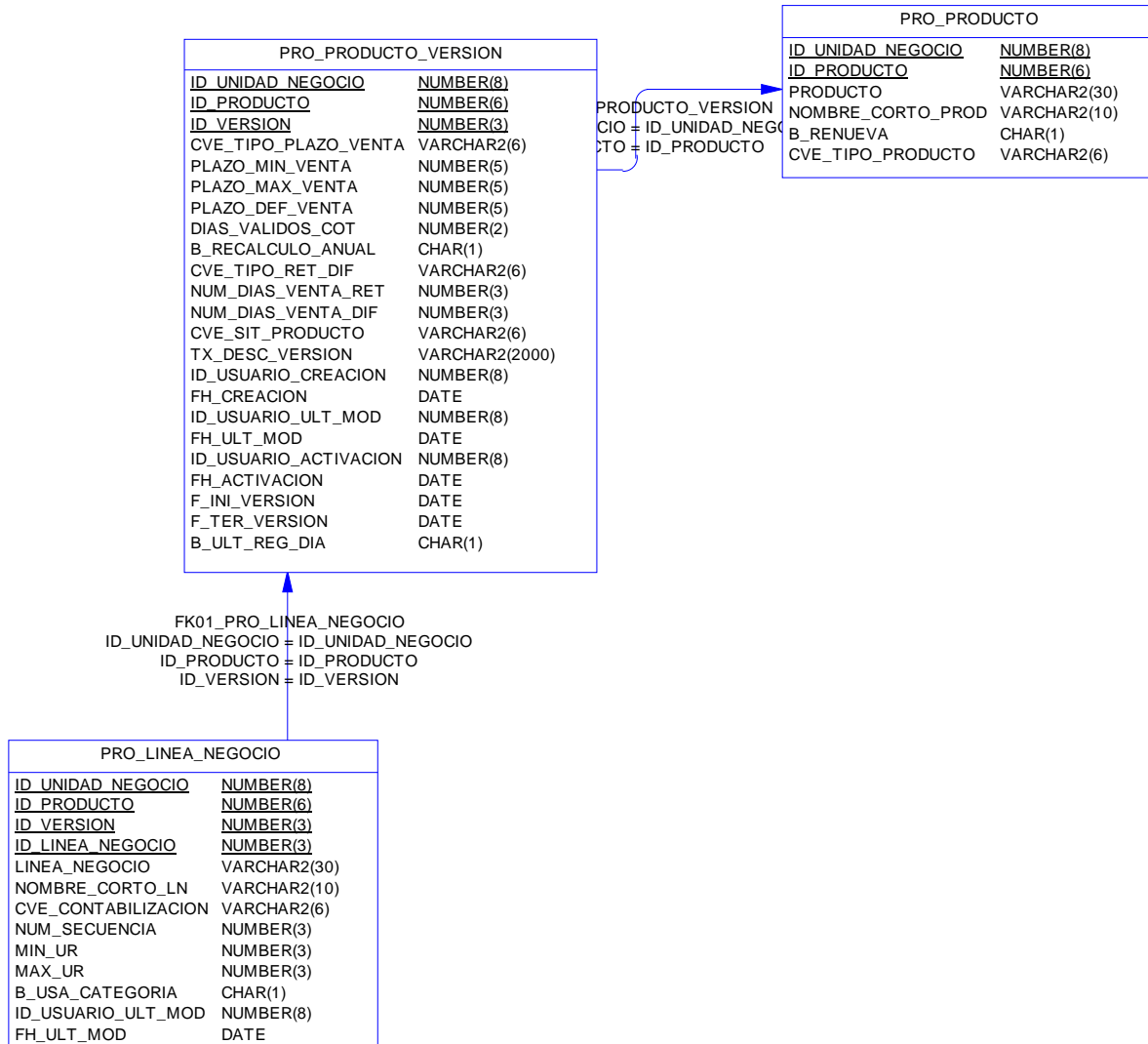


Figura 34. Diagrama entidad-relación para la entidad PRO_PRODUCTO

4.2.3 Diseño de interfaz

El primer diseño se basa en la forma de acceso al sistema, la seguridad de la información es primordial para tener confiabilidad, por lo tanto cada vez que se ejecute el sistema la interfaz de acceso será activada como se muestra en la figura 35.

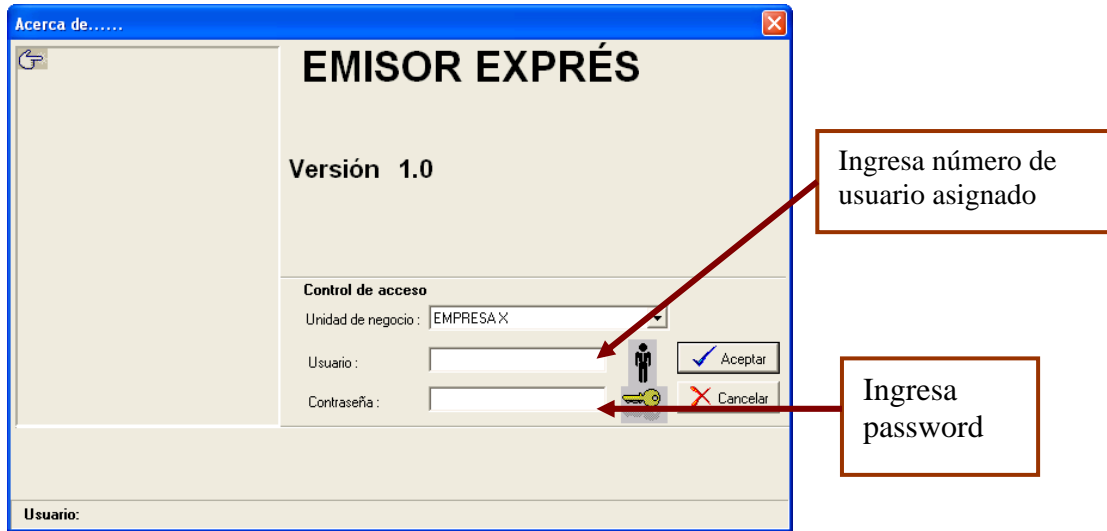


Figura 35. Acceso al sistema punto de venta portátil

Internamente, la clase que maneja el control de acceso contendrá los algoritmos que validarán contra la base de datos que el usuario ingresado es válido para operar el sistema.

En caso de que se intente más de tres veces acceder, el sistema se desactivará y se cerrará la aplicación primero mostrando los siguientes mensajes, ver figura 36.

La primera validación indicará que la contraseña es incorrecta.

La segunda validación indicará que se sobrepasó el número de intentos.

La tercera validación cierra la aplicación y muestra un mensaje de conexión fallida.

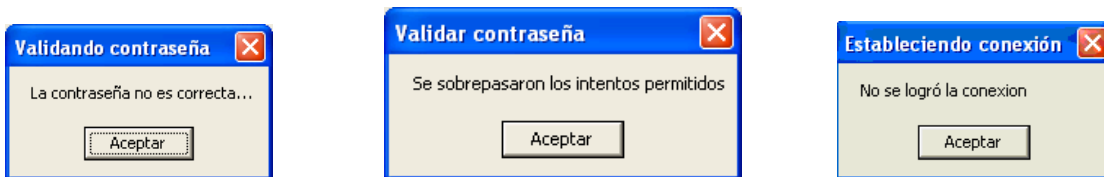


Figura 36. Validación de usuario y contraseña

Una vez ingresado al sistema con un usuario válido, se muestra un menú principal visible al usuario en cualquier navegación, ver figura 37.

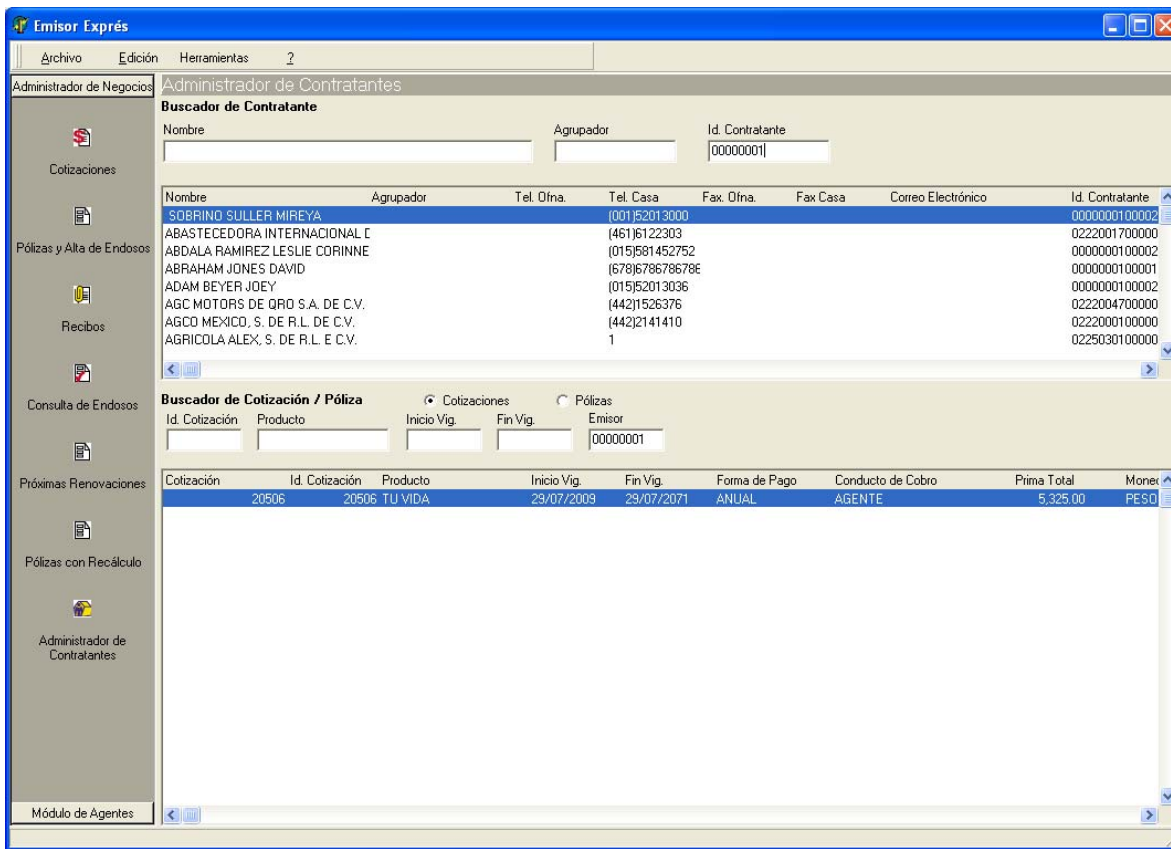


Figura 37. Pantalla principal del sistema

En las siguientes descripciones se mostrarán las demás interfaces involucradas con la pantalla principal.

Diseño de navegación de información

En caso de que los datos de usuario y password sean correctos el sistema mostrará la primera interfaz de navegación de tipo corporativo, donde la información de personas es registrada, esta interfaz es el *Buscador de Contratante*, ver figura 38.

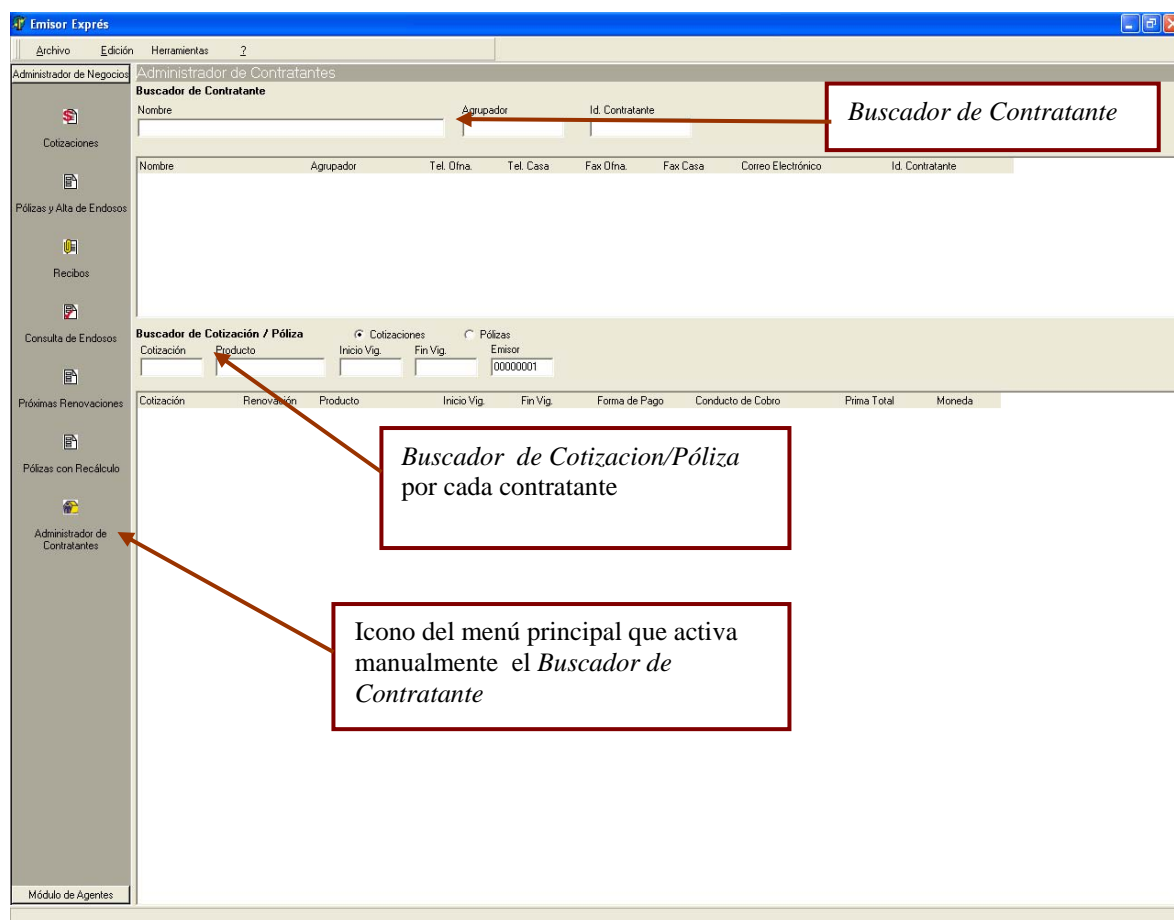


Figura 38. Administrador de Contratantes

Como se puede observar, este diseño del *Buscador de Contratante* permite hacer una búsqueda rápida a través de filtros.

Adicionalmente, contiene un subnavegador de cotizaciones y pólizas. También incluye sus propios filtros para realizar búsquedas específicas.

Dentro del diseño del menú principal se mostrarán los iconos principales de navegación y de operación para el usuario.

El icono del menú izquierdo, como el nombre lo indica, *Cotizaciones*, cada que sea activado con un clic derecho mostrará un *Buscador de Cotización*, ver figura 39.

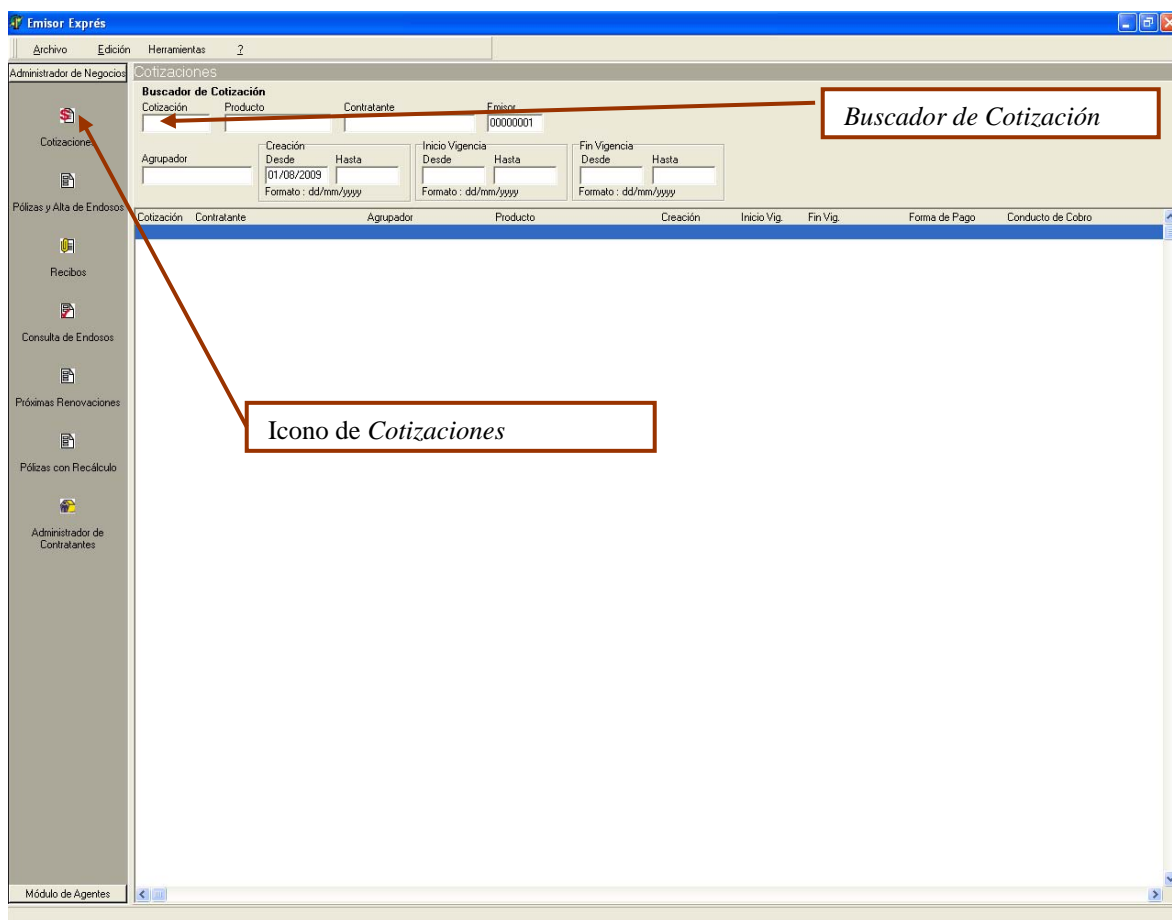


Figura 39. Buscador de Cotización

Sobre el *Buscador de Cotización* se mostrarán todas las propuestas que el agente realizó hacia sus clientes potenciales.

El segundo icono del menú, como el nombre lo indica, *Pólizas y Alta de Endosos*, cada que sea activado con un clic derecho mostrará un *Buscador de Póliza*, ver figura 40.

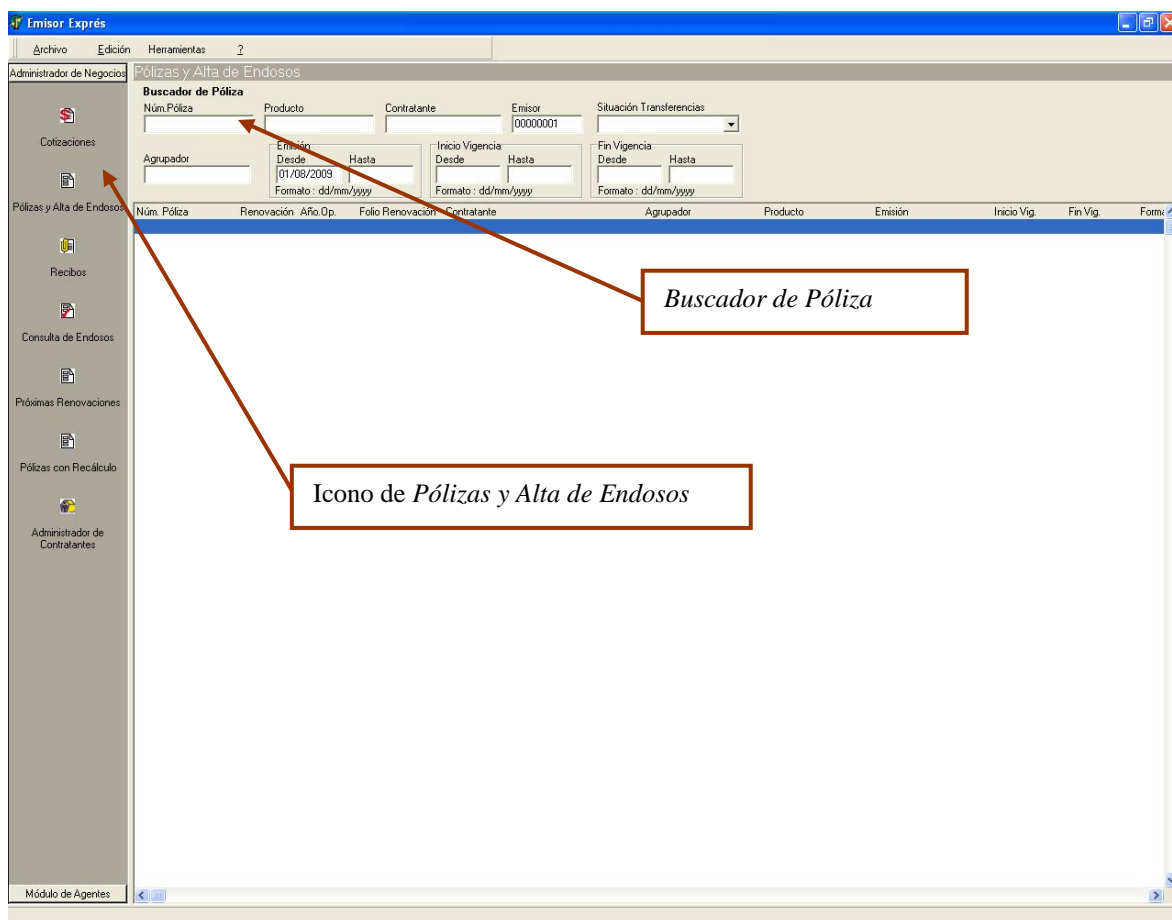


Figura 40. Buscador de Póliza

Una vez que una cotización es aceptada y emitida, se crea una póliza la cual se verá reflejada en la interfaz de pólizas.

El tercer icono del menú izquierdo, como el nombre lo indica, *Recibos*, activará el *Buscador de Recibo* por póliza filtrada, ver figura 41.

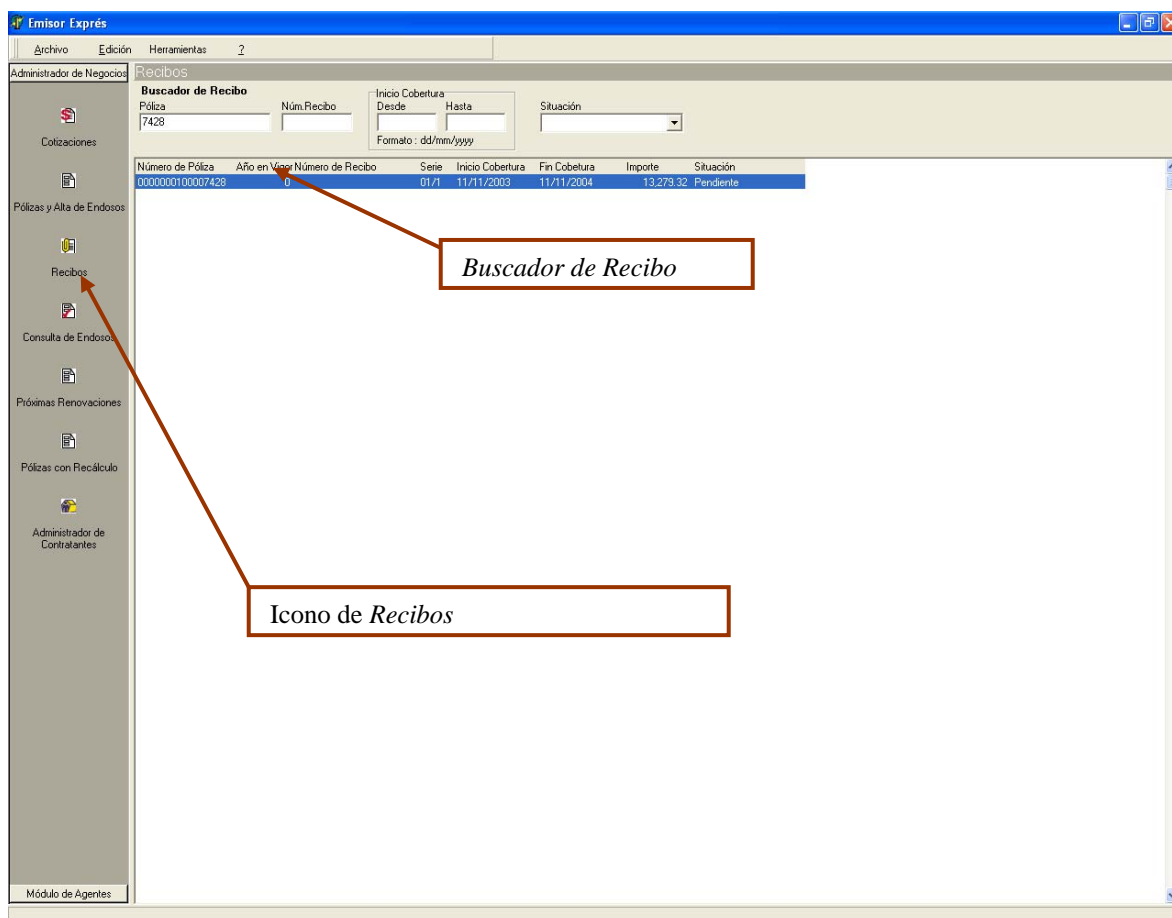


Figura 41. Buscador de Recibo

Toda póliza emitida genera una tira de recibos de acuerdo a su forma de pago:

- Mensual – 12 recibos
- Trimestral – 4 recibos
- Semestral – 2 recibos
- Anual – 1 recibo

La información de cada recibo por póliza filtrada se reflejará en la interfaz de recibos.

El cuarto icono del menú izquierdo, como el nombre lo indica, *Consulta de Endosos*, activará el *Buscador de Endoso* por póliza filtrada, ver figura 42.

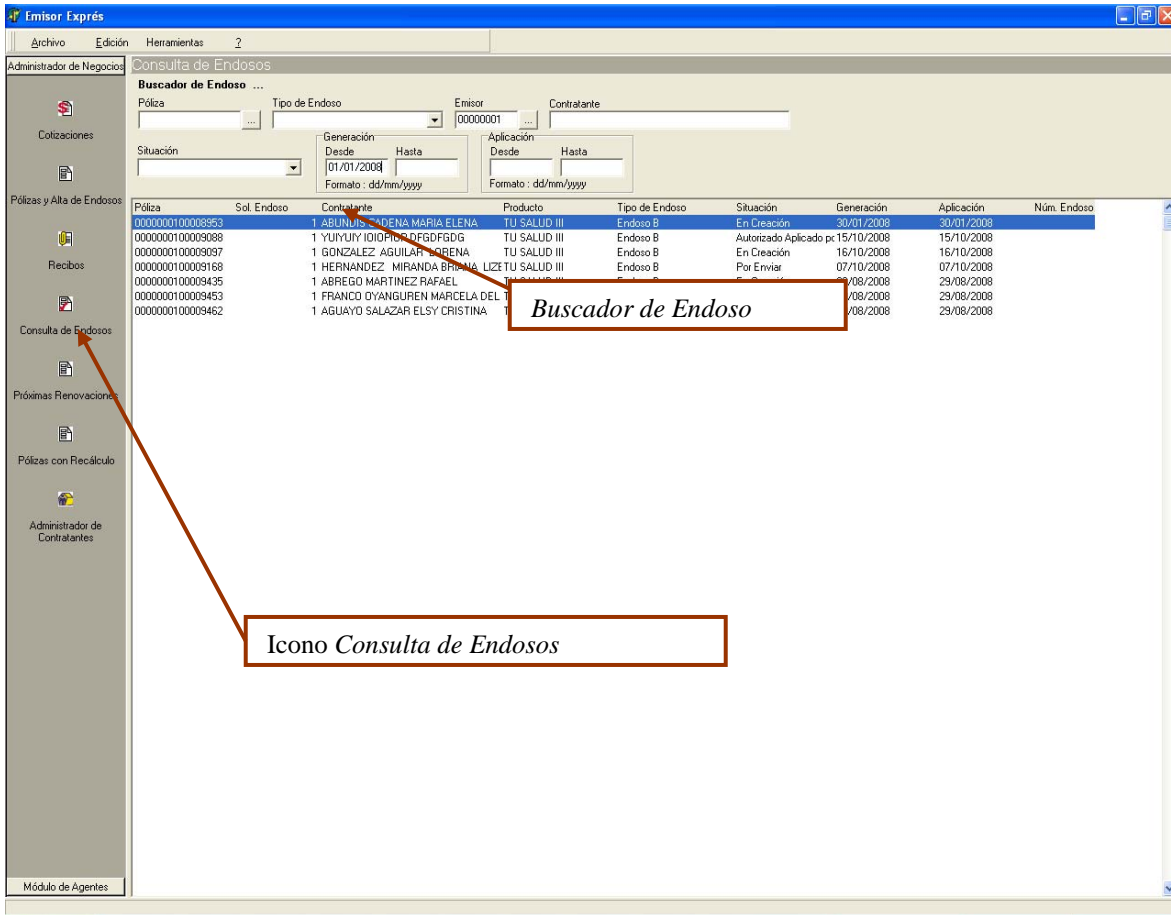


Figura 42. Buscador de Endoso

Una vez que inicia la vigencia de cualquier póliza, se puede aplicar endosos que modifiquen información de la misma. Estos registros de endosos se verán reflejados en la interfaz *Buscador de Endoso*.

Sobre el quinto icono del menú izquierdo, como el nombre lo indica, *Próximas Renovaciones*, activa el *Buscador de Renovaciones*. Cuando una póliza esté por terminar su vigencia puede ser renovada con el mismo número de póliza y con la vigencia correspondiente al año de operación, ver figura 43.

Esta herramienta le ayudará al agente para notificarle qué pólizas puede renovar.

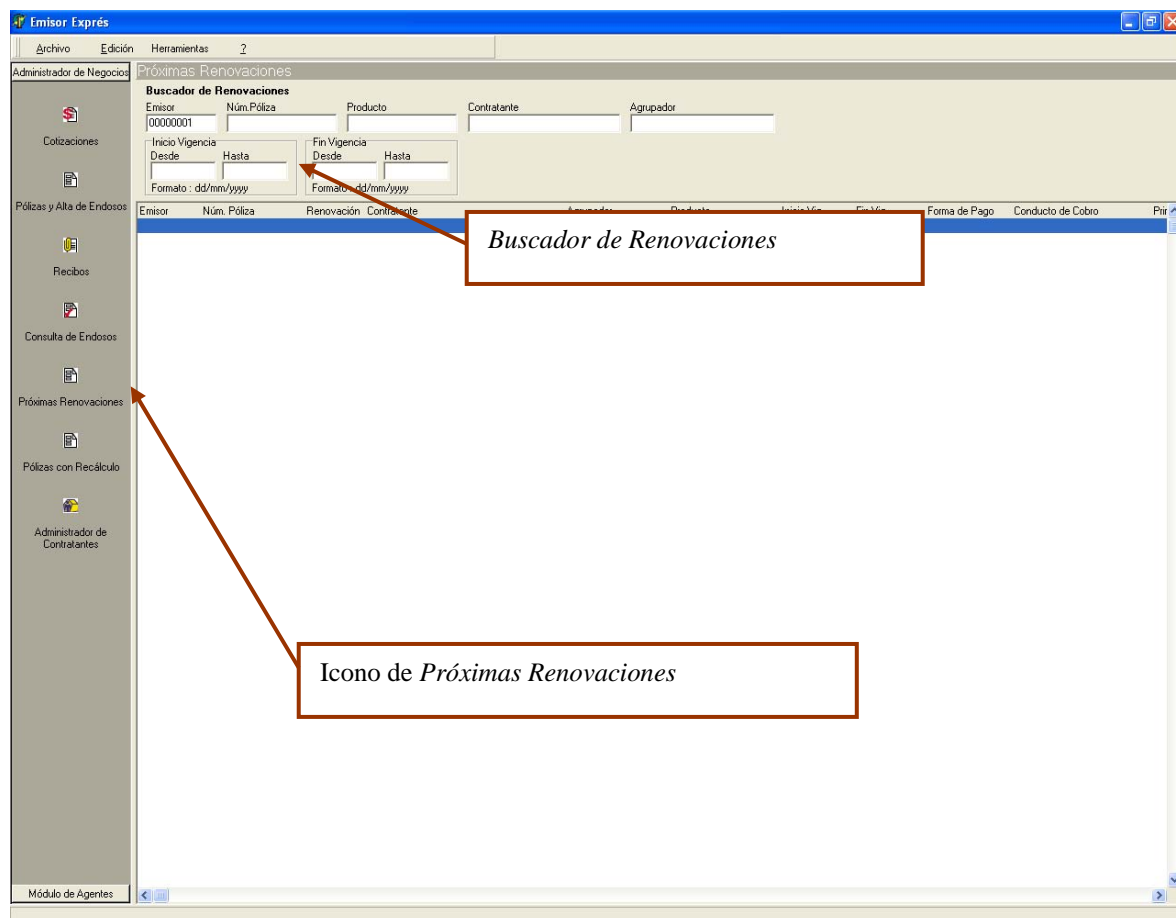


Figura 43. Buscador de Renovaciones

Cuando una póliza termina su vigencia el sistema permitirá renovarla; sobre la interfaz mostrará aquellas polizas que estén por terminar su vigencia y permitirá realizar nuevamente la emisión conservando el mismo número de póliza.

El sexto icono del menú izquierdo, como el nombre lo indica, *Pólizas con Recálculo*, activa el *Buscador de Pólizas Recalculadas*. Una póliza cuando esté por terminar su año de operación, puede ser recalculada para continuar con su cobranza, ver figura 44.

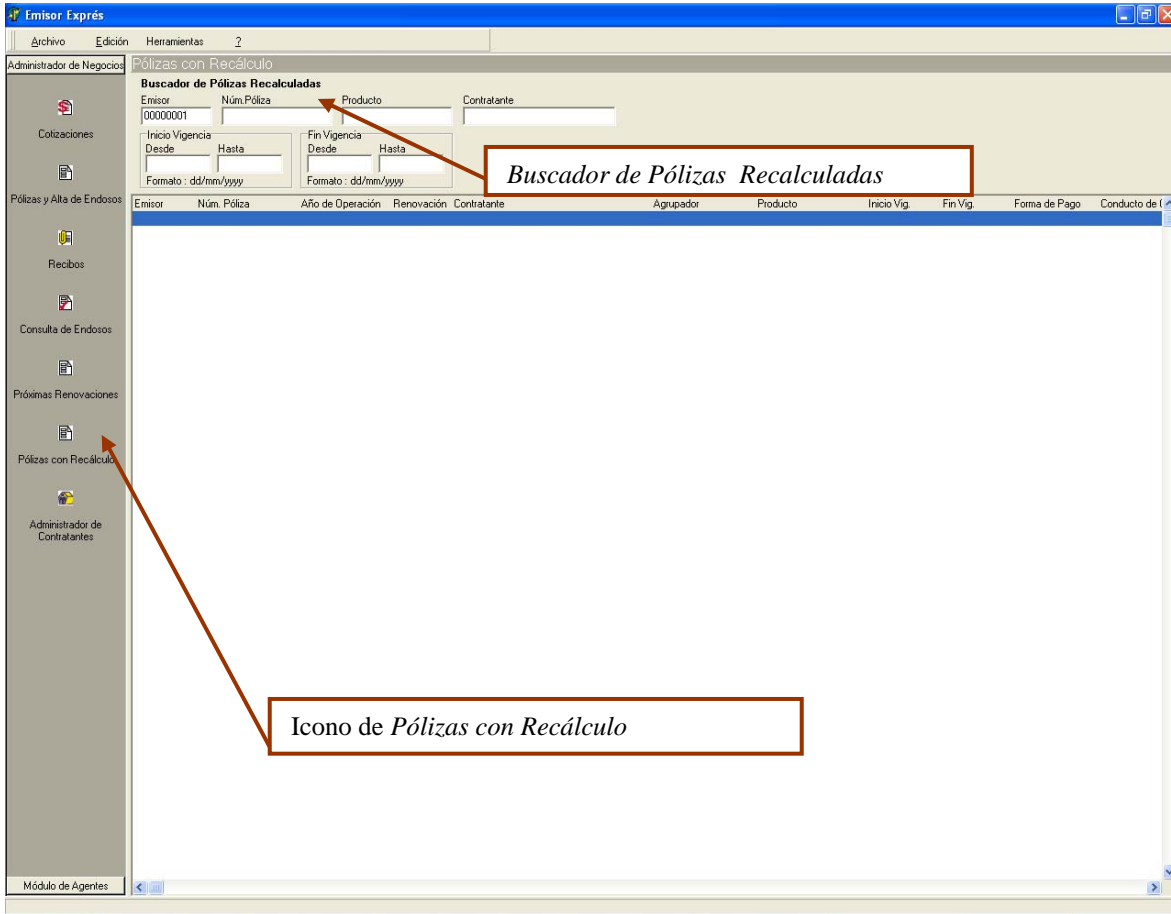


Figura 44. Buscador de Pólizas Recalculadas

En la primera parte se mostró el diseño de navegación inicial a través del menú principal del sistema.

4.2.4 Diseño de operación

Sobre la interfaz que se llama *Buscador de Contratante*, el modo de operación es de la siguiente manera, ver figura 45.

Si se desea consultar información sobre algún contratante, sólo al dar un clic derecho sobre el registro y un submenú se mostrará que permitirá hacer alguna acción sobre un contratante.

Administrador de Contratantes

Buscador de Contratante

Nombre Agrupador Id. Contratante

Nombre	Agrupador	Tel. Ofna.	Tel. Casa	Fax Ofna.	Fax Casa	Correo Electrónico	Id. Contratante
SOBRINO SULLER MIREYA			(001)52013000				0000000100002862
ABASTECEDORA INTERNACIONAL C			(461)6122303				0222001700000001
ABDALA RAMIREZ LESLIE CORINNE			(015)581452752				0000000100002343
ABRAHAM JONES DAVID			(678)6786786786				0000000100001173
ADAM BEYER JOEY			(015)52013036				0000000100002621
AGC MOTORS DE QRO S.A. DE C.V.			(442)1526376				0222004700000001
AGCO MEXICO, S. DE R.L. DE C.V.			(442)2141410				0222000100000001
AGRICOLA ALEX, S. DE R.L. E C.V.			1				0225030100000001
AGUILAR CERVERA ALEJANDRO M			(019)999469083				0231008000000001

(1)

A través del filtro señalado se pueden buscar contratantes dados de alta, sólo escribiendo el nombre del contratante

Aquí se encuentra la información de contratantes dados de alta en el sistema: nombre, teléfono etc.

Figura 45. Buscador de Contratante

Las acciones de los submenús dependen de qué tipo de operación quiera hacer sobre algún contratante escogido, ver figura 46.

Una vez seleccionado el contratante:

- 1 Se puede elegir una opción para buscar una póliza o cotización.
- 2 El sistema arrojará la información de qué producto se tiene cotizado o emitido: las fechas de vigencia, la forma de pago, el conducto de cobro y la prima total.
- 3 Si se desea dar de alta una nueva póliza al dar un clic derecho se mostrará otro submenú asociado al buscador de pólizas y seleccionando.

The screenshot shows the 'Administrador de Contratantes' interface. At the top, there is a search bar for 'Buscador de Contratante' with fields for 'Nombre', 'Agrupador', and 'Id. Contratante' (value: 00000001). Below this is a table of search results with columns: Nombre, Agrupador, Tel. Ofna., Tel. Casa, Fax Ofna., Fax Casa, Correo Electrónico, and Id. Contratante. The first row is highlighted in blue and corresponds to 'SOBRINO SULLER MIREYA'. Below the table is another search section, 'Buscador de Cotización / Póliza', with radio buttons for 'Cotizaciones' (selected) and 'Pólizas', and fields for 'Id. Cotización', 'Producto', 'Inicio Vig.', 'Fin Vig.', and 'Emisor' (value: 00000001). Below this is a table of search results with columns: Cotización, Id. Cotización, Producto, Inicio Vig., Fin Vig., Forma de Pago, Conducto de Cobro, Prima Total, and Moneda. The first row is highlighted in blue and corresponds to '20506', '20506 TU VIDA', '29/07/2009', '29/07/2071', 'ANUAL', 'AGENTE', '5,225.00', and 'PESOS MEX'. A context menu is open over the first row of this table, showing options: 'Nueva Cotización/Póliza', 'Consultar Cotización', and 'Eliminar Cotización'. Three numbered boxes (1, 2, 3) are placed on the screen with arrows pointing to the search bar, the first row of the second table, and the context menu respectively.

Figura 46. Buscador de Cotización / Póliza

Para dar de alta un nuevo registro de un contratante:

Al dar clic derecho sobre el *Buscador de Contratante*, se activa el submenú con las opciones requeridas y se selecciona, *Nuevo Contratante*, ver figura 47.

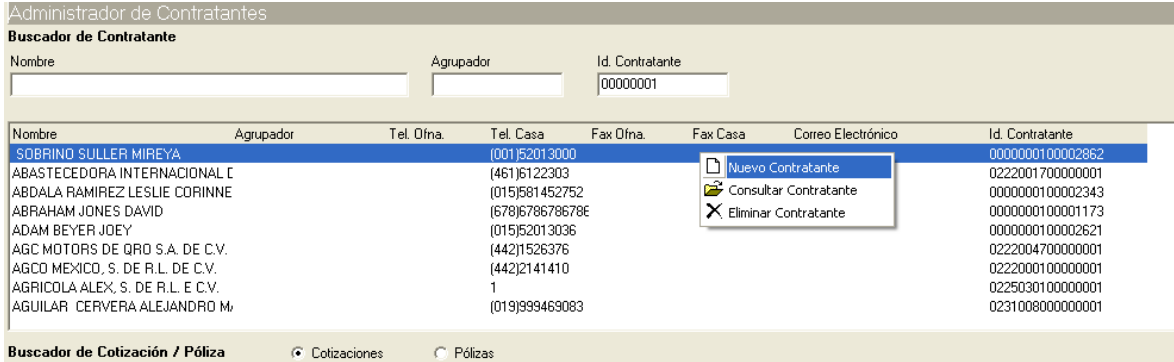


Figura 47. Nuevo Contratante

Una vez seleccionado aparece la siguiente interfaz para ingresar los datos del contratante, ver figura 48.

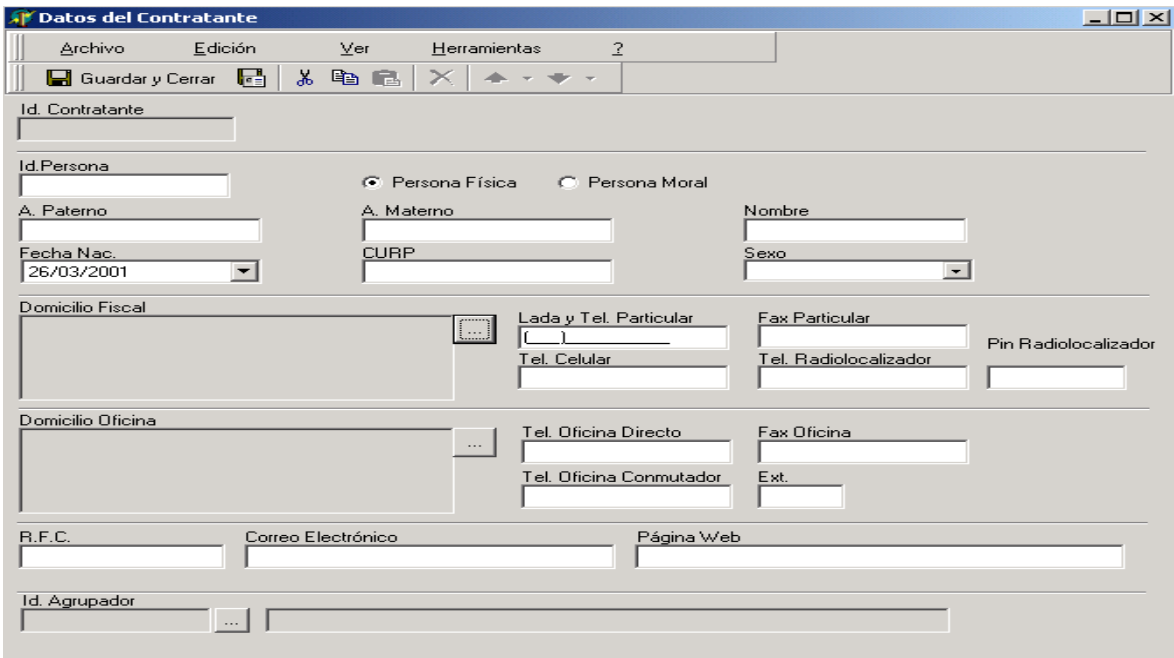


Figura 48. Datos del Contratante

Para ingresar el domicilio, se debe activar la interfaz; con sólo ingresar el código postal, el sistema automáticamente dará el estado, municipio y ubicación postal, se teclea la calle y el número, con ello se limita a que la información introducida es verídica, al terminar se da un clic en *Aceptar* y automáticamente la información queda capturada, ver figura 49.

Figura 49. Datos domiciliarios

Al ingresar el domicilio se regresa a la interfaz de la figura 48 y se procede a llenar los demás datos del contratante, una vez que la información sea la correcta, dar clic sobre el boton, *Guardar y Cerrar*, como se muestra en la figura 50. De lo contrario la información ingresada se perderá y no se guardará en la base de datos del sistema.

Figura 50. Guarda el registro del contratante

Una vez que el agente empiece a dar de alta un contratante, podrá realizar una cotización.

Para hacer la cotización existen dos caminos: A través del administrador de contratantes, como ya se describió anteriormente, o desde el menú principal, ver figura 51.

Sobre esta interfaz se tienen los siguientes elementos:

1. El buscador de cotizaciones se hace a través de filtros como: Número de cotización, producto o el nombre del contratante.
2. Al ingresar información sobre los filtros, el sistema busca información y en caso de encontrarla se reflejará sobre la interfaz.
3. Para consultar alguna cotización, sólo selecciona la cotización y al dar clic derecho mostrará un submenú y tendrá la opción, *Consultar Cotización*.
4. Si se desea eliminar la cotización, ésta se selecciona y al dar clic derecho en el submenú, seleccionar la opción, *Eliminar Cotización*.

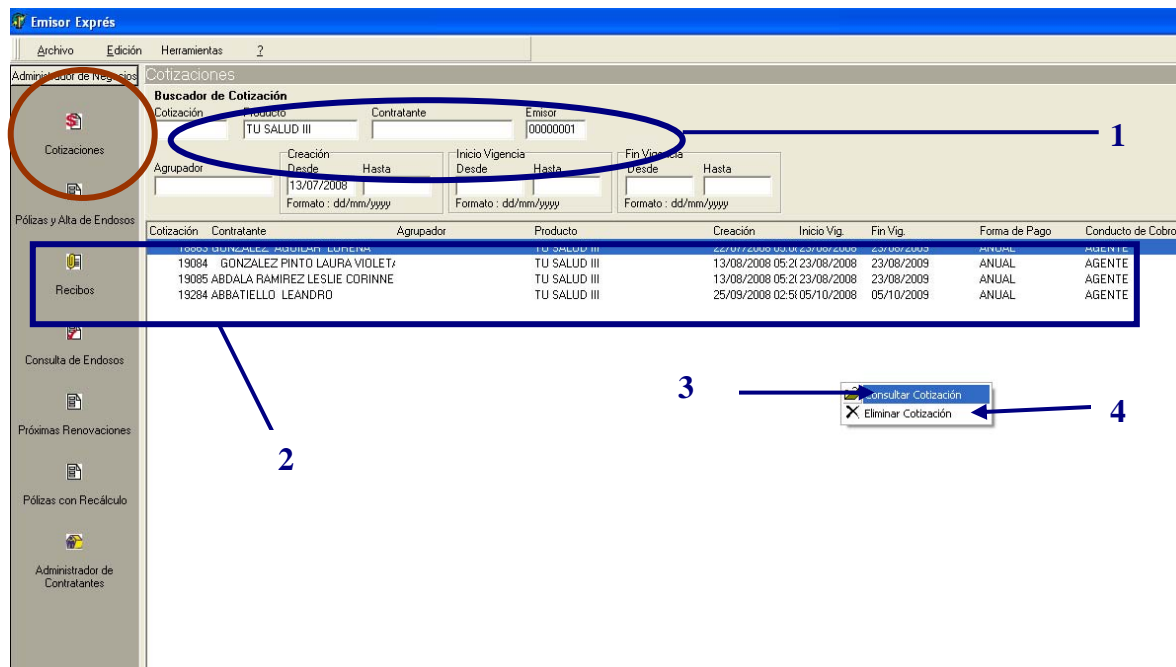


Figura 51. Cotización registrada

Sobre el menú principal al manejar la barra de herramientas superior, una vez que se esté posicionado sobre el buscador de cotizaciones se activa el botón, *Archivo*, y se desplegará un submenú con tres opciones, ver figura 52.

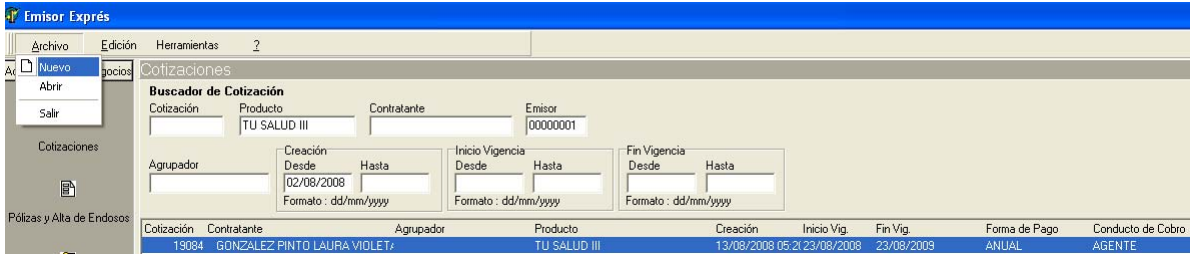


Figura 52. Creación de una cotización

Una vez que se active la opción, *Nuevo*, se desplegará la interfaz siguiente, donde se podrá elegir el producto que se desea vender o cotizar y que esté configurado en el sistema, ver figura 53.

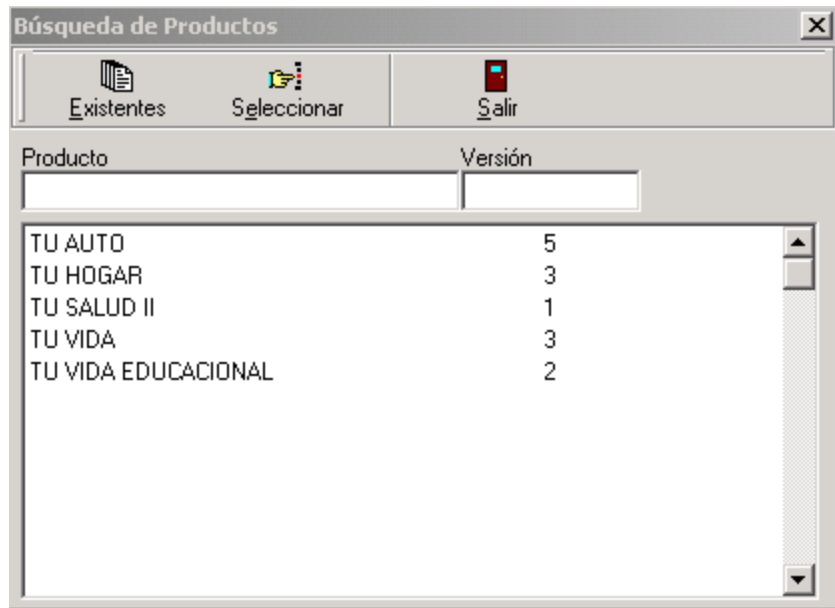


Figura 53. Productos configurados en el sistema

Una vez que se escoja el producto a cotizar, se muestra la siguiente interfaz, recargando la información dada de alta anteriormente como el nombre del contratante y la configuración del producto a cotizar, ver figura 54.

La interfaz muestra información como:

1. La fecha de inicio y fin de vigencia
2. Información del contratante
3. Información del producto
4. Resumen de costos y botones de aceptación
5. Datos de conducto de cobro, desglose de recibos y forma de pago

The screenshot shows the 'Cotización' application window. At the top, there is a menu bar with 'Archivo', 'Edición', 'Ver', 'Herramientas', and '?'. Below the menu is a toolbar with icons for 'Guardar y Cerrar', 'Cortar', 'Copiar', 'Pegar', 'Cancelar', 'Aceptar', 'Volver', and 'Avanzar'. The main data entry area includes fields for 'Cotiza' (150), 'Inicio de Vigencia' (22/03/2003), 'Fin de Vigencia' (22/03/2004), and 'Contratante' (AHUATZIN SANCHEZ FERNANDO). Below this are sections for 'Información del Producto' (TU SALUD II), 'Información del Agente' (SALDAÑA HERRERA JO), 'Resumen de Costos' (Prima Neta, Gastos de Emisión, Financiamiento, IVA, Prima Total), and 'Conducto de Cobro y Forma de Pago' (AGENTE, ANUAL). A table of 'Recibos' shows one entry with 'FOLIO' 1, 'Fecha' 22/03/2003, and 'Prima' 0.00. The bottom section contains 'Información del Conducto de Cobro' with fields for 'CUENTAHABIENTE', 'NÚMERO DE CUENTA', 'NÚMERO DE CHEQUE', and 'BANCO EMISOR'. Numbered callouts 1 through 5 point to the 'Inicio de Vigencia' field, the 'Contratante' field, the 'Información del Producto' section, the 'Resumen de Costos' table and buttons, and the 'Información del Agente' field respectively.

Prima Neta	\$0.00
Gastos de Emisión	\$0.00
Financiamiento	\$0.00
IVA	\$0.00
Prima Total	\$0.00

FOLIO	Fecha	Prima
1	22/03/2003	0.00

Figura 54. Interfaz para crear una cotización

La valoración del riesgo empieza cuando se introduce información en la siguiente interfaz, ver figura 55.

Esta interfaz tiene los siguientes elementos a valorar para el cálculo de la prima:

1. Información del contratante como titular
2. Datos de los dependientes económicos del titular
3. Selección de coberturas adicionales que forman parte del producto

SALUD

Archivo Edición Ver Herramientas ?

Guardar y Cerrar

Cotización/Póliza Núm. Inciso Situación Motivo de Situación Inicio Vig. Término Vig.
 0000000100020763 Cotización Cotización 11/10/2009 11/10/2010

FAMILIA:

LOS DATOS EN COLOR AZUL O NEGRITAS SON INDISPENSABLES PARA COTIZAR

DATOS DEL ASEGURADO

FECHA DE ALTA: 11/10/2009 EDAD: 0 PESO (KGS): 0 ESTATURA (MTS): 0
 EJEMPLO: 1.70

DOMICILIO: ESTADO CIVIL: SOLTERO PRIMA NETA DEL TITULAR: 10.941,05 OCUPACIÓN TITULAR:

ZONA CORRESPONDIENTE AL ASEGURADO: 1 (NO SE PUEDE ESCOGER UNA ZONA MAS BAJA A LA CORRESPONDIENTE)
 TIPO DE EMISIÓN: CASO ESPECIAL: PÓLIZA ANTERIOR: ZONA A ESCOGER PARA TARIFICAR: ZONA 1

CONTRASEÑA: Se deberá de ingresar los 16 dígitos de la póliza
 Eje. 0101010100001234 ES RENOVACIÓN.?: NO SI ES RENOVACIÓN: TECLEE EL ÚLTIMO NÚMERO DE PÓLIZA

DATOS DE LOS DEPENDIENTES ECONÓMICOS

DATOS GENERALES	EDAD	PARENTESCO	OCUPACIÓN	PESO (KGS)	ESTATURA (MTS)	PRIMA NETA POR DEPENDIENTE	FECHA DE ALTA
	0			1	1	0	11/10/2009
	0			1	1	0	11/10/2009
	0			1	1	0	11/10/2009
	0			1	1	0	11/10/2009
	0			1	1	0	11/10/2009
	0			1	1	0	11/10/2009
	0			1	1	0	11/10/2009
	0			1	1	0	11/10/2009

COBERTURAS ADICIONALES

CLÁUSULA DE PROTECCIÓN FAMILIAR: NO Allianz Plus únicamente se puede contratar cuando el plan sea Master o Amplio, diferente a zona 1.
 EI "ECI" se incluye sin costo en el plan Internacional ALLIANZ PLUS: NO

ENFERMEDADES CATASTRÓFICAS (CEC): NO AMPAR SE INCLUYE SIN COSTO EN PLAN INTERNACIONAL REDUCCIÓN DE COASEGURO: NO

MUERTE ACCIDENTAL EN VIAJES AÉREOS: NO CLIENTE WORLD CLASS: NO

GASTOS FUNERARIOS: NO ACTIVIDADES PELIGROSAS: NO

URGENCIAS EN EL EXTRANJERO (CUE): NO Para EDA, la S.A. máxima es de \$25,000 y será la misma que el deducible elegido en el plan contratado RESPALDO HOSPITALARIO: 0 Indemnización diaria

ELIMINACIÓN DE DEDUCIBLE POR ACCIDENTE (EDA): NO

Versión E (04/09/2006)

BENEFICIARIOS PARA MUERTE ACCIDENTAL EN VIAJES AÉREOS

En caso de que desee endoso de antigüedad de otras Compañías necesita enviar la siguiente documentación:
 Carátula de Póliza (última renovación)
 Copia de último recibo de pago (con sello de pagado o leyenda respectiva)

Figura 55. Selección de riesgos

Una vez que se termina de llenar la información, tanto en producto, conducto de cobro y forma de pago, se reflejará sobre la siguiente interfaz, la prima total a pagar, ver figura 56.

En caso de que se acepte el contrato se procede a emitir la póliza.

Resumen de Costo

Prima Neta	\$31,389.51
Gastos de Emisión	\$480.00
Financiamiento	\$0.00
IVA	\$4,780.42
Prima Total	\$36,649.93

Información del Conducto de Cobro

FOLIO	Fecha	Prima
1	25/11/2007	36,649.93

Figura 56. Monto total de la cotización realizada

Una vez que se realiza la emisión, el contratante acepta el costo y las condiciones, por lo que el sistema procederá a imprimir la información del contratante y asegurados en una carátula principal especificando todas las condiciones de acuerdo a coberturas contratadas.

4.2.5 Diseño de salida

A continuación se muestra el diseño de estas carátulas entregables al contratante de una póliza, ver figura 57.

PÓLIZA DE SEGURO							
NOMBRE Y DIRECCION DEL CONTRATANTE LAURA VIOLETA GONZALEZ PINTO PAX CHRISTI #20 COL. LOS OLVERA CORREGIDORA					G. M. M. LÍNEAS PERSONALES Plan Clásico		
					C.P. 76902 QUERETARO		
No. PÓLIZA	VIGENCIA		FECHA DE EMISIÓN		No. PÓLIZA ANT.		
00000001.00008944	Del: 25/11/2007	Al: 25/11/2008	15/11/2007				
ASEGURADOS		SEXO	FECHA ALTA	FECHA DE NACIMIENTO	MATERIIDAD	PRIMA ANUAL	
LAURA VIOLETA GONZÁLEZ PINTO		Fem.	25/11/2007	20/11/1956	EXCLUIDA	8,989.78	
DFDS SDF SDDSE		Fem.	25/11/2007	29/03/1960	EXCLUIDA	8,372.36	
DFE SDF SDFE		Fem.	25/11/2007	16/11/1982	AMPARADA	4,167.16	
DFGEF DFGR DFGRF		Fem.	25/11/2007	17/05/1985	AMPARADA	4,167.16	
SDFEF SDFEDF SEDF		Masc.	25/11/2007	19/06/1986	EXCLUIDA	3,231.32	
DSFEF XXDRC SDFESD		Fem.	25/11/2007	04/12/1988	AMPARADA	2,461.72	
						0.00	
						0.00	
						0.00	
PLANI	ZONA	GUA	SUMA ASEG. BÁSICA	DEDUCIBLE	COASEGURO	CONTRATO	
Clásico	III	100%	Sin Límite	4,000.00	5%	SPV-GMM022007	
COBERTURAS ADICIONALES							
Urgencias en el Extranjero	AMPARADO		Enf. Catastróficas Internacional		EXCLUIDO		
Protección Familiar	EXCLUIDO		Reducción de Coaseguro		EXCLUIDO		
Eliminación de Deducible por Accidente	EXCLUIDO		Allianz Plus		EXCLUIDO		
Actividades Peligrosas	EXCLUIDO		Respaldo Hospitalario		0.00		
World Class	EXCLUIDO						
Muerte en Accidentes Aéreos	Suma Asegurada		Gastos Funerarios		Suma Asegurada		
Asegurado	NO CUBIERTO		Asegurado		NO CUBIERTO		
Cónyuge	NO CUBIERTO		Cónyuge		NO CUBIERTO		
Dependiente > 12 años	NO CUBIERTO		Dependiente > 1 año		NO CUBIERTO		
Cláusulas que se Anexan a esta Póliza							
,GMM-022-0207 ,GMM-024-0207 ,GMM-016-0207 GMM-005-0206, GMM-008-0207, GMM-017-0207- Cláusula de Inicio de Vigencia de la Póliza							
No. AGENTE	10000001	NOMBRE AGENTE					
Prima Anual	36,649.93	Forma de Pago	ANUAL	Derecho de Póliza	480.00	I.V.A.	4,780.42
Primer Pago	36,649.93	Pagos Subsecuentes	0.00	Conducto de Cobro	AGENTE	Cuenta	0

Figura 57. Impresión de póliza emitida

4.3 Fase de codificación

Como primera instancia se empezará por la implementación del manejador de base de datos DBMS. Oracle es un sistema de gestión de base de datos relacional o RDBMS por el acrónimo en inglés de Relational Data Base Management System, fabricado por Oracle Corporation, el cual se considera como uno de los sistemas de bases de datos más completos, destacado por su:

- Soporte de transacciones
- Estabilidad
- Escalabilidad
- Multiplataforma

Una vez que la base de datos sea instalada de acuerdo al Anexo 1, se procede a la generación de entidades y características del modelo entidad-relación, como se muestra en el Anexo 2.

Se utilizará un navegador *PL/SQL Developer* de base de datos externo, para tener la facilidad de generar todas las entidades necesarias que se ocuparán en la codificación, ver figura 58.

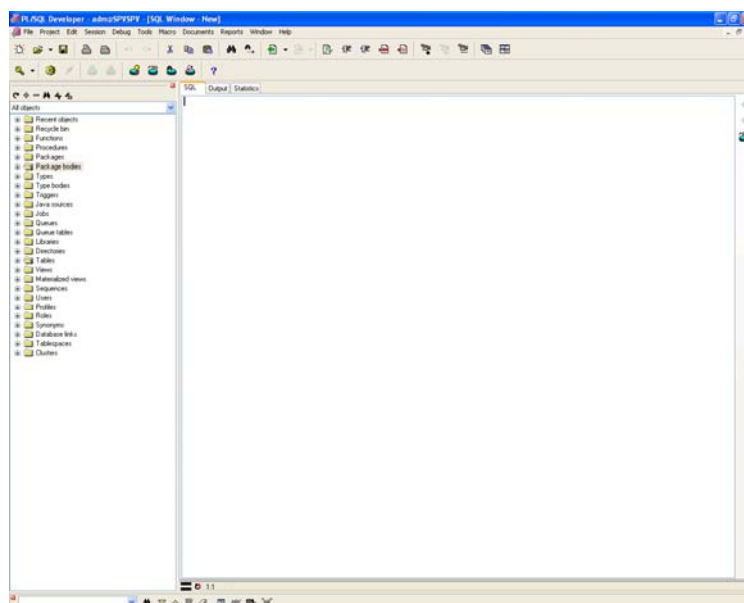


Figura 58. Pantalla principal del navegador PL/SQL Developer

Con ayuda de esta herramienta se ejecutan los scripts que se describen en el Anexo 2, de las tablas diseñadas para el sistema, al final de la ejecución se tendrá el modelo entidad-relación de la base de datos.

4.3.1 Preparación del entorno de desarrollo

Se elige Delphi, ya que proporciona las siguientes ventajas:

- Genera ficheros ejecutables compilados.
- Puede utilizar bibliotecas de enlace dinámico, fichero tipo DLL.
- Puede usar controles OCX de Visual Basic y además permite crear sus propios controles llamados VCL.
- Tiene todos los controles disponibles.
- Si se necesita controles especiales, es posible obtenerlo de Internet, hay cientos de ellos y muchos son freeware.
- Se pueden enviar mensajes a controles (*botones, etiquetas*), para reaccionar a ciertas situaciones.

Una de las características principales que hace a Delphi diferente de otros lenguajes de programación es que permite programar en un entorno totalmente visual. Lo hace utilizando como base el lenguaje de programación Object Pascal (Pascal orientado a objetos).

Una de las ventajas más destacables de Delphi respecto a lenguajes en entornos no visuales como pueden ser C, C++, etc; son las facilidades que ofrece para crear aplicaciones con interfaces gráficas y bajo entorno Windows.

Las especificaciones técnicas para instalar Delphi están en el Anexo 3.

4.3.2 Estandarización de codificación

1. Nomenclatura de codificación

2. Declaración de nombre de proyecto

Ejemplo: prNombreProyecto
Se anteponen el prefijo (pr)

3. Declaración de unidades

- Unit caseNombreUnidadGeneral
- Unit unNombreUnidadSimple

4. Declaración de objetos

- TcasePoliza = class
- TwcasePoliza = class(TForm)
- Es una clase no visual
- Es una clase visual, cuya herencia proviene de un objeto TForm

5. Instanciación de objetos

- Objeto: TcasePoliza

Se declara el nombre del objeto con la primera letra mayúscula.
Los objetos se declaran debajo de la directiva (Var).

6. Declaración de variables

- iNombrevariable: Integer
- sNombreVariable : String

Se antepone la primera letra minúscula de acuerdo al tipo de variable.

7. Identación

Ejemplo:

```
While sCluster <> '' Do Begin
  sDato := Copy(sCluster,1,Pos('||',sCluster)-2);
  sCluster := Copy(sCluster,Pos('||',sCluster)+2, 10000);
  sNombreDato := Copy(sDato,1,Pos('<',sDato)-1);
  If sNombreDato = sSalud Then Begin
    While sDato <> '' Do Begin
      sPregCuest := Copy(sDato,Pos('<',sDato)+1,10000);
      sDato := Copy(sDato,Pos('|',sDato)+1,10000);
      sPregRenov := Copy(sPregCuest,1,Pos('=' ,sPregCuest)-1);
      If sPregRenov = 'PREG19' Then Begin
        sRespRenov := Copy(sPregCuest,Pos('=' ,sDato),2);
        Self.ActualizaRenov(qyQuery.FieldName('id_unidad_negocio').AsInteger,
                           qyQuery.FieldName('id_emisor').AsString,
                           qyQuery.FieldName('id_cotizacion').AsInteger,
                           qyQuery.FieldName('id_renovacion').AsInteger,
                           sRespRenov);
      End;
    End;
  End;
End;
```

La indentación debe ser aplicada de manera visible y linealmente entre cada sentencia. Esto permitirá la facilidad de lectura entre cada bloque programado.

8. Cuerpo sintáctico de las unidades

```
unit Unit1;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs;

type
  TForm1 = class(TForm)
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;

Implementation
```

Unit: Especificación de la unidad asignando un nombre de acuerdo a la nomenclatura establecida.

Interfase: Directiva que especifica que la unidad es de tipo visual, es decir es una forma.

Uses: Especificación de librerías, objetos de tipo componentes y controles; disponibles para instanciar sus propiedades y utilizarlos.

Type: Directiva donde se declara el nombre de los objetos o clases que serán instanciados de un objeto de tipo TForm, clase de tipo visual.

Private: Directiva donde se declaran funciones, procedimientos y propiedades con carácter privado. No se puede tener acceso desde otras clases.

Public: Directiva donde se declaran funciones, procedimientos y propiedades con carácter público. Se puede tener acceso desde otras clases.

Var: Directiva donde se hará la instancia de los objetos a utilizar.

Implementation: Directiva donde se procede a la creación de algoritmos a través de funciones, procedimientos o eventos.

Con todos estos elementos descritos, se puede tener una estandarización de codificación estable, manejable y de fácil lectura para futuros mantenimientos del sistema.

4.3.3 Codificación de interfaces visuales

La codificación para el control de acceso hacia el sistema se validará mediante el siguiente código, una vez que el agente introduzca sus datos en la interfaz, ver figura 59.



Figura 59. Pantalla control de acceso

Código de verificación correspondiente a la figura 59:

```
Function TwExpEmisor.ValidaInfoNecesaria : Boolean;
Var
  sSQL : String;
  qyQuery : TQuery;
  dFechaUltCon : TDateTime;
  dFechaBD : TDateTime;
  sEmisor : String;
  iNumDiasCEmision : Integer;
  iNumDiasSEmision : Integer;
  iDiasDif : Integer;
  bActFecUltAcceso : Boolean;
Begin
  Result := True;
  sSQL := ' Select Count(*) CUANTOS'+
    ' From SEG_USUARIO'+
    ' WHERE ID_USUARIO <> 1';

  qyQuery := ObtenerQuery(Self.dbSPV, sSQL);
  If qyQuery <> nil Then Begin
    If qyQuery.FieldByName('CUANTOS').AsInteger = 0 Then Begin
      Application.MessageBox('No existen usuarios creados para operar el sistema,'+ chr(13)+
        'Debe realizar una conexión con su oficina asignada previo a iniciar operación','Iniciando Emisor Expres',0);
      Result := False;
      qyQuery.Free;
      Exit;
    End;
    qyQuery.Free;
  End;
  //Else Begin
  sSQL := 'Select Count(*) CUANTOS'+
    ' From EML_PARAM_OPERACION';

  qyQuery := ObtenerQuery(Self.dbSPV, sSQL);
  If qyQuery <> nil Then Begin
```

Sistema de Punto de Venta Portátil de Seguros

```
If qyQuery.FieldName('CUANTOS').AsInteger = 0 Then Begin
  Application.MessageBox('El sistema no cuenta con un identificador de emisor asignado,'+ chr(13)+
  'Debe realizar una conexión con su oficina asignada previo a iniciar operación','Iniciando Emisor Exprés',0);
  Result := False;
  qyQuery.Free;
  Exit;
End;
qyQuery.Free;
End;
sSQL := 'Select Count(*) CUANTOS'+
  ' From AE_UN_EMISOR_AGENTE';
qyQuery := ObtenerQuery(Self.dbSPV, sSQL);
If qyQuery <> nil Then Begin
  If qyQuery.FieldName('CUANTOS').AsInteger = 0 Then Begin
    Application.MessageBox('El sistema no cuenta con agentes asignados,'+ chr(13)+
    'Debe realizar una conexión con su oficina asignada previo a iniciar operación','Iniciando Emisor Exprés',0);
    Result := False;
    qyQuery.Free;
    Exit;
  End;
  qyQuery.Free;
End;

bActFecUltAcceso := False;
//Obtiene la fecha de la BD
qyQuery := ObtenerQuery(Self.dbSPV, 'Select sysdate from dual');
If qyQuery <> Nil Then Begin
  dFechaBD := qyQuery.FieldName('sysdate').AsDateTime;
  qyQuery.Free;
End;

// Valida fecha de última conexión
sSQL := ' SELECT * From EMI_PARAM_OPERACION';
qyQuery := ObtenerQuery(Self.dbSPV, sSQL);
If qyQuery <> Nil Then Begin
  iF (qyQuery.FieldName('B_BLOQUEADO').AsString = 'F') or (qyQuery.FieldName('B_BLOQUEADO').AsString = "") Then
  Begin
    //dFechaUltCon := qyQuery.FieldName('FH_ULT_ACCESO').AsDateTime;
    sEmisor := qyQuery.FieldName('ID_EMISOR').AsString;
    If qyQuery.FieldName('FH_ULT_ACCESO').AsDateTime <> 0 Then Begin
      If dFechaBD < qyQuery.FieldName('FH_ULT_ACCESO').AsDateTime Then Begin
        sSQL := ' UPDATE EMI_PARAM_OPERACION SET B_BLOQUEADO = "V" ';
        EjecutarQuery(Self.dbSPV, sSQL);
        Application.MessageBox('La fecha actual es menor a la de la última conexión,'+ chr(13)+
        'Debe realizar una conexión con su oficina asignada previo a iniciar operación','Iniciando Emisor Exprés',0);
        Result := False;
        qyQuery.Free;
        Exit;
      End;
    End;
  End;
  Else If qyQuery.FieldName('B_BLOQUEADO').AsString = 'V' Then Begin
    Application.MessageBox('La fecha actual es menor a la de la última conexión,'+ chr(13)+
    'Debe realizar una conexión con su oficina asignada previo a iniciar operación','Iniciando Emisor Exprés',0);
    Result := False;
    qyQuery.Free;
    Exit;
  End;
  qyQuery.Free;
  bActFecUltAcceso := True;
End;

// Valida fecha de última conexión con emisión y sin emisión
// DRP 01032004 - No hacia la validación contra la fecha de emi_bit_conexion
sSQL := 'select max(f_ini_conexion) AS FHULTCON from emi_bit_conexion ' +
  'where cve_sit_conexion="SC_EXI"';
qyQuery := ObtenerQuery(Self.dbSPV, sSQL);
If qyQuery <> Nil then begin
  dFechaUltCon := qyQuery.FieldName('FHULTCON').AsDateTime;
```

```

    qyQuery.Close;
    qyQuery.Free;
end
else begin
    dFechaUltCon := 0;
end;
// -----
If dFechaUltCon <> 0 Then Begin
    sSQL := ' SELECT * From AE_UN_EMITOR ' +
        ' WHERE ID_EMITOR = ' + sEmisor + '';
    qyQuery := ObtenerQuery(Self.dbSPV, sSQL);
    If qyQuery <> Nil Then Begin
        If qyQuery.FieldName('CVE_TIPO_CONEXION').AsString = 'TC_CRC' Then Begin
            iNumDiasCEmision := qyQuery.FieldName('NUM_DIAS_CONEXI_CEMISION').AsInteger;
            iNumDiasSEmision := qyQuery.FieldName('NUM_DIAS_CONEXI_SEMISION').AsInteger;
        End;
        qyQuery.Free;

        // Obtiene la diferencia entre hoy y la fecha de última conexión
        iDiasDif := Trunc(dFechaBD - dFechaUltCon);
        // Recupera pólizas desde la fecha de última conexión
        sSQL := ' SELECT count(*) Cuantos From EMI_POLIZA ' +
            ' WHERE NUM_POLIZA IS NOT NULL ' +
            ' AND FH_EMITION >= ' +
            ' TO_DATE('' + FormatDateTime('dd/mm/yyyy',dFechaUltCon) + '' ,''DD/MM/YYYY'' ) ' ;
        qyQuery := ObtenerQuery(Self.dbSPV, sSQL);
        If qyQuery <> Nil Then Begin
            If qyQuery.FieldName('Cuantos').AsInteger > 0 Then Begin
                If iNumDiasCEmision <> 0 Then Begin
                    //Valida contra NUM_DIAS_CONEXI_CEMISION
                    If iDiasDif >= iNumDiasCEmision Then Begin
                        Application.MessageBox('Se excedió el número de días permitidos desde la última emisión, '+ chr(13)+
                            'Debe realizar una conexión con su oficina asignada previo a iniciar operación', 'Iniciando Emisor Exprés',0);
                        Result := False;
                        qyQuery.Free;
                        Exit;
                    End;
                End;
            End;
        End;
    Else Begin
        If iNumDiasSEmision <> 0 Then Begin
            //Valida contra NUM_DIAS_CONEXI_SEMISION
            If iDiasDif >= iNumDiasSEmision Then Begin
                Application.MessageBox('Se excedió el número de días permitidos sin emisión, '+ chr(13)+
                    'Debe realizar una conexión con su oficina asignada previo a iniciar operación', 'Iniciando Emisor Exprés',0);
                Result := False;
                qyQuery.Free;
                Exit;
            End;
        End;
    End;
    qyQuery.Free;
End;
End;
End;

If bActFecUltAcceso = True Then Begin
    sSQL := ' UPDATE EMI_PARAM_OPERACION SET FH_ULT_ACCESO = ' +
        ' TO_DATE('' + FormatDateTime('dd/mm/yyyy HH:nn:SS',dFechaBD) +
        '' ,''DD/MM/YYYY HH24:MI:SS'' ) ' ;
    EjecutarQuery(Self.dbSPV, sSQL);
End;
End;

```

Dentro de esta codificación se ingresa a la base de datos, en el esquema de seguridad, validando:

- Que exista al menos un registro de usuario para operar el sistema.
- Si existe un usuario, verifica el identificador de operación (clave de emisor).
- Si existe un identificador de operación verifica el registro del agente asignado.
- Si existe un agente asignado verifica la fecha de última conexión realizada al sistema central.
- Verifica la fecha de emisión de la última póliza, en caso de sobrepasar el número de días sin emitir obliga el sistema a realizar una conexión al sistema central.

Una vez que, a través de la función, verifica dicha información de acceso, se procede abriendo el sistema mediante la siguiente sentencia:

```
Self.caseConexion := TcaseConnection.create(Self);
Self.caseConexion.BasePrincipal := Self.dbSPV;
Self.caseConexion.BaseSeguridades := Self.dbSPV;
Self.caseConexion.AsignarSistema('EMISOR EXPRES');
If Not Self.caseConexion.Open Then Begin
  Application.MessageBox('No se logró la conexión', 'Estableciendo Conexión',0);
  Application.Terminate;
End
```

En la siguiente codificación se valida el USUARIO sobre la base de datos:

```
procedure TwAbout.edUsuarioExit(Sender: TObject);
var
  qyNomUsuario : TQuery;
  sQuery       : String;
  sIdUnidadNegocio : String;
Begin
  If Self.edUsuario.Text = '' Then
    Exit;
  Self.LimpiarControles;
  sIdUnidadNegocio := Self.FUnidadNegocio.Strings [Self.cbxUnidadNegocio.ItemIndex];
  sQuery := 'SELECT SU.ID_USUARIO, SU.CONTRASENIA, P.NOMBRE, P.A_PATERN, P.A_MATERNO' +
    ' FROM SEG_SIST_UN_USUARIO SS, SEG_USUARIO SU, PERSONA P' +
    ' WHERE SS.ID_SISTEMA = ' + intToStr(Self.IdSistema) +
    ' AND SS.ID_UNIDAD_NEGOCIO = ' + sIdUnidadNegocio +
    ' AND SU.NOMBRE_USUARIO = ''' + Self.edUsuario.Text + ''' +
    ' AND SS.ID_USUARIO = SU.ID_USUARIO' +
    ' AND SU.ID_PERSONA = P.ID_PERSONA';
  qyNomUsuario := TQuery.Create(nil);
  try
    qyNomUsuario.DatabaseName := Self.DatabaseName;
    qyNomUsuario.SQL.Text := sQuery;
    qyNomUsuario.Active := True;
    if qyNomUsuario.Eof = True Then Begin
      Self.lbUsuario.Caption := 'La clave del usuario no se encontró.';
      //Application.MessageBox ('La clave del usuario no se encontró.','Validando usuario',0);
      //Self.edUsuario.SetFocus;
    End;
    while not qyNomUsuario.Eof do begin
      Self.IDUsuario := qyNomUsuario.FieldByName('ID_USUARIO').AsInteger;
      Self.Contrasenia := qyNomUsuario.FieldByName('CONTRASENIA').AsString;
      Self.lbUsuario.Caption := qyNomUsuario.FieldByName('NOMBRE').AsString +
        '' + qyNomUsuario.FieldByName('A_PATERN').AsString +
        '' + qyNomUsuario.FieldByName('A_MATERNO').AsString;
      qyNomUsuario.Next;
    end;
  end;
end;
```

```
    end;  
  finally  
    qyNomUsuario.Free;  
  end;  
end;
```

Después de ingresar un usuario válido, verifica la contraseña en la siguiente codificación:

```
procedure TwAbout.btnAceptarClick(Sender: TObject);  
begin  
  {Valida la contraseña}  
  if Self.edPwd.Text <> " Then Begin  
    if Self.edPwd.Text = Self.Contrasenia Then Begin  
      {indica que la contraseña es valida}  
      Self.Objeto.IdSistema := Self.IdSistema;  
      Self.Objeto.IdUnidadNegocio := strToInt(Self.FUnidadNegocio.Strings [Self.cbxUnidadNegocio.ItemIndex]);  
      Self.Objeto.IdUsuario := Self.IDUsuario;  
      Self.Objeto.NombreUsuario := Self.edUsuario.Text;  
      Self.Objeto.Contrasenia := Self.Contrasenia;  
      Self.Objeto.UsuarioValido := True;  
      Self.Close;  
    End  
  Else Begin  
    if Self.FIntentos = 2 Then Begin  
      Application.MessageBox('Se sobrepasaron los intentos permitidos','Validar contraseña',0);  
      Self.Close;  
    End  
  Else Begin  
    {Incrementa el contador de intentos}  
    Self.FIntentos := Self.FIntentos + 1;  
    Application.MessageBox('La contraseña no es correcta... ','Validando contraseña',0);  
    Self.edPwd.SetFocus;  
  End;  
End;  
End;  
end;
```

Codificación de menú principal y exploradores de información

Una vez validada la información de manera correcta, se carga la siguiente interfaz de operación, donde incluye el menú principal para navegar sobre distintos paneles.

Con este procedimiento se tendrá el control del menú principal al ejecutar el evento sobre cada botón.

Codificación de la siguiente interfaz mostrada en el sistema y llamada *Buscador de Contratante*, ver figura 60.

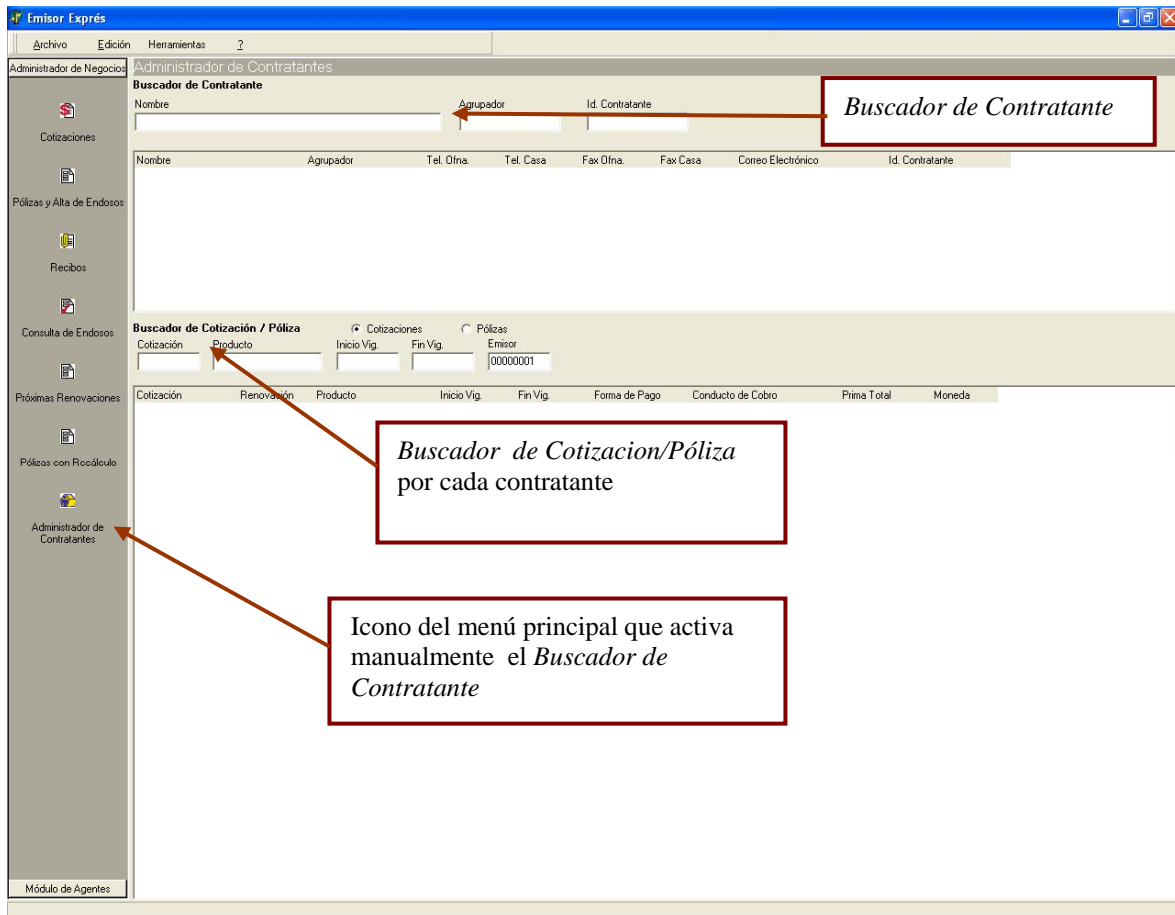


Figura 60. Buscador de Contratante

Cada buscador o navegador de información mencionado en las siguientes codificaciones están basados en una consulta principal, codificado en lenguaje SQL plus.

Estas consultas describen los campos necesarios para obtener la información solicitada por el usuario.

Procedimiento para extraer información de contratantes de la base de datos y mostrarla en los GRID's, se muestra en la siguiente codificación:

```
// OBTIENE LOS DATOS DE LOS CLIENTES
Procedure TwcaseClientes.RecuperaClientes;
Var
  sSQLBase : String;
  sNombre: String;
  sCond : String;
  sSQL1 : String;
  sSQL2 : String;
  sSQLUnion : String;
Begin
  {Consulta para obtener los campos de la tabla}
  sSQLBase := ' SELECT P.ID_PERSONA, P.RAZON_SOCIAL NOMBRE, EC.ID_AGRUPADOR, ' +
    ' AGRUPA.RAZON_SOCIAL AGRUPADOR, P.TEL_OFNA_DIRECTO, P.TEL_PARTICULAR, ' +
    ' P.FAX_OFNA, P.FAX_PARTICULAR, P.EMAIL, EC.ID_CONTRATANTE ' +
    ' FROM EMI_CONTRATANTE EC, PERSONA P, PERSONA AGRUPA ' +
    ' WHERE EC.ID_PERSONA = P.ID_PERSONA ' +
    ' AND EC.ID_AGRUPADOR = AGRUPA.ID_PERSONA (+) ';
  Self.qyClientes.DatabaseName := Self.Objeto.caseConexion.BasePrincipal.DatabaseName;

  // Validar Acceso Cartera *****
  If Self.Objeto.BAccCartR Then Begin
    sCond := "";
    sCond := ' AND EC.ID_USUARIO_CREACION ='+ IntToStr(Self.Objeto.caseConexion.IdUsuario);
    sSQLBase := sSQLBase + sCond;

  End;

  If Self.edIdCliente.Text <> " Then Begin
    sCond := ' AND UPPER(EC.ID_CONTRATANTE) LIKE "%'+ Trim(UpperCase(Self.edIdCliente.Text))+ '%";
    sSQLBase := sSQLBase + sCond;
  End;

  If Self.edIdAgrupador.Text <> " Then Begin
    sCond := ' AND UPPER(AGRUPA.RAZON_SOCIAL) LIKE "%'+ Trim(UpperCase(Self.edIdAgrupador.Text))+ '%";
    sSQLBase := sSQLBase + sCond;
  End;

  If Self.edNomCliente.Text <> " Then Begin
    sCond := "";
    sCond := ' AND UPPER(P.RAZON_SOCIAL) LIKE "%'+ Trim(UpperCase(Self.edNomCliente.Text))+ '%";
    sSQLBase := sSQLBase + sCond;
  End;

  // Se ordena por Nombre
  sCond := "";
  sCond := ' ORDER BY NOMBRE ';
  sSQLBase := sSQLBase + sCond;

  Self.qyClientes.SQL.Text := sSQLBase;
  Self.qyClientes.Close;
  Self.qyClientes.Open;
  If Self.qyClientes.Eof Then Begin
    Self.bQuery1Act := False;
    Self.qyCotizaPoliza.Close;
    Self.bQuery2Act := False;
  End
  Else
    Self.bQuery1Act := True;
End;
```

Por cada valor introducido en el área de filtros del buscador de contratantes se activará el procedimiento para extraer información, como se muestra en la siguiente codificación:

```
//RECUPERA LAS COTIZACIONES Y PÓLIZAS
Procedure TwcaseClientes.RecuperaCotizaPoliza(sContratante : String);
Var
  sQuery : String;
  sDerivacion: String;
  sSQLBase : String;
  sCond : String;
Begin
  If Self.rbCotizaciones.Checked Then Begin
    // Cotizaciones *****
    sSQLBase :=
      'SELECT DISTINCT EP.ID_COTIZACION,EP.ID_UNIDAD_NEGOCIO, EP.ID_EMISOR, ' +
      ' EP.ID_RENOVACION, P.PRODUCTO , EP.F_INICIO_VIG, EP.F_FIN_VIG, ' +
      ' SPOL.CLAVE, M.MONEDA, FP.FORMA_PAGO , ' +
      ' To_Char(NVL(DIMP.IMP_PT_SAI,0) + NVL(DIMP.IMP_PT_NAI,0), "999,999,990.00") PRIMATOT,' +
      ' PCCOB.CONDUCTO_COBRO ' +
      ' FROM EMI_POLIZA EP, PRO_PRODUCTO P,(SELECT GC.CVE_CLAVE, GC.CLAVE FROM INF_GRUPO G,' +
      ' INF_GRUPO_CLAVE GC WHERE G.GRUPO = "CVE_MTVO_SIT_POLIZA" AND G.ID_GRUPO = GC.ID_GRUPO)
SPOL,' +
      ' MONEDA M, PRO_FORMA_PAGO FP, EMI_DESGLOSEIMPORTE DIMP, PRO_CONDUCTO_COBRO PCCOB'+
      ' WHERE EP.ID_UNIDAD_NEGOCIO ='+ IntToStr(Self.Objeto.caseConexion.IdUnidadNegocio) +
      ' AND EP.ID_CONTRATANTE = '' + sContratante + '' +
      ' AND EP.ID_UNIDAD_NEGOCIO = P.ID_UNIDAD_NEGOCIO'+
      ' AND EP.ID_PRODUCTO = P.ID_PRODUCTO'+
      ' AND EP.CVE_SIT_POLIZA = "SP_COT"+
      ' AND EP.CVE_MTVO_SIT_POLIZA = SPOL.CVE_CLAVE'+
      ' AND EP.ID_MONEDA = M.ID_MONEDA'+
      ' AND EP.ID_FORMA_PAGO = FP.ID_FORMA_PAGO'+
      // Desglose Importe ***** Aop ***
      ' AND EP.ID_UNIDAD_NEGOCIO = DIMP.ID_UNIDAD_NEGOCIO (+)' +
      ' AND EP.ID_EMISOR = DIMP.ID_EMISOR (+)' +
      ' AND EP.ID_COTIZACION = DIMP.ID_COTIZACION (+)' +
      ' AND EP.ID_RENOVACION = DIMP.ID_RENOVACION (+)' +
      ' AND EP.ID_DESGLOSE_IMPORTE = DIMP.ID_DESGLOSE_IMPORTE (+)' +
      ' AND EP.ID_CONDUCTO_COBRO = PCCOB.ID_CONDUCTO_COBRO';

    If Self.Objeto.IdAgente <> 0 Then Begin
      sCond := ' AND EP.ID_AGENTE = '+ IntToStr(Self.Objeto.IdAgente);
      sSQLBase := sSQLBase + sCond;
    End;
    // .....
    // ***** Condiciones *****

    If Self.edEmisor.Text <> '' Then Begin
      sCond := ' AND UPPER(EP.ID_EMISOR) LIKE "%'+ Trim(UpperCase(Self.edEmisor.Text))+ '%";
      sSQLBase := sSQLBase + sCond;
    End;

    If Self.edIdCotizacion.Text <> '' Then Begin
      sCond := ' AND EP.ID_COTIZACION = '+ Trim(Self.edIdCotizacion.Text);
      sSQLBase := sSQLBase + sCond;
    End;
  End;

  // Pólizas *****
  If Self.rbPolizas.Checked Then Begin
    sSQLBase :=
      'SELECT DISTINCT EP.NUM_POLIZA, EP.ID_UNIDAD_NEGOCIO, EP.ID_EMISOR, ' +
      ' EP.ID_COTIZACION, EP.ID_RENOVACION, P.PRODUCTO , EP.F_INICIO_VIG, EP.F_FIN_VIG,' +
      ' SPOL.CLAVE, M.MONEDA, FP.FORMA_PAGO , ' +
      ' To_Char(NVL(DIMP.IMP_PT_SAI,0) + NVL(DIMP.IMP_PT_NAI,0), "999,999,990.00") PRIMATOT,' +
      ' PCCOB.CONDUCTO_COBRO'+
      ' FROM EMI_POLIZA EP, PRO_PRODUCTO P,(SELECT GC.CVE_CLAVE, GC.CLAVE FROM INF_GRUPO G,' +
      ' INF_GRUPO_CLAVE GC WHERE G.GRUPO = "CVE_MTVO_SIT_POLIZA" AND G.ID_GRUPO = GC.ID_GRUPO)
SPOL,' +
```



```

' MONEDA M, PRO_FORMA_PAGO FP, EMI_DESGLOSEIMPORTE DIMP, PRO_CONDUCTO_COBRO PCCOB'+
' WHERE EP.ID_UNIDAD_NEGOCIO = ' + IntToStr(Self.Objeto.caseConexion.IdUnidadNegocio) +
' AND EP.ID_CONTRATANTE = '' + sContratante + '' +
' AND EP.ID_PRODUCTO = P.ID_PRODUCTO'+
' AND EP.CVE_SIT_POLIZA <> "SP_COT"+
' AND EP.CVE_MTVO_SIT_POLIZA = SPOL.CVE_CLAVE'+
' AND EP.ID_MONEDA = M.ID_MONEDA'+
' AND EP.ID_FORMA_PAGO = FP.ID_FORMA_PAGO'+
// Desglose Importe ***** Aop ***
' AND EP.ID_UNIDAD_NEGOCIO = DIMP.ID_UNIDAD_NEGOCIO '+
' AND EP.ID_EMITOR = DIMP.ID_EMITOR'+
' AND EP.ID_COTIZACION = DIMP.ID_COTIZACION'+
' AND EP.ID_RENOVACION = DIMP.ID_RENOVACION'+
' AND EP.ID_DESGLOSE_IMPORTE = DIMP.ID_DESGLOSE_IMPORTE'+
' AND EP.ID_CONDUCTO_COBRO = PCCOB.ID_CONDUCTO_COBRO';
// ***** Condiciones *****

If Self.edEmisor.Text <> " Then Begin
    sCond := ' AND UPPER(EP.ID_EMITOR) LIKE "%'+ Trim(UpperCase(Self.edEmisor.Text))+ "%";
    sSQLBase := sSQLBase + sCond;
End;

If Self.edNumPoliza.Text <> " Then Begin
    sCond := ' AND UPPER(EP.NUM_POLIZA) LIKE "%'+ Trim(UpperCase(Self.edNumPoliza.Text))+ "%";
    sSQLBase := sSQLBase + sCond;
End;
End;

// Condiciones en común para Cotizaciones/Pólizas *****
If Self.edProducto.Text <> " Then Begin
    sCond := ' AND UPPER(P.PRODUCTO) LIKE "%'+ Trim(UpperCase(Self.edProducto.Text))+ "%";
    sSQLBase := sSQLBase + sCond;
End;

// ***** CODIGO OK ***** SFF 08082001
If Self.FfecIniVig <> 0 Then Begin
    sCond := ' AND EP.F_INICIO_VIG = (TO_DATE(''+ FormatDateTime('DD/MM/YYYY',Self.FfecIniVig) + '' ,''dd/mm/yyyy''));
    sSQLBase := sSQLBase + sCond;
End;

If Self.FfecFinVig <> 0 Then Begin
    sCond := ' AND EP.F_FIN_VIG = (TO_DATE(''+ FormatDateTime('DD/MM/YYYY',Self.FfecFinVig) + '' ,''dd/mm/yyyy''));
    sSQLBase := sSQLBase + sCond;
End;

If (Self.cbxSitPoliza.ItemIndex <> -1) AND (Self.cbxSitPoliza.Text <> ") Then Begin
    sCond := ' AND EP.CVE_MTVO_SIT_POLIZA ='''+
    Self.FListSitPol.Strings[Self.cbxSitPoliza.ItemIndex] + ''";
    sSQLBase := sSQLBase + sCond;
End;

// Datos para ejecutar query
Self.qyCotizaPoliza.DatabaseName := Self.Objeto.caseConexion.BasePrincipal.DatabaseName;
Self.qyCotizaPoliza.SQL.Text := sSQLBase;
Self.qyCotizaPoliza.Close;
Self.qyCotizaPoliza.Open;

If Self.qyCotizaPoliza.Eof Then
    Self.bQuery2Act := False
Else
    Self.bQuery2Act := True;

If (Self.FfecIniVig = 0) And (Self.edFecIniVig.Text <> ") Then
    Self.edFecIniVig.Text := "";
If (Self.FfecFinVig = 0) And (Self.edFecFinVig.Text <> ") Then
    Self.edFecFinVig.Text := "";
End;

```

En la siguiente interfaz, como se muestra en la figura 61, se puede realizar una cotización, ingresando cada valor sobre cada campo asociado a ésta.

La interfaz está ligada a varias clases de negocio, cuya función es importante para realizar cálculos y arrojar la prima total que un cliente debe pagar por su póliza.

The screenshot shows a software window titled 'Cotización' with a menu bar (Archivo, Edición, Ver, Herramientas, ?) and a toolbar. The main interface is divided into several sections:

- Header Fields:** Cotización (15000215), Inicio de Vigencia (22/03/2003), Fin de Vigencia (004), and Contratante (AHUATZIN SANCHEZ FERNAN). There are also buttons for 'Contratantes Adicionales' and 'Domicilio de Correspondencia'.
- Información del Producto:** Product (TU SALUD II), Version (1), and Currency (PESOS MEX). A button 'Para ver el producto' is present.
- Información del Agente:** Cve. Agente (00000911) and Agente (SALDAÑA HERRERA JOSE LUIS).
- Resumen de Costos:** A table showing:

Prima Neta	\$0.00
Gastos de Emisión	\$0.00
Financiamiento	\$0.00
IVA	\$0.00
Prima Total	\$0.00
- Conducto de Cobro y Forma de Pago:** Tipo de Conducto de Cobro (AGENTE), Conducto de Cobro (AGENTE), Forma de Pago (ANUAL), and % de Recargo (0.00).
- Recibos:** A table with columns FOLIO, Fecha, and Prima. It shows one entry: FOLIO 1, Fecha 22/03/2003, Prima 0.00.
- Información del Conducto de Cobro:** Fields for CUENTAHABIENTE, NÚMERO DE CUENTA, NÚMERO DE CHEQUE, and BANCO EMISOR.

Buttons at the bottom include 'Emitir Póliza' and 'Imprimir Cotización'.

Figura 61. Pantalla para realizar cotización con el producto salud

A continuación se describe el código de esas funciones siguiendo la numeración de acuerdo a la interfaz de la figura 61:

1. La información de este campo es automática y fija, es generada y formateada por una sentencia sobre la base de datos:

```
Procedure TwcasePoliza.GeneraIdCotizacion;
Var
  qyQuery : TQuery;
  sQuery : String;

Begin
  Self.Objeto.DomicilioCorr.Nuevo;
  // Genera el identificador de la cotización
  If Self.Objeto.SitFrame = sitCatNuevo Then Begin
    sQuery := 'SELECT SQ01_EMI_POLIZA.NEXTVAL as ID_COTIZACION From sys.dual';
    qyQuery := ObtenerQuery(Self.Objeto.Conexion.BasePrincipal,sQuery);
    if qyQuery <> Nil Then Begin
      //Aplica mascara a cotización
      Self.AplicaMascaraACotizacion(qyQuery.fieldbyname('ID_COTIZACION').AsString);

      Self.Objeto.Campos.FieldByName('ID_COTIZACION').AsInteger := qyQuery.fieldbyname('ID_COTIZACION').AsInteger;
      //Recupera el emisor
      Self.edIdEmisor.Text := Self.Objeto.RecuperaEmisor;
      Self.edIdRenovacion.Text := '0';
      qyQuery.Free;
    End;
  End;

Procedure TwcasePoliza.AplicaMascaraACotizacion(sIdCotiza :String);
var
  sCotiza : String;
  SMascara: String;
Begin
  sMascara:= '00000000';
  sCotiza:= sIdCotiza;
  sCotiza:= copy (sMascara, 1, length (sMascara)- length (trim (sCotiza))) + sCotiza;
  Self.edIdCotizacion.Text:= sCotiza;
End;
```

La sentencia principal `SQ01_EMI_POLIZA.NEXTVAL as ID_COTIZACION From sys.dual`, es un objeto creado en la base de datos y el cual, al ejecutarlo, siempre nos dará un número consecutivo válido para la tabla `emi_poliza`, ya que el valor es una llave primaria.

2. La información de inicio y fin de vigencia también será generada automáticamente a través del siguiente procedimiento:

```
// ACTUALIZA FECHA INICIO Y FIN DE VIGENCIA
Procedure TwcasePoliza.ActualizarFecIniFinVidaT1 (qyQuery: TQuery);
var.
  iNumDias: Integer;
  iNumDiasRet: Integer;
  iIncremento: Integer;
  dFechaMin: TDate;
  dFechaMax: TDate;
Begin
  // Fecha de inicio
  iNumDias:= qyQuery.fieldByName ('NUM_DIAS_VENTA_RET').asInteger;
  If qyQuery.fieldByName('CVE_TIPO_RET_DIF').asString = 'TRD_RO' Then Begin
    // TRD_RO = RETROACTIVO OBLIGATORIO hoy -numdias
    Self.dtpIniVig.Date:= Self.Objeto.Conexion.Fecha - iNumDias;
    Self.dtpIniVig.Enabled := False;
```

```

End;
If qqQuery.fieldByName('CVE_TIPO_RET_DIF').asString = 'TRD_DO' Then Begin
  // TRD_DO = DIFERIDO OBLIGATORIO hoy + numdias
  iNumDias:= qqQuery.fieldByName ('NUM_DIAS_VENTA_DIF').asInteger;
  iNumDiasRet:= qqQuery.fieldByName ('NUM_DIAS_VENTA_RET').asInteger;
  //Validación si el producto tiene un rango de fechas
  If (qqQuery.fieldByName('ID_PRODUCTO').AsInteger = 2) or
    (qqQuery.fieldByName('ID_PRODUCTO').AsInteger = 127) or
    (qqQuery.fieldByName('ID_PRODUCTO').AsInteger = 67) or
    (qqQuery.fieldByName('ID_PRODUCTO').AsInteger = 187) OR
    (qqQuery.fieldByName('ID_PRODUCTO').AsInteger = 247) or
  // se agrega el nuevo id producto 227 tu salud exceso EMA 16032006
  (qqQuery.fieldByName('ID_PRODUCTO').AsInteger = 227) Then Begin
    DFechaMin:= Trunc (Self.Objeto.Conexion.Fecha) + iNumDiasRet;
    dFechaMax := dFechaMin + iNumDias;
    Self.dtpIniVig.Date := dFechaMin;
  End
  Else Begin
    Self.dtpIniVig.Date := Self.Objeto.Conexion.Fecha + iNumDias;
    Self.dtpIniVig.Enabled := False;
  End;
End;
End;
If qqQuery.fieldByName('CVE_TIPO_RET_DIF').asString = 'TRD_OP' Then Begin
  // TRD_OP = OPCIONAL hoy
  Self.dtpIniVig.Date:= Self.Objeto.Conexion.Fecha;
  Self.dtpIniVig.Enabled := True;
End;
// Actualiza la fecha fin
iIncremento:= qqQuery.fieldByName ('PLAZO_DEF_VENTA').asInteger;
If qqQuery.fieldByName('CVE_TIPO_PLAZO_VENTA').asString = 'PV_DIA' Then Begin
  Self.dtpFinVig.Date := Self.dtpIniVig.Date + iIncremento;
End;
If qqQuery.fieldByName('CVE_TIPO_PLAZO_VENTA').asString = 'PV_MES' Then Begin
  Self.dtpFinVig.Date := IncMonth(Self.dtpIniVig.Date, iIncremento );
End;
If qqQuery.fieldByName('CVE_TIPO_PLAZO_VENTA').asString = 'PV_AÑO' Then Begin
  Self.dtpFinVig.Date := IncMonth(Self.dtpIniVig.Date, iIncremento *12 );
End;
If Self.Objeto.Campos.FieldByName('ID_RENOVACION').AsInteger > 0 then Begin
  dtpIniVig.Enabled := False;
  dtpFinVig.Enabled := False;
end;
End;

```

Este procedimiento obtiene la fecha del sistema, le incrementa el tiempo diferido correspondiente y fija la fecha de inicio de acuerdo al tipo de producto y después, la fecha de fin es calculada sumando los 12 meses a la fecha de inicio.

Para el producto de GMM siempre la vigencia de la póliza será anual.

3. Dentro de esta parte de la interfaz se considera la más importante del desarrollo, ya que hace instancia de componentes e interfaces que tienen la responsabilidad de realizar los costos correctos de una póliza de acuerdo a la información ingresada.

4. Una vez que haya cumplido su función el motor de cálculo, el total de los costos se reflejará en la siguiente interfaz de cotización.

```
// ACTUALIZA EL RESUMEN DE LOS COSTOS DEL MOTOR
```

```
Procedure TwcasePoliza.ActualizaResumenDeCostosDeMotor;
```

```
var.
```

```
  sReferencia: String;
```

```
Begin
```

Sistema de Punto de Venta Portátil de Seguros

```
//Prima Neta
sReferencia:= Self.Objeto.FTTarifaGral.DameReferencia ('CostoPol.Imp_Prima_Neta_SAI', False);
Self.f1ResumenCosto.NumberRC [1,2]:= Self.Objeto.FTTarifaGral.ObtenerValorDatoNumero (sReferencia);
sReferencia:= Self.Objeto.FTTarifaGral.DameReferencia ('CostoPol.Imp_Prima_Neta_NAI', False);
Self.f1ResumenCosto.NumberRC [1,3]:= Self.Objeto.FTTarifaGral.ObtenerValorDatoNumero (sReferencia);
sReferencia:= Self.Objeto.FTTarifaGral.DameReferencia ('CostoPol.Imp_Prima_Neta_Total', False);
Self.f1ResumenCosto.NumberRC [1,4]:= Self.Objeto.FTTarifaGral.ObtenerValorDatoNumero (sReferencia);

//Gastos de Emisión
sReferencia:= Self.Objeto.FTTarifaGral.DameReferencia ('CostoPol.Imp_DerEmis_SAI', False);
Self.f1ResumenCosto.NumberRC [2,2]:= Self.Objeto.FTTarifaGral.ObtenerValorDatoNumero (sReferencia);
sReferencia:= Self.Objeto.FTTarifaGral.DameReferencia ('CostoPol.Imp_DerEmis_NAI', False);
Self.f1ResumenCosto.NumberRC [2,3]:= Self.Objeto.FTTarifaGral.ObtenerValorDatoNumero (sReferencia);
sReferencia:= Self.Objeto.FTTarifaGral.DameReferencia ('CostoPol.Imp_DerEmis_Tot', False);
Self.f1ResumenCosto.NumberRC [2,4]:= Self.Objeto.FTTarifaGral.ObtenerValorDatoNumero (sReferencia);

//Financiamiento
sReferencia:= Self.Objeto.FTTarifaGral.DameReferencia ('CostoPol.Imp_Rcgo_FPago_SAI', False);
Self.f1ResumenCosto.NumberRC [3,2]:= Self.Objeto.FTTarifaGral.ObtenerValorDatoNumero (sReferencia);
sReferencia:= Self.Objeto.FTTarifaGral.DameReferencia ('CostoPol.Imp_Rcgo_FPago_NAI', False);
Self.f1ResumenCosto.NumberRC [3,3]:= Self.Objeto.FTTarifaGral.ObtenerValorDatoNumero (sReferencia);
sReferencia:= Self.Objeto.FTTarifaGral.DameReferencia ('CostoPol.Imp_Rcgo_FPago_Tot', False);
Self.f1ResumenCosto.NumberRC [3,4]:= Self.Objeto.FTTarifaGral.ObtenerValorDatoNumero (sReferencia);

//Impuestos
sReferencia:= Self.Objeto.FTTarifaGral.DameReferencia ('CostoPol.Imp_Otros_Imptos_SAI', False);
Self.f1ResumenCosto.NumberRC [4,2]:= Self.Objeto.FTTarifaGral.ObtenerValorDatoNumero (sReferencia);
sReferencia:= Self.Objeto.FTTarifaGral.DameReferencia ('CostoPol.Imp_Otros_Imptos_NAI', False);
Self.f1ResumenCosto.NumberRC [4,3]:= Self.Objeto.FTTarifaGral.ObtenerValorDatoNumero (sReferencia);
sReferencia:= Self.Objeto.FTTarifaGral.DameReferencia ('CostoPol.Imp_Otros_Imptos_Tot', False);
Self.f1ResumenCosto.NumberRC [4,4]:= Self.Objeto.FTTarifaGral.ObtenerValorDatoNumero (sReferencia);

//IVA
sReferencia:= Self.Objeto.FTTarifaGral.DameReferencia ('CostoPol.Imp_IVA', False);
Self.f1ResumenCosto.NumberRC [5,2]:= Self.Objeto.FTTarifaGral.ObtenerValorDatoNumero (sReferencia);
Self.f1ResumenCosto.NumberRC[5,3] := 0;
Self.f1ResumenCosto.NumberRC[5,4] := Self.f1ResumenCosto.NumberRC[5,2];

//Prima Total
sReferencia:= Self.Objeto.FTTarifaGral.DameReferencia ('CostoPol.Imp_PT_SAI', False);
Self.f1ResumenCosto.NumberRC [6,2]:= Self.Objeto.FTTarifaGral.ObtenerValorDatoNumero (sReferencia);
sReferencia:= Self.Objeto.FTTarifaGral.DameReferencia ('CostoPol.Imp_PT_NAI', False);
Self.f1ResumenCosto.NumberRC [6,3]:= Self.Objeto.FTTarifaGral.ObtenerValorDatoNumero (sReferencia);
sReferencia:= Self.Objeto.FTTarifaGral.DameReferencia ('CostoPol.Imp_PT_Tot', False);
Self.f1ResumenCosto.NumberRC [6,4]:= Self.Objeto.FTTarifaGral.ObtenerValorDatoNumero (sReferencia);

End;
```

5. La información sobre el agente es reflejada automáticamente y registrada desde el momento que un emisor es dado de alta y lo realiza a través de una consulta sobre la base de datos.

```
// FIJAR EL AGENTE DE LA COLECCIÓN
Procedure TwcasePoliza.FijarAgente;
var
  qyQuery : TQuery;
  sSQL : String;
Begin

  Self.Objeto.Campos.PasaInfoDeControlesACampos;
  If Self.Objeto.IdAgteConexion > 0 then Begin

    //Fija el agente
```

```

Self.edIdAgente.Text:= intToStr (Self.Objeto.IdAgteConexion);
Self.Objeto.Campos.FieldByName ('ID_AGENTE').PasaInfoDeControlACampo;
sSQL:= ' SELECT NUM_AGENTE FROM AE_AGENTE WHERE ID_AGENTE = ' + intToStr (Self.Objeto.IdAgteConexion);
qyQuery:= ObtenerQuery (Self.Objeto.Conexion.BasePrincipal, sSQL);
If qyQuery <> Nil Then Begin
    Self.edNumAgente.Text := qyQuery.FieldByName('NUM_AGENTE').AsString;
    qyQuery.free;
End;

Self.edNumAgente.Enabled := False;
Self.btnAgente.Enabled := False;
End;

//Valida si el agente tiene al producto asignado para prueba
sSQL := ' SELECT * FROM AE_PROD_AGTE_PRUEBA ' +
    ' WHERE ID_UNIDAD_NEGOCIO = ' + Self.Objeto.Campos.FieldByName ('ID_UNIDAD_NEGOCIO').AsString +
    ' AND ID_PRODUCTO = ' + Self.Objeto.Campos.FieldByName('ID_PRODUCTO').AsString +
    ' AND ID_AGENTE = ' + intToStr(Self.Objeto.IdAgteConexion) +
    ' AND B_DELETE = "F" ';

qyQuery:= ObtenerQuery (Self.Objeto.Conexion.BasePrincipal, sSQL);
if qyQuery <> Nil Then Begin
    Self.Objeto.Campos.FieldByName('B_PRUEBA_AGTE').AsString := 'V';
    qyQuery.free;

```

6. De acuerdo al tipo de conducto de cobro y la forma de pago se reflejará la información requerida, ver figura 62.

CONDUCTO AGENTE:

Conducto de Cobro y Forma de Pago

Tipo de Conducto de Cobro: AGENTE Conducto de Cobro: AGENTE Forma de Pago: ANUAL % de Recargo: 0.00

Recibos: Enviar Recibos ... Periodicidad: Según Forma de Pago

FOLIO	Fecha	Prima
1	21/01/2008	54,409.26

Información del Conducto de Cobro

CONDUCTO CHEQUES

Conducto de Cobro y Forma de Pago

Tipo de Conducto de Cobro: CARGO A CHEQUERA Conducto de Cobro: CUENTA_CHEQUES Forma de Pago: ANUAL % de Recargo: 0.00

Recibos: Enviar Recibos ... Periodicidad: Según Forma de Pago

FOLIO	Fecha	Prima
1	21/01/2008	54,409.26

Información del Conducto de Cobro

BANCO: BBVA BANCOMER

CUENTAHABIENTE: "Mismo nombre del contratante de la póliza"

CLABE: _____

CONDUCTO TARJETA DE CRÉDITO

Conducto de Cobro y Forma de Pago

Tipo de Conducto de Cobro: TARJETA DE CREDITO Conducto de Cobro: AMERICAN_EXPRESS Forma de Pago: ANUAL % de Recargo: 0.00

Recibos: Enviar Recibos ... Periodicidad: Según Forma de Pago

FOLIO	Fecha	Prima
1	21/01/2008	54,409.26

Información del Conducto de Cobro

TARJETAHABIENTE: "Mismo nombre del contratante de la póliza"

NÚMERO DE TARJETA: 0 VIGENCIA: 20/08/2001

CÓDIGO DE 4 NÚMEROS: _____

Figura 62. Tipos de conducto de cobro

4.4 Fase de pruebas

Esta fase se enfoca hacia el funcionamiento preciso del sistema siguiendo con la forma de operación, con ello se pretende detectar tempranamente errores e inconsistencias, ya sea de los componentes desarrollados o en los documentos de impresión de pólizas.

Para esta fase se realizan las siguientes actividades.

- Ejecutar el sistema
- Probar las entradas
- Observar las salidas
- Identificar los errores
- Corregir el sistema

Las pruebas de integración se utilizan para verificar que los componentes desarrollados interaccionan entre sí de forma apropiada después de haber sido integrados en el desarrollo.

Para llevar el control de las pruebas de funcionalidad y de cada elemento sobre los módulos diseñados en el sistema, se realizó a través de matrices.

Matriz de pruebas para control de acceso, ver figura 63.

Emisor	Usuario	Password	Ingreso de usuario válido	Ingreso de usuario inválido	Validación de usuario	Validación de password	Identificación de nombre de usuario	Botón de aceptar	Botón de cancelar	Aprobación de usuario
01926845	00005635	00005635	si	no	si	si	si	si	si	si
01926845	00005635	00001234	no	si	si	si	si	si	si	no

Figura 63. Matriz de prueba para control de acceso

Matriz de pruebas para el menú principal, ver figura 64.

Elemento del menú	Muestra navegador	Muestra información	Muestra filtros de búsqueda
Botón Cotizaciones	si	si	si
Botón Pólizas y Alta Endosos	si	si	si
Botón Recibos	si	si	si
Botón Consulta de Endosos	si	si	si
Botón Renovaciones	si	si	si
Botón Recálculos	si	si	si
Botón Administrador de contratantes	si	si	si

Figura 64. Matriz de prueba para el menú principal

Matriz de pruebas para pólizas, ver figura 65.

Filtros de búsqueda	Resultado	Información del Explorador	Resultado
Número de Póliza	01010135000045	Número de Póliza	01010135000045
Nombre de Producto	Tu salud	Número de Renovación	0
Nombre de Contratante	Mario González	Año de Operación	0
Id Emisor	01010135	Folio de Renovación	0
Situación de Transferencia	V	Contratante	Mario González
Agrupador	sin	Agrupador	sin
Rango de fechas de vigencia	01/01/2009 01/03/2009	Producto	Tu salud
		Fecha Emisión	01/02/2009
		Fh Inicio Vig	01/02/2009
		Fh Fin Vig	01/02/2010
		Forma de pago	anual
		Conducto de Cobro	agente
		Prima Total	\$ 5,250.00
		Emisor	01010135
		Póliza Enviada	V
		Póliza de Prueba	V

Figura 65. Matriz de prueba para hacer una cotización

Pruebas de integración

Se utilizan para verificar que los componentes interaccionan entre si de forma apropiada después de haber sido integrados en el desarrollo.

Para este proyecto nos basamos en la elaboración de pantallas, esto es, una vez que se concluía una pantalla se probaba junto con las que ya estaban implementadas para verificar que:

- La secuencia entre las pantallas fuera la correcta.
- Que los casos de uso arrojaran los resultados esperados.
- Pruebas de sistema.
- Se usan para probar que el sistema funciona correctamente como un todo.
- Cada una de estas pruebas examina las combinaciones de casos de uso bajo condiciones diferentes.

Evaluación de pruebas

Dentro de la evaluación de pruebas se realizaron las siguientes actividades:

- Se documentaron los tipos de errores encontrados para ser evaluados y realizar el mantenimiento respectivo.
- Se mantuvo informado a los usuarios de los errores.
- El mecanismo de seguridad del sistema, pudo validar la información de los campos requeridos y obligatorios manteniendo la integridad de datos.
- El manejo de mensajes de error, fue una manera de avisar hacia los usuarios que la o las acciones realizadas eran correctas o incorrectas.

4.5 Fase de mantenimiento

Para realizar el mantenimiento se deben considerar los requerimientos mínimos de hardware con el objetivo de mantener un ambiente estable para todos los componentes que integran el sistema:

- Tipo de PC
- Velocidad procesador
- Memoria RAM
- Capacidad de almacenamiento (disco duro)

Una vez que se cumple con lo anterior se procede a:

- Instalación del software de desarrollo
- Configurar componentes y librerías necesarias para su funcionamiento correcto
- Tener las últimas actualizaciones de código del sistema

Se opta por manejar software y hardware que satisfaga la capacidad de procesamiento, almacenamiento, velocidad y manejo de memoria.

Una vez implementado el ambiente de mantenimiento, debe seguirse el siguiente plan dependiendo de la demanda de usuarios:

1. **Mantenimiento correctivo:** Se debe tener activo el ambiente de desarrollo, disponible para corregir las primeras incidencias evaluadas y que se pueden manejar de manera inmediata durante el inicio de su operación, debido a que no es razonable asumir que en la fase de pruebas se hayan descubierto todos los errores del sistema.
2. **Mantenimiento preventivo:** Realizar una bitácora y registrar todos los cambios hechos sobre el sistema, esto ayudará a realizar el mantenimiento preventivo.

3. Mantenimiento perfectivo: Se llevará de acuerdo a las situaciones presentadas con el fin de mejorar el performance del sistema. Este tipo de situaciones deberán ser analizadas y evaluadas, verificando costo-beneficio de implementación.
4. Mantenimiento adaptativo: Ante el constante crecimiento tecnológico de hardware y del sistema operativo, debe manejarse de manera cuidadosa; ya que se deben tomar en cuenta tanto librerías como componentes funcionen correctamente ante los nuevos cambios.

Este proceso de mantenimiento dependerá del equipo de trabajo, debe ser personal, calificado con amplia experiencia, tanto en el negocio como en las herramientas de desarrollo.

Capítulo 5

Resultados, impacto y conclusiones

5.1 Resultados

Los resultados obtenidos a raíz del nuevo sistema, favorecieron el crecimiento competitivo de la empresa aseguradora. Para obtener dichos resultados se realizó una comparación entre el sistema antiguo y el nuevo sistema, siguiendo los indicadores principales que se muestran en la figura 66.

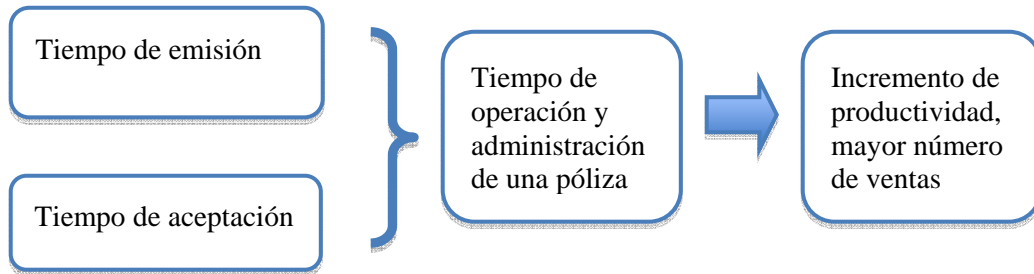


Figura 66. Indicadores de comparación

Para realizar la medición del sistema fue necesario recopilar el tiempo transcurrido en cada una de la operaciones realizadas por el agente, a éstas se les asignó una variable, la cual ayudará a identificar qué parte del proceso ocupa la mayor cantidad de tiempo. De esta forma será más fácil realizar la comparación del sistema antiguo con el nuevo sistema.

En la siguiente tabla se muestran los tiempos obtenidos del sistema antiguo para cada proceso de operación que realizaba un agente, ver figura 67.

Operación que realiza el agente de ventas	Variable	Respuesta Hrs	Total Días
Construcción de una cotización	A	6	0.25
Recopilación de documentación, como requisitos del cliente	B	24	1
Envío de documentación vía mensajería	C	24	1
Aceptación de póliza por parte de la aseguradora	D	120	5
Envío de contrato de aceptación de la póliza vía mensajería	E	24	1
Registro de información sobre el sistema antiguo	F	2	0.08
Total		200	8.33

Figura 67. Proceso de emisión de una póliza con el sistema antiguo

Gráficamente se observa el resultado obtenido, el tiempo total en días de cada uno de los procesos para poder emitir una póliza hacia el cliente con el sistema antiguo, donde la variable D que es la aceptación de la póliza por parte de la aseguradora, ocupa el mayor tiempo en procesar la información, ver figura 68.

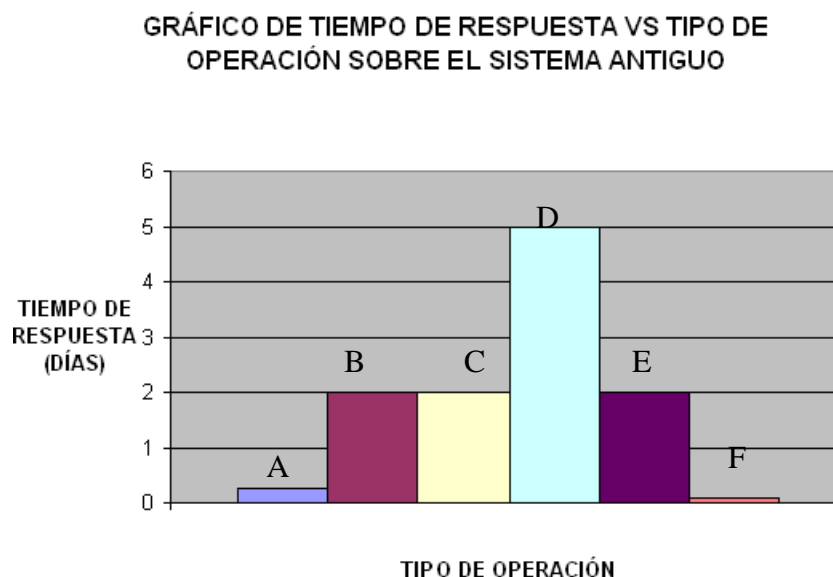


Figura 68. Gráfica de tiempos obtenidos con el sistema antiguo

Para el nuevo sistema también se realizó la recopilación del tiempo para cada proceso en la emisión de pólizas, se obtuvo una medición del tiempo de respuesta hacia el cliente, arrojando los siguientes datos, ver figura 69.

Operación que realiza el agente de ventas	Variable	Respuesta Hrs	Total Días
Construcción de una cotización	A	1	0.0416
Captura de información del cliente	B	2	0.0833
Envío de documentación vía electrónica	C	0.08	0.0035
Impresión de contrato de póliza	D	0.01	0.0006
Aceptación de la póliza por parte de la aseguradora	E	12	0.5
Total		15.09	0.629

Figura 69. Proceso de emisión de una póliza con el nuevo sistema

El resultado obtenido, es el tiempo total en horas y días para poder emitir una póliza hacia el cliente con el nuevo sistema, donde se puede observar gráficamente qué proceso ocupa la mayor capacidad de tiempo. En este caso, la variable E representa la aceptación de la póliza por la aseguradora, como se muestra en la figura 70.

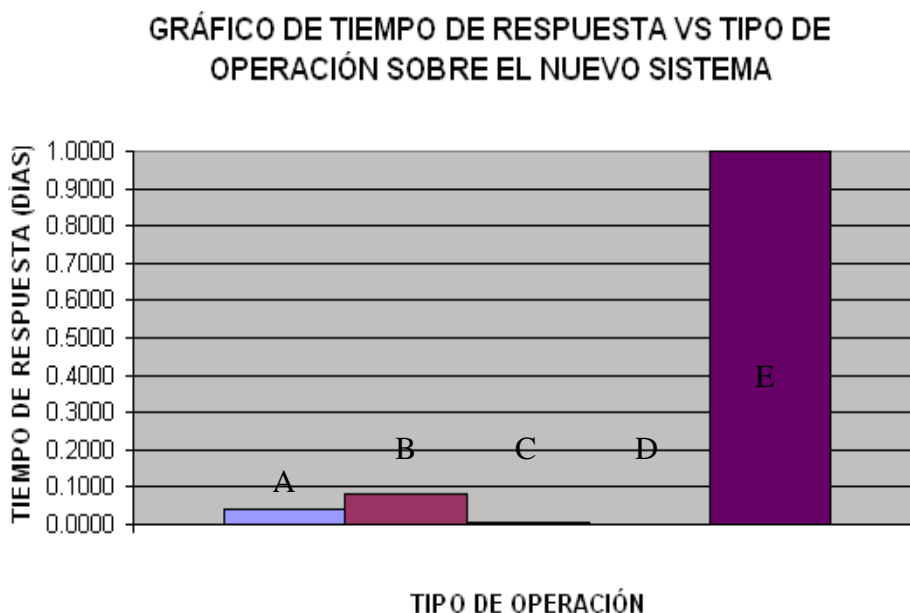


Figura 70. Gráfica de tiempos obtenidos con el nuevo sistema

Resultado comparativo de producción

Con los tiempos obtenidos en cada sistema, se pudo observar que el proceso de aceptación de una póliza se redujo drásticamente con el nuevo sistema, de acuerdo a las cifras mostradas en las figuras 69 y 70, se redujo aproximadamente un 80% del tiempo total comparándolo con el sistema antiguo.

Después de haber obtenido los tiempos de emisión por cada sistema, se calculó el número de ventas por cada emisor en un periodo anual para poder medir el nivel de productividad, se obtuvo un muestreo de varios emisores para representarlos gráficamente.

Con el sistema antiguo se observa que el máximo número de pólizas vendidas fue de 48 en un periodo anual, ver figura 71.

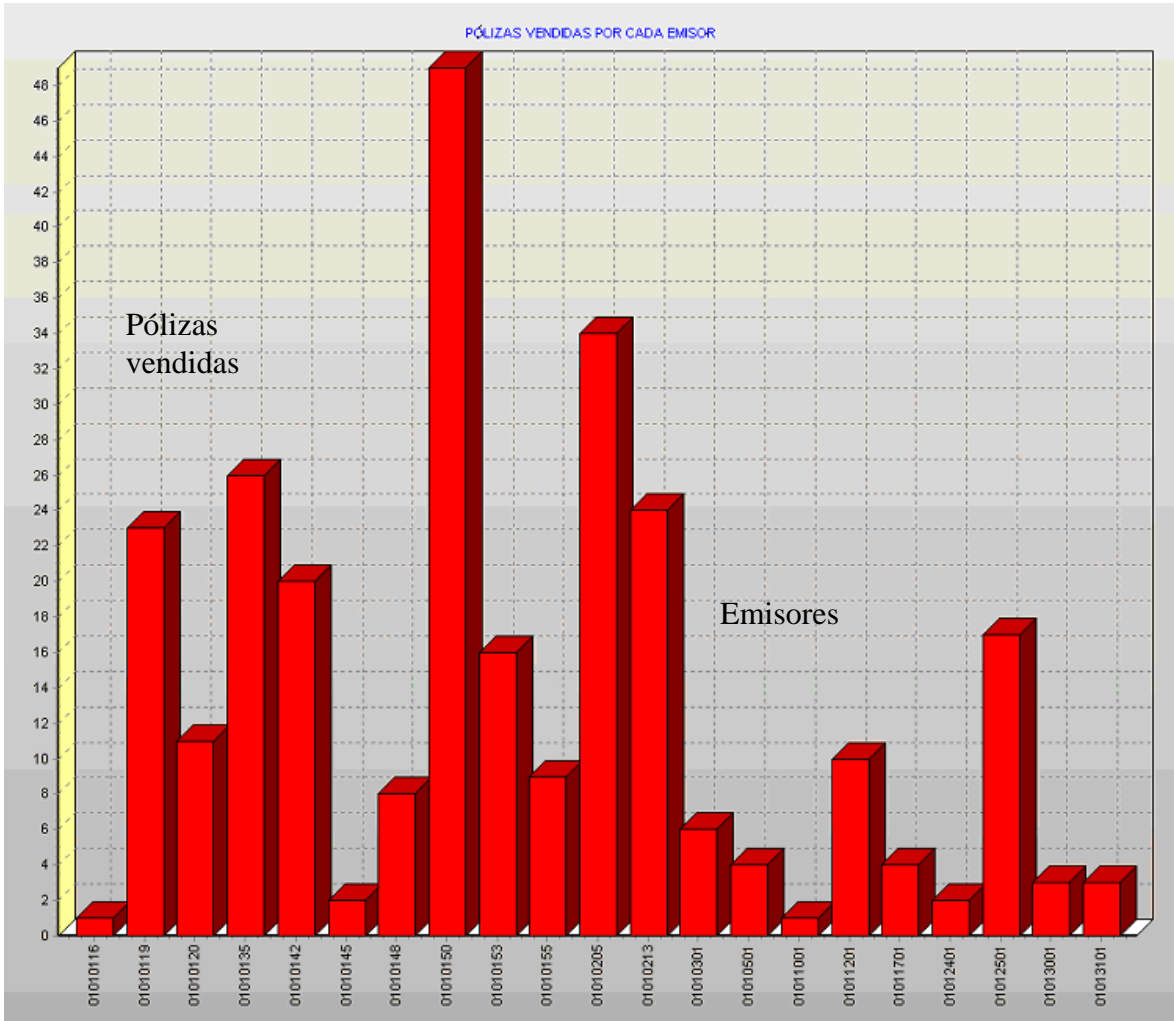


Figura 71. Gráfica de ventas por cada emisor en un periodo anual con el sistema antiguo

Posteriormente se realizó la extracción de información con el nuevo sistema por cada emisor, gráficamente se observó que el máximo número de pólizas vendidas fue de 146 en un periodo anual, ver figura 72.

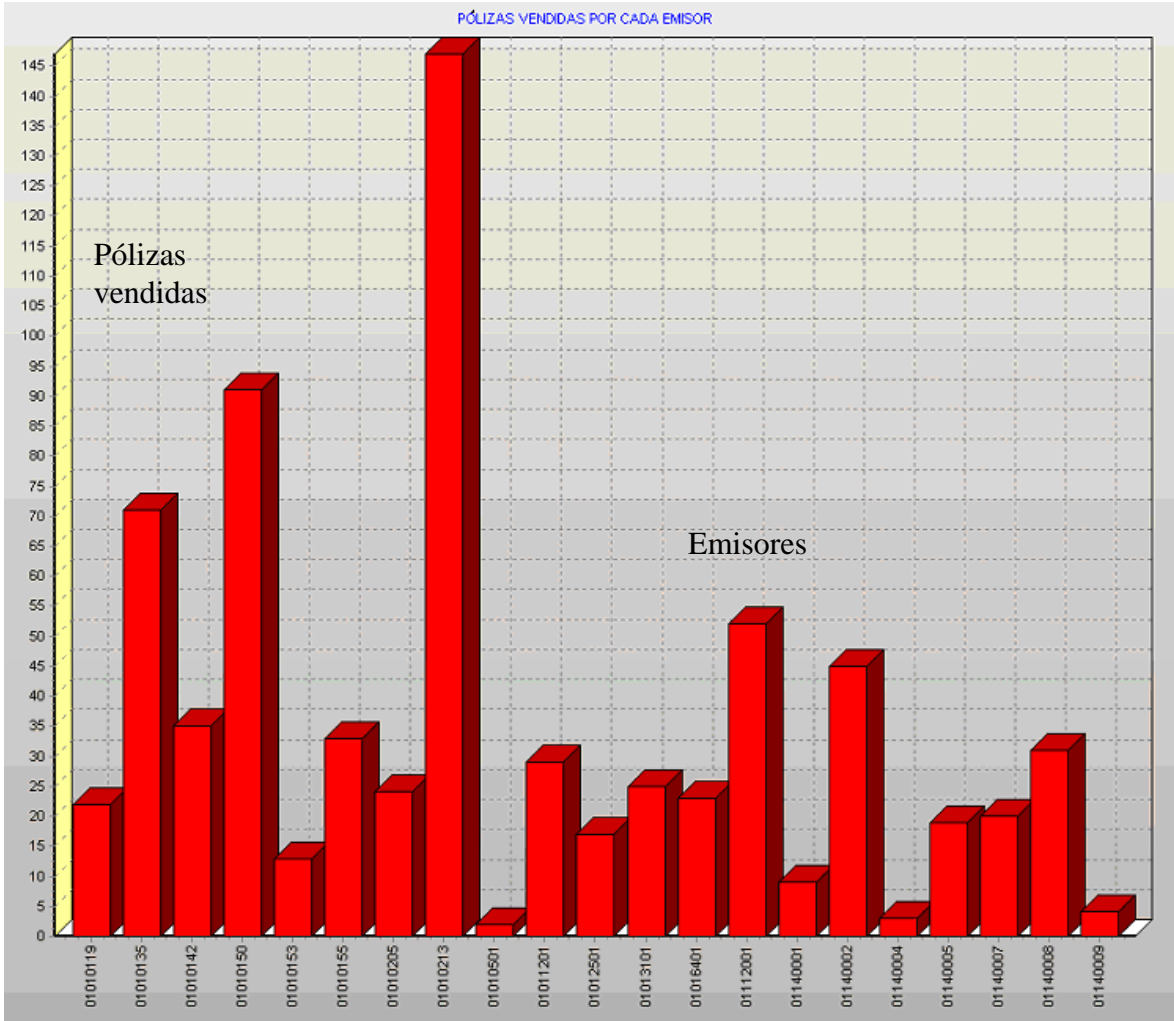


Figura 72. Gráfica de ventas en un periodo anual con el nuevo sistema

Analizando las dos gráficas, figura 71 y figura 72, es visible que hubo un incremento de aproximadamente un 200% en ventas, por parte de los agentes de seguros que utilizaron el nuevo sistema, arrojando resultados positivos para la empresa.

Entre otros resultados cualitativos observados por el nuevo sistema son:

- Proceso de cobranza ágil.
- La atención personalizada a clientes incrementó el número de ventas, porque el vendedor a través del nuevo sistema siempre brindó una atención constante, dando confianza a sus necesidades para que fueran cubiertas de forma servicial y eficaz.
- El cliente sabe cuánto debe de pagar antes de aprobar su compra, dando confiabilidad, el nuevo sistema ofreció las mejores ofertas de formas pago, indispensables para adecuarse a las situaciones del comprador.
- Consultas de información administrativa de manera rápida y confiable.
- Registro en tiempo real de la venta.
- Reutilización de equipo portátil para otras actividades.

En general, el sistema ayudó a lograr una relación estrecha cliente-vendedor originando fructíferas y continuas ventas.

El diseño del nuevo sistema permite el almacenamiento de una mayor cantidad de información en una base de datos personal, dando un performance de respuesta más rápido en los procesos, por ello se puede decir que la información almacenada es confiable y segura.

Para reducir los costos se implementó un motor de cálculo poderoso y dinámico que refleja al instante cualquier cambio en la elaboración de la cotización, en consecuencia se reduce el tiempo en la emisión de una póliza.

Se crearon interfaces amigables para que el agente de ventas tenga gran facilidad en el manejo del sistema.

El tiempo de operación del sistema fue un factor importante para cumplir con toda una gama de expectativas de superación y crecimiento dentro de la empresa, dando como resultado operaciones complejas en el menor tiempo posible y adicionalmente automáticas.

El desarrollo del nuevo sistema concluyó satisfactoriamente, siempre siguiendo el objetivo general mencionado en el capítulo uno y enfocándose hacia un modelo de ingeniería de software como se detalla en el capítulo tres.

5.2 Impacto

Actualmente con el sistema implementado sobre toda la fuerza de ventas de la empresa aseguradora, ha incrementado el porcentaje de ventas considerablemente como se mostró en la parte de resultados, cumpliendo así con las expectativas y objetivos propuestos en el capítulo uno.

De acuerdo con el análisis de impacto en ambos sistemas se pudo observar que el nuevo sistema muestra ventaja en los siguientes puntos:

1. Impacto sobre tareas manuales por parte del agente de ventas de seguros, ya que para realizar la cotización hasta la emisión de una póliza en el sistema antiguo era:

- Capturar manualmente la información sobre papelería prediseñada por la empresa aseguradora con el formato para realizar el archivado correspondiente.
- La cotización de una póliza era realizada a través de una hoja de cálculo Excel, que estaba diseñada por el área técnica de actuaría.
- El agente de ventas debía recabar toda la documentación necesaria del contratante posteriormente enviarla por medio de la empresa de Telégrafos de México a la oficina matriz de la empresa aseguradora.
- El agente esperaba un tiempo indeterminado para saber si era aceptada su póliza.

Ahora con el nuevo sistema, en un tiempo muy reducido se puede hacer entrega de la póliza emitida hacia el cliente, ya que elimina las tareas manuales para cotizar y enviar documentación física.

2. Impacto en horas de procesamiento de información cliente-aseguradora, ya que el tiempo era variable y extenso en el sistema antiguo, debido a que:

- La información era enviada a través de Telégrafos de México, por lo que el tiempo de entrega dependía de la empresa de correos.
- Recibiendo la documentación en la empresa aseguradora, era clasificada y enviada a cada recurso correspondiente para realizar el análisis de riesgo de la póliza y poder obtener el dictamen.
- El tiempo de respuesta para la administración de una póliza por parte de la empresa aseguradora hacia sus agentes era variable, ya que en caso de un error de captura de información por parte del agente, la aceptación era pospuesta hasta que el agente hiciera la corrección y volviera a enviar por correo toda la información y documentación corregida.

Ahora con el nuevo sistema, se cuenta con un módulo de selección de riesgos que verifica si la persona titular es elegible para contratar dicha póliza. Una vez que el sistema acepta el riesgo, basta con una conexión hacia Internet para vaciar la información en la base central de la empresa aseguradora.

5.3 Conclusiones

Anteriormente, la compañía de seguros contaba con un sistema para aceptar el riesgo muy rudimentario, manual, no era confiable y el tiempo de respuesta para aceptar una póliza era muy largo. Estos factores producían márgenes de error muy frecuentes que se traducían en una deficiente competitividad en el mercado de seguros, principalmente hacia al cliente.

Esta falta de competitividad lleva a la aseguradora a buscar una reestructuración en su proceso de emisión para disminuir los tiempos de respuesta ante los próximos asegurados y de esta forma incrementar las ventas.

Con la estrategia que propuso la empresa aseguradora, se lograron reducir tiempos de respuesta, errores humanos, facilitar las ventas de los agentes, seguridad en la información, disponibilidad de la información y capacidad de almacenamiento, objetivos específicos mencionados en el capítulo uno. El resultado fue un impacto directamente sobre la productividad de la empresa aseguradora, ya que fue notable el incremento en ventas en un periodo muy corto. El éxito de éste fue producto de utilizar una disciplina llamada ingeniería de software, que proporcionó las bases para desarrollar y crear el sistema de manera efectiva de acuerdo al marco teórico descrito en el capítulo tres.

De acuerdo a los resultados, las ventajas que ofrece el sistema de venta son grandes porque redujo drásticamente el tiempo de espera hacia los clientes, es decir el tiempo de servicio hacia el cliente reflejó un incremento de ventas.

Los alcances logrados en el sistema son los siguientes:

- Desarrollo de una herramienta computacional (Sistema de punto de venta portátil de seguros) que facilita y agiliza el flujo de operación para la venta de seguros.
- Elaboración de interfaces de presentación y recepción de información, permitiendo al usuario facilidad de control y administración de negocios.
- La información del sistema de punto de venta es personalizada y se puede instalar desde arquitecturas de PC hasta laptop, siempre y cuando estén dentro de los requisitos mínimos de sistema operativo.
- Los usuarios pueden realizar la instalación sin la asesoría de un técnico especializado.
- El sistema cuenta con una base de datos relacional que facilita el mantenimiento del sistema.
- Los negocios realizados por los usuarios se pueden mostrar de manera impresa.

- Los tiempos de respuesta para la emisión de una póliza se redujeron en gran manera en los procesos de cálculo de primas.

La experiencia recibida durante el desarrollo del sistema fue enriquecida tanto en el aspecto tecnológico como en el humano, debido a que los usuarios (*agentes*) transmitían sus conocimientos en el área de seguros, así como sus necesidades para facilitarles el trabajo de ventas; con ello exigía realizar investigaciones de tecnologías de desarrollo y poder cubrir los objetivos mencionados en el capítulo uno.

La parte crucial del éxito del proyecto estuvo en aplicar un modelo de ingeniería de software, que permitió tener el control en cada fase del desarrollo del sistema, reflejando una alta capacidad y satisfacción hacia los usuarios. Con el modelo, ciclo de vida clásico del software, se pudo comprobar que siguiendo una serie de etapas simples se puede obtener un producto de alta calidad. La teoría de este concepto, permitió traducir todas las necesidades del cliente en un sistema capaz de resolver altos conceptos del negocio de seguros.

En cuestiones económicas es predecible observar que COSTO/BENEFICIO será amortizado por el nuevo sistema en un lapso muy corto de tiempo, ya que de acuerdo a los resultados mostrados en la figura 72, el número de ventas se incrementó en un porcentaje aproximadamente del 200% en un periodo anual.

La visión futurista hacia la realización de este tipo de sistemas computacionales pretende eliminar cada vez más la intervención humana en operaciones repetitivas y de cálculo, con ello generando una mayor competitividad para las empresas, también cada vez más y conforme pasa el tiempo, las soluciones empresariales basadas en tecnologías de la información, permiten al vendedor establecer una relación más estrecha hacia sus clientes logrando la confiabilidad necesaria para vender sus productos.

La realización de este proyecto permitió aplicar en gran parte los conocimientos adquiridos durante el desarrollo profesional de la Facultad de Ingeniería. También es una satisfacción proporcionar una herramienta que será de utilidad y que traerá beneficios. Por todo lo anterior, se puede decir que este proyecto permitió crecer tanto en el ámbito profesional como personal.

Anexos

Anexo 1 Instalación Oracle Server 9i

El inicio de la instalación es con la siguiente interfaz, indicando rutas de carga del software, ver figura 73.



Figura 73 Ruta de instalación

Como siguiente opción, seleccionar el tipo de producto de Oracle9i Database 9.2.0.1.0, ver figura 74.

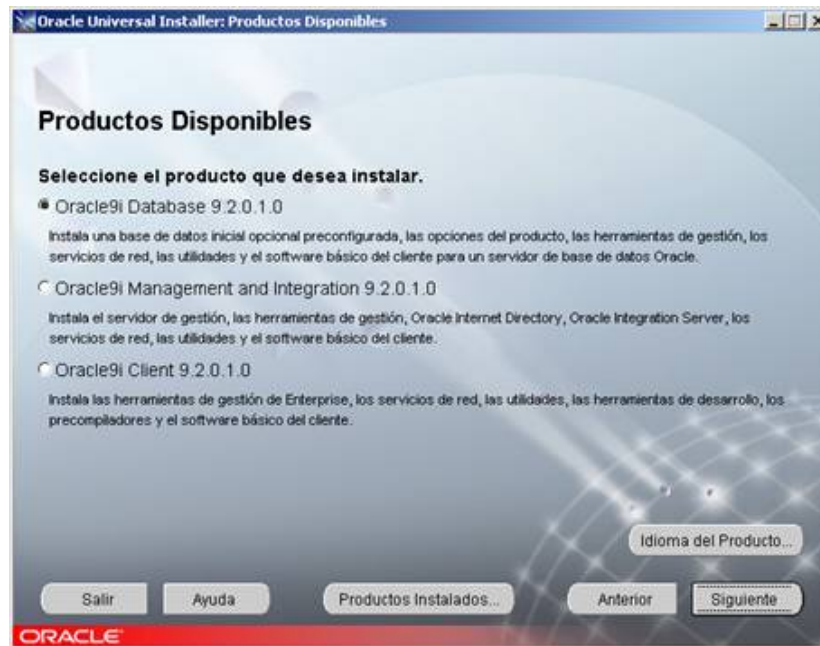


Figura 74. Producto a instalar

Después se debe seleccionar el tipo de instalación, ver figura 75.

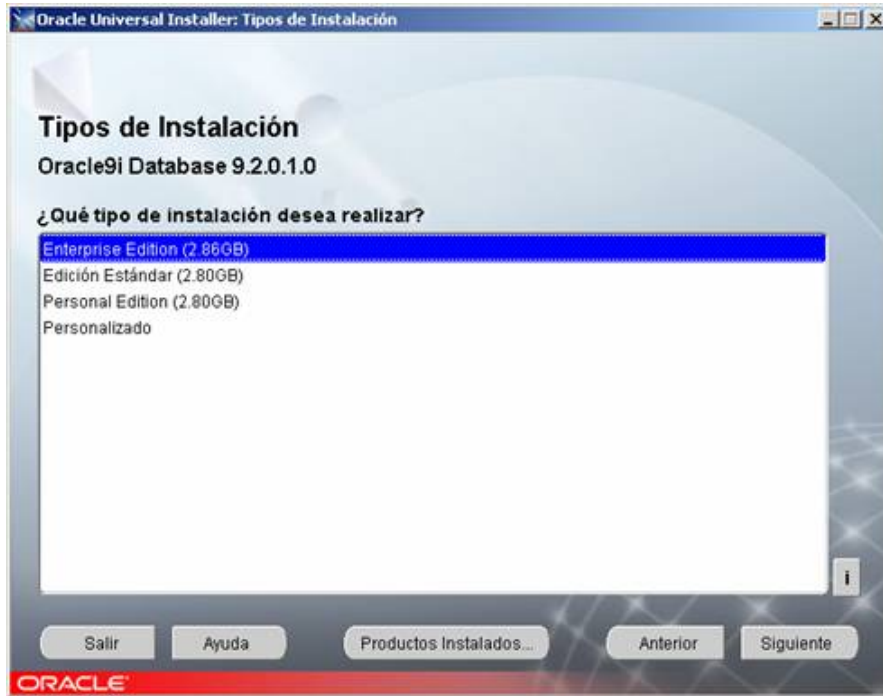


Figura 75. Tipo de instalación

Seleccionar *Enterprise Edition*, para realizar una instalación más avanzada (especificando manualmente las opciones a instalar) seleccionar modo *Personalizado*, ver figura 76.



Figura 76. Componentes a instalar

Seleccionar una opción de configuración dependiendo del uso que se quiera dar a la base de datos, normalmente es para *Uso General*. Si no queremos crear una base de datos en el proceso de instalación se puede crear en otro momento, seleccionar *Sólo software*, ver figura 77.

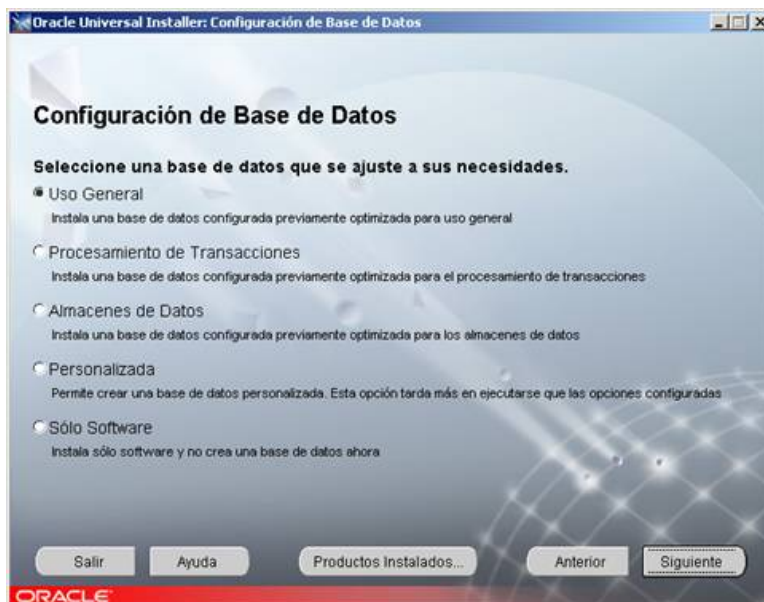


Figura 77. Opción uso general

Seleccionar el puerto de comunicación para Oracle. Este parámetro es muy importante pues, si decidimos cambiar el puerto, cuando queramos que un cliente se conecte al servidor deberemos especificar el puerto que hayamos seleccionado en este punto de la instalación, ver figura 78.



Figura 78. Puerto de comunicación

Especificar el nombre de la base de datos con un máximo de 8 caracteres, ver figura 79.



Figura 79. Nombramiento de la base de datos

Nota: el *SID* es el identificador interno que utilizará Oracle para referenciar a nuestra base de datos, se puede elegir uno diferente al del nombre de la Base de Datos, aunque se suele utilizar el mismo.

En este punto de la instalación seleccionaremos la ubicación de los archivos de la base de datos que la instalación creará.

En nuestro caso, instalaremos los archivos de la base de datos localmente para el desarrollo del sistema de punto de venta, ver figura 80.



Figura 80. Ubicación de los archivos de bases de datos

Seleccionar el juego de caracteres por default para Oracle, ya que utilizará el idioma del sistema operativo, es la opción recomendada. Este parámetro sólo sería importante en el caso de querer exportar la base de datos a otro servidor de Oracle, ver figura 81.



Figura 81. Juego de caracteres de la base de datos

Como último paso aparece una ventana, especificando el software que se va a instalar, tras comprobar que es correcto pulsar en *Instalar*, ver figura 82.



Figura 82. Verificación de software y características

Tras la instalación de Oracle y la creación de la base de datos aparecerá una ventana indicando que el proceso de instalación ha finalizado, ver figura 83.



Figura 83. Finaliza la instalación

Anexo 2 Scripts para creación de base de datos

SCRIPT PARA TABLA (PERSONA)

```
CREATE TABLE "ADM"."PERSONA"  
(  
    "ID_PERSONA" VARCHAR2(16) NOT NULL ENABLE,  
    "A_PATERNO" VARCHAR2(30),  
    "A_MATERNO" VARCHAR2(30),  
    "NOMBRE" VARCHAR2(30),  
    "F_NACIMIENTO" DATE,  
    "CVE_GENERO" VARCHAR2(6) NOT NULL ENABLE,  
    "CURP" VARCHAR2(30),  
    "RAZON_SOCIAL" VARCHAR2(90) NOT NULL ENABLE,  
    "ID_CONTACTO" VARCHAR2(16),  
    "CVE_TIPO_PERSONA" VARCHAR2(6) NOT NULL ENABLE,  
    "ID_DOMICILIO_PARTICULAR" VARCHAR2(16),  
    "ID_DOMICILIO_OFNA" VARCHAR2(16),  
    "RFC" VARCHAR2(15),  
    "TEL_PARTICULAR" VARCHAR2(30),  
    "TEL_OFNA_DIRECTO" VARCHAR2(30),  
    "TEL_OFNA_CONMUTADOR" VARCHAR2(30),  
    "EXT_OFNA" VARCHAR2(5),  
    "TEL_MOVIL" VARCHAR2(30),  
    "FAX_PARTICULAR" VARCHAR2(30),  
    "FAX_OFNA" VARCHAR2(30),  
    "TEL_RLOCALIZADOR" VARCHAR2(30),  
    "PIN_RLOCALIZADOR" VARCHAR2(10),  
    "EMAIL" VARCHAR2(100),  
    "EPAGE_WWW" VARCHAR2(50),  
    "CVE_NACIONALIDAD" VARCHAR2(6),  
    "TX_PROFESION" VARCHAR2(100),  
    "TX_GIROACTIVIDAD" VARCHAR2(100),  
    "FH_CONSTITUCION" DATE,  
    CONSTRAINT "PK_PERSONA" PRIMARY KEY ("ID_PERSONA")  
    CONSTRAINT "FK01_PERSONA" FOREIGN KEY ("ID_DOMICILIO_PARTICULAR")  
    REFERENCES "ADM"."DOMICILIO" ("ID_DOMICILIO") ENABLE,  
    CONSTRAINT "FK02_PERSONA" FOREIGN KEY ("ID_DOMICILIO_OFNA")  
    REFERENCES "ADM"."DOMICILIO" ("ID_DOMICILIO") ENABLE  
)  
CREATE UNIQUE INDEX "ADM"."PK_PERSONA" ON "ADM"."PERSONA" ("ID_PERSONA")  
ALTER TABLE "ADM"."PERSONA" ADD CONSTRAINT "PK_PERSONA" PRIMARY KEY  
("ID_PERSONA")
```

SCRIPT PARA TABLA (UNIDAD_NEGOCIO):

```
CREATE TABLE "ADM"."UNIDAD_NEGOCIO"
(
  "ID_UNIDAD_NEGOCIO" NUMBER (8,0) NOT NULL ENABLE,
  "UNIDAD_NEGOCIO" VARCHAR2 (30) NOT NULL ENABLE,
  "ID_PERSONA" VARCHAR2 (16),
  "ID_UNIDAD_PADRE" NUMBER (8,0),
  "CVE_TIPO_UNIDAD_NEGOCIO" VARCHAR2 (6) NOT NULL ENABLE,
  CONSTRAINT "PK_UNIDAD_NEGOCIO" PRIMARY KEY ("ID_UNIDAD_NEGOCIO")
  CONSTRAINT "FK01_UNIDAD_NEGOCIO" FOREIGN KEY ("ID_PERSONA")
  REFERENCES "ADM"."PERSONA" ("ID_PERSONA") ENABLE,
  CONSTRAINT "FK02_UNIDAD_NEGOCIO" FOREIGN KEY ("ID_UNIDAD_PADRE")
  REFERENCES "ADM"."UNIDAD_NEGOCIO" ("ID_UNIDAD_NEGOCIO") ENABLE
)

CREATE UNIQUE INDEX "ADM"."PK_UNIDAD_NEGOCIO" ON "ADM"."UNIDAD_NEGOCIO"
("ID_UNIDAD_NEGOCIO")
ALTER TABLE "ADM"."UNIDAD_NEGOCIO" ADD CONSTRAINT "PK_UNIDAD_NEGOCIO"
PRIMARY KEY ("ID_UNIDAD_NEGOCIO")
```

SCRIPT PARA TABLA (DOMICILIO):

```
CREATE TABLE "ADM"."DOMICILIO"
(
  "ID_DOMICILIO" VARCHAR2(16) NOT NULL ENABLE,
  "ID_PAIS" NUMBER(4,0) NOT NULL ENABLE,
  "ID_ESTADO" NUMBER(3,0) NOT NULL ENABLE,
  "CODIGO_POSTAL" CHAR (5),
  "ID_USUARIO_CP" NUMBER (2,0),
  "COLONIA" VARCHAR2 (50),
  "CALLE" VARCHAR2 (50),
  "NUMERO_OFICIAL" VARCHAR2 (5),
  "NUMERO_INTERIOR" VARCHAR2 (5),
  "TX_REFERENCIA" VARCHAR2 (2000),
  CONSTRAINT "PK_DOMICILIO" PRIMARY KEY ("ID_DOMICILIO")
  CONSTRAINT "FK01_DOMICILIO" FOREIGN KEY ("ID_PAIS", "ID_ESTADO")
  REFERENCES "ADM"."ESTADO" ("ID_PAIS", "ID_ESTADO") ENABLE,
  CONSTRAINT "FK02_DOMICILIO" FOREIGN KEY ("ID_PAIS", "CODIGO_POSTAL",
  "ID_USUARIO_CP")
  REFERENCES "ADM"."CODIGO_POSTAL" ("ID_PAIS", "CODIGO_POSTAL",
  "ID_USUARIO_CP") ENABLE
)

CREATE UNIQUE INDEX "ADM"."PK_DOMICILIO" ON "ADM"."DOMICILIO" ("ID_DOMICILIO")
ALTER TABLE "ADM"."DOMICILIO" ADD CONSTRAINT "PK_DOMICILIO" PRIMARY KEY
("ID_DOMICILIO")
```

SCRIPT PARA TABLA (PAIS) :

```
CREATE TABLE "ADM"."PAIS"  
  ("ID_PAIS" NUMBER(4,0) NOT NULL ENABLE,  
   "PAIS" VARCHAR2(50) NOT NULL ENABLE,  
   CONSTRAINT "PK_PAIS" PRIMARY KEY ("ID_PAIS")  
  )  
CREATE UNIQUE INDEX "ADM"."PK_PAIS" ON "ADM"."PAIS" ("ID_PAIS")  
ALTER TABLE "ADM"."PAIS" ADD CONSTRAINT "PK_PAIS" PRIMARY KEY ("ID_PAIS")
```

SCRIPT PARA TABLA (ESTADO):

```
CREATE TABLE "ADM"."ESTADO"  
  ("ID_PAIS" NUMBER(4,0) NOT NULL ENABLE,  
   "ID_ESTADO" NUMBER(3,0) NOT NULL ENABLE,  
   "ESTADO" VARCHAR2(50) NOT NULL ENABLE,  
   CONSTRAINT "PK_ESTADO" PRIMARY KEY ("ID_PAIS", "ID_ESTADO")  
   CONSTRAINT "FK01_ESTADO" FOREIGN KEY ("ID_PAIS")  
   REFERENCES "ADM"."PAIS" ("ID_PAIS") ENABLE  
  )  
CREATE UNIQUE INDEX "ADM"."PK_ESTADO" ON "ADM"."ESTADO" ("ID_PAIS",  
"ID_ESTADO")  
ALTER TABLE "ADM"."ESTADO" ADD CONSTRAINT "PK_ESTADO" PRIMARY KEY ("ID_PAIS",  
"ID_ESTADO")
```

SCRIPT PARA TABLA (MUNICIPIO):

```
CREATE TABLE "ADM"."MUNICIPIO"  
  ("ID_PAIS" NUMBER(4,0) NOT NULL ENABLE,  
   "ID_ESTADO" NUMBER(3,0) NOT NULL ENABLE,  
   "ID_MUNICIPIO" NUMBER(5,0) NOT NULL ENABLE,  
   "MUNICIPIO" VARCHAR2(100) NOT NULL ENABLE,  
   CONSTRAINT "PK_MUNICIPIO" PRIMARY KEY ("ID_PAIS", "ID_ESTADO", "ID_MUNICIPIO")  
   CONSTRAINT "FK01_MUNICIPIO" FOREIGN KEY ("ID_PAIS", "ID_ESTADO")  
   REFERENCES "ADM"."ESTADO" ("ID_PAIS", "ID_ESTADO") ENABLE  
  )  
  
CREATE UNIQUE INDEX "ADM"."PK_MUNICIPIO" ON "ADM"."MUNICIPIO" ("ID_PAIS",  
"ID_ESTADO", "ID_MUNICIPIO")  
ALTER TABLE "ADM"."MUNICIPIO" ADD CONSTRAINT "PK_MUNICIPIO" PRIMARY KEY  
("ID_PAIS", "ID_ESTADO", "ID_MUNICIPIO")
```

SCRIPT PARA TABLA (CODIGO_POSTAL):

```
CREATE TABLE "ADM"."CODIGO_POSTAL"
(
  "ID_PAIS" NUMBER(4,0) NOT NULL ENABLE,
  "CODIGO_POSTAL" CHAR(5) NOT NULL ENABLE,
  "ID_USUARIO_CP" NUMBER(2,0) NOT NULL ENABLE,
  "ID_ESTADO" NUMBER(3,0) NOT NULL ENABLE,
  "ID_MUNICIPIO" NUMBER(5,0) NOT NULL ENABLE,
  "USUARIO_CP" VARCHAR2(100) NOT NULL ENABLE,
  "PCT_IVA" NUMBER(8,4) NOT NULL ENABLE,
  "ZONA_TEV" NUMBER(3,0) NOT NULL ENABLE,
  "ZONA_SISMICA" VARCHAR2(5) NOT NULL ENABLE,
  "ZONA_HURACAN" NUMBER(3,0) NOT NULL ENABLE,
  "ZONA_AUTO" NUMBER(5,0) NOT NULL ENABLE,
  CONSTRAINT "PK_CODIGO_POSTAL" PRIMARY KEY ("ID_PAIS", "CODIGO_POSTAL",
"ID_USUARIO_CP")
  CONSTRAINT "FK01_CODIGO_POSTAL" FOREIGN KEY ("ID_PAIS", "ID_ESTADO",
"ID_MUNICIPIO")
  REFERENCES "ADM"."MUNICIPIO" ("ID_PAIS", "ID_ESTADO", "ID_MUNICIPIO") ENABLE
)

CREATE INDEX "ADM"."IDX_PAIS_EDO_MUN" ON "ADM"."CODIGO_POSTAL" ("ID_PAIS",
"ID_ESTADO", "ID_MUNICIPIO")
CREATE UNIQUE INDEX "ADM"."PK_CODIGO_POSTAL" ON "ADM"."CODIGO_POSTAL"
("ID_PAIS", "CODIGO_POSTAL", "ID_USUARIO_CP")
ALTER TABLE "ADM"."CODIGO_POSTAL" ADD CONSTRAINT "PK_CODIGO_POSTAL"
PRIMARY KEY ("ID_PAIS", "CODIGO_POSTAL", "ID_USUARIO_CP")
```

SCRIPT PARA TABLA (EMI_POLIZA):

```
CREATE TABLE "ADM"."EMI_POLIZA"
(
  "ID_UNIDAD_NEGOCIO" NUMBER(8,0) NOT NULL ENABLE,
  "ID_EMITOR" VARCHAR2(8) NOT NULL ENABLE,
  "ID_COTIZACION" NUMBER(8,0) NOT NULL ENABLE,
  "ID_RENOVACION" NUMBER(2,0) NOT NULL ENABLE,
  "ANIO_OPERACION" NUMBER(2,0) NOT NULL ENABLE,
  "ID_CONTRATANTE" VARCHAR2 (16),
  "ID_AGENTE" NUMBER (8,0),
  "F_INICIO_VIG" DATE NOT NULL ENABLE,
  "F_FIN_VIG" DATE NOT NULL ENABLE,
  "ID_PRODUCTO" NUMBER(6,0) NOT NULL ENABLE,
  "ID_VERSION" NUMBER(3,0) NOT NULL ENABLE,
  "B_RENOVABLE" CHAR(1) NOT NULL ENABLE,
  "B_RENOVADA" CHAR(1) NOT NULL ENABLE,
  "B_RECALCULO_ANUAL" CHAR(1) NOT NULL ENABLE,
  "FH_ULT_RECALCULO" DATE,
  "CVE_TIPO_ADMON" VARCHAR2(6) NOT NULL ENABLE,
  "ID_MONEDA" NUMBER(3,0) NOT NULL ENABLE,
  "ID_CONDUCTO_COBRO" NUMBER(5,0) NOT NULL ENABLE,
  "ID_FORMA_PAGO" NUMBER(5,0) NOT NULL ENABLE,
  "CVE_NIVEL_COBRO" VARCHAR2(6),
  "ID_DESGLOSE_IMPORTE" NUMBER(12,0) NOT NULL ENABLE,
  "PCT_DESCTO_POL" NUMBER(8,4),
  "PCT_BON_COMIS" NUMBER(8,4),
```

```

"PCT_RCGO_FPAGO" NUMBER(8,4),
"PCT_IVA" NUMBER (8,4),
"CVE_SIT_POLIZA" VARCHAR2 (6),
"CVE_MTVO_SIT_POLIZA" VARCHAR2 (6),
"NUM_POLIZA" VARCHAR2 (16),
"NUM_POLIZA_ANT" VARCHAR2 (16),
"ID_BITACORA" VARCHAR2(16),
"B_DOM_CORRESP_CTE" CHAR(1) NOT NULL ENABLE,
"ID_DOM_CORRESP" VARCHAR2 (16),
"CVE_ENVIO_RECIBO" VARCHAR2 (6),
"CVE_PER_ENVIO_RECIBO" VARCHAR2 (6),
"FH_CREACION" DATE NOT NULL ENABLE,
"ID_USUARIO_CREACION" NUMBER (8,0),
"FH_EMISION" DATE,
"ID_USUARIO_EMISION" NUMBER (8,0),
"FH_CANCELACION" DATE,
"FH_SOL_AUTORIZACION" DATE,
"FH_SOL_PROCESADA" DATE,
"ID_USUARIO_PROCESO" NUMBER (8,0),
"FH_ULT_MOD" DATE NOT NULL ENABLE,
"TX_REFERENCIA" VARCHAR2(2000),
"B_DERIVACION_AE" CHAR(1) NOT NULL ENABLE,
"FH_DERIVACION_AE" DATE,
"CVE_DERIVACION_SC" VARCHAR2(6),
"FH_DERIVACION_SC" DATE,
"FH_ULT_MOD_AE" DATE,
"CVE_MOD_AE" VARCHAR2(6),
"TX_COND_COB_CLUSTER" LONG,
"COND_COB_NUM_CUENTA" VARCHAR2 (50),
"COND_COB_DATO1" VARCHAR2 (100),
"COND_COB_DATO2" VARCHAR2 (100),
"COND_COB_DATO3" VARCHAR2 (100),
"COND_COB_DATO4" VARCHAR2 (100),
"COND_COB_DATO5" VARCHAR2 (100),
"COND_COB_DATO6" VARCHAR2(100),
"B_ERROR" CHAR(1) NOT NULL ENABLE,
"B_AUTORIZAR" CHAR(1) NOT NULL ENABLE,
"B_PRUEBA_PROD" CHAR (1) NOT NULL ENABLE,
"B_PRUEBA_AGTE" CHAR (1) NOT NULL ENABLE,
"FOLIO_RENOVACION" VARCHAR2 (16),
"B_POLSOSPECHOSA_LD" VARCHAR2 (1),
"B_DOCUMENTACION" VARCHAR2 (1),
CONSTRAINT "PK_EMI_POLIZA" PRIMARY KEY ("ID_UNIDAD_NEGOCIO", "ID_EMITOR",
"ID_COTIZACION", "ID_RENOVACION")
CONSTRAINT "FK_EMI_POLI_REF_34456_AE_UN_EM" FOREIGN KEY
("ID_UNIDAD_NEGOCIO", "ID_EMITOR")
REFERENCES "ADM"."AE_UN_EMITOR" ("ID_UNIDAD_NEGOCIO", "ID_EMITOR") ENABLE,
CONSTRAINT "FK_EMI_POLI_REF_41430_EMI_CONT" FOREIGN KEY ("ID_CONTRATANTE")
REFERENCES "ADM"."EMI_CONTRATANTE" ("ID_CONTRATANTE") ENABLE,
CONSTRAINT "FK_EMI_POLI_REF_5328_UNIDAD_N" FOREIGN KEY ("ID_UNIDAD_NEGOCIO")
REFERENCES "ADM"."UNIDAD_NEGOCIO" ("ID_UNIDAD_NEGOCIO") ENABLE,
CONSTRAINT "FK_EMI_POLI_REF_5349_PRO_PROD" FOREIGN KEY ("ID_UNIDAD_NEGOCIO",
"ID_PRODUCTO", "ID_VERSION")
REFERENCES "ADM"."PRO_PRODUCTO_VERSION" ("ID_UNIDAD_NEGOCIO",
"ID_PRODUCTO", "ID_VERSION") ENABLE,

```



```

CONSTRAINT "FK_EMI_POLI_REF_5365_MONEDA" FOREIGN KEY ("ID_MONEDA")
REFERENCES "ADM"."MONEDA" ("ID_MONEDA") ENABLE,
CONSTRAINT "FK_EMI_POLI_REF_5369_PRO_PROD" FOREIGN KEY ("ID_UNIDAD_NEGOCIO",
"ID_PRODUCTO", "ID_VERSION", "ID_MONEDA", "ID_CONDUCTO_COBRO",
"ID_FORMA_PAGO")
REFERENCES "ADM"."PRO_PROD_MON_COND_COB_FPAGO" ("ID_UNIDAD_NEGOCIO",
"ID_PRODUCTO", "ID_VERSION", "ID_MONEDA", "ID_CONDUCTO_COBRO",
"ID_FORMA_PAGO") ENABLE,
CONSTRAINT "FK_EMI_POLI_REF_5548_AE_AGENTE" FOREIGN KEY ("ID_AGENTE")
REFERENCES "ADM"."AE_AGENTE" ("ID_AGENTE") ENABLE
)

```

```

CREATE INDEX "ADM"."IX01_EMI_POLIZA" ON "ADM"."EMI_POLIZA" ("FH_EMISION")
CREATE INDEX "ADM"."IX02_EMI_POLIZA" ON "ADM"."EMI_POLIZA" ("NUM_POLIZA")
CREATE UNIQUE INDEX "ADM"."PK_EMI_POLIZA" ON "ADM"."EMI_POLIZA"
("ID_UNIDAD_NEGOCIO", "ID_EMISOR", "ID_COTIZACION", "ID_RENOVACION")
ALTER TABLE "ADM"."EMI_POLIZA" ADD CONSTRAINT "PK_EMI_POLIZA" PRIMARY KEY
("ID_UNIDAD_NEGOCIO", "ID_EMISOR", "ID_COTIZACION", "ID_RENOVACION")

```

SCRIPT PARA TABLA (EMI_ASEGURADO) :

```

CREATE TABLE "ADM"."EMI_ASEGURADO"
(
    "ID_UNIDAD_NEGOCIO" NUMBER(8,0) NOT NULL ENABLE,
    "ID_EMISOR" VARCHAR2(8) NOT NULL ENABLE,
    "ID_COTIZACION" NUMBER(8,0) NOT NULL ENABLE,
    "ID_RENOVACION" NUMBER(2,0) NOT NULL ENABLE,
    "ID_PERSONA" VARCHAR2(16) NOT NULL ENABLE,
    "NUM_SECUENCIA" NUMBER(3,0) NOT NULL ENABLE,
    CONSTRAINT "PK_EMI_ASEGURADO" PRIMARY KEY ("ID_UNIDAD_NEGOCIO", "ID_EMISOR",
"ID_COTIZACION", "ID_RENOVACION", "ID_PERSONA")
    CONSTRAINT "FK01_EMI_ASEGURADO" FOREIGN KEY ("ID_UNIDAD_NEGOCIO",
"ID_EMISOR", "ID_COTIZACION", "ID_RENOVACION")
    REFERENCES "ADM"."EMI_POLIZA" ("ID_UNIDAD_NEGOCIO", "ID_EMISOR",
"ID_COTIZACION", "ID_RENOVACION") ON DELETE CASCADE ENABLE,
    CONSTRAINT "FK_EMI_ASEG_REF_41443_PERSONA" FOREIGN KEY ("ID_PERSONA")
    REFERENCES "ADM"."PERSONA" ("ID_PERSONA") ENABLE
)
CREATE UNIQUE INDEX "ADM"."PK_EMI_ASEGURADO" ON "ADM"."EMI_ASEGURADO"
("ID_UNIDAD_NEGOCIO", "ID_EMISOR", "ID_COTIZACION", "ID_RENOVACION",
"ID_PERSONA")
ALTER TABLE "ADM"."EMI_ASEGURADO" ADD CONSTRAINT "PK_EMI_ASEGURADO"
PRIMARY KEY ("ID_UNIDAD_NEGOCIO", "ID_EMISOR", "ID_COTIZACION", "ID_RENOVACION",
"ID_PERSONA")

```

SCRIPT PARA TABLA (EMI_CONTRATANTE):

```
CREATE TABLE "ADM"."EMI_CONTRATANTE"
(
  "ID_CONTRATANTE" VARCHAR2 (16) NOT NULL ENABLE,
  "ID_PERSONA" VARCHAR2 (16) NOT NULL ENABLE,
  "ID_AGRUPADOR" VARCHAR2 (16),
  "FH_CREACION" DATE NOT NULL ENABLE,
  "ID_USUARIO_CREACION" NUMBER(8,0) NOT NULL ENABLE,
  CONSTRAINT "PK_EMI_CONTRATANTE" PRIMARY KEY ("ID_CONTRATANTE")
  CONSTRAINT "FK_EMI_CONT_REF_27293_PERSONA" FOREIGN KEY ("ID_PERSONA")
  REFERENCES "ADM"."PERSONA" ("ID_PERSONA") ENABLE,
  CONSTRAINT "FK_EMI_CONT_REF_30623_EMI_AGRU" FOREIGN KEY ("ID_AGRUPADOR")
  REFERENCES "ADM"."EMI_AGRUPADOR" ("ID_AGRUPADOR") ENABLE
)
CREATE UNIQUE INDEX "ADM"."PK_EMI_CONTRATANTE" ON "ADM"."EMI_CONTRATANTE"
("ID_CONTRATANTE")
ALTER TABLE "ADM"."EMI_CONTRATANTE" ADD CONSTRAINT "PK_EMI_CONTRATANTE"
PRIMARY KEY ("ID_CONTRATANTE")
```

SCRIPT PARA TABLA (EMI_CONTRATANTE):

```
CREATE TABLE "ADM"."EMI_AGENTE"
(
  "ID_UNIDAD_NEGOCIO" NUMBER(8,0) NOT NULL ENABLE,
  "ID_EMITOR" VARCHAR2(8) NOT NULL ENABLE,
  "ID_COTIZACION" NUMBER(8,0) NOT NULL ENABLE,
  "ID_RENOVACION" NUMBER(2,0) NOT NULL ENABLE,
  "ID_AGENTE" NUMBER(8,0) NOT NULL ENABLE,
  "PCT_PARTICIPACION" NUMBER(8,4) NOT NULL ENABLE,
  "IMP_COMIS_PN" NUMBER(16,2),
  "IMP_COMIS_FPAGO" NUMBER(16,2),
  "B_PRINCIPAL" CHAR(1) NOT NULL ENABLE,
  CONSTRAINT "PK_EMI_AGENTE" PRIMARY KEY ("ID_UNIDAD_NEGOCIO", "ID_EMITOR",
  "ID_COTIZACION", "ID_RENOVACION", "ID_AGENTE")
  CONSTRAINT "FK01_EMI_AGENTE" FOREIGN KEY ("ID_UNIDAD_NEGOCIO", "ID_EMITOR",
  "ID_COTIZACION", "ID_RENOVACION")
  REFERENCES "ADM"."EMI_POLIZA" ("ID_UNIDAD_NEGOCIO", "ID_EMITOR",
  "ID_COTIZACION", "ID_RENOVACION") ON DELETE CASCADE ENABLE,
  CONSTRAINT "FK_EMI_AGEN_REF_5571_AE_AGENT" FOREIGN KEY ("ID_AGENTE")
  REFERENCES "ADM"."AE_AGENTE" ("ID_AGENTE") ENABLE
)
CREATE UNIQUE INDEX "ADM"."PK_EMI_AGENTE" ON "ADM"."EMI_AGENTE"
("ID_UNIDAD_NEGOCIO", "ID_EMITOR", "ID_COTIZACION", "ID_RENOVACION", "ID_AGENTE")
ALTER TABLE "ADM"."EMI_AGENTE" ADD CONSTRAINT "PK_EMI_AGENTE" PRIMARY KEY
("ID_UNIDAD_NEGOCIO", "ID_EMITOR", "ID_COTIZACION", "ID_RENOVACION", "ID_AGENTE")
```

SCRIPT PARA TABLA (EMI_AGENTE):

```
CREATE TABLE "ADM"."EMI_AGENTE"
(
  "ID_UNIDAD_NEGOCIO" NUMBER(8,0) NOT NULL ENABLE,
  "ID_EMITOR" VARCHAR2(8) NOT NULL ENABLE,
  "ID_COTIZACION" NUMBER(8,0) NOT NULL ENABLE,
  "ID_RENOVACION" NUMBER(2,0) NOT NULL ENABLE,
  "ID_AGENTE" NUMBER(8,0) NOT NULL ENABLE,
  "PCT_PARTICIPACION" NUMBER(8,4) NOT NULL ENABLE,
  "IMP_COMIS_PN" NUMBER(16,2),
  "IMP_COMIS_FPAGO" NUMBER(16,2),
  "B_PRINCIPAL" CHAR(1) NOT NULL ENABLE,
  CONSTRAINT "PK_EMI_AGENTE" PRIMARY KEY ("ID_UNIDAD_NEGOCIO", "ID_EMITOR",
  "ID_COTIZACION", "ID_RENOVACION", "ID_AGENTE")
  CONSTRAINT "FK01_EMI_AGENTE" FOREIGN KEY ("ID_UNIDAD_NEGOCIO", "ID_EMITOR",
  "ID_COTIZACION", "ID_RENOVACION")
  REFERENCES "ADM"."EMI_POLIZA" ("ID_UNIDAD_NEGOCIO", "ID_EMITOR",
  "ID_COTIZACION", "ID_RENOVACION") ON DELETE CASCADE ENABLE,
  CONSTRAINT "FK_EMI_AGEN_REF_5571_AE_AGENT" FOREIGN KEY ("ID_AGENTE")
  REFERENCES "ADM"."AE_AGENTE" ("ID_AGENTE") ENABLE
)
CREATE UNIQUE INDEX "ADM"."PK_EMI_AGENTE" ON "ADM"."EMI_AGENTE"
("ID_UNIDAD_NEGOCIO", "ID_EMITOR", "ID_COTIZACION", "ID_RENOVACION", "ID_AGENTE")
ALTER TABLE "ADM"."EMI_AGENTE" ADD CONSTRAINT "PK_EMI_AGENTE" PRIMARY KEY
("ID_UNIDAD_NEGOCIO", "ID_EMITOR", "ID_COTIZACION", "ID_RENOVACION", "ID_AGENTE")
```

SCRIPT PARA TABLA (EMI_RECIBO):

```
CREATE TABLE "ADM"."EMI_RECIBO"
(
  "NUM_RECIBO" VARCHAR2(18) NOT NULL ENABLE,
  "ID_UNIDAD_NEGOCIO" NUMBER(8,0) NOT NULL ENABLE,
  "ID_EMITOR" VARCHAR2(8) NOT NULL ENABLE,
  "ID_COTIZACION" NUMBER(8,0) NOT NULL ENABLE,
  "ID_RENOVACION" NUMBER(2,0) NOT NULL ENABLE,
  "ANIO_OPERACION" NUMBER(2,0) NOT NULL ENABLE,
  "ID_RECIBO" NUMBER(10,0) NOT NULL ENABLE,
  "F_INICIO_COBERTURA" DATE NOT NULL ENABLE,
  "F_FIN_COBERTURA" DATE NOT NULL ENABLE,
  "F_LIM_PAGO" DATE NOT NULL ENABLE,
  "F_CONTABLE" DATE,
  "ID_MONEDA" NUMBER(3,0) NOT NULL ENABLE,
  "SERIE" VARCHAR2(5),
  "FOLIO" NUMBER(10,0),
  "CVE_SIT_RECIBO" VARCHAR2(6),
  "NUM_RECIBO_SC" VARCHAR2(18),
  "IMP_COMIS_BON_PN" NUMBER(16,2),
  "IMP_PN_BON_SAI" NUMBER(16,2),
  "IMP_PN_BON_NAI" NUMBER(16,2),
  "IMP_DEREMIS_TOT_SAI" NUMBER(16,2),
  "IMP_DEREMIS_TOT_NAI" NUMBER(16,2),
  "IMP_RCGO_FPAGO_SAI" NUMBER(16,2),
  "IMP_RCGO_FPAGO_NAI" NUMBER(16,2),
  "IMP_COMIS_BON_FPAGO" NUMBER(16,2),
```

```

"IMP_COMIS_BON_TOT" NUMBER(16,2),
"IMP_IVA" NUMBER(16,2),
"IMP_OTROS_IMPTOS_SAI" NUMBER(16,2),
"IMP_OTROS_IMPTOS_NAI" NUMBER(16,2),
"IMP_PT_SAI" NUMBER(16,2),
"IMP_PT_NAI" NUMBER(16,2),
"FH_ULT_MOD" DATE NOT NULL ENABLE,
CONSTRAINT "PK_EMI_RECIBO" PRIMARY KEY ("NUM_RECIBO")
CONSTRAINT "AK_AK01_EMI_RECIBO_EMI_RECIBO" UNIQUE ("ID_UNIDAD_NEGOCIO",
"ID_EMITOR", "ID_COTIZACION", "ID_RENOVACION", "ID_RECIBO")
CONSTRAINT "FK01_EMI_RECIBO" FOREIGN KEY ("ID_UNIDAD_NEGOCIO",
"ID_EMITOR", "ID_COTIZACION", "ID_RENOVACION")
REFERENCES "ADM"."EMI_POLIZA" ("ID_UNIDAD_NEGOCIO", "ID_EMITOR",
"ID_COTIZACION", "ID_RENOVACION") ON DELETE CASCADE ENABLE
)
CREATE UNIQUE INDEX "ADM"."AK_AK01_EMI_RECIBO_EMI_RECIBO" ON
"ADM"."EMI_RECIBO" ("ID_UNIDAD_NEGOCIO", "ID_EMITOR", "ID_COTIZACION",
"ID_RENOVACION", "ID_RECIBO")
CREATE UNIQUE INDEX "ADM"."PK_EMI_RECIBO" ON "ADM"."EMI_RECIBO"
("NUM_RECIBO")
ALTER TABLE "ADM"."EMI_RECIBO" ADD CONSTRAINT "PK_EMI_RECIBO" PRIMARY KEY
("NUM_RECIBO")
ALTER TABLE "ADM"."EMI_RECIBO" ADD CONSTRAINT "AK_AK01_EMI_RECIBO_EMI_RECIBO"
UNIQUE ("ID_UNIDAD_NEGOCIO", "ID_EMITOR", "ID_COTIZACION", "ID_RENOVACION",
"ID_RECIBO")

```

SCRIPT PARA TABLA (EMI_RECIBO_DESGLOSADO):

```

CREATE TABLE "ADM"."EMI_RECIBO_DESGLOSADO"
(
"NUM_RECIBO" VARCHAR2(18) NOT NULL ENABLE,
"ID_LINEA_NEGOCIO" NUMBER(3,0) NOT NULL ENABLE,
"ID_INCISO" NUMBER(5,0) NOT NULL ENABLE,
"ID_COBERTURA" NUMBER(8,0) NOT NULL ENABLE,
"ID_UNIDAD_NEGOCIO" NUMBER(8,0) NOT NULL ENABLE,
"ID_EMITOR" VARCHAR2(8) NOT NULL ENABLE,
"ID_COTIZACION" NUMBER(8,0) NOT NULL ENABLE,
"ID_RENOVACION" NUMBER(2,0),
"ID_RECIBO" NUMBER(10,0),
"IMP_COMIS_BON_PN" NUMBER(16,2),
"IMP_PN_BON_SAI" NUMBER(16,2),
"IMP_PN_BON_NAI" NUMBER(16,2),
"IMP_DEREMIS_TOT_SAI" NUMBER(16,2),
"IMP_DEREMIS_TOT_NAI" NUMBER(16,2),
"IMP_RCGO_FPAGO_SAI" NUMBER(16,2),
"IMP_RCGO_FPAGO_NAI" NUMBER(16,2),
"IMP_COMIS_BON_FPAGO" NUMBER(16,2),
"IMP_COMIS_BON_TOT" NUMBER(16,2),
"IMP_IVA" NUMBER(16,2),
"IMP_OTROS_IMPTOS_SAI" NUMBER(16,2),
"IMP_OTROS_IMPTOS_NAI" NUMBER(16,2),
"IMP_PT_SAI" NUMBER(16,2),
"IMP_PT_NAI" NUMBER(16,2),

```

```

CONSTRAINT "PK_EMI_RECIBO_DESGLOSADO" PRIMARY KEY ("NUM_RECIBO",
"ID_LINEA_NEGOCIO", "ID_INCISO", "ID_COBERTURA")
CONSTRAINT "AK01_EMI_RECIBO_DESGLOSADO" UNIQUE ("ID_UNIDAD_NEGOCIO",
"ID_EMITOR", "ID_COTIZACION", "ID_RENOVACION", "ID_RECIBO", "ID_LINEA_NEGOCIO",
"ID_COBERTURA")
CONSTRAINT "FK01_EMI_RECIBO_DESGLOSADO" FOREIGN KEY ("NUM_RECIBO")
REFERENCES "ADM"."EMI_RECIBO" ("NUM_RECIBO") ON DELETE CASCADE ENABLE,
CONSTRAINT "FK02_EMI_RECIBO_DESGLOSADO" FOREIGN KEY ("ID_UNIDAD_NEGOCIO",
"ID_EMITOR", "ID_COTIZACION", "ID_RENOVACION", "ID_LINEA_NEGOCIO", "ID_INCISO",
"ID_COBERTURA")
REFERENCES "ADM"."EMI_COBERTURA" ("ID_UNIDAD_NEGOCIO", "ID_EMITOR",
"ID_COTIZACION", "ID_RENOVACION", "ID_LINEA_NEGOCIO", "ID_INCISO",
"ID_COBERTURA") ON DELETE CASCADE ENABLE
)

```

```

CREATE UNIQUE INDEX "ADM"."AK01_EMI_RECIBO_DESGLOSADO" ON
"ADM"."EMI_RECIBO_DESGLOSADO" ("ID_UNIDAD_NEGOCIO", "ID_EMITOR",
"ID_COTIZACION", "ID_RENOVACION", "ID_RECIBO", "ID_LINEA_NEGOCIO",
"ID_COBERTURA")
CREATE UNIQUE INDEX "ADM"."PK_EMI_RECIBO_DESGLOSADO" ON
"ADM"."EMI_RECIBO_DESGLOSADO" ("NUM_RECIBO", "ID_LINEA_NEGOCIO", "ID_INCISO",
"ID_COBERTURA")
ALTER TABLE "ADM"."EMI_RECIBO_DESGLOSADO" ADD CONSTRAINT
"PK_EMI_RECIBO_DESGLOSADO" PRIMARY KEY ("NUM_RECIBO", "ID_LINEA_NEGOCIO",
"ID_INCISO", "ID_COBERTURA")
ALTER TABLE "ADM"."EMI_RECIBO_DESGLOSADO" ADD CONSTRAINT
"AK01_EMI_RECIBO_DESGLOSADO" UNIQUE ("ID_UNIDAD_NEGOCIO", "ID_EMITOR",
"ID_COTIZACION", "ID_RENOVACION", "ID_RECIBO", "ID_LINEA_NEGOCIO",
"ID_COBERTURA")

```

SCRIPT PARA TABLA (EMI_COBERTURA):

```

CREATE TABLE "ADM"."EMI_COBERTURA"
(
    "ID_UNIDAD_NEGOCIO" NUMBER(8,0) NOT NULL ENABLE,
    "ID_EMITOR" VARCHAR2(8) NOT NULL ENABLE,
    "ID_COTIZACION" NUMBER(8,0) NOT NULL ENABLE,
    "ID_RENOVACION" NUMBER(2,0) NOT NULL ENABLE,
    "ID_LINEA_NEGOCIO" NUMBER(3,0) NOT NULL ENABLE,
    "ID_INCISO" NUMBER(5,0) NOT NULL ENABLE,
    "ID_COBERTURA" NUMBER(8,0) NOT NULL ENABLE,
    "CVE_SIT_COBERTURA" VARCHAR2(6) NOT NULL ENABLE,
    "CVE_MTVO_SIT_COBERTURA" VARCHAR2(6) NOT NULL ENABLE,
    "F_INICIO_VIG" DATE NOT NULL ENABLE,
    "F_FIN_VIGENCIA" DATE NOT NULL ENABLE,
    "IMP_SUMAASEG_LIMITE" NUMBER(16,2),
    "IMP_SUMAASEG_ELEMENTO" NUMBER(16,2),
    "PCT_DESCTO_COB" NUMBER(8,4),
    "IMP_DEDUCIBLE_1SIN" NUMBER(16,2),
    "IMP_DEDUCIBLE_2SIN" NUMBER(16,2),
    "IMP_DEDUCIBLE_3SIN" NUMBER(16,2),
    "IMP_DEDUCIBLE_4SIN" NUMBER(16,2),
    "IMP_DEDUCIBLE_MIN" NUMBER(16,2),
    "IMP_DEDUCIBLE_MAX" NUMBER(16,2),

```

```

"IMP_COASEGURO" NUMBER(16,2),
"ID_DESGLOSE_IMPORTE" NUMBER(12,0),
"TX_CLUSTER" LONG,
"TX_REFERENCIA" VARCHAR2 (2000),
"B_ERROR" CHAR (1) NOT NULL ENABLE,
CONSTRAINT "PK_EMI_COBERTURA" PRIMARY KEY ("ID_UNIDAD_NEGOCIO", "ID_EMITOR",
"ID_COTIZACION", "ID_RENOVACION", "ID_LINEA_NEGOCIO", "ID_INCISO",
"ID_COBERTURA")
CONSTRAINT "FK01_EMI_COBERTURA" FOREIGN KEY ("ID_UNIDAD_NEGOCIO",
"ID_EMITOR", "ID_COTIZACION", "ID_RENOVACION", "ID_LINEA_NEGOCIO", "ID_INCISO")
REFERENCES "ADM"."EMI_INCISO" ("ID_UNIDAD_NEGOCIO", "ID_EMITOR",
"ID_COTIZACION", "ID_RENOVACION", "ID_LINEA_NEGOCIO", "ID_INCISO") ON DELETE
CASCADE ENABLE,
CONSTRAINT "FK02_EMI_COBERTURA" FOREIGN KEY ("ID_UNIDAD_NEGOCIO",
"ID_EMITOR", "ID_COTIZACION", "ID_RENOVACION", "ID_DESGLOSE_IMPORTE")
REFERENCES "ADM"."EMI_DESGLOSEIMPORTE" ("ID_UNIDAD_NEGOCIO", "ID_EMITOR",
"ID_COTIZACION", "ID_RENOVACION", "ID_DESGLOSE_IMPORTE") ON DELETE CASCADE
ENABLE
)
CREATE UNIQUE INDEX "ADM"."PK_EMI_COBERTURA" ON "ADM"."EMI_COBERTURA"
("ID_UNIDAD_NEGOCIO", "ID_EMITOR", "ID_COTIZACION", "ID_RENOVACION",
"ID_LINEA_NEGOCIO", "ID_INCISO", "ID_COBERTURA")
ALTER TABLE "ADM"."EMI_COBERTURA" ADD CONSTRAINT "PK_EMI_COBERTURA"
PRIMARY KEY ("ID_UNIDAD_NEGOCIO", "ID_EMITOR", "ID_COTIZACION", "ID_RENOVACION",
"ID_LINEA_NEGOCIO", "ID_INCISO", "ID_COBERTURA")

```

SCRIPT PARA TABLA (EMI_COBERTURA_DATO):

```

CREATE TABLE "ADM"."EMI_COBERTURA_DATO"
(
"ID_UNIDAD_NEGOCIO" NUMBER(8,0) NOT NULL ENABLE,
"ID_EMITOR" VARCHAR2(8) NOT NULL ENABLE,
"ID_COTIZACION" NUMBER(8,0) NOT NULL ENABLE,
"ID_RENOVACION" NUMBER(2,0) NOT NULL ENABLE,
"ID_LINEA_NEGOCIO" NUMBER(3,0) NOT NULL ENABLE,
"ID_INCISO" NUMBER(5,0) NOT NULL ENABLE,
"ID_COBERTURA" NUMBER(8,0) NOT NULL ENABLE,
"DATO" VARCHAR2(30) NOT NULL ENABLE,
"VALOR" VARCHAR2 (200),
CONSTRAINT "PK_EMI_COBERTURA_DATO" PRIMARY KEY ("ID_UNIDAD_NEGOCIO",
"ID_EMITOR", "ID_COTIZACION", "ID_RENOVACION", "ID_LINEA_NEGOCIO", "ID_INCISO",
"ID_COBERTURA", "DATO")
CONSTRAINT "FK01_EMI_COBERTURA_DATO" FOREIGN KEY ("ID_UNIDAD_NEGOCIO",
"ID_EMITOR", "ID_COTIZACION", "ID_RENOVACION", "ID_LINEA_NEGOCIO", "ID_INCISO",
"ID_COBERTURA")
REFERENCES "ADM"."EMI_COBERTURA" ("ID_UNIDAD_NEGOCIO", "ID_EMITOR",
"ID_COTIZACION", "ID_RENOVACION", "ID_LINEA_NEGOCIO", "ID_INCISO",
"ID_COBERTURA") ON DELETE CASCADE ENABLE
)
CREATE UNIQUE INDEX "ADM"."PK_EMI_COBERTURA_DATO" ON
"ADM"."EMI_COBERTURA_DATO" ("ID_UNIDAD_NEGOCIO", "ID_EMITOR", "ID_COTIZACION",
"ID_RENOVACION", "ID_LINEA_NEGOCIO", "ID_INCISO", "ID_COBERTURA", "DATO")
ALTER TABLE "ADM"."EMI_COBERTURA_DATO" ADD CONSTRAINT
"PK_EMI_COBERTURA_DATO" PRIMARY KEY ("ID_UNIDAD_NEGOCIO", "ID_EMITOR",

```

"ID_COTIZACION", "ID_RENOVACION", "ID_LINEA_NEGOCIO", "ID_INCISO",
"ID_COBERTURA", "DATO")

SCRIPT PARA TABLA (EMI_DESGLOSEIMPORTE):

```
CREATE TABLE "ADM"."EMI_DESGLOSEIMPORTE"  
(  
    "ID_UNIDAD_NEGOCIO" NUMBER(8,0) NOT NULL ENABLE,  
    "ID_EMITOR" VARCHAR2(8) NOT NULL ENABLE,  
    "ID_COTIZACION" NUMBER(8,0) NOT NULL ENABLE,  
    "ID_RENOVACION" NUMBER(2,0) NOT NULL ENABLE,  
    "ID_DESGLOSE_IMPORTE" NUMBER(12,0) NOT NULL ENABLE,  
    "IMP_PN_REAL_SAI" NUMBER(16,2),  
    "IMP_PN_REA_NAI" NUMBER(16,2),  
    "IMP_DESCTO_POL_SAI" NUMBER(16,2),  
    "IMP_DESCTO_POL_NAI" NUMBER(16,2),  
    "IMP_DESCTO_VOL_SAI" NUMBER(16,2),  
    "IMP_DESCTO_VOL_NAI" NUMBER(16,2),  
    "IMP_DESCTO_B SIN_SAI" NUMBER(16,2),  
    "IMP_DESCTO_B SIN_NAI" NUMBER(16,2),  
    "IMP_DESCTO_LIN_SAI" NUMBER(16,2),  
    "IMP_DESCTO_LIN_NAI" NUMBER(16,2),  
    "IMP_DESCTO_COB_SAI" NUMBER(16,2),  
    "IMP_DESCTO_COB_NAI" NUMBER(16,2),  
    "IMP_DESCTO_OTROS_SAI" NUMBER(16,2),  
    "IMP_DESCTO_OTROS_NAI" NUMBER(16,2),  
    "IMP_DESCTO_TOT_SAI" NUMBER(16,2),  
    "IMP_DESCTO_TOT_NAI" NUMBER(16,2),  
    "IMP_PN_DESC_SAI" NUMBER(16,2),  
    "IMP_PN_DESC_NAI" NUMBER(16,2),  
    "IMP_COMIS_PN" NUMBER(16,2),  
    "IMP_BON_COMIS_PN_SAI" NUMBER(16,2),  
    "IMP_BON_COMIS_PN_NAI" NUMBER(16,2),  
    "IMP_COMIS_BON_PN" NUMBER(16,2),  
    "IMP_PN_BON_SAI" NUMBER(16,2),  
    "IMP_PN_BON_NAI" NUMBER(16,2),  
    "IMP_DEREMIS_PROD_SAI" NUMBER(16,2),  
    "IMP_DEREMIS_PROD_NAI" NUMBER(16,2),  
    "IMP_DEREMIS_LIN_SAI" NUMBER(16,2),  
    "IMP_DEREMIS_LIN_NAI" NUMBER(16,2),  
    "IMP_DEREMIS_TOT_SAI" NUMBER(16,2),  
    "IMP_DEREMIS_TOT_NAI" NUMBER(16,2),  
    "IMP_RCGO_FPAGO_SAI" NUMBER(16,2),  
    "IMP_RCGO_FPAGO_NAI" NUMBER(16,2),  
    "IMP_COMIS_FPAGO" NUMBER(16,2),  
    "IMP_COMIS_TOT" NUMBER(16,2),  
    "IMP_BON_COMIS_FPAGO_SAI" NUMBER(16,2),  
    "IMP_BON_COMIS_FPAGO_NAI" NUMBER(16,2),  
    "IMP_BON_COMIS_TOT_SAI" NUMBER(16,2),  
    "IMP_BON_COMIS_TOT_NAI" NUMBER(16,2),  
    "IMP_COMIS_BON_FPAGO" NUMBER(16,2),  
    "IMP_COMIS_BON_TOT" NUMBER(16,2),  
    "IMP_STOT_PN_SAI" NUMBER(16,2),  
    "IMP_STOT_PN_NAI" NUMBER(16,2),
```

```

"IMP_IVA" NUMBER(16,2),
"IMP_OTROS_IMPTOS_SAI" NUMBER(16,2),
"IMP_OTROS_IMPTOS_NAI" NUMBER(16,2),
"IMP_PT_SAI" NUMBER(16,2),
"IMP_PT_NAI" NUMBER(16,2),
CONSTRAINT "PK_EMI_DESGLOSEIMPORTE" PRIMARY KEY ("ID_UNIDAD_NEGOCIO",
"ID_EMITOR", "ID_COTIZACION", "ID_RENOVACION", "ID_DESGLOSE_IMPORTE")
CONSTRAINT "FK01_EMI_DESGLOSEIMPORTE" FOREIGN KEY ("ID_UNIDAD_NEGOCIO",
"ID_EMITOR", "ID_COTIZACION", "ID_RENOVACION")
REFERENCES "ADM"."EMI_POLIZA" ("ID_UNIDAD_NEGOCIO", "ID_EMITOR",
"ID_COTIZACION", "ID_RENOVACION") ON DELETE CASCADE ENABLE
)
CREATE UNIQUE INDEX "ADM"."PK_EMI_DESGLOSEIMPORTE" ON
"ADM"."EMI_DESGLOSEIMPORTE" ("ID_UNIDAD_NEGOCIO", "ID_EMITOR", "ID_COTIZACION",
"ID_RENOVACION", "ID_DESGLOSE_IMPORTE")
ALTER TABLE "ADM"."EMI_DESGLOSEIMPORTE" ADD CONSTRAINT
"PK_EMI_DESGLOSEIMPORTE" PRIMARY KEY ("ID_UNIDAD_NEGOCIO", "ID_EMITOR",
"ID_COTIZACION", "ID_RENOVACION", "ID_DESGLOSE_IMPORTE")

```

SCRIPT PARA TABLA (EMI_INCISO):

```

CREATE TABLE "ADM"."EMI_INCISO"
(
  "ID_UNIDAD_NEGOCIO" NUMBER(8,0) NOT NULL ENABLE,
  "ID_EMITOR" VARCHAR2(8) NOT NULL ENABLE,
  "ID_COTIZACION" NUMBER(8,0) NOT NULL ENABLE,
  "ID_RENOVACION" NUMBER(2,0) NOT NULL ENABLE,
  "ID_LINEA_NEGOCIO" NUMBER(3,0) NOT NULL ENABLE,
  "ID_INCISO" NUMBER(5,0) NOT NULL ENABLE,
  "ID_DESGLOSE_IMPORTE" NUMBER(12,0),
  "ID_INCISO_POLIZA_ANT" NUMBER(5,0),
  "CVE_TIPO_UR" VARCHAR2(6),
  "ID_PAQUETE" NUMBER(3,0),
  "F_INICIO_VIG" DATE NOT NULL ENABLE,
  "F_FIN_VIG" DATE NOT NULL ENABLE,
  "NUM_UR" NUMBER(5,0),
  "PCT_DESCTO_B SIN" NUMBER(8,4),
  "PCT_DESCTO_OTROS" NUMBER(8,4),
  "CVE_SIT_INCISO" VARCHAR2(6) NOT NULL ENABLE,
  "CVE_MTVO_SIT_INCISO" VARCHAR2(6) NOT NULL ENABLE,
  "TX_REFERENCIA" VARCHAR2(2000),
  "B_ERROR" CHAR(1) NOT NULL ENABLE,
  "B_AUTORIZAR" CHAR(1) NOT NULL ENABLE,
  "B_OBLIGA_COT" CHAR(1) NOT NULL ENABLE,
  "B_OBLIGA_POL" CHAR(1) NOT NULL ENABLE,
  CONSTRAINT "PK_EMI_INCISO" PRIMARY KEY ("ID_UNIDAD_NEGOCIO", "ID_EMITOR",
"ID_COTIZACION", "ID_RENOVACION", "ID_LINEA_NEGOCIO", "ID_INCISO")
  CONSTRAINT "FK01_EMI_INCISO" FOREIGN KEY ("ID_UNIDAD_NEGOCIO", "ID_EMITOR",
"ID_COTIZACION", "ID_RENOVACION", "ID_LINEA_NEGOCIO")
REFERENCES "ADM"."EMI_LINEA" ("ID_UNIDAD_NEGOCIO", "ID_EMITOR",
"ID_COTIZACION", "ID_RENOVACION", "ID_LINEA_NEGOCIO") ON DELETE CASCADE
ENABLE,
  CONSTRAINT "FK02_EMI_INCISO" FOREIGN KEY ("ID_UNIDAD_NEGOCIO", "ID_EMITOR",
"ID_COTIZACION", "ID_RENOVACION", "ID_DESGLOSE_IMPORTE")

```



```

REFERENCES "ADM"."EMI_DESGLOSEIMPORTE" ("ID_UNIDAD_NEGOCIO", "ID_EMISOR",
"ID_COTIZACION", "ID_RENOVACION", "ID_DESGLOSE_IMPORTE") ON DELETE CASCADE
ENABLE
)
CREATE UNIQUE INDEX "ADM"."PK_EMI_INCISO" ON "ADM"."EMI_INCISO"
("ID_UNIDAD_NEGOCIO", "ID_EMISOR", "ID_COTIZACION", "ID_RENOVACION",
"ID_LINEA_NEGOCIO", "ID_INCISO")
ALTER TABLE "ADM"."EMI_INCISO" ADD CONSTRAINT "PK_EMI_INCISO" PRIMARY KEY
("ID_UNIDAD_NEGOCIO", "ID_EMISOR", "ID_COTIZACION", "ID_RENOVACION",
"ID_LINEA_NEGOCIO", "ID_INCISO")

```

SCRIPT PARA TABLA (EMI_LINEA):

```

CREATE TABLE "ADM"."EMI_LINEA"
(
    "ID_UNIDAD_NEGOCIO" NUMBER(8,0) NOT NULL ENABLE,
    "ID_EMISOR" VARCHAR2(8) NOT NULL ENABLE,
    "ID_COTIZACION" NUMBER(8,0) NOT NULL ENABLE,
    "ID_RENOVACION" NUMBER(2,0) NOT NULL ENABLE,
    "ID_LINEA_NEGOCIO" NUMBER(3,0) NOT NULL ENABLE,
    "NUM_UR" NUMBER(5,0),
    "ID_DESGLOSE_IMPORTE" NUMBER(12,0),
    "PCT_DESCTO_VOL" NUMBER(8,4),
    "PCT_DESCTO_LIN" NUMBER(8,4),
    CONSTRAINT "PK_EMI_LINEA" PRIMARY KEY ("ID_UNIDAD_NEGOCIO", "ID_EMISOR",
"ID_COTIZACION", "ID_RENOVACION", "ID_LINEA_NEGOCIO")
    CONSTRAINT "FK01_EMI_LINEA" FOREIGN KEY ("ID_UNIDAD_NEGOCIO", "ID_EMISOR",
"ID_COTIZACION", "ID_RENOVACION")
REFERENCES "ADM"."EMI_POLIZA" ("ID_UNIDAD_NEGOCIO", "ID_EMISOR",
"ID_COTIZACION", "ID_RENOVACION") ON DELETE CASCADE ENABLE,
    CONSTRAINT "FK02_EMI_LINEA" FOREIGN KEY ("ID_UNIDAD_NEGOCIO", "ID_EMISOR",
"ID_COTIZACION", "ID_RENOVACION", "ID_DESGLOSE_IMPORTE")
REFERENCES "ADM"."EMI_DESGLOSEIMPORTE" ("ID_UNIDAD_NEGOCIO", "ID_EMISOR",
"ID_COTIZACION", "ID_RENOVACION", "ID_DESGLOSE_IMPORTE") ON DELETE CASCADE
ENABLE
)
CREATE UNIQUE INDEX "ADM"."PK_EMI_LINEA" ON "ADM"."EMI_LINEA"
("ID_UNIDAD_NEGOCIO", "ID_EMISOR", "ID_COTIZACION", "ID_RENOVACION",
"ID_LINEA_NEGOCIO")
ALTER TABLE "ADM"."EMI_LINEA" ADD CONSTRAINT "PK_EMI_LINEA" PRIMARY KEY
("ID_UNIDAD_NEGOCIO", "ID_EMISOR", "ID_COTIZACION", "ID_RENOVACION",
"ID_LINEA_NEGOCIO")

```

SCRIPT PARA TABLA (SEG_USUARIO):

```

CREATE TABLE "ADM"."SEG_USUARIO"
(
    "ID_USUARIO" NUMBER(8,0) NOT NULL ENABLE,
    "ID_PERSONA" VARCHAR2(16) NOT NULL ENABLE,
    "NOMBRE_USUARIO" VARCHAR2(15) NOT NULL ENABLE,
    "CONTRASENIA" VARCHAR2(15) NOT NULL ENABLE,
    "CVE_SIT_USUARIO" VARCHAR2(6) NOT NULL ENABLE,
    CONSTRAINT "PK_SEG_USUARIO" PRIMARY KEY ("ID_USUARIO")
    CONSTRAINT "FK01_SEG_USUARIO" FOREIGN KEY ("ID_PERSONA")

```

```
REFERENCES "ADM"."PERSONA" ("ID_PERSONA") ENABLE
)
CREATE UNIQUE INDEX "ADM"."PK_SEG_USUARIO" ON "ADM"."SEG_USUARIO"
("ID_USUARIO")
ALTER TABLE "ADM"."SEG_USUARIO" ADD CONSTRAINT "PK_SEG_USUARIO" PRIMARY
KEY ("ID_USUARIO")
```

SCRIPT PARA TABLA (SEG_SIST_UN_USUARIO):

```
CREATE TABLE "ADM"."SEG_SIST_UN_USUARIO"
(
  "ID_SISTEMA" NUMBER(8,0) NOT NULL ENABLE,
  "ID_UNIDAD_NEGOCIO" NUMBER(8,0) NOT NULL ENABLE,
  "ID_USUARIO" NUMBER(8,0) NOT NULL ENABLE,
  CONSTRAINT "PK_SEG_SIST_UN_USUARIO" PRIMARY KEY ("ID_SISTEMA",
"ID_UNIDAD_NEGOCIO", "ID_USUARIO")
  CONSTRAINT "FK01_SEG_SIST_UN_USUARIO" FOREIGN KEY ("ID_SISTEMA",
"ID_UNIDAD_NEGOCIO")
  REFERENCES "ADM"."SEG_SIST_UN" ("ID_SISTEMA", "ID_UNIDAD_NEGOCIO") ENABLE,
  CONSTRAINT "FK02_SEG_SIST_UN_USUARIO" FOREIGN KEY ("ID_USUARIO")
  REFERENCES "ADM"."SEG_USUARIO" ("ID_USUARIO") ENABLE
)
CREATE UNIQUE INDEX "ADM"."PK_SEG_SIST_UN_USUARIO" ON
"ADM"."SEG_SIST_UN_USUARIO" ("ID_SISTEMA", "ID_UNIDAD_NEGOCIO", "ID_USUARIO")
ALTER TABLE "ADM"."SEG_SIST_UN_USUARIO" ADD CONSTRAINT
"PK_SEG_SIST_UN_USUARIO" PRIMARY KEY ("ID_SISTEMA", "ID_UNIDAD_NEGOCIO",
"ID_USUARIO")
```

SCRIPT PARA TABLA (SEG_SIST_UN):

```
CREATE TABLE "ADM"."SEG_SIST_UN"
(
  "ID_SISTEMA" NUMBER(8,0) NOT NULL ENABLE,
  "ID_UNIDAD_NEGOCIO" NUMBER(8,0) NOT NULL ENABLE,
  CONSTRAINT "PK_SEG_SIST_UN" PRIMARY KEY ("ID_SISTEMA", "ID_UNIDAD_NEGOCIO")
  CONSTRAINT "FK01_SEG_SISTEMA_UN" FOREIGN KEY ("ID_UNIDAD_NEGOCIO")
  REFERENCES "ADM"."UNIDAD_NEGOCIO" ("ID_UNIDAD_NEGOCIO") ENABLE,
  CONSTRAINT "FK02_SEG_SISTEMA_UN" FOREIGN KEY ("ID_SISTEMA")
  REFERENCES "ADM"."SEG_SISTEMA" ("ID_SISTEMA") ENABLE
)
CREATE UNIQUE INDEX "ADM"."PK_SEG_SIST_UN" ON "ADM"."SEG_SIST_UN"
("ID_SISTEMA", "ID_UNIDAD_NEGOCIO")
ALTER TABLE "ADM"."SEG_SIST_UN" ADD CONSTRAINT "PK_SEG_SIST_UN" PRIMARY KEY
("ID_SISTEMA", "ID_UNIDAD_NEGOCIO")
```

SCRIPT PARA TABLA (SEG_SISTEMA):

```
CREATE TABLE "ADM"."SEG_SISTEMA"
(
  "ID_SISTEMA" NUMBER(8,0) NOT NULL ENABLE,
  "SISTEMA" VARCHAR2(30) NOT NULL ENABLE,
  "VERSION" VARCHAR2(5) NOT NULL ENABLE,
```

```
"NOMBRE_EXE" VARCHAR2 (50) NOT NULL ENABLE,
"TX_DERECHOS" VARCHAR2 (200),
"TX_IMAGEN" LONG RAW,
"TX_REFERENCIA" VARCHAR2(200),
CONSTRAINT "PK_SEG_SISTEMA" PRIMARY KEY ("ID_SISTEMA")
)
CREATE UNIQUE INDEX "ADM"."PK_SEG_SISTEMA" ON "ADM"."SEG_SISTEMA"
("ID_SISTEMA")
ALTER TABLE "ADM"."SEG_SISTEMA" ADD CONSTRAINT "PK_SEG_SISTEMA" PRIMARY KEY
("ID_SISTEMA")
```

SCRIPT PARA TABLA (SEG_SISTEMA_OBJETO):

```
CREATE TABLE "ADM"."SEG_SISTEMA_OBJETO"
(
  "ID_SISTEMA" NUMBER(8,0) NOT NULL ENABLE,
  "ID_OBJETO" NUMBER(6,0) NOT NULL ENABLE,
  CONSTRAINT "PK_SEG_SISTEMA_OBJETO" PRIMARY KEY ("ID_SISTEMA", "ID_OBJETO")
  CONSTRAINT "FK01_SEG_SISTEMA_OBJETO" FOREIGN KEY ("ID_SISTEMA")
  REFERENCES "ADM"."SEG_SISTEMA" ("ID_SISTEMA") ENABLE,
  CONSTRAINT "FK02_SEG_SISTEMA_OBJETO" FOREIGN KEY ("ID_OBJETO")
  REFERENCES "ADM"."SEG_OBJETO" ("ID_OBJETO") ENABLE
)
CREATE UNIQUE INDEX "ADM"."PK_SEG_SISTEMA_OBJETO" ON
"ADM"."SEG_SISTEMA_OBJETO" ("ID_SISTEMA", "ID_OBJETO")
ALTER TABLE "ADM"."SEG_SISTEMA_OBJETO" ADD CONSTRAINT
"PK_SEG_SISTEMA_OBJETO" PRIMARY KEY ("ID_SISTEMA", "ID_OBJETO")
```

SCRIPT PARA TABLA (SEG_OBJETO):

```
CREATE TABLE "ADM"."SEG_OBJETO"
(
  "ID_OBJETO" NUMBER(6,0) NOT NULL ENABLE,
  "OBJETO" VARCHAR2 (30) NOT NULL ENABLE,
  "TX_DESCRIPCION" VARCHAR2 (200),
  CONSTRAINT "PK_SEG_OBJETO" PRIMARY KEY ("ID_OBJETO")
)
CREATE UNIQUE INDEX "ADM"."PK_SEG_OBJETO" ON "ADM"."SEG_OBJETO" ("ID_OBJETO")
ALTER TABLE "ADM"."SEG_OBJETO" ADD CONSTRAINT "PK_SEG_OBJETO" PRIMARY KEY
("ID_OBJETO")
```

SCRIPT PARA TABLA (SEG_METODO):

```
CREATE TABLE "ADM"."SEG_METODO"
(
  "ID_OBJETO" NUMBER(6,0) NOT NULL ENABLE,
  "ID_METODO" NUMBER(3,0) NOT NULL ENABLE,
  "METODO" VARCHAR2(30) NOT NULL ENABLE,
  "CVE_TIPO_METODO" VARCHAR2(6) NOT NULL ENABLE,
  CONSTRAINT "PK_SEG_METODO" PRIMARY KEY ("ID_OBJETO", "ID_METODO")
  CONSTRAINT "FK01_SEG_METODO" FOREIGN KEY ("ID_OBJETO")
  REFERENCES "ADM"."SEG_OBJETO" ("ID_OBJETO") ENABLE
)
)
```

```
CREATE UNIQUE INDEX "ADM"."PK_SEG_METODO" ON "ADM"."SEG_METODO"  
("ID_OBJETO", "ID_METODO")  
ALTER TABLE "ADM"."SEG_METODO" ADD CONSTRAINT "PK_SEG_METODO" PRIMARY KEY  
("ID_OBJETO", "ID_METODO")
```

SCRIPT PARA TABLA (SEG_USUARIO_PERFIL):

```
CREATE TABLE "ADM"."SEG_USUARIO_PERFIL"  
(  
    "ID_UNIDAD_NEGOCIO" NUMBER(8,0) NOT NULL ENABLE,  
    "ID_SISTEMA" NUMBER(8,0) NOT NULL ENABLE,  
    "ID_USUARIO" NUMBER(8,0) NOT NULL ENABLE,  
    "ID_PERFIL" NUMBER(5,0) NOT NULL ENABLE,  
    CONSTRAINT "PK_SEG_USUARIO_PERFIL" PRIMARY KEY ("ID_UNIDAD_NEGOCIO",  
"ID_SISTEMA", "ID_USUARIO", "ID_PERFIL")  
    CONSTRAINT "FK03_SEG_USUARIO_PERFIL" FOREIGN KEY ("ID_SISTEMA", "ID_PERFIL")  
REFERENCES "ADM"."SEG_PERFIL" ("ID_SISTEMA", "ID_PERFIL") ENABLE,  
    CONSTRAINT "FK_SEG_USUA_REF_13623_SEG_SIST" FOREIGN KEY ("ID_SISTEMA",  
"ID_UNIDAD_NEGOCIO", "ID_USUARIO")  
REFERENCES "ADM"."SEG_SIST_UN_USUARIO" ("ID_SISTEMA", "ID_UNIDAD_NEGOCIO",  
"ID_USUARIO") ENABLE  
)  
CREATE UNIQUE INDEX "ADM"."PK_SEG_USUARIO_PERFIL" ON  
"ADM"."SEG_USUARIO_PERFIL" ("ID_UNIDAD_NEGOCIO", "ID_SISTEMA", "ID_USUARIO",  
"ID_PERFIL")  
ALTER TABLE "ADM"."SEG_USUARIO_PERFIL" ADD CONSTRAINT  
"PK_SEG_USUARIO_PERFIL" PRIMARY KEY ("ID_UNIDAD_NEGOCIO", "ID_SISTEMA",  
"ID_USUARIO", "ID_PERFIL")
```

SCRIPT PARA TABLA (SEG_PERFIL):

```
CREATE TABLE "ADM"."SEG_PERFIL"  
(  
    "ID_SISTEMA" NUMBER(8,0) NOT NULL ENABLE,  
    "ID_PERFIL" NUMBER(5,0) NOT NULL ENABLE,  
    "PERFIL" VARCHAR2(15) NOT NULL ENABLE,  
    "TX_PERFIL" VARCHAR2(200),  
    CONSTRAINT "PK_SEG_PERFIL" PRIMARY KEY ("ID_SISTEMA", "ID_PERFIL")  
    CONSTRAINT "FK01_SEG_PERFIL" FOREIGN KEY ("ID_SISTEMA")  
REFERENCES "ADM"."SEG_SISTEMA" ("ID_SISTEMA") ENABLE  
)  
CREATE UNIQUE INDEX "ADM"."PK_SEG_PERFIL" ON "ADM"."SEG_PERFIL" ("ID_SISTEMA",  
"ID_PERFIL")  
ALTER TABLE "ADM"."SEG_PERFIL" ADD CONSTRAINT "PK_SEG_PERFIL" PRIMARY KEY  
("ID_SISTEMA", "ID_PERFIL")
```

SCRIPT PARA TABLA (SEG_PERFIL_OBJETO):

```
CREATE TABLE "ADM"."SEG_PERFIL_OBJETO"  
(  
    "ID_SISTEMA" NUMBER(8,0) NOT NULL ENABLE,  
    "ID_PERFIL" NUMBER(5,0) NOT NULL ENABLE,  
    "ID_OBJETO" NUMBER(6,0) NOT NULL ENABLE,
```

```
CONSTRAINT "PK_SEG_PERFIL_OBJETO" PRIMARY KEY ("ID_SISTEMA", "ID_PERFIL",
"ID_OBJETO")
CONSTRAINT "FK01_SEG_PERFIL_OBJETO" FOREIGN KEY ("ID_SISTEMA", "ID_PERFIL")
REFERENCES "ADM"."SEG_PERFIL" ("ID_SISTEMA", "ID_PERFIL") ENABLE,
REFERENCES "ADM"."SEG_SISTEMA_OBJETO" ("ID_SISTEMA", "ID_OBJETO") ENABLE
)
CREATE UNIQUE INDEX "ADM"."PK_SEG_PERFIL_OBJETO" ON "ADM"."SEG_PERFIL_OBJETO"
("ID_SISTEMA", "ID_PERFIL", "ID_OBJETO")
ALTER TABLE "ADM"."SEG_PERFIL_OBJETO" ADD CONSTRAINT "PK_SEG_PERFIL_OBJETO"
PRIMARY KEY ("ID_SISTEMA", "ID_PERFIL", "ID_OBJETO")
```

SCRIPT PARA TABLA (SEG_PERFIL_METODO):

```
CREATE TABLE "ADM"."SEG_PERFIL_METODO"
(
    "ID_SISTEMA" NUMBER(8,0) NOT NULL ENABLE,
    "ID_PERFIL" NUMBER(5,0) NOT NULL ENABLE,
    "ID_OBJETO" NUMBER(6,0) NOT NULL ENABLE,
    "ID_METODO" NUMBER(3,0) NOT NULL ENABLE,
    CONSTRAINT "PK_SEG_PERFIL_METODO" PRIMARY KEY ("ID_SISTEMA", "ID_PERFIL",
"ID_OBJETO", "ID_METODO")
    CONSTRAINT "FK01_SEG_PERFIL_METODO" FOREIGN KEY ("ID_SISTEMA", "ID_PERFIL",
"ID_OBJETO")
REFERENCES "ADM"."SEG_PERFIL_OBJETO" ("ID_SISTEMA", "ID_PERFIL", "ID_OBJETO")
ENABLE,
    CONSTRAINT "FK02_SEG_PERFIL_METODO" FOREIGN KEY ("ID_OBJETO", "ID_METODO")
REFERENCES "ADM"."SEG_METODO" ("ID_OBJETO", "ID_METODO") ENABLE
)
CREATE UNIQUE INDEX "ADM"."PK_SEG_PERFIL_METODO" ON
"ADM"."SEG_PERFIL_METODO" ("ID_SISTEMA", "ID_PERFIL", "ID_OBJETO", "ID_METODO")
ALTER TABLE "ADM"."SEG_PERFIL_METODO" ADD CONSTRAINT
"PK_SEG_PERFIL_METODO" PRIMARY KEY ("ID_SISTEMA", "ID_PERFIL", "ID_OBJETO",
"ID_METODO")
```

SCRIPT PARA TABLA (PRO_PRODUCTO):

```
CREATE TABLE "ADM"."PRO_PRODUCTO"
(
    "ID_UNIDAD_NEGOCIO" NUMBER(8,0) NOT NULL ENABLE,
    "ID_PRODUCTO" NUMBER(6,0) NOT NULL ENABLE,
    "PRODUCTO" VARCHAR2(30) NOT NULL ENABLE,
    "NOMBRE_CORTO_PROD" VARCHAR2(10) NOT NULL ENABLE,
    "B_RENUEVA" CHAR(1) NOT NULL ENABLE,
    "CVE_TIPO_PRODUCTO" VARCHAR2(6) NOT NULL ENABLE,
    CONSTRAINT "CK01_PRO_PRODUCTO" CHECK ( CVE_TIPO_PRODUCTO in
(TP_TRA', 'TP_DEC', 'TP_TP', 'TP_TR', 'TP_AB', 'TP_PP')) ENABLE,
    CONSTRAINT "PK_PRO_PRODUCTO" PRIMARY KEY ("ID_UNIDAD_NEGOCIO",
"ID_PRODUCTO")
    CONSTRAINT "FK01_PRO_PRODUCTO" FOREIGN KEY ("ID_UNIDAD_NEGOCIO")
REFERENCES "ADM"."UNIDAD_NEGOCIO" ("ID_UNIDAD_NEGOCIO") ENABLE
)
CREATE UNIQUE INDEX "ADM"."PK_PRO_PRODUCTO" ON "ADM"."PRO_PRODUCTO"
("ID_UNIDAD_NEGOCIO", "ID_PRODUCTO")
```

```
ALTER TABLE "ADM"."PRO_PRODUCTO" ADD CONSTRAINT "PK_PRO_PRODUCTO" PRIMARY  
KEY ("ID_UNIDAD_NEGOCIO", "ID_PRODUCTO")
```

SCRIPT PARA TABLA (PRO_PRODUCTO_VERSION):

```
CREATE TABLE "ADM"."PRO_PRODUCTO_VERSION"  
(  
    "ID_UNIDAD_NEGOCIO" NUMBER(8,0) NOT NULL ENABLE,  
    "ID_PRODUCTO" NUMBER(6,0) NOT NULL ENABLE,  
    "ID_VERSION" NUMBER(3,0) NOT NULL ENABLE,  
    "CVE_TIPO_PLAZO_VENTA" VARCHAR2 (6) NOT NULL ENABLE,  
    "PLAZO_MIN_VENTA" NUMBER (5,0),  
    "PLAZO_MAX_VENTA" NUMBER (5,0),  
    "PLAZO_DEF_VENTA" NUMBER (5,0),  
    "DIAS_VALIDOS_COT" NUMBER (2,0),  
    "B_RECALCULO_ANUAL" CHAR(1) NOT NULL ENABLE,  
    "CVE_TIPO_RET_DIF" VARCHAR2(6) NOT NULL ENABLE,  
    "NUM_DIAS_VENTA_RET" NUMBER(3,0) NOT NULL ENABLE,  
    "NUM_DIAS_VENTA_DIF" NUMBER(3,0) NOT NULL ENABLE,  
    "CVE_SIT_PRODUCTO" VARCHAR2(6) NOT NULL ENABLE,  
    "TX_DESC_VERSION" VARCHAR2(2000),  
    "ID_USUARIO_CREACION" NUMBER(8,0) NOT NULL ENABLE,  
    "FH_CREACION" DATE NOT NULL ENABLE,  
    "ID_USUARIO_ULT_MOD" NUMBER(8,0) NOT NULL ENABLE,  
    "FH_ULT_MOD" DATE NOT NULL ENABLE,  
    "ID_USUARIO_ACTIVACION" NUMBER (8,0),  
    "FH_ACTIVACION" DATE,  
    "F_INI_VERSION" DATE NOT NULL ENABLE,  
    "F_TER_VERSION" DATE NOT NULL ENABLE,  
    "B_ULT_REG_DIA" CHAR(1) NOT NULL ENABLE,  
    CONSTRAINT "PK_PRO_PRODUCTO_VERSION" PRIMARY KEY ("ID_UNIDAD_NEGOCIO",  
"ID_PRODUCTO", "ID_VERSION")  
    CONSTRAINT "FK01_PRO_PRODUCTO_VERSION" FOREIGN KEY ("ID_UNIDAD_NEGOCIO",  
"ID_PRODUCTO")  
    REFERENCES "ADM"."PRO_PRODUCTO" ("ID_UNIDAD_NEGOCIO", "ID_PRODUCTO") ENABLE  
)  
CREATE UNIQUE INDEX "ADM"."PK_PRO_PRODUCTO_VERSION" ON  
"ADM"."PRO_PRODUCTO_VERSION" ("ID_UNIDAD_NEGOCIO", "ID_PRODUCTO",  
"ID_VERSION")  
ALTER TABLE "ADM"."PRO_PRODUCTO_VERSION" ADD CONSTRAINT  
"PK_PRO_PRODUCTO_VERSION" PRIMARY KEY ("ID_UNIDAD_NEGOCIO", "ID_PRODUCTO",  
"ID_VERSION")
```

SCRIPT PARA TABLA (PRO_PRODUCTO_VERSION):

```
CREATE TABLE "ADM"."PRO_LINEA_NEGOCIO"  
(  
    "ID_UNIDAD_NEGOCIO" NUMBER(8,0) NOT NULL ENABLE,  
    "ID_PRODUCTO" NUMBER(6,0) NOT NULL ENABLE,  
    "ID_VERSION" NUMBER(3,0) NOT NULL ENABLE,  
    "ID_LINEA_NEGOCIO" NUMBER(3,0) NOT NULL ENABLE,  
    "LINEA_NEGOCIO" VARCHAR2(30) NOT NULL ENABLE,  
    "NOMBRE_CORTO_LN" VARCHAR2(10) NOT NULL ENABLE,
```

```
"CVE_CONTABILIZACION" VARCHAR2(6) NOT NULL ENABLE,  
"NUM_SECUENCIA" NUMBER(3,0) NOT NULL ENABLE,  
"MIN_UR" NUMBER(3,0) NOT NULL ENABLE,  
"MAX_UR" NUMBER(3,0) NOT NULL ENABLE,  
"B_USA_CATEGORIA" CHAR(1) NOT NULL ENABLE,  
"ID_USUARIO_ULT_MOD" NUMBER(8,0) NOT NULL ENABLE,  
"FH_ULT_MOD" DATE NOT NULL ENABLE,  
CONSTRAINT "PK_PRO_LINEA_NEGOCIO" PRIMARY KEY ("ID_UNIDAD_NEGOCIO",  
"ID_PRODUCTO", "ID_VERSION", "ID_LINEA_NEGOCIO")  
CONSTRAINT "FK01_PRO_LINEA_NEGOCIO" FOREIGN KEY ("ID_UNIDAD_NEGOCIO",  
"ID_PRODUCTO", "ID_VERSION")  
REFERENCES "ADM"."PRO_PRODUCTO_VERSION" ("ID_UNIDAD_NEGOCIO",  
"ID_PRODUCTO", "ID_VERSION") ENABLE  
)  
CREATE UNIQUE INDEX "ADM"."PK_PRO_LINEA_NEGOCIO" ON  
"ADM"."PRO_LINEA_NEGOCIO" ("ID_UNIDAD_NEGOCIO", "ID_PRODUCTO", "ID_VERSION",  
"ID_LINEA_NEGOCIO")  
ALTER TABLE "ADM"."PRO_LINEA_NEGOCIO" ADD CONSTRAINT  
"PK_PRO_LINEA_NEGOCIO" PRIMARY KEY ("ID_UNIDAD_NEGOCIO", "ID_PRODUCTO",  
"ID_VERSION", "ID_LINEA_NEGOCIO")
```

Anexo 3 Instalación Delphi

Para instalar Delphi se requiere cumplir con las siguientes características:

- Procesador: Intel Pentium 90 ó superior (recomendado Pentium 166)
- Sistema operativo: Microsoft Windows 95, 98, Me, XP, o NT4.0 ó 2000 con Service Pack 3 ó posterior
- Memoria RAM: 32Mb (recomendado 64Mb)
- Espacio disponible en disco duro: 80 MB para la instalación compacta y 230MB para la completa

Proceso de instalación de Delphi:

Inicialmente aparece la pantalla de bienvenida, dar clic en el botón next, ver figura 84.

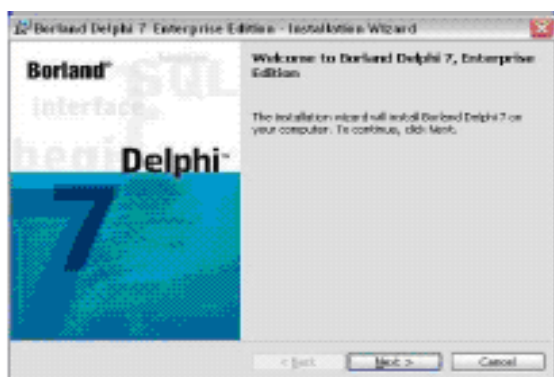


Figura 84. Pantalla de inicio

Seguido aparecerá la pantalla donde se ingresa el número de serie y la llave de autorización, ver figura 85.

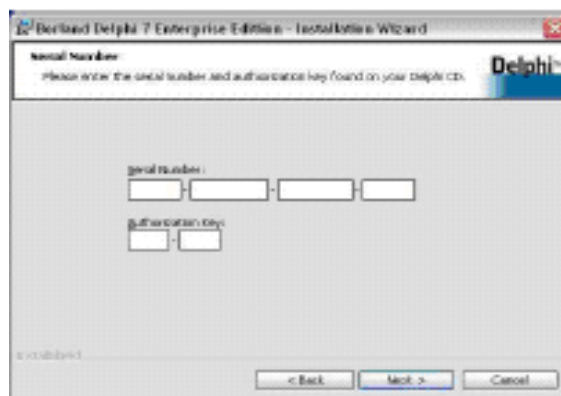


Figura 85. Pantalla número serial

Posteriormente muestra la pantalla de los términos y condiciones de la licencia, ver figura 86.

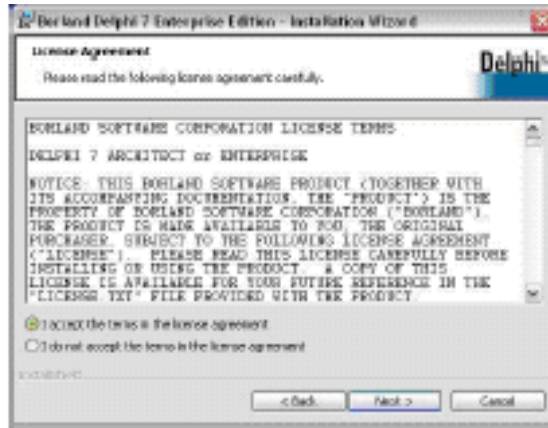


Figura 86. Aceptación de términos

Seleccionar modo de instalación *Typical* y posteriormente dar clic en next para continuar con la instalación, ver figura 87.

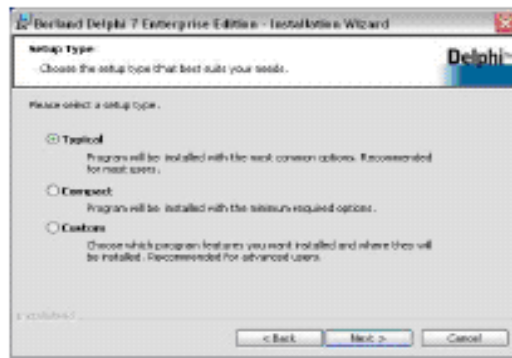


Figura 87. Tipo de instalación

Aceptar la opción de objetos deseados, seguido de next, ver figura 88.

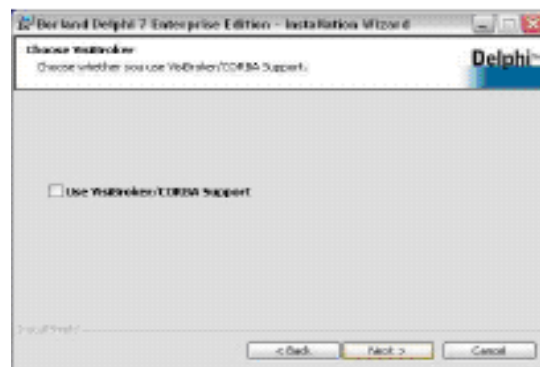


Figura 88. Objetos deseados

Para continuar con la instalación, seleccionar las carpetas donde se descargará el software, ver figura 89.

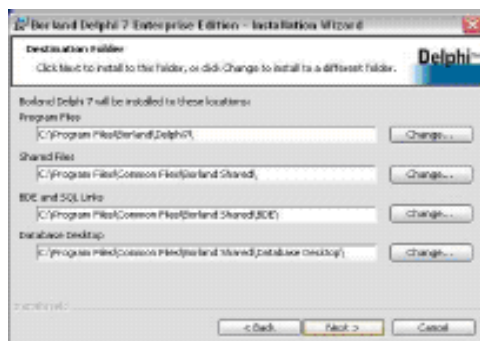


Figura 89. Ruta de instalación

Guardar un instante mientras se instala el programa, ver figura 90.

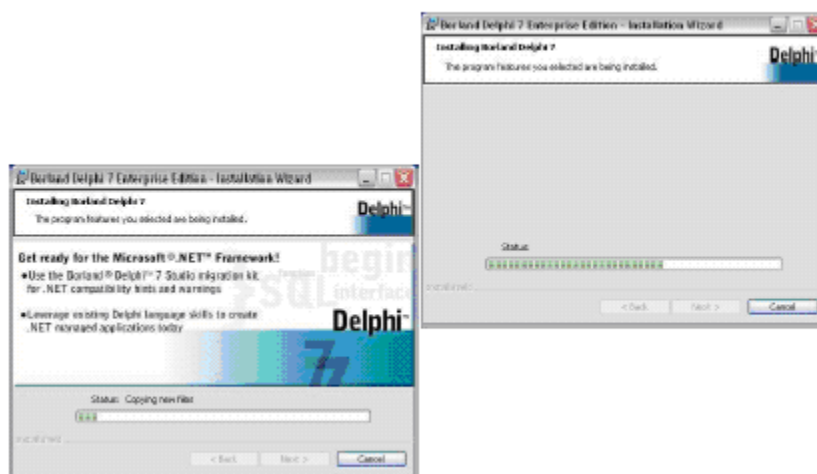


Figura 90. Ejecución del programa

Una vez instalado el programa, se muestra el fin de la instalación, ver figura 91.

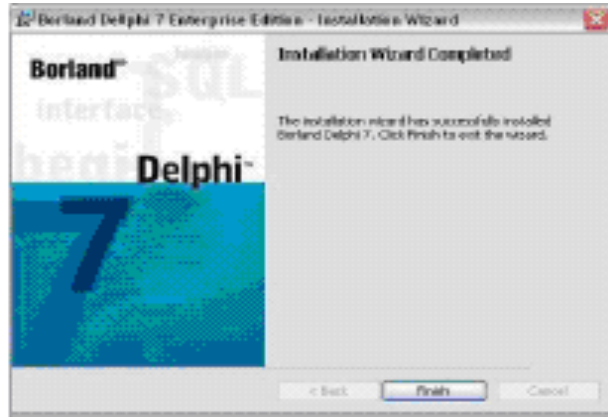


Figura 91. Fin de la instalación

Anexo 4 Pasos para crear disco autoejecutable

Para generar el disco de instalación, se utilizará la herramienta llamada INSTALSHIELD EXPRESS FOR DELPHI, la herramienta es instalada por default junto con el IDE de desarrollo para DELPHI, ver figura 92.

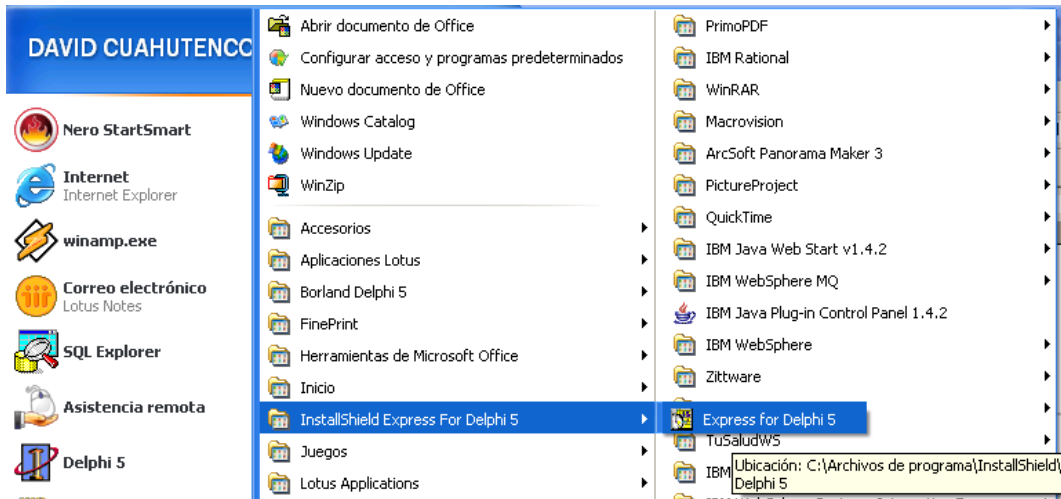


Figura 92. Herramienta para generar disco de instalación

Una vez que se ejecuta la herramienta para generar el disco autoinstalable, es necesario realizar la configuración de la interfaz de inicio, ver figura 93.

Setup Checklist	
Set the Visual Design	
<input checked="" type="checkbox"/>	<input type="checkbox"/> Application Information
<input checked="" type="checkbox"/>	<input type="checkbox"/> Main Window
<input checked="" type="checkbox"/>	<input type="checkbox"/> Features
Specify InstallShield Objects for Delphi 5	
<input checked="" type="checkbox"/>	<input type="checkbox"/> General Options
<input checked="" type="checkbox"/>	<input type="checkbox"/> Advanced Options
Specify Components and Files	
<input checked="" type="checkbox"/>	<input type="checkbox"/> Groups and Files
<input checked="" type="checkbox"/>	<input type="checkbox"/> Components
<input checked="" type="checkbox"/>	<input type="checkbox"/> Setup Types
Select User Interface Components	
<input checked="" type="checkbox"/>	<input type="checkbox"/> Dialog Boxes
Make Registry Changes	
<input checked="" type="checkbox"/>	<input type="checkbox"/> Keys
<input checked="" type="checkbox"/>	<input type="checkbox"/> Values
Specify Folders and Icons	
<input checked="" type="checkbox"/>	<input type="checkbox"/> General Settings
<input checked="" type="checkbox"/>	<input type="checkbox"/> Advanced Settings
Run Disk Builder	
<input checked="" type="checkbox"/>	<input type="checkbox"/> Disk Builder
Test the Installation	
<input type="checkbox"/>	<input type="checkbox"/> Test Run
Create Distribution Media	
<input type="checkbox"/>	<input type="checkbox"/> Copy to Floppy

Figura 93. Elementos a configurar

Por cada elemento de este CheckList, mostrado en la figura 93, se tendrá que realizar una configuración:

- Set Visual Design: Aquí el ejecutable prEmisor.exe resultado del desarrollo por completo del sistema, se agrega en el directorio principal de la instalación: C:\Archivos de programa\InterCase, ver figura 94.

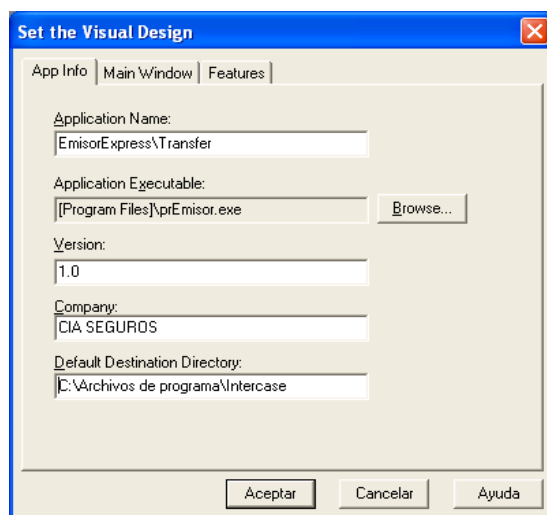


Figura. 94 Elementos para el archivo ejecutable

- Specify InstallShield Objects for Delphi: Seleccionar los objetos principales del funcionamiento del sistema. Éstos son objetos propios para operar base de datos locales, ver figura 95.

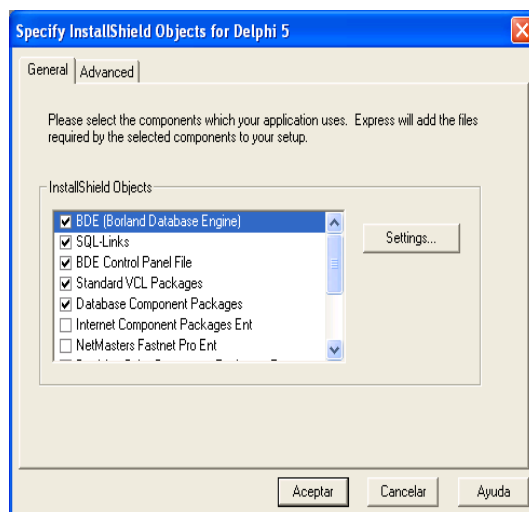


Figura 95. Elementos de la base de datos

- BDE Installation Type: Especificar el tipo de base de datos sobre el cual operará el sistema, ver figura 96.

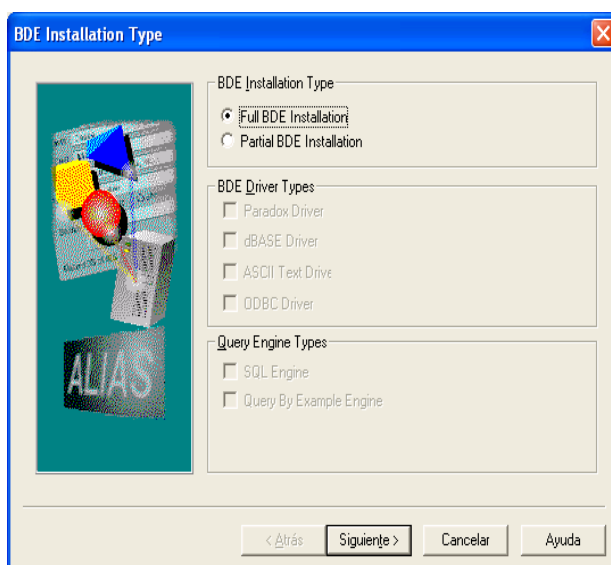


Figura 96. Tipo de base a usar en el sistema

- BDE Alias: Nombrar la base de datos, ver figura 97.

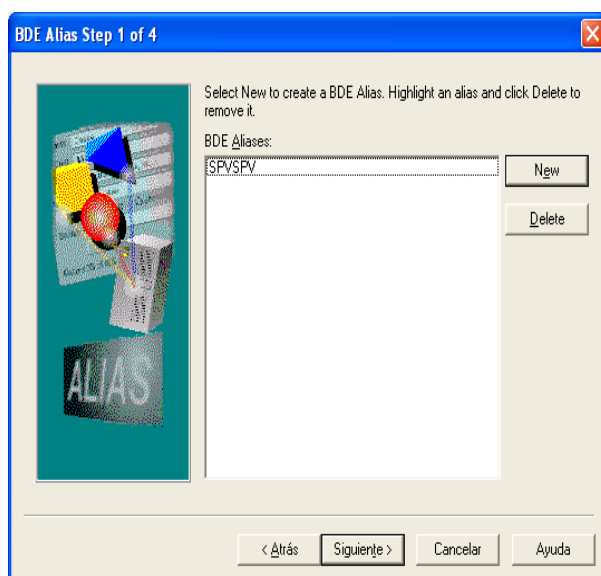


Figura 97. Especificación de la base de datos

- BDE Especification Alias: Especificar el tipo de Driver para la conexión del sistema hacia la base de datos con sus respectivos parámetros, ver figura 98.

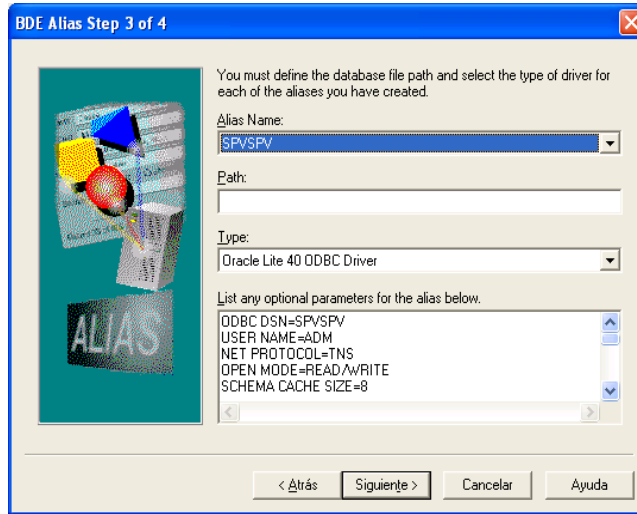


Figura 98. Driver para la conexión del sistema a la base de datos

- Specify Components and Files: Crear las carpetas donde se depositarán componentes y archivos del sistema, ver figura 99.

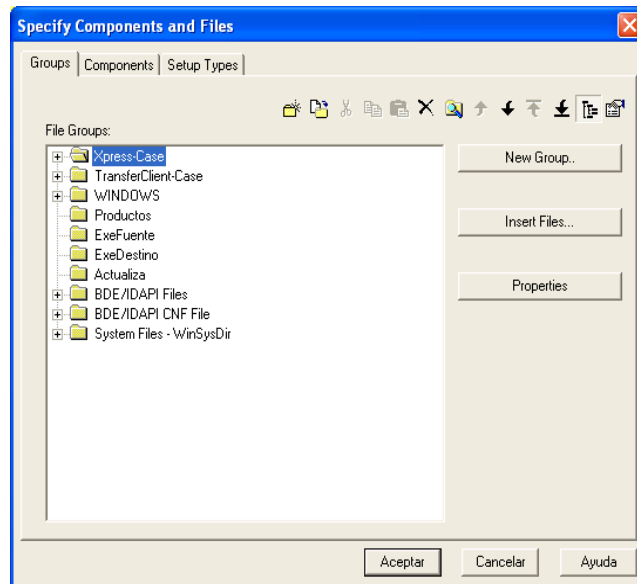


Figura 99. Creación de carpetas

- Una vez que se crearon las carpetas de instalación como se muestra en la figura 99, se agregarán sobre cada una de ellas los archivos que componen el sistema, ver figura 100.

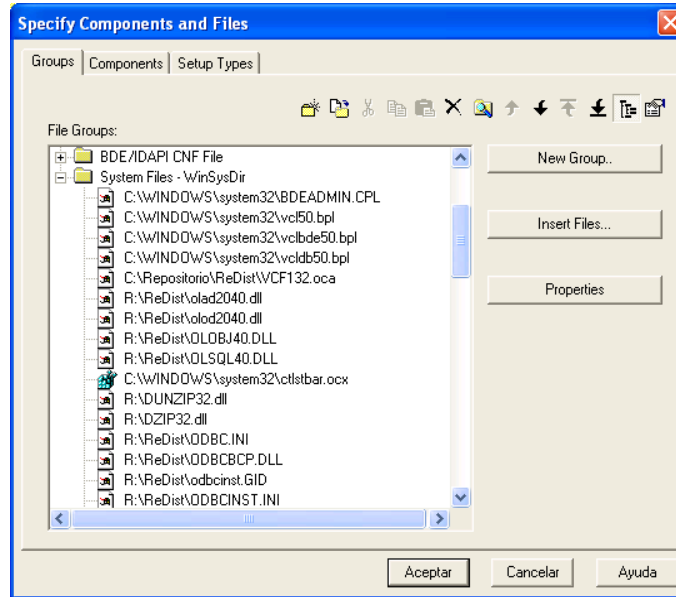


Figura 100. Fusión de carpetas al sistema operativo

- Dialog Boxes: Seleccionar los tipos de mensajes que se mostrarán al usuario durante la instalación, ver figura 101.

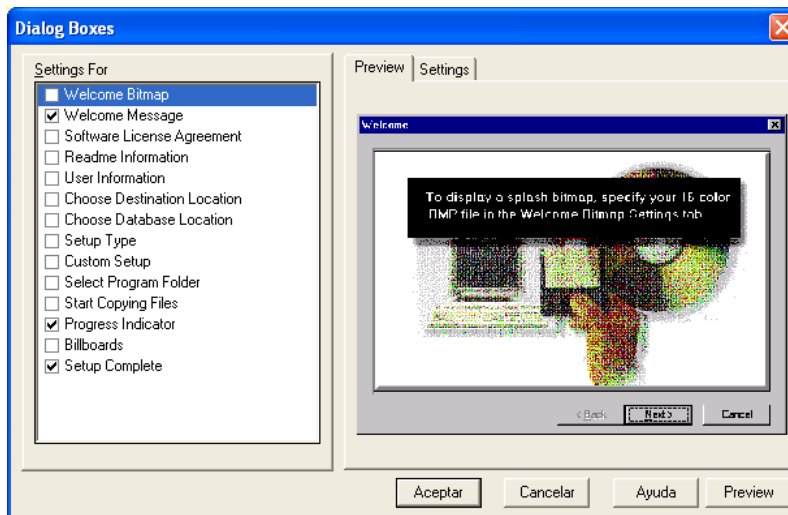


Figura 101. Mensajes de instalación hacia el usuario

- Make Registry Changes: Configurar el registro de librerías especiales sobre el sistema operativo WINDOWS, recomendable usarlo si se tiene conocimiento especializado en el sistema operativo, ver figura 102.

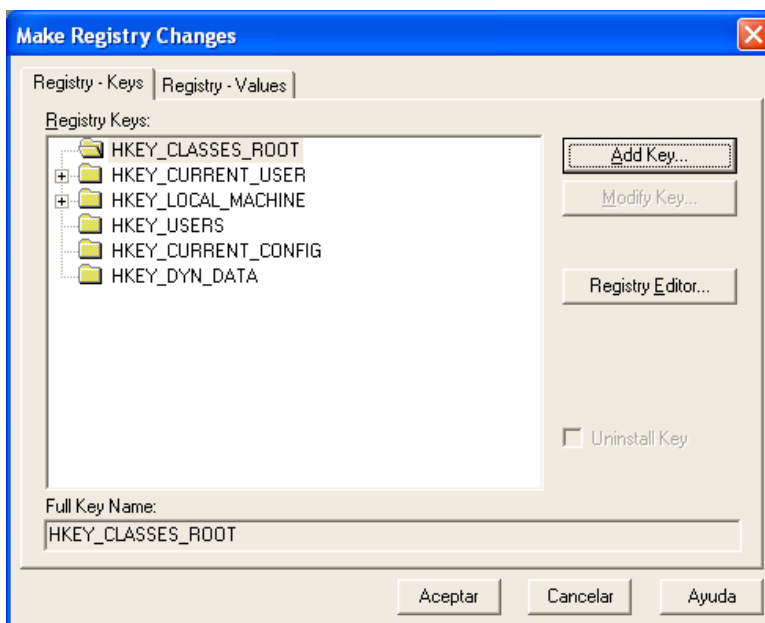


Figura 102. Configuración para el registro de librerías

- Specify Folders and Icons: Agregar los iconos relacionados a cada componente principal del sistema, ver figura 103.

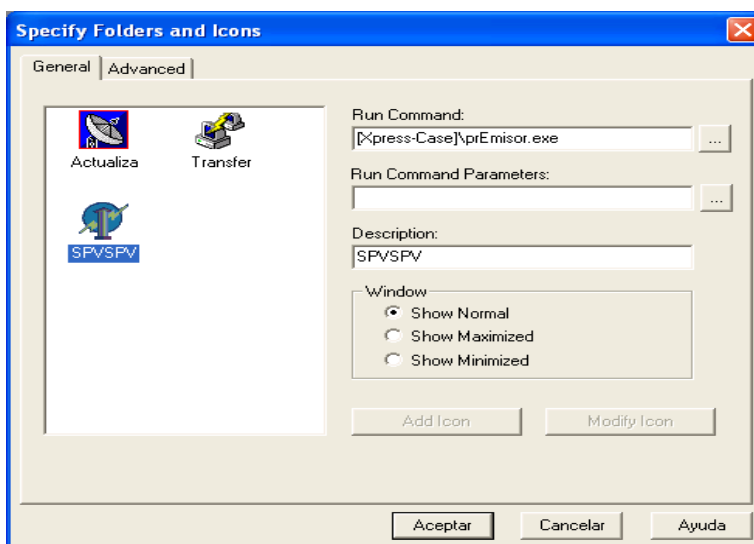


Figura 103. Creación de iconos para el sistema

- Disk Builder: Crear la encriptación secuencial de todos los elementos configurados, arrojando una serie de archivos listos para grabarse en un CD, ver figura 104.

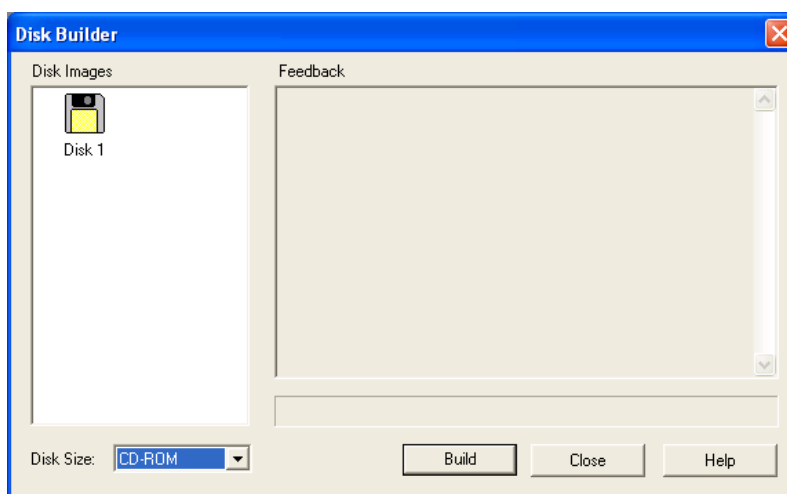


Figura 104. Encriptación de archivos para descargar en CD

El resultado es un CD de autoinstalación listo para operar en cualquier PC que cumpla con los requerimientos mínimos.

Con este CD se cumplirá con la parte portátil del sistema, ya que requerirá de la mínima intervención de técnicos especializados para realizar la instalación. Además de que el mantenimiento será más fácil de llevar a cabo.

Bibliografía

IMESFAC. *Inducción a la actividad aseguradora*,
Editorial Imesfac, 2a Edición, México, 2007.

Jacobson, I.; Booch, G.; Rumbaugh, J.; *El Proceso unificado de desarrollo*,
Addision Wesley, 4a Edición, España, 2000.

Presuman, R.S.; *Ingeniería del software. Un enfoque práctico*,
McGraw-Hill, 5a Edición, España, 2002

Steve Texteira & Xavier Pacheco. *Guía desarrollo Delphi*,
Prentice Hall, 2a Edición, España, 2000.