UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES

TEMA

Sistema de Administración de Bróker de Seguros utilizando Software Libre para la Institución "Asesores Productores De Seguros Álvaro Cazar"

APLICATIVO

Sistema de Administración del Bróker de Seguros (ACBróker 1.0)

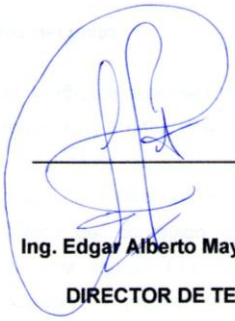
Autor: Jhoanna Paola Villegas Estévez
Director: ING. EDGAR MAYA OLALLA

Ibarra - Ecuador

CERTIFICACIÓN

En mi calidad de Director del Trabajo de Grado presentado por la egresada JHOANNA PAOLA VILLEGAS ESTÉVEZ, para optar por el título de INGENIERA EN SISTEMAS COMPUTACIONALES, cuyo tema es SISTEMA DE ADMINISTRACIÓN DE BRÓKER DE SEGUROS UTILIZANDO SOFTWARE LIBRE PARA LA INSTITUCIÓN "ASESORES PRODUCTORES DE SEGUROS ÁLVARO CAZAR". Considero que el presente trabajo reúne requisitos y méritos suficientes para ser sometido a la presentación pública y evaluación por parte del tribunal examinador que se designe.

En la ciudad de Ibarra a los 25 días del mes de octubre del 2011.



Ing. Edgar Alberto Maya Olalla

DIRECTOR DE TESIS

CERTIFICACIÓN DE LA EMPRESA

AC Broker
ASESORIA y SEGUROS

CERTIFICA:

Que, la Srta. Jhoanna Paola Villegas Estévez, con Número de cedula de Ciudadanía 1002601043, estudiante de la Facultad de Ingeniería en Ciencias Aplicadas, Carrera de Ingeniería en Sistemas Computacionales de la Universidad Técnica del Norte; ha realizado su Trabajo de Grado, **Sistema de Administración de Bróker de Seguros utilizando Software Libre para la Institución "Asesores Productores De Seguros Álvaro Cazar";** cumpliendo con todos los requisitos reglamentos de aprobación de la institución, con cualidades de responsabilidad y profesionalismo.

Para el efecto se extiende el presente CERTIFICADO DE CULMINACIÓN DE TRABAJO DE GRADO, en la ciudad de Ibarra a los 27 días del mes diciembre del 2011.

Atentamente,

ALVARO CAZAR G.
RUC: 1001709375001
AGENTE DE SEGUROS

Ing. Alvaro Cazar G.
GERENTE

Dir: Bolívar 8-91 Y Velasco (esq.) **Telf:** 062 611-750 **Telefax:** 062 601-945. **Cels.**090 918-852 / 087 278-251 / 094 959-635
e-mai: asecseguros@andinanet.ne / alvarocazar@hotmail.com - Ibarra – Ecuador

UNIVERSIDAD TÉCNICA DEL NORTE

CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE GRADO A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

Yo, JHOANNA PAOLA VILLEGAS ESTÉVEZ con Cédula de Ciudadanía Nº 100260104-3, manifiesto mi voluntad de ceder a la Universidad Técnica del Norte los derechos patrimoniales consagrados en la Ley de Propiedad Intelectual del Ecuador, artículos 4,5, y 6, en calidad de autora del Trabajo de Grado denominado **SISTEMA DE ADMINISTRACIÓN DE BRÓKER DE SEGUROS UTILIZANDO SOFTWARE LIBRE PARA LA INSTITUCIÓN "ASESORES PRODUCTORES DE SEGUROS ÁLVARO CAZAR"**, que ha sido desarrollado para optar el título de Ingeniera en Sistemas Computacionales en la Universidad Técnica del Norte, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En mi condición de autor me reservo los derechos morales de la obra antes citada. En concordancia suscribo este documento en el momento que hago entrega del trabajo final en formato impreso y digital a la biblioteca de la Universidad Técnica del Norte.

Firma: _____

Nombre: Johanna Paola Villegas Estévez

Cédula: 100260104-3

Ibarra, a los 16 del mes de Julio del 2012

UNIVERSIDAD TÉCNICA DEL NORTE

CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE GRADO

A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

1. Identificación de la obra

La Universidad Técnica del Norte dentro del proyecto Repositorio Digital institucional, determinó la necesidad de disponer de textos completos en formato digital con la finalidad de apoyar los procesos de investigación, docencia y extensión de la Universidad.

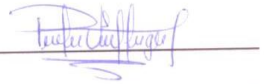
Por medio del presente documento dejo sentada mi voluntad de participar en este proyecto, para lo cual pongo a disposición la siguiente información:

DATOS CONTACTO			
CÉDULA DE IDENTIDAD:	100260104 – 3		
APELLIDOS Y NOMBRES:	VILLEGAS ESTÉVEZ JHOANNA PAOLA		
DIRECCIÓN:	LA FLORIDA, AZUCENAS S/N Y GARDENIAS		
EMAIL:	jhpao_villegas@hotmail.com		
TELÉFONO FIJO:	2631603	**TELÉFONO MOVIL:**	0987416034

DATOS DE LA OBRA	
TÍTULO:	Sistema de Administración de Bróker de Seguros utilizando Software Libre para la Institución “Asesores Productores De Seguros Álvaro Cazar”
AUTOR:	VILLEGAS ESTÉVEZ JHOANNA PAOLA
FECHA:	21/05/2012
TÍTULO POR EL QUE OPTA:	INGENIERO EN SISTEMAS COMPUTACIONALES
DIRECTOR:	ING. EDGAR MAYA

2. Autorización de uso a favor de la universidad

Yo, VILLEGAS ESTÉVEZ JHOANNA PAOLA, con cédula de identidad 100260104-3, en calidad de autor y titular de los derechos patrimoniales del trabajo de grado descrito anteriormente, hago la entrega del ejemplar respectivo en formato digital y autorizo a la Universidad Técnica del Norte la publicación de la obra en el repositorio digital y uso del archivo digital en la Biblioteca de la Universidad con fines académicos, para ampliar la disponibilidad del material y como apoyo a la educación, investigación y extensión; en concordancia con la Ley de Educación Superior Articulo 144.

Firma: 

Nombre: Johanna Paola Villegas Estévez

Cédula: 100260104-3

Ibarra, a los 16 del mes de Julio del 2012

DEDICATORIA

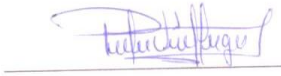
El presente trabajo de tesis es una demostración de la constancia y la tenacidad y está dedicado:

A Dios, por haberme permitido llegar hasta este momento tan especial en mi vida. Por haberme cuidado, dado las fuerzas y las ganas para llegar a cumplir este triunfo y disfrutarlo junto a las personas que tienen un lugar especial en mi vida.

A mi Padres y Hermanos, por haberme apoyado en todo momento, por sus consejos, sus valores, por la constante motivación que me brindaron para impulsarme a cumplir mis sueños los cuales me permiten crecer como persona y profesionalmente.

A mis maestros por su tiempo, apoyo y sabiduría que me transmitieron en el desarrollo de mi formación profesional, así como a la motivación que me brindaron para poder culminar mis estudios, en especial al Ing. y amigo Edgar Maya por su incondicional ayuda y por compartir sus conocimientos que fueron fundamentales para el desarrollo de esta tesis.

A mis amigos con los cuales mutuamente nos apoyamos en nuestra formación tanto profesional al igual que como personas de bien, con los cuales seguimos siendo amigos.



Villegas Estévez Johanna Paola

CI: 100260104-3

AGRADECIMIENTO

Agradeciendo a Dios, a mis padres Nancy y Abrahan, por haberme educado y ayudado cuando cometía errores y en los momentos más difíciles, apoyarme cada instante de mi vida e inculcar en mi la responsabilidad, la bondad hacia los demás, y por todo su amor.

A mis hermanos Cristina, Héctor y Franklin porque siempre he contado con ellos para todo, gracias a la confianza que nos hemos tenido, a su amistad y cariño.

Mi gratitud a la **UNIVERSIDAD TÉCNICA DEL NORTE** y a la **FACULTAD DE CIENCIAS APLICADAS** por la oportunidad de realizar mis estudios en sus instalaciones.

A toda mi familia, amigos y personas que con su apoyo y ayuda participaron directa o indirectamente en la elaboración de esta tesis.

Espero nunca se alejen de mí...

Villegas Estévez Johanna Paola

CI: 100260104-3

ÍNDICE DE CONTENIDOS

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

RESUMEN

En el mundo de los seguros, los Bróker representan un rol significativo para los clientes, quienes confían en ellos sus necesidades.

El asesoramiento en el proceso de contratación de seguros, la administración de sus pólizas y el seguimiento de los siniestros, son las principales funciones que desempeñan los Bróker de Seguros. Para poder cumplir con estas funciones, se deben realizar otras actividades que sólo serán efectivas si se lleva un seguimiento ordenado.

El Sistema AC Bróker 1.0 está diseñado para cubrir estas exigencias y requisitos de los Bróker de Seguros, y así, en el interés por satisfacer más las necesidades de sus clientes, se ayudarán y sentirán la optimización de sus funciones.

En la actualidad la mayoría de los negocios referentes a brókers no cuentan con sistemas para automatizar o mejorar los procesos, es por ello que un sistema de control desarrollado con nuevos métodos, con interfaces hará que se explote el negocio al máximo aumentando sus ventas y captando mayor cantidad de clientes; y en cuanto al negocio interno permita llevar un control personal con mayor eficiencia.

SUMMARY

In the world of insurance, Broker representing a significant role for customers who rely on them their needs.

The advice in the process of insurance contracts, the administration of their policies and tracking of claims, are the main functions performed by insurance brokers. To fulfill these functions, you must perform other activities that will only be effective if it takes them a tidy up.

Broker AC System 1.0 is designed to meet these demands and requirements of the Insurance Broker, and thus, more interest in meeting the needs of their clients, and feel will help optimize their functions.

At present the majority of business brokers regarding no systems to automate or improve processes, which is why we developed a control system with new methods, interfaces will be exploited to the maximum business increasing sales and capturing more customers, and as for the internal business enables to keep personal control with greater efficiency.

CAPITULO I

ASESORES PRODUCTORES DE SEGUROS ALVARO CAZAR

1. ASESORES PRODUCTORES DE SEGUROS ÁLVARO CAZAR

1.1 Antecedentes de la Institución

El Bróker de Seguros actúa como intermediario entre Clientes y Aseguradoras, es responsable de conjuntar a dichas partes y su objetivo es proporcionar al cliente un producto (Programa de Seguros) que se ajuste a sus necesidades.

Su principal función es el de velar por los intereses de los asegurados / clientes frente a las Aseguradoras, para lo cual desde el inicio de la relación el bróker es el encargado de realizar la inspección del riesgo para con esta información entregar al cliente el producto que requiere, el mismo que deberá contemplar las cláusulas y coberturas que cubran los riesgos potenciales; también es el responsable de verificar que las liquidaciones de los siniestros sean acordes a las condiciones establecidas en los contratos de seguros a fin de que el cliente esté satisfecho con la liquidación recibida.

El compromiso del bróker "Asesores Productores De Seguros Álvaro Cazar", es brindar un asesoramiento integral y personalizado a los clientes, con la precisión y la dinámica de un Bróker de sólida experiencia en la administración y elaboración de Programas de Seguros a la medida de cada cliente, basándose primordialmente en la transparencia, la ética y la permanente innovación del mercado asegurador.

1.2 Misión

ÁLVARO CAZAR AGENTE DE SEGUROS, responde por la gestión eficiente de generar, administrar, asesorar y comercializar programas de seguros competitivos a nivel nacional, para satisfacer las necesidades de los clientes, con personal calificado y comprometido, cumpliendo con las normativas legales y contribuyendo al desarrollo del país.

1.3 Visión

Ser una de las agencias asesoras, productoras de seguros más importantes y sólidas del país, de reconocido prestigio, confianza y credibilidad en el ámbito nacional, como resultado de sus crecientes niveles de productividad, administración transparente y alto nivel de tecnificación.

1.4 Objetivo

Conocer amplia y profundamente los riesgos de los clientes para proteger sus intereses, diseñando programas de seguros y administrándolos de acuerdo a sus necesidades, negociando con las aseguradoras las mejores condiciones y otorgando constante asesoría al momento de presentarse un siniestro.

1.5 Servicios

Principalmente es brindar un servicio transparente y útil, alternado con constante asesoría, teniendo como meta que los clientes tengan la libertad de elegir su compañía de seguros contando con la tranquilidad de tener profesionales en seguros a su lado en todo momento.

El Bróker se encuentra formado por los mejores profesionales en el ámbito de los seguros, interactuando en función de las necesidades de sus clientes. Todos y cada uno forman parte de un mismo objetivo que es la Satisfacción Total, ofreciendo:

- Consultoría a Empresas
- Asesoría Jurídica
- Administración de Carteras
- Análisis de Riesgos Especiales

AC BROKER tiene una especialización en el área de seguros que permite ofrecer un servicio ágil y eficaz en varios ramos.

1.5.1 Ramos Generales

Accidentes personales

Tiene por objeto la prestación de indemnizaciones en caso de accidentes que ocasionen la muerte o incapacidad del asegurado, y también los gastos de asistencia médica que precise el asegurado accidentado para su total curación con los límites y condiciones que se estipulen en la póliza.

Enfermedad

Su objetivo es garantizar parcial o totalmente, el pago de los gastos médicos – farmacéuticos incluso quirúrgicos, necesarios para la curación del enfermo.

Vehículos

Aquel que tiene por objeto la prestación de indemnizaciones derivadas de accidentes producidos a consecuencia de la circulación de vehículos, al igual en caso de robos de los vehículos.

Robo y /o Asalto

En éste el asegurador se compromete a indemnizar al asegurado por los daños sufridos a consecuencia de la desaparición, destrucción o deterioro de los objetos asegurados, a causa de robo o tentativa de robo.

Incendio y Aliadas

Es aquel que garantiza al asegurado la entrega de una indemnización en caso de incendio en los bienes determinados en la póliza o a la reparación de las piezas averiadas.

Lucro Cesante

Garantiza al asegurado la pérdida de rendimiento económico que hubiera podido alcanzarse en un acto o actividad, en caso de no haberse producido el siniestro descrito en la póliza.

Transporte

En este ramo la entidad aseguradora se compromete al pago de determinadas indemnizaciones a consecuencia de los daños sobrevenidos durante el trasporte de las mercaderías.

Casco Marítimo

Garantiza en general los riesgos de navegación que puedan afectar tanto al buque como transportador o a la carga transportada.

Dinero y Valores

Garantiza la indemnización del dinero perdido en caso de robo, que se encontraba en ventanilla o caja fuerte.

Responsabilidad Civil

El asegurador se compromete a indemnizar al asegurado del daño que pueda experimentar su patrimonio a consecuencia de la reclamación que le efectúe un tercero, por la responsabilidad que haya podido incurrir, tanto el propio asegurado como aquellas personas de quienes él deba responder civilmente.

Agropecuario

La actividad agrícola además está expuesta a los riesgos climáticos, que por su intensidad, frecuencia o por ser inoportunos provocan daños totales o parciales en cultivos y producciones de fruta, y con ello ocasionan pérdidas económicas a los agricultores. Son riesgos que el agricultor no puede prevenir ni evitar, sólo a veces mitigar, por lo que debe transferirlo a una empresa especializada en asumir las consecuencias económicas de los riesgos de terceros, una Compañía de Seguros, contratando el Seguro Agrícola contra fenómenos climáticos.

En caso de sufrir un siniestro por un riesgo cubierto por la póliza, mediante el Seguro Agrícola, el agricultor recuperará los costos directos de producción, esto es, el capital de trabajo invertido en el cultivo o producción de fruta. Así, mediante el Seguro Agrícola se logra que el agricultor tenga una mayor estabilidad y solvencia financiera,

sea un mejor sujeto de crédito, le permite la continuidad en la actividad agrícola y protege el trabajo y a la familia del agricultor.

SOAT

Garantiza la indemnización de un determinado valor ya sea en caso de muerte, invalidez o gastos médicos de las personas que se encuentren dentro del vehículo accidentado o de las terceras personas que resulten afectadas en un accidente de tránsito.

1.5.2 Ramo Vida

Vida individual

El seguro de Vida es uno de los tipos del Seguro de Personas en el que el pago por el asegurador de la cantidad estipulada en el contrato se hace depender del fallecimiento o supervivencia del asegurado en un momento determinado.

Vida en grupo

Los seguros de Vida en Grupo son planes abiertos que pueden adaptarse a las necesidades de cada uno de los clientes en función de las características propias de cada colectividad mismas que se representan como Seguro de Grupo Empresarial y Seguro de Grupo Diversos.

La cobertura básica consiste en el pago de la suma asegurada contratada en caso de ocurrir el fallecimiento del asegurado.

También pueden negociarse el pago de gastos médicos o invalidez total o permanente que se den a causa de accidentes.

1.5.3 Ramos Técnicos

Todo Riesgo para Contratista

El seguro de Todo Riesgo para Contratistas es un seguro de daños materiales, especialmente diseñado para proteger las obras civiles durante el periodo de su

construcción. Su protección está limitada al “Sitio de Construcción” tal como se define en el contrato de obra y / o en los planos del proyecto.

Las coberturas proporcionadas por el seguro TRC son muy flexibles y se adaptan a las necesidades propias de cada asegurado; entre ellas tenemos las siguientes:

- Incendio, Rayo, Explosión
- Descuido, Negligencia, Falla Humana
- Robo, Hurto, Acto Malintencionado
- Daño causado por material defectuoso
- Daño causado por defecto de mano armada
- Temblor, erupción volcánica, maremoto,
- Responsabilidad Civil Extracontractual
- Remoción de Escombros

Equipo y Maquinaria de Contratista

El seguro de Equipo y Maquinaria de Contratistas es un seguro diseñado para proteger todos aquellos equipos y maquinarias que se encuentran asignadas a un determinado proyecto en el sitio o región geográfica determinado por el asegurado.

Algunas de las coberturas proporcionadas por el seguro de Equipo y Maquinaria son las siguientes.

- Vuelcos Accidentales o Colisiones
- Incendio, Ignición o Rayo, Robo Total, Huelgas, y Alborotos Populares
- Desbordamiento de ríos o esteros, Ciclón, Huracán, Granizo, Temblor
- Terremoto, Erupción Volcánica, Inundación

Rotura de Maquinaria

Su denominación correcta es Avería o Rotura de Máquina.

Tiene por objeto garantizar los daños que puedan sufrir las máquinas, equipos o plantas industriales descritas en el contrato por hechos de carácter accidental inherentes a su funcionamiento o por manejo (defectos de fabricación, materiales o diseño de cualquier elemento de las mismas, impacto o entrada de cuerpos extraños, sobrecalentamiento por los fallos de los circuitos de refrigeración, impericia o negligencia en su manejo, etc.). Normalmente se excluyen los riesgos de carácter convencional, así como los derivados del propio desgaste del uso de los equipos.

Pérdida de Beneficio por Rotura de Maquinaria

Este seguro es una extensión del seguro de rotura de maquinaria. Indemniza las pérdidas financieras por interrupción del negocio como consecuencia de un evento amparado en el ramo antes citado. Puede tener coberturas adicionales como: suspensión de energía eléctrica, agua o gas, horarios de auditores o contadores.

Obra Civil Terminada

La póliza de obras civiles terminadas indemnizará al asegurado en la forma y hasta los límites estipulados y durante el período del seguro o durante otros períodos sucesivos por los que el asegurado abonó la prima correspondiente, todos los daños o pérdidas que los bienes asegurados o partes de los mismos sufran a causa de un siniestro súbito e imprevisto en forma tal que exijan la reparación o reposición de los bienes asegurados o de sus partes.

La cobertura operará cuando dichos daños o pérdidas sean causados por: incendio, impacto de rayo, explosión, colisión de vehículos terrestres o embarcaciones acuáticas, caída de aviones u otras naves aéreas o caída de objetos de los mismos, terremoto, volcanismo, tsunami (ola sísmica), viento huracanado (movimientos del aire superiores a la intensidad 8 de la escala Beaufort), avenida o inundación, acción de las olas o de aguas, hundimiento del terreno, corrimiento de tierra, caída de rocas u otros movimientos de la tierra, helada, aludes, hielo, vandalismo de personas aisladas.

Equipo Electrónico

Garantiza los daños que puedan sufrir los equipos electrónicos como computadores, impresoras, cámaras digitales, aparatos de medicina, telecomunicaciones, etc. que sean descritos en el contrato, e incluso sus instalaciones auxiliares, a consecuencia de cualquier hecho de carácter accidental, incendios, robo, excepto los expresamente excluidos.

Generalmente se excluyen los daños derivados del desgaste, montaje o desmontaje, influencia de la temperatura y aquellos que deban ser soportados por el fabricante, de acuerdo con el contrato de mantenimiento que se exige que sea celebrado con el fabricante.

1.5.4 Ramo Fianzas

Es aquel por el que el asegurador se obliga, en caso de incumplimiento por el tomador del seguro de sus obligaciones legales o contractuales, a indemnizar al asegurado a título de resarcimiento o penalidad de los daños patrimoniales sufridos dentro de los límites establecidos en el contrato.

De acuerdo con la legislación ecuatoriana, existen los siguientes contratos de seguros de esta clase:

- **Seriedad de Oferta**
- **Cumplimiento de Contrato**
- **Buen Uso de Anticipo**
- **Garantía Aduanera**

Todo pago hecho por el asegurador deberá serle reembolsado por el tomador del seguro.

Mediante esta clase de seguro, el tomador garantiza al asegurado el cumplimiento de determinadas obligaciones contraídas con este último.

Esta modalidad de seguro es utilizado especialmente por el Estado, al momento de la contratación de obras públicas en las cuales intervienen particulares. El Estado ecuatoriano es quién garantiza a través de este tipo de seguro la realización y buen cumplimiento de la obra encomendada.

Los seguros de Fianzas, según lo que establece la ley de Contratación Pública son Irrevocables, Incondicionales y de cobro inmediato.

El seguro de Fianza depende de un contrato principal, que es el que ha celebrado la persona natural o jurídica con el Estado o un particular.

1.6 Funcionamiento

AC BROKER es una empresa encargada en la venta de seguros el cual tiene ya cierto tiempo de funcionamiento, y se encarga de tramitar todas las ventas de seguros que ofrecen las diferentes compañías con las que trabaja, el medio por el cual llega a

sus clientes es personalizado, esto hace que los clientes adquieran productos con mucha más confianza.

El Bróker de Seguros trabaja para conseguir las ofertas que mejor se adapten a las necesidades de sus clientes, gestionando la contratación definitiva de sus contratos de seguros y asistiéndoles, asesorándoles y ayudándoles en todos los trámites y gestiones posteriores, especialmente en caso de producirse un siniestro.

Para conseguir la mejor oferta para su cliente, el Corredor realiza su tarea mediante un asesoramiento objetivo.

En caso de siniestro, el Corredor representa a su cliente ante las Compañías de Seguros tramitando, gestionando y defendiendo sus intereses para que las Entidades Aseguradoras cumplan con sus obligaciones según los términos y condiciones acordados en su Contrato de Seguro.

1.6.1 Identificación

Nuestro servicio comienza por identificar las áreas – riesgo de nuestro cliente, para evaluarlas correctamente y así preparar el Programa de Seguros en forma adecuada.

1.6.2 Negociación

Diseñado el Programa de Seguros, procedemos con su respectiva cotización, la cual dispondrá de las mejores tasas, coberturas, amparos y beneficios.

1.6.3 Emisión

Luego de la respectiva autorización del cliente, solicitamos la emisión de las pólizas en las Cías de Seguros, las mismas que serán debidamente revisadas y entregadas.

1.6.4 Seguimiento

Periódicamente nos preocupamos de realizar reuniones con el cliente, con el objeto de evaluar el cumplimiento de nuestro servicio y observar cualquier modificación en el riesgo, para mantener permanentemente actualizados los seguros contratados.

1.6.5 Siniestros

La asesoría técnica se refleja de gran manera en este punto, ya que dicha asistencia facilita la pronta recuperación o restitución de los bienes del cliente.

1.6.7 Renovación

El servicio de renovación nunca termina, por lo tanto con la debida anticipación se elabora un actualizado programa de seguros, con el afán de mantener una larga relación basada en la mutua confianza, siendo este el producto de un trabajo profesional y contando con el respaldo de las mejores aseguradoras del país.

1.7 Ventajas de contratar los Seguros a través de un Bróker

Figura 1 Ventajas de contratar un Bróker de Seguros

Fuente: RisckSeguros

1.7.1 Independencia: Rasgo diferenciador y único del Bróker de seguros.

El Bróker de Seguros NO mantiene relación o vinculación contractual con las Compañías Aseguradoras que suponga afección de ningún tipo con ellas, lo que garantiza su TOTAL INDEPENDENCIA y su TOTAL IMPARCIALIDAD a la hora de

seleccionar para sus clientes las opciones que consideren más adecuadas a sus necesidades.

La actividad no está mediatizada por acuerdos exclusivos con Aseguradoras concretas, ni está limitada a los intereses de éstas ni afectada por compromisos de producción adquiridos. De esta independencia del Bróker de Seguros se benefician desde el primer momento sus clientes, en la contratación de las pólizas y en la gestión y tramitación de los siniestros. En todo caso, el Corredor trabaja a favor de los intereses de su cliente.

1.7.2 Amplitud de oferta: Un Bróker todas las compañías

Puede ofrecer a sus clientes productos de cualquier Compañía de Seguros que opere en el Mercado. De esta forma, el cliente se asegura el libre acceso a cualquier producto de seguro del mercado, optando por el que mejor se adapte a sus necesidades y pudiendo acceder al mejor precio y a las mejores coberturas en cada momento gracias, además, a la gestión de negociación que llevan a cabo el Bróker con las Aseguradoras.

1.7.3 Servicio sin coste para el cliente

Por regla general, los Brókers reciben sus ingresos económicos de las Entidades aseguradoras y se obtienen normalmente de los corretajes de las pólizas contratadas por su intervención, lo que no supone ningún coste extra para sus clientes.

Pero además, se puede considerar que el importe de la prima de un seguro contratado a través de un corredor es indirectamente más económico ya que se habrá conseguido ese precio gracias al servicio de búsqueda de ofertas y a las negociaciones realizadas por el Bróker.

1.7.4 Seguridad: una actividad permanentemente supervisada y controlada

Todos los Brókers de Seguros, están inscritos en la Superintendencia de Bancos y Seguros, por lo cual son sometidos a su permanente supervisión e inspección.

La Superintendencia de Bancos y Seguros controla directamente toda la actividad de los Brókers de Seguros, velando porque su operativa se sujete a las normas de actuación establecidas por la Ley, revisando periódicamente toda su información

económica y contable. En ese sentido, revisa su capacidad financiera, comprueba la validez de su seguro de responsabilidad civil y verifica la realización de los programas de formación obligatorios. Este control permanente la actividad de los Brókers de Seguros es una garantía más de seguridad para sus clientes.

1.7.5 Asesoramiento personalizado: el Análisis Objetivo

Un rasgo exclusivo y único de los Brókers de Seguros es el Análisis Objetivo.

Antes de llevar a cabo la contratación de un seguro, está obligado a facilitar a su cliente información suficientemente motivada acerca de las opciones de seguro seleccionadas en función de su buen criterio profesional.

El cliente tiene la seguridad de que va a analizar siempre un mínimo de tres ofertas antes de presentarle aquella que mejor se adapta a sus necesidades, siendo capaz de acreditarle el análisis realizado si el asegurado se lo pide.

1.7.6 Gestión eficaz y servicio post-venta

Una de las funciones básicas es la asistencia permanente a su cliente a lo largo de toda la vida del Contrato. Asiste a su cliente en los trámites de gestión y ejecución de los contratos por primera vez y en las futuras renovaciones, revisando que se sigan adaptando a sus necesidades y vigilando el cumplimiento de los Contratos por parte de las Compañías de Seguros.

Por otra parte, el rasgo más característico del servicio que ofrecen los Brókers de Seguros es su labor de ayuda a los asegurados en las gestiones para la tramitación de sus siniestros con las Compañías de Seguros.

El Bróker agiliza los trámites, aporta los argumentos técnicos necesarios y se preocupa en agilizar el cobro de las prestaciones. Es muy habitual que los propios clientes no lleguen a conocer las numerosas gestiones que ha tenido que realizar su Corredor de Seguros con una Compañía para que finalmente se haya podido resolver satisfactoriamente un siniestro. Esta es una labor de servicio perfectamente asumida por los Brókers sin necesidad de evidenciar lo que ellos consideran como parte de su trabajo. Es en el momento del siniestro cuando un Bróker se encuentra más próximo de su cliente.

1.7.7 Trato personal y humano

En un mundo dominado cada día más por la frialdad de los operadores automáticos y de internet, el Bróker de Seguros tiene a gala ofrecer siempre un trato personalizado a sus clientes.

Ante la competencia de los canales de venta directa, el aportará siempre esa cercanía que permite al cliente expresar sus necesidades en un ambiente de tranquilidad y confianza.

Incluso los empleados de los Brókers de mayor tamaño tienen entre sus prioridades mantener un trato personalizado con todos y cada uno de sus clientes.

1.7.8 Representa al cliente ante la aseguradora

Otra de las funciones básicas es el papel de representación y defensa de los intereses de sus clientes frente a las aseguradoras.

La ley le otorga esta facultad con plena capacidad de negociación con las compañías en nombre de sus clientes en todo lo relacionado con sus contratos de seguros.

Es tan importante este papel como representante de los asegurados que incluso, cualquier comunicación que un Corredor dirija a una Compañía de Seguros en nombre de su cliente se considera exactamente igual, a todos los efectos, que si la hubiese realizado efectivamente el propio asegurado, salvo para suscribir un nuevo contrato, o para modificar o rescindir el que ya tiene.

1.8 Compañías Aseguradoras con las que Trabaja

Como la tranquilidad del cliente es la mayor preocupación AC BROKER trabaja sólo con las mejores compañías del mercado.

Mantenemos contrato de agenciamiento con las siguientes Compañías de Seguros:

- SEGUROS EQUINOCCIAL
- AIG METROPOLITANA
- COOPSEGUROS DEL ECUADOR
- ASEGURADORA DEL SUR
- SEGUROS BOLIVAR
- SEGUROS COLVIDA
- SEGUROS PICHINCHA
- SEGUROS COLONIAL
- SEGUROS ALIANZA
- INTEROCEÁNICA DE SEGUROS
- SEGUROS EQUIVIDA
- HISPANA DE SEGUROS
- CENTRO SEGUROS
- LATINA DE SEGUROS Y REASEGUROS
- LATINA VIDA
- PANAMERICAN LIFE
- B.M.I.
- SALUD
- HUMANA

CAPITULO II

ESTUDIOS DE LAS HERRAMIENTAS DE DESARROLLO

2. ESTUDIOS DE LAS HERRAMIENTAS DE DESARROLLO

SOFTWARE LIBRE

En los últimos años hemos venido escuchando cada vez más los términos Software Libre (Free Software) y, más recientemente Software de fuentes abiertas (Open Source Software).

Estos términos se refieren al modelo de desarrollo y de distribución del software desarrollado cooperativamente. En vez de que el código del sistema o de cada uno de los programas sea un secreto celosamente guardado por la empresa que lo produce, éste es puesto a disposición del público, para que puedan modificar, mejorar o corregir.

Razones para usar Software Libre:

Libre: Entre otras cosas es libre para usar, modificar, regalar o vender los programas de software libre.

La copia es legal: Es legal repartir software libre a otras personas. Usar un sistema libre evita en gran medida los problemas de la piratería. Si lo natural es compartir programas con otras personas, con software libre es legal.

Abierto: Se puede usar el código de los programas y modificarlo.

Colaborativo: El modelo de desarrollo de software libre es colaborativo y participativo. Todo se puede modificar o criticar.

Ayuda: Existen innumerables grupos de usuarios que se ayudan entre sí a través de Internet. Es decir si surge un problema es muy probable que a otras personas les ha ocurrido y su ayuda será la más valiosa que puedas encontrar.

Auditable: El software libre se puede inspeccionar al disponer de su código fuente.

Robusto frente a los virus: Existen muy pocos virus para GNU/Linux ya que el problema de los virus se debe al diseño del sistema operativo.

Personalizable: Puede personalizar toda la interfaz que el sistema le presenta al usuario.

Bajo costo: De estudios realizados para empresas se han encontrado reducciones de costo de hasta un 30% en TCO (Costo total de propiedad).

Reutilización de equipos: GNU/Linux puede ejecutarse perfectamente en equipos que han sido desechados por las nuevas versiones de sistemas operativos propietarios.

Ventajas

- Existen aplicaciones para todas las plataformas (Linux, Windows, Mac Os)
- El precio de las aplicaciones es mucho menor, la mayoría de las veces son gratuitas
- Libertad de copia
- Libertad de modificación y mejora
- Libertad de uso con cualquier fin
- Libertad de redistribución
- Facilidad a la hora de traducir una aplicación en varios idiomas
- Mayor seguridad y fiabilidad
- El usuario no depende del autor del software
- Escrutinio Público ya que muchas personas que tienen acceso al código fuente, eso lleva a un proceso de corrección de errores muy dinámico, no hace falta esperar que el proveedor del software saque una nueva versión.

Desventajas

- Algunas aplicaciones (bajo Linux) pueden llegar a ser algo complicadas de instalar
- Inexistencia de garantía por parte del autor
- Interfaces gráficas menos amigables
- Poca estabilidad y flexibilidad en el campo de multimedia y juegos
- Menor compatibilidad con el hardware
- Dificultad en la implementación y en la interoperabilidad (aprendizaje, instalación, migración, etc)
- La curva de aprendizaje es mayor
- La mayoría de la configuración de hardware no es intuitiva

2.1 Apache Tomcat

Apache es un programa que permite crear un servidor http gratuito en tu propio ordenador de una forma rápida y sencilla, robusto y de código abierto.

También llamado Jakarta Tomcat o simplemente Tomcat, funciona como un contenedor de servlets desarrollado bajo el proyecto Jakarta en la Apache Software Foundation. Tomcat, implementa las especificaciones de los servlets y de JavaServer Pages (JSP) de Sun Microsystems.

Es el software más usado mundialmente para crear servidores http (bajo linux claro).

Dispone de libre acceso a su código fuente como se mencionó anteriormente. Las primeras distribuciones de Tomcat fueron las versiones 3.0.x. Las versiones más recientes son las 7.x, que implementan las especificaciones de Servlet 3.0 y de JSP 2.2. A partir de la versión 4.0, Jakarta Tomcat utiliza el contenedor de servlets Catalina.

Tomcat es un servidor web con soporte de servlets y JSPs. No es un servidor de aplicaciones, incluye el compilador Jasper, que compila JSPs convirtiéndolas en servlets. El motor de servlets de Tomcat a menudo se presenta en combinación con el servidor web Apache.

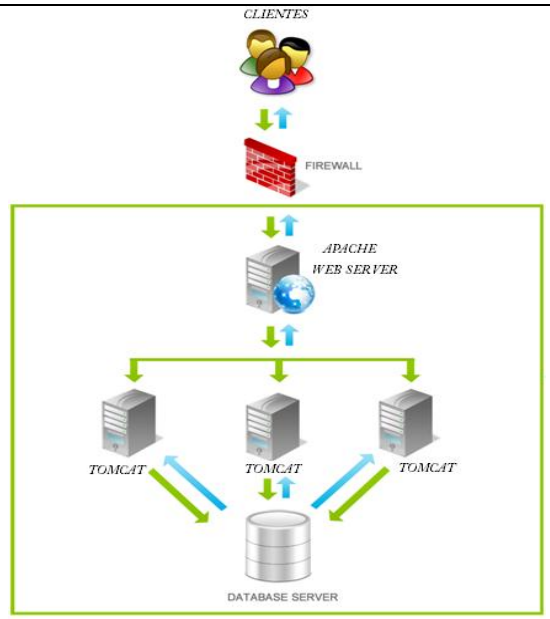


Figura 2 Funcionamiento Apache Tomcat

Fuente: Propia

Puede funcionar como servidor web por sí mismo. En sus inicios existió la percepción de que el uso de Tomcat de forma autónoma era sólo recomendable para entornos de desarrollo y entornos con requisitos mínimos de velocidad y gestión de transacciones. Pero hoy en día ya no existe esa percepción y Tomcat es usado como servidor web autónomo en entornos con alto nivel de tráfico y alta disponibilidad.

Dado que Tomcat fue escrito en Java, funciona en cualquier sistema operativo que disponga de la máquina virtual Java.

Estructura de directorios

La jerarquía de directorios de instalación de Tomcat incluye:

- **bin** - arranque, cierre, y otros scripts y ejecutables
- **common** - clases comunes que pueden utilizar Catalina y las aplicaciones web
- **conf** - ficheros XML y los correspondientes DTD para la configuración de Tomcat
- **logs** - logs de Catalina y de las aplicaciones
- **server** - clases utilizadas solamente por Catalina
- **shared** - clases compartidas por todas las aplicaciones web
- **webapps** - directorio que contiene las aplicaciones web
- **work** - almacenamiento temporal de ficheros y directorios

2.2 JSP Y JSF

Los servlets y Java Server Pages (JSPs) son dos métodos de creación de páginas web dinámicas en servidor usando el lenguaje Java. En ese sentido son similares a otros métodos o lenguajes tales como el PHP, los CGIs (common gateway interface), programas que generan páginas web en el servidor, o los ASP (Active Server Pages), un método específico de Microsoft. etc.

Los JSPs y servlets se ejecutan en una máquina virtual Java, lo cual permite que, en principio, se puedan usar en cualquier tipo de ordenador, siempre que exista una máquina virtual Java para él. Cada servlet (o JSP, a partir de ahora lo usaremos de forma indistinta) se ejecuta en su propia hebra, es decir, en su propio contexto; pero no se comienza a ejecutar cada vez que recibe una petición, sino que persiste de una petición a la siguiente, de forma que no se pierde tiempo en invocarlo (cargar programa + intérprete). Su persistencia le permite también hacer una serie de cosas

de forma más eficiente: conexión a bases de datos y manejo de sesiones, por ejemplo.

Los JSPs son en realidad servlets: un JSP se compila a un programa en Java la primera vez que se invoca, y del programa en Java se crea una clase que se empieza a ejecutar en el servidor como un servlet. La principal diferencia entre los servlets y los JSPs es el enfoque de la programación: un JSP es una página Web con etiquetas especiales y código Java incrustado, mientras que un servlet es un programa que recibe peticiones y genera a partir de ellas una página web

Ambos necesitan un programa que los contenga, y sea el que envíe efectivamente páginas web al servidor, y reciba las peticiones, las distribuya entre los servlets, y lleve a cabo todas las tareas de gestión propias de un servidor web. Mientras que servidores como el Apache están especialmente pensados para páginas web estáticas CGIs, y programas ejecutados por el servidor, tales como el PHP, hay otros servidores específicos para servlets y JSPs llamados contenedores de servlets (servlet containers) o servlet engines. Los principales son los siguientes:

- Resin, de Caucho Technologies, un motor especialmente enfocado al servicio de páginas XML, con una licencia libre para desarrolladores. Dice ser bastante rápido. Incluye soporte para Javascript además de Java. Incluye también un lenguaje de templates llamado XTP. Es bastante fácil de instalar, y en dos minutos, se pueden empezar a servir páginas JSP.

- BEA Weblogic es un servidor de aplicaciones de alto nivel, y también de alto precio. Está escrito íntegramente en Java, y se combina con otra serie de productos, tales como Tuxedo, un servidor de bases de datos para XML.

- JRun, de Macromedia, un servidor de aplicaciones de Java, de precio medio y probablemente prestaciones medias. Se puede bajar una versión de evaluación gratuita.

- Lutris Enhydra, otro servidor gratuito y Open Source, aunque tiene una versión de pago. También enfocado a servir XML, y para plataformas móviles. Las versiones más actualizadas son de pago, como es natural.

- El más popular, Open Source, y continuamente en desarrollo, es el Jakarta Tomcat, del consorcio Apache, un contenedor de servlets con muchos desarrollos adicionales alrededor; por ejemplo, Cocoon para servir páginas XML. Puede servir páginas sólo o bien como un añadido al servidor Apache. Es Open Source, relativamente rápido, y fácil de instalar.

JSF

JavaServer Faces (JSF) es una tecnología y framework para aplicaciones Java basadas en web que simplifica el desarrollo de interfaces de usuario en aplicaciones Java EE. JSF usa JavaServer Pages (JSP) como la tecnología que permite hacer el despliegue de las páginas, pero también se puede acomodar a otras tecnologías como XUL.

JSF incluye:

- Un conjunto de APIs para representar componentes de una interfaz de usuario y administrar su estado, manejar eventos, validar entrada, definir un esquema de navegación de las páginas y dar soporte para internacionalización y accesibilidad.

- Un conjunto por defecto de componentes para la interfaz de usuario.

- Dos bibliotecas de etiquetas personalizadas para JavaServer Pages que permiten expresar una interfaz JavaServer Faces dentro de una página JSP.

- Un modelo de eventos en el lado del servidor.

- Administración de estados.

- Beans administrados.

Objetivos

Definir un conjunto simple de clases base de Java para componentes de la interfaz de usuario, estado de los componentes y eventos de entrada. Estas clases tratarán los aspectos del ciclo de vida de la interfaz de usuario, controlando el estado de un componente durante el ciclo de vida de su página.

Proporcionar un conjunto de componentes para la interfaz de usuario, incluyendo los elementos estándares de HTML para representar un formulario. Estos componentes se obtendrán de un conjunto básico de clases base que se pueden utilizar para definir componentes nuevos.

Proporcionar un modelo de JavaBeans para enviar eventos desde los controles de la interfaz de usuario del lado del cliente a la aplicación del servidor.

Definir APIs para la validación de entrada, incluyendo soporte para la validación en el lado del cliente.

Especificar un modelo para la internacionalización y localización de la interfaz de usuario.

Automatizar la generación de salidas apropiadas para el objetivo del cliente, teniendo en cuenta todos los datos de configuración disponibles del cliente, como versión del navegador

COMPARACIÓN ENTRE JSP Y JSF

JSP (Java Server Pages) son las páginas webs dinámicas de java, dinámicas porque con ellas puedes crear contenido que puedes sacar de una base de datos.

JSF (Java Server Faces) son componentes que se utilizan en las páginas JSP para facilitar la programación, dibujar formularios o mostrar datos que están en una Base en una tabla.

JSF es una aplicación web que se utiliza para simplificar la integración de desarrollo de interfaces de usuario basada en web.

JSP es una tecnología basada en Java que se utiliza específicamente para ayudar a los desarrolladores de software crear páginas web dinámicas.

JSF contiene varias características fundamentales, incluyendo pero no limitado a, beans gestionados, un sistema de plantillas basado en componentes, y dos XML basado en las bibliotecas de etiquetas, JSP debe ser compilado en bytecode de Java para funcionar correctamente.

Para comenzar JSP trabaja de una conocida forma en base Request/Response por el contrario JSF tiene una lógica de trabajo tipo aplicación Desktop en base a eventos.

JSF Es una ampliación del JSP ya que utiliza Ajax.

JSF Trae componentes de diseño que facilitan la labor del desarrollador, y estos componentes pueden abarcar el uso de AJAX.

JSF se establece como un estándar para la construcción de interfaces de usuario ubicándose en el lado del servidor.

JSP	JSF
Trabaja en base a request y response.	Trabaja en base a aplicaciones de escritorio en base a eventos.
Al utilizar context, request, response se realiza de manera automática mediante Servlet.	Se tiene que simular el manejo de los mismos no lo hace automático.
Es dirigido a un evento por cada solicitud.	Puede tener varios controladores de eventos en una página.
Utiliza html simplemente.	Es una interfaz de usuario orientada a objetos.
Debe ser compilado en bytecode de Java para funcionar correctamente.	Contienen varias funciones básicas, pero no limitado a, beans administradas, un sistema de plantillas basado en componentes, y dos XML basado en librerías de etiquetas

Tabla 1 Comparación entre JSP y JSF

Fuente: Propia

2.3 Framework Spring

El Spring Framework (también conocido simplemente como Spring) es un framework de código abierto de desarrollo de aplicaciones para la plataforma Java. La primera versión fue escrita por Rod Jonhson, quien lo lanzó primero con la publicación de su libro Expert One-on-One Java EE Design and Development (Wrox Press, octubre 2002). También hay una versión para la plataforma .NET, Spring .NET.

Facilita la creación de aplicaciones para nuestra empresa. Diseñado en módulos, con funcionalidades específicas y consistentes con otros módulos, facilita el desarrollo de funcionalidades específicas y hace que la curva de aprendizaje sea favorable para el desarrollador.

Dentro de las ventajas que nos ofrece Spring, nos encontramos con que facilita la manipulación de nuestros objetos se usen EJBs o no, reduce la proliferación de Singletons, elimina la necesidad de usar distintos y variados tipos de ficheros de configuración, mejora la práctica de programación, permite el uso o no de EJBs, realizando el mismo tipo de funciones sin ellos.

El framework fue lanzado inicialmente bajo Apache 2.0 License en junio de 2003. El primer gran lanzamiento hito fue la versión 1.0, que apareció en marzo de 2004 y fue seguida por otros hitos en septiembre de 2004 y marzo de 2005.

A pesar de que Spring Framework no obliga a usar un modelo de programación en particular, se ha popularizado en la comunidad de programadores en Java al ser considerado como una alternativa y sustituto del modelo de Enterprise JavaBean. Por su diseño el framework ofrece mucha libertad a los desarrolladores en Java y soluciones muy bien documentadas y fáciles de usar para las prácticas comunes en la industria.

VERSIONES

Primeramente fue creado en Sourceforge en febrero de 2003.

Se lanza una primera versión (1.0) en marzo de 2004. Después de este lanzamiento Spring ganó mucha popularidad en la comunidad Java, debido en parte al uso de Javadoc y de una documentación de referencia por encima del promedio de un proyecto de código abierto.

La meta de diseño de Spring Framework es su facilidad de integración con los estándares J2EE y herramientas comerciales existentes.

Spring Framework hizo que aquellas técnicas que resultaban desconocidas para la mayoría de programadores se volvieran populares en un periodo muy corto de tiempo. El ejemplo más notable es la inversión de control. En el año 2004, Spring disfrutó de unas altísimas tasas de adopción y al ofrecer su propio framework de programación orientada a aspectos (aspect-oriented programming, AOP) consiguió hacer más popular su paradigma de programación en la comunidad Java.

En 2005 Spring superó las tasas de adopción del año anterior como resultado de nuevos lanzamientos y más características fueron añadidas.

Tiene mucha demanda. Incluso ya no es exclusivo de Java, pues ya hay versión para .NET, bautizada como Spring.NET.

CARACTERÍSTICAS

La característica fundamental de este framework es que puede emplearse en cualquier aplicación hecha en Java, citamos las siguientes:

Facilita el desarrollo de aplicaciones J2EE, promoviendo buenas prácticas de diseño y programación. En concreto se trata de manejar patrones de diseño como Factory, Abstract Factory, Builder, Decorator, Service Locator, etc; que son ampliamente reconocidos dentro de la industria del desarrollo de software.

- Es código abierto
- Enfoque en el manejo de objetos de negocio, dentro de una arquitectura en capas
- Una ventaja de Spring es su modularidad, pudiendo usar algunos de los módulos sin comprometerse con el uso del resto.
- El Core Container o Contenedor de Inversión de Control (Inversion of Control, IoC) es el núcleo del sistema. Responsable de la creación y configuración de los objetos.
- Aspect-Oriented Programming Framework, que trabaja con soluciones que son utilizadas en numerosos lugares de una aplicación, lo que se conoce como asuntos transversales (cross-cutting concerns).
- Data Access Framework, que facilita el trabajo de usar un API com JDBC, Hibernate, etc.
- Transaction Management Framework.

- Remote Access framework. Facilita la existencia de objetos en el servidor que son exportados para ser usados como servicios remotos.
- Spring Web MVC. Maneja la asignación de peticiones a controladores y desde estos a las vistas. Implica el manejo y validación de formularios.

Una característica de Spring es que puede actuar como pegamento de integración entre diferentes APIs (JDBC, JNDI, etc.) y frameworks.

VENTAJAS

- A continuación se detallan las ventajas de utilizar Spring.
- Su configuración está basada en archivos XML.
- Uso de JTA para el manejo de transacciones.
- Transacciones distribuidas.
- La instanciación de las clases las realiza Spring a través de la inyección de dependencias y es el servidor de aplicaciones quien resuelve estas instancias, de igual manera decide cuando reservar memoria para dicha instancia y cuando liberarla.
- Facilidad al momento de integrarse con un framework Web.
- Al controlar la transaccionabilidad el servidor de aplicaciones, es el quien decide cuando abrir y cerrar la conexión a la base de datos.
- Está diseñado en interfaces, por lo que los desarrolladores pueden utilizarlas en diferentes partes del proyecto, reutilizando el código.
- Puede ejecutarse en cualquier contenedor Web o en una aplicación de escritorio normal.

DESVENTAJAS

- La curva de aprendizaje depende de la experiencia del desarrollador en el lenguaje Java.
- La inyección de dependencias no se puede probar sino al momento de la ejecución.
- Configuración compleja ya que utiliza archivos XML.

COMPONENTES DE SPRING

Módulo CORE O Núcleo.- La parte fundamental del framework es el módulo Core o "Núcleo", este provee toda la funcionalidad de Inyección de Dependencias permitiéndole administrar la funcionalidad del contenedor de beans. El concepto básico de este módulo es el BeanFactory, que implementa el patrón de diseño Factory eliminando la necesidad de crear singletons programáticamente permitiéndole desligar la configuración y especificación de las dependencias de la lógica de programación.

Encima del módulo core se encuentra el módulo Context (Contexto), el cual provee de herramientas para acceder a los beans de una manera elegante, similar a un registro JNDI. El paquete de contexto hereda sus características del paquete de beans y añade soporte para mensajería de texto, como son resource bundles (para internacionalización), propagación de eventos, carga de recursos y creación transparente de contextos por contenedores (como el contenedor de servlets, por ejemplo).

El paquete DAO.- Provee una capa de abstracción de JDBC que elimina la necesidad de teclear código JDBC tedioso y redundante así como el parseo de códigos de error específicos de cada proveedor de base de datos. También, el paquete JDBC provee de una manera de administrar transacciones tanto declarativas como programáticas, no solo para clases que implementen interfaces especiales, pero para todos sus POJOs.

El paquete ORM.- Provee capas de integración para APIs de mapeo objeto - relacional, incluyendo, JDO, Hibernate e iBatis. Usando el paquete ORM es posible usar esos mapeadores en conjunto con otras características que Spring ofrece, como la administración de transacciones.

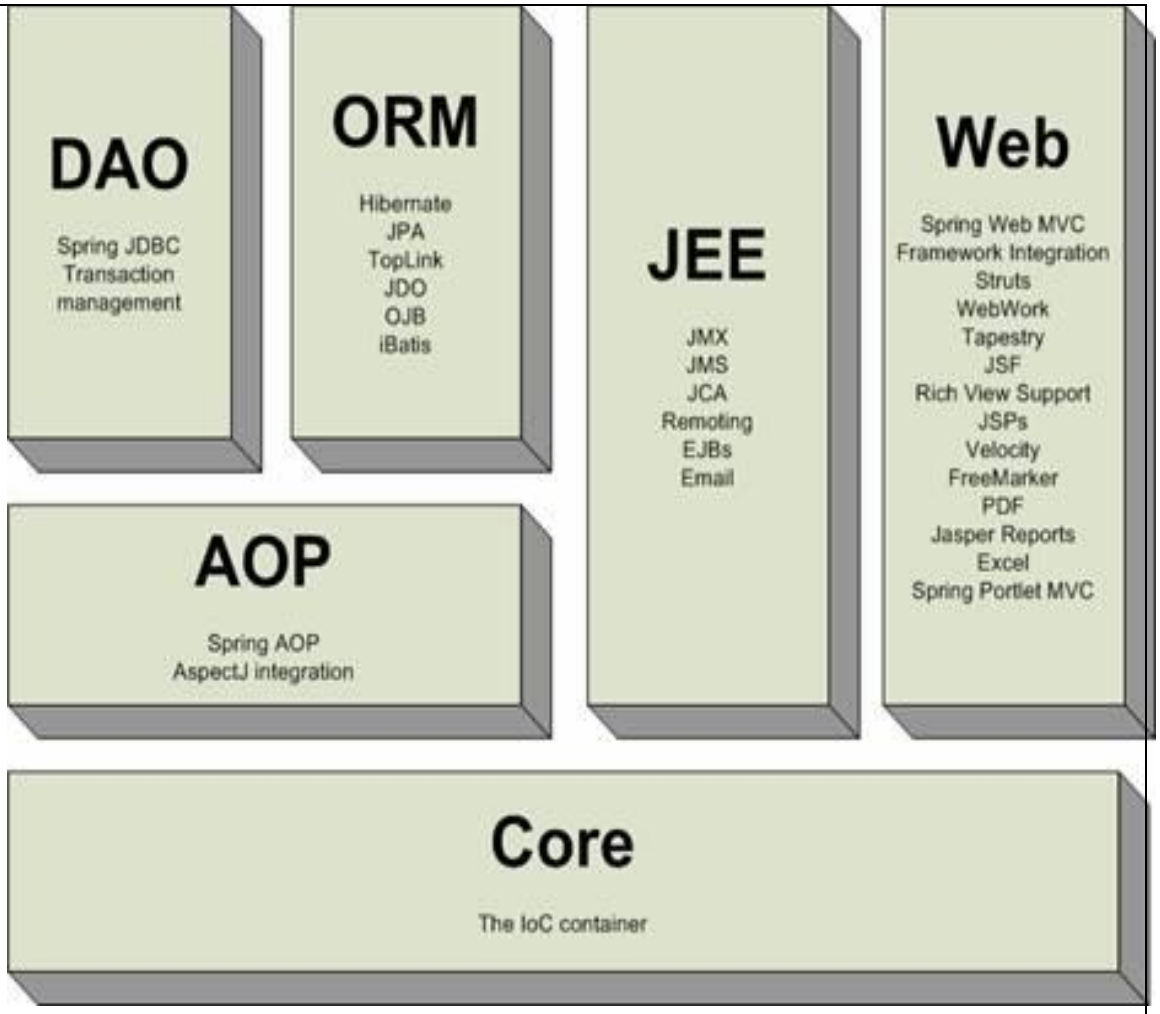


Figura 3 Diagrama de Componentes de Spring

Fuente: [WEB14]

El paquete AOP.- Provee una implementación de programación orientada a aspectos compatible con AOP Alliance, permitiendo definir pointcuts e interceptores de métodos para desacoplar el código de una manera limpia implementando funcionalidad que por lógica y claridad debería estar separada. Usando metadatos a nivel de código fuente se pueden incorporar diversos tipos de información y comportamiento al código, un poco similar a los atributos de .NET

El paquete Web.- Provee características básicas de integración orientadas a la web, como funcionalidad multipartes (para realizar la carga de archivos), inicialización de contextos mediante servlet listeners y un contexto de aplicación orientado a web. Cuando se usa Spring junto con WebWork o Struts, este es el paquete que te permite una integración sencilla.

Spring Web MVC

El Modelo Vista Controlador (MVC) es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. El patrón MVC se ve frecuentemente en aplicaciones web, donde la vista es la página HTML y el código que provee de datos dinámicos a la página, el modelo es el Sistema de Gestión de Base de Datos y el controlador representa la Lógica de negocio.

Descripción del patrón

Modelo: Esta es la representación específica de la información con la cual el sistema opera. La lógica de datos asegura la integridad de estos y permite derivar nuevos datos; por ejemplo, no permitiendo comprar un número de unidades negativo, calculando si hoy es el cumpleaños del usuario o los totales, impuestos o importes en un carrito de la compra.

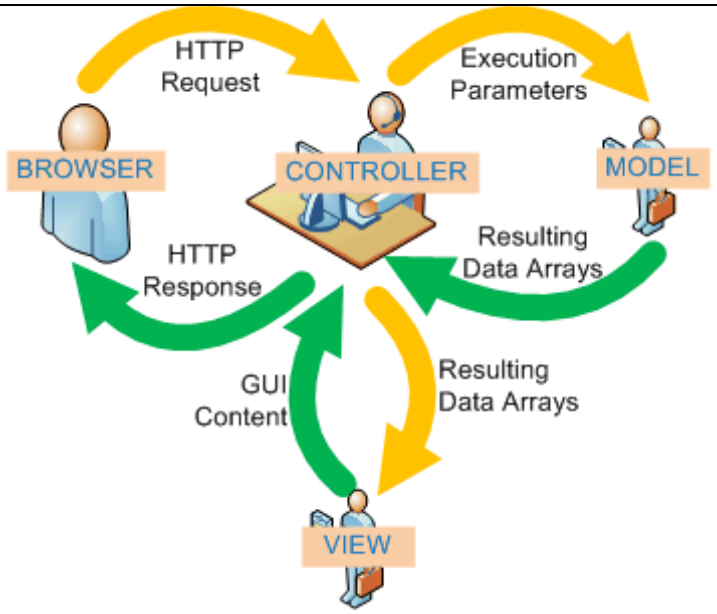


Figura 4 Patrón MVC

Fuente: [WEB09]

Vista: Este presenta el modelo en un formato adecuado para interactuar, usualmente la interfaz de usuario.

Controlador: Este responde a eventos, usualmente acciones del usuario e invoca cambios en el modelo y probablemente en la vista.

Muchos sistemas informáticos utilizan un Sistema de Gestión de Base de Datos para gestionar los datos. En MVC corresponde al modelo.

Aunque se pueden encontrar diferentes implementaciones de MVC, el flujo que sigue el control generalmente es el siguiente:

- El usuario interactúa con la interfaz de usuario de alguna forma (por ejemplo, el usuario pulsa un botón, enlace)

- El controlador recibe (por parte de los objetos de la interfaz-vista) la notificación de la acción solicitada por el usuario. El controlador gestiona el evento que llega, frecuentemente a través de un gestor de eventos (handler) o callback.

- El controlador accede al modelo, actualizándolo, posiblemente modificándolo de forma adecuada a la acción solicitada por el usuario (por ejemplo, el controlador actualiza el carro de la compra del usuario). Los controladores complejos están a menudo estructurados usando un patrón de comando que encapsula las acciones y simplifica su extensión.

- El controlador delega a los objetos de la vista la tarea de desplegar la interfaz de usuario.

- La vista obtiene sus datos del modelo para generar la interfaz apropiada para el usuario donde se refleja los cambios en el modelo (por ejemplo, produce un listado del contenido del carro de la compra).

- El modelo no debe tener conocimiento directo sobre la vista. Sin embargo, el patrón de observador puede ser utilizado para proveer cierta indirección entre el modelo y la vista, permitiendo al modelo notificar a los interesados de cualquier cambio. Un objeto vista puede registrarse con el modelo y esperar a los cambios, pero aun así el modelo en sí mismo sigue sin saber nada de la vista. El controlador no pasa objetos de dominio (el modelo) a la vista aunque puede dar la orden a la vista para que se actualice.

- La interfaz de usuario espera nuevas interacciones del usuario, comenzando el ciclo nuevamente.

Spring Web es el componente específico para el desarrollo de aplicaciones web en Spring. El desarrollo de aplicaciones web con este módulo se realiza utilizando el patrón MVC y puede usarse conjuntamente con todas las funcionalidades que proporciona el framework. Algunas características de Spring Web són:

- Proporciona una jerarquía de controladores para ser usados en diferentes escenarios.
- Su libreria de tags JSP permite la escritura de las páginas JSP mucho más clara y sencilla.
- Mapeador de peticiones muy configurable.
- Permite trabajar con cualquier tecnología en la vista como JSP/JSTL, XSL, Freemarker, Velocity o PDF.
- Se puede integrar con otros frameworks web como Tapestry o Struts.

Que es Struts?

Struts es una herramienta de soporte para el desarrollo de aplicaciones Web bajo el patrón MVC bajo la plataforma Java EE (Java Enterprise Edition). Struts se desarrollaba como parte del proyecto Jakarta de la Apache Software Foundation, pero actualmente es un proyecto independiente conocido como Apache Struts.

Struts permite reducir el tiempo de desarrollo. Su carácter de "software libre" y su compatibilidad con todas las plataformas en las que Java Enterprise esté disponible lo convierten en una herramienta altamente disponible.

Características

Provee una API con sus métodos y sus clases, las cuales tienes que extender para implementar la lógica de negocio. Esto hace que tengas que tener un conocimiento aceptable de dichas clases, sus responsabilidades y sus interacciones, lo que retrasa la hora de empezar a desarrollar. Sin embargo no presenta un modelo demasiado complicado y gracias a la cantidad de documentación se hace más fácil.

Debido a su gran difusión existen innumerables artículos, foros y similar con información sobre su funcionamiento, integración y cualquier problema que pueda surgir durante un desarrollo basado en Struts.

En cuanto a madurez puede decirse que es el líder indiscutible. Ha sido el estándar de facto durante años, y ha sido usado por grandes y pequeñas empresas en multitud de proyectos.

La extensibilidad es uno de los puntos sus fuertes. Struts no fue diseñado para ofrecer una solución web completa, si no como un núcleo que define la comunicación entre la capa de presentación y negocio. Es fácil añadirle extensiones para manejar la seguridad, el flujo de páginas, plantillas de vistas… y hacerle cooperar con otros frameworks como hibernate o Spring.

La configuración está basada 100% en archivos XML y prácticamente centralizada en el fichero de configuración principal: struts-config.xml. Para aplicaciones grandes este fichero suele volverse inmanejable.

En Struts no viene integrado Ajax por defecto, cosa lógica ya que Struts es mucho anterior a Ajax.

Sin embargo es posible su integración, pero es laboriosa, ya que requiere reescribir código en la capa de presentación y negocio.

Struts no tiene Mapeo Objeto Relacional, se centra en las capas de negocio y presentación, pero debido a su diseño es fácil hacerlo cooperar con soluciones ORM.

Las últimas versiones de Struts vienen con la extensión Struts-Tiles integrada. Este es un sistema de plantillas que sigue el patrón Composite View . Esto consiste en utilizar vistas compuestas que se componen de varias subvistas atómicas. Cada componente de la plantilla se podría incluir dinámicamente dentro del total y la distribución de la página se maneja independientemente del contenido.

Ventajas

- Promueve la reutilización de código java y de código abstracto en JSP. Esto permite una integración agradable en la herramienta de desarrollo basada en JSP que permite autoría con tags.

- Struts provee un punto de partida si estás comenzando a aprender la tecnología de tags JSP.

- Al ser Software Libre está disponible el código y todos pueden usar las revisiones del código.

- Divide y vencerás es una forma agradable de resolver el problema y hacerlo manejable. Por su puesto, el problema se hace más complejo y necesita más manejo.

Desventajas

- El framework está sufriendo una rápida cantidad de cambios. Han ocurrido una gran cantidad de cambios desde Struts 0.5 y 1.0. Se debe descargar la versión más reciente para evitar métodos desaprobados. Se deben modificar muchas veces los códigos a causa de los cambios en Struts.

- Una de las librerías de tags de Struts es el Tag Logic, el cual maneja generación de condicionales de salida, pero no previene al desarrollador de los problemas con el código Java. Cual sea el tipo de framework que decidas usar, debes entender el entorno en el cual se despliega y se mantiene. Pero es más fácil decirlo que hacerlo.

- Separar el problema en partes introduce complejidad. Con los constantes cambios ocurriendo, esto puede ser frustrante a veces.

Que es JBoss Seam?

Es un framework desarrollado por JBoss, una división de Red Hat. El líder del proyecto es Gavin King, también autor del framework para mapeo objeto relacional Hibernate. Combina las dos tecnologías Enterprise JavaBeans EJB3 y Java Server Faces JSF. Se puede acceder a cualquier componente EJB desde la capa de presentación refiriéndote a él mediante su nombre de componente seam.

Características

JBoss Seam sigue un modelo de programación muy sencillo y uniforme, basado en componentes y anotaciones en todas las capas de la aplicación. En ningún momento hace falta extender una clase ni implementar un interfaz, no tienes que aprenderte

ningún API, simplemente un par de conceptos. Esto hace que la curva de aprendizaje sea muy suave.

JBoss Seam en la distribución de Seam se incluyen varias aplicaciones de ejemplo que cubren casi todas las funcionalidades del framework. El tutorial oficial es muy extenso. En él se detallan todos los aspectos del framework con códigos de ejemplo. Explica los conceptos, las diferentes configuraciones posibles, las anotaciones, la integración con otras tecnologías, etc.

En el diseño de JBoss Seam han participado pesos pesados del desarrollo de software como Gavi King (creador de Hibernate) y se asienta sobre principios que han popularizado frameworks como RubyOnRails o Spring. Además, es una de los frameworks que se están tomando como referencia para futuros estándares como WebBeans.

El diseño basado en componentes le permite integrarse con numerosas tecnologías tan dispares como Google Web Toolkit, Spring, Groovy… además de las propias de JBoss que vienen integradas por defecto.

La configuración puede realizarse en su práctica totalidad mediante anotaciones, lo que ayuda a simplificar esta tediosa tarea. Aún así, hay una pequeña cantidad de configuración que debe hacerse con XML (como la relativa a JSF). Afortunadamente, esta configuración es autogenerada por la herramienta SeamGen. En caso de no hacer uso de ella, la configuración puede copiarse de las aplicaciones de ejemplo. Además, el framework sigue la filosofía de “configuración por omisión”: Todos los componentes vienen con un comportamiento por defecto que solo hace falta redefinir si queremos personalizarlos. También permite la configuración de los componentes internos del framework a través del archivo seam.properties.

El diseño de JBoss Seam ha sido concebido con Ajax en mente. Es capaz de manejar peticiones simultáneas de distintos usuarios preservando las condiciones de aislamiento e integridad de los datos. El sistema de cache evita la sobrecarga de tráfico con la base de datos que afecta a muchas aplicaciones que usan Ajax. El soporte en la parte del cliente está a cargo de la implementación de JSF que se elija. En cualquier caso, mediante la librería Ajax4jsf de JBoss se pueden añadir funcionalidades Ajax a los componentes ya existentes.

Proporciona soporte completo para las dos arquitecturas de persistencia más populares: hibernate 3 y el Java Persistence API introducido con EJB 3.0.

A través de el contexto de conversación y los componentes con estado asociados a él, es capaz de manejar eficientemente las operaciones transaccionales e implementar el patrón de carga perezosa (LazyLoad) sin los errores presentes en otras soluciones ORM.

Al ser POJOs todos los componentes, los test de unidad son triviales. También se proporcionan anotaciones para facilitar la recreación del entorno.

Para los test de integración proporciona un contenedor de EJBs y declaración de objetos mock mediante anotaciones. La herramienta SeamGen crea automáticamente test de unidad estándar y test TestNG que simulan peticiones y respuestas JSF para cada acción mediante scripting.

Viene integrado por defecto con Facelets. Facelets es un sistema de plantillas para JSF. Permite definir la vista mediante xml en forma de árbol de manera que se pueden definir componentes como composición de otros componentes. Entre sus características destacar que soporta el lenguaje EL, composición a partir de diferentes archivos, decorators y reporte de errores indicando la etiqueta/linea precisa.

Ventajas

- Ofrece una solución completa para el desarrollo web que abarca todas las capas, desde la de presentación hasta la de integración.

- Posee la habilidad de integrar tecnologías de diferentes ámbitos a través de la inyección de dependencias.

- A eso se le agrega un trabajo más fino de los contextos, tenemos más contextos lo que no quiere decir que ocupemos más memoria, si no que todo lo contrario vamos a tener más opciones para elegir donde guardar nuestras instancias y estas van a permanecer en memoria el tiempo indicado.

Desventajas

- Al generar mucho código transparente para el usuario, se dificulta la ubicación y resolución de problemas en dicho código.

- Los parámetros que construyen el modelo podrían no ser los adecuados para el caso específico que se busca programar, cambiar estos parámetros resulta demasiado complicado.

- No existe el soporte suficiente para la capa de negocio, y sus archivos xml son demasiados extensos.

CUADRO COMPARATIVO ENTRE STRUTS, SPRING Y JBOSS SEAM

A continuación se detallan las semejanzas y diferencias que existen entre las 3 herramientas.

	Struts	**Spring**	**JBoss Seam**
Curva de Aprendizaje	Dura necesita conocimiento aceptable de las clases del API, sus responsabilidades y sus interacciones.	Dura para usuarios sin experiencia en JEE.	Suave
Documentación	Innumerables artículos, foros y similares.	Buena documentación oficial y una nutrida comunidad.	Tutorial oficial y varios ejemplos incluidos en la distribución.
Madurez	Ha sido el estándar de facto durante años, y ha sido usado por grandes y pequeñas empresas en multitud de proyectos.	Una arquitectura sólida y, sobre todo, muy flexible, capaz de adaptarse a los requerimientos de proyectos grandes y pequeños.	Se asienta sobre principios que han popularizado frameworks como RubyOnRails o Spring.

	Struts	Spring	JBoss Seam
Extensibilidad	No fue diseñado para ofrecer una solución web completa, si no como un núcleo que define la comunicación entre la capa de presentación y negocio	Está expresamente concebido para que deba ser extensible y acoplable con otros frameworks.	El diseño basado en componentes le permite integrarse con numerosas tecnologías.
Configuración	Está basada 100% en archivos XML y prácticamente centralizada en el fichero de configuración principal: struts-config.xml	La configuración de Spring está basada en XML. El no tener anotaciones le hace independiente de JEE.	Puede realizarse en su práctica totalidad mediante anotaciones
Ajax	No posee soporte.	Posee soporte.	Posee soporte.
Mapeo Objeto Relacional	No por cuanto se centra en las capas de negocio y presentación.	Proporciona integración con varias implementaciones ORM (SpringDAO, Hibernate, JPA, Ibatis, etc.)	Soporte completo para: Hibernate y JPA únicamente.
Testeabilidad	No provee de un mecanismo nativo.	Proporciona un estupendo marco para el testing.	Al ser POJOs todos los componentes, los test de unidad son triviales.

	Struts	Spring	JBoss Seam
Sistema de Plantillas	Utiliza la extensión Struts-Tiles.	Permite elegir la tecnología a utilizar para la vista (JSP, JSF, velocity, PDF, ..)	Viene integrado por defecto con Facelets.

Tabla 2 CUADRO COMPARATIVO ENTRE STRUTS, SPRING Y JBOSS SEAM

Fuente: Propia

Los resultados de la comparación muestran que Spring Framework es especialmente preferible para usarse en compañías pequeñas, en conjunto con otros productos Open-Source. Es un framework muy simple, conveniente y flexible, pero al mismo tiempo muy poderoso. Se recomienda el uso de Spring en los casos donde un contenedor de aplicaciones pesado no es necesario.

2.4 PostgreSQL

PostgreSQL es un sistema de gestión de base de datos relacional orientada a objetos y libre, publicado bajo la licencia BSD.

Es de código abierto, el desarrollo de PostgreSQL no es manejado por una empresa y/o persona, sino que es dirigido por una comunidad de desarrolladores que trabajan de forma desinteresada, altruista, libre y/o apoyada por organizaciones comerciales.

Características

Algunas de sus principales características son, entre otras:

Alta concurrencia.- Mediante un sistema denominado MVCC (Acceso concurrente multiversión, por sus siglas en inglés) PostgreSQL permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos. Cada usuario obtiene una visión consistente de lo último a lo que se le hizo commit.

Esta estrategia es superior al uso de bloqueos por tabla o por filas común en otras bases, eliminando la necesidad del uso de bloqueos explícitos.

Amplia variedad de tipos nativos

PostgreSQL provee nativamente soporte para:
Números de precisión arbitraria.
Texto de largo ilimitado.
Figuras geométricas (con una variedad de funciones asociadas).
Direcciones IP (IPv4 e IPv6).
Bloques de direcciones estilo CIDR.
Direcciones MAC.
Arrays.

Adicionalmente los usuarios pueden crear sus propios tipos de datos, los que pueden ser por completo indexables gracias a la infraestructura GiST de PostgreSQL. Algunos ejemplos son los tipos de datos GIS creados por el proyecto PostGIS.

Otras características

Claves ajenas también denominadas Llaves ajenas o Claves Foráneas (foreign keys).

Disparadores (triggers): Un disparador o trigger se define como una acción específica que se realiza de acuerdo a un evento, cuando éste ocurra dentro de la base de datos. En PostgreSQL esto significa la ejecución de un procedimiento almacenado basado en una determinada acción sobre una tabla específica. Ahora todos los disparadores se definen por seis características:

- El nombre del disparador o trigger
- El momento en que el disparador debe arrancar
- El evento del disparador deberá activarse sobre...
- La tabla donde el disparador se activará
- La frecuencia de la ejecución
- La función que podría ser llamada

Entonces combinando estas seis características, PostgreSQL le permitirá crear una amplia funcionalidad a través de su sistema de activación de disparadores (triggers).

- Vistas.
- Integridad transaccional.
- Herencia de tablas.
- Tipos de datos y operaciones geométricas.
- Soporte para transacciones distribuidas. Permite a PostgreSQL integrase en un sistema distribuido formado por varios recursos (p.ej, una base de datos PostgreSQL, otra Oracle, una cola de mensajes IBM MQ JMS y un ERP SAP) gestionado por un servidor de aplicaciones donde el éxito ("commit") de la transacción globlal es el resultado del éxito de las transacciones locales.

Funciones

Bloques de código que se ejecutan en el servidor. Pueden ser escritos en varios lenguajes, con la potencia que cada uno de ellos da, desde las operaciones básicas de programación, tales como bifurcaciones y bucles, hasta las complejidades de la programación orientada a objetos o la programación funcional.

Los disparadores (triggers en inglés) son funciones enlazadas a operaciones sobre los datos.

PostgreSQL soporta funciones que retornan "filas", donde la salida puede tratarse como un conjunto de valores que pueden ser tratados igual a una fila retornada por una consulta (query en inglés).

Las funciones pueden ser definidas para ejecutarse con los derechos del usuario ejecutor o con los derechos de un usuario previamente definido. El concepto de funciones, en otros DBMS, son muchas veces referidas como "procedimientos almacenados".

Las características positivas:

- Soporta transacciones y desde la version 7.0, claves ajenas (con comprobaciones de integridad referencial).
- Tiene mejor soporte para triggers y procedimientos en el servidor.
- Incorpora una estructura de datos array.
- Incluye herencia entre tablas (aunque no entre objetos, ya que no existen), por lo que a este gestor de bases de datos se le incluye entre los gestores objeto-relacionales.

- Se pueden realizar varias opreraciones al mismo tiempo sobre la misma tabla sin necesidad de bloquearla.
- Posee una gran escalabilidad. Es capaz de ajustarse al número de CPUs y a la cantidad de memoria que posee el sistema de forma óptima, haciéndole capaz de soportar una mayor cantidad de peticiones simultáneas de manera correcta (en algunos benchmarks se dice que ha llegado a soportar el triple de carga de lo que soporta MySQL).
- Implementa el uso de rollback's, subconsultas y transacciones, haciendo su funcionamiento mucho más eficaz, y ofreciendo soluciones en campos en las que MySQL no podría.
- Tiene la capacidad de comprobar la integridad referencial, así como también la de almacenar procedimientos en la propia base de datos, equiparándolo con los gestores de bases de datos de alto nivel, como puede ser Oracle.

Desventajas:

- Consume gran cantidad de recursos.
- Tiene un límite de 8K por fila, aunque se puede aumentar a 32K, con una disminución considerable del rendimiento.
- Es de 2 a 3 veces más lento que MySQL.

MySQL

Es un sistema de gestión de bases de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones, de Oracle Corporation desde abril de 2009— desarrolla MySQL como software libre en un esquema de licenciamiento dual.

Por un lado se ofrece bajo la GNU GPL para cualquier uso compatible con esta licencia, pero para aquellas empresas que quieran incorporarlo en productos privativos deben comprar a la empresa una licencia específica que les permita este uso. Está desarrollado en su mayor parte en ANSI C.

Ventajas

- Es evidente que la gran mayoría de gente usa este gestor en Internet, por lo que encontrar opiniones favorables no ha resultado en absoluto complicado:

- Sin lugar a duda, lo mejor de MySQL es su velocidad a la hora de realizar las operaciones, lo que le hace uno de los gestores que ofrecen mayor rendimiento.

- Tiene una probabilidad muy reducida de corromper los datos, incluso en los casos en los que los errores no se produzcan en el propio gestor, sino en el sistema en el que está.

Desventajas

- Carece de soporte para transacciones, rollback's y subconsultas.

- El hecho de que no maneje la integridad referencial, hace de este gestor una solución pobre para muchos campos de aplicación, sobre todo para aquellos programadores que provienen de otros gestores que sí que poseen esta característica.

- No es viable para su uso con grandes bases de datos, a las que se acceda continuamente, ya que no implementa una buena escalabilidad.

Ambos gestores son excelentes pero si uno quiere migrar desde Oracle, Sybase, o Microsoft SQL Server, utilizar claves foráneas, vistas, subconsultas, la mejor alternativa es PostgreSQL aunque para el desarrollo Web y otro tipo de aplicaciones en las que la rapidez de respuesta y el bajo consumo de recursos son indispensables, lo mejor es MySQL. Cada uno de estos gestores posee características que los convierten en una gran opción en su respectivo campo al momento de elegir ya que fueron concebidos para una determinada implementación, por lo que en este caso utilizaremos PostgreSQL por la potencia que posee.

2.5 HTML y CSS

HTML (HyperText Markup Language)

Es un lenguaje muy sencillo que permite describir hipertexto, es decir, texto presentado de forma estructurada y agradable, con *enlaces* (*hyperlinks*) que conducen a otros documentos o fuentes de información relacionadas, y con *inserciones* multimedia (gráficos, sonido...) La descripción se basa en especificar en el texto la estructura lógica del contenido (títulos, párrafos de texto normal, enumeraciones, definiciones, citas, etc) así como los diferentes efectos que se quieren dar (especificar los lugares del documento donde se debe poner cursiva, negrita, o un gráfico determinado) y dejar que luego la presentación final de dicho hipertexto se realice por un programa especializado (como Mosaic, o Netscape).

Cómo especificar efectos del texto

La mayoría de los efectos se especifican de la misma forma: rodeando el texto que se quiere marcar entre dos *etiquetas* o *directivas* (*tags*, en inglés), que definen el efecto o unidad lógica que se desea. Las etiquetas están formadas por determinados códigos metidos entre los signos < y >, y con la barra / cuando se trata de la segunda etiqueta de un efecto (la de cierre).

Por ejemplo: <efecto> para abrir y </efecto> para cerrar. Ciertas directivas sólo se ponen una vez en el lugar del texto donde queramos que aparezca el efecto concreto. Esto es lo que ocurre, por ejemplo, cuando queremos poner un gráfico, caso en el que se usa algo parecido a <poner_gráfico_aquí> (más adelante ya veremos la directiva concreta que se utiliza).

A veces es necesario ofrecer datos adicionales en una directiva. Por ejemplo, cuando se define un *hiperenlace* hay que especificar su destino. Para ello se incluyen parámetros en la directiva inicial (la de apertura), de la siguiente forma: <efecto parametro1 parametro2 ...>. La directiva de cierre, caso de ser necesaria, queda como antes: </efecto>.

Estructura básica de un documento html: Comienza con la etiqueta <html>, y termina con </html>. Dentro del documento (entre las etiquetas de principio y fin de html), hay dos zonas bien diferenciadas: el *encabezamiento*, delimitado por <head> y

</head>, que sirve para definir diversos valores válidos en todo el documento; y el *cuerpo*, delimitado por <body> y </body>, donde reside la información del documento.

La única utilidad del encabezamiento en la que nos detendremos es la directiva <title>, que permite especificar el título de un documento HTML. Este título no forma parte del documento en sí: no aparece, por ejemplo, al principio del documento una vez que este se presenta con un programa adecuado, sino que suele servir como título de la ventana del programa que nos la muestra. Por ejemplo, en el encabezamiento se ha especificado:

<title> **HTML**</title>

El título que encabeza este texto se ha escrito con mayúsculas, para distinguirlo.

El cuerpo de un documento HTML contiene el texto que, con la presentación y los efectos que se decidan, se presentará ante el *hiperlector*. Dentro del cuerpo son aplicables todos los efectos mismos que se especifican exclusivamente a través de directivas. Esto quiere decir que los espacios, tabulaciones y retornos de carro que se introduzcan en el fichero fuente no tienen ningún efecto a la hora de la presentación final del documento.

En resumen, la estructura básica de un documento HTML queda de la forma siguiente:

<html>
<head>
<title>
</title>
</head>
<body>
</body>
</html>

Etiqueta <html>: es básica en un documento HTML, ya que denota que estamos escribiendo un documento de hipertexto con este lenguaje. Siempre debe ir al principio y al final de todo documento HTML.

Etiqueta <head>: En esta parte del documento aparece implícita la etiqueta <title>, etiqueta que veremos a continuación. Además, aquí vendrán dadas las etiquetas <meta> y <link>. Con la etiqueta <link> haremos un link a la hoja CSS en nuestra página HTML, de la siguiente manera:

<link rel="stylesheet" type="text/css" href="enlacerelativodelahojacss">

Etiqueta <title>: Aquí escribiremos el nombre que queremos que aparezca en la ventana del navegador cuando dicho navegador esté mostrando esta web, es decir, el nombre que aparecerá en la "barrita" correspondiente a este navegador que aparece en la barra de tareas. A resumidas cuentas, el título que mostrará el navegador.

Etiqueta <body>: "cuerpo". En efecto, este apartado es el cuerpo de la página, ya que a partir de aquí aparecerá, en un navegador, todo lo que escribamos en la página web. Aquí es donde introduciremos el texto, los párrafos, las listas, los enlaces externos o internos (es decir, a otra página o a la misma página, respectivamente), las tablas, los formularios, etc.

Características del lenguaje HTML

El lenguaje HTML es un estándar reconocido en todo el mundo y cuyas normas define un organismo sin ánimo de lucro llamado World Wide Web Consortium, más conocido como W3C. Como se trata de un estándar reconocido por todas las empresas relacionadas con el mundo de Internet, una misma página HTML se visualiza de forma muy similar en cualquier navegador de cualquier sistema operativo.

Reconocido universalmente y que permite publicar información de forma global". Desde su creación, el lenguaje HTML ha pasado de ser un lenguaje utilizado exclusivamente para crear documentos electrónicos a ser un lenguaje que se utiliza en muchas aplicaciones electrónicas como buscadores, tiendas online y banca electrónica.

HTML es un lenguaje utilizado únicamente para dar estructura a una página web. El estilo de la propia página web vendrá dado por un enlace a una hoja CSS (Hojas de Estilo en Cascada), de las que se hablará aparte. Es decir, que toda etiqueta que defina estilo y no estructura no se podrá considerar perteneciente al lenguaje HTML. Algunos ejemplos serían: <b>, <i> (Negrita y Cursiva).

CSS Cascading Style Sheets (*hojas de estilo en cascada*)

CSS se utiliza para dar estilo a documentos HTML y XML, separando el contenido de la presentación. Los Estilos definen la forma de mostrar los elementos HTML y XML. CSS permite a los desarrolladores Web controlar el estilo y el formato de múltiples páginas Web al mismo tiempo. Cualquier cambio en el estilo marcado para un elemento en la CSS afectará a todas las páginas vinculadas a esa CSS en las que aparezca ese elemento.

La idea que se encuentra detrás del desarrollo de CSS es separar la estructura de un documento de su presentación.

Es un mecanismo simple que describe cómo se va a mostrar un documento en la pantalla, o cómo se va a imprimir, o incluso cómo va a ser pronunciada la información presente en ese documento a través de un dispositivo de lectura. Esta forma de descripción de estilos ofrece a los desarrolladores el control total sobre estilo y formato de sus documentos.

CSS funciona a base de reglas, es decir, declaraciones sobre el estilo de uno o más elementos. Las hojas de estilo están compuestas por una o más de esas reglas aplicadas a un documento HTML o XML. La regla tiene dos partes: un selector y la declaración. A su vez la declaración está compuesta por una propiedad y el valor que se le asigne.

El modo de funcionamiento de las CSS consiste en definir, mediante una sintaxis especial, la forma de presentación que le aplicaremos a:

- Un sitio entero, de modo que se puede definir el diseño de todo el sitio de una sola vez.

- Un documento **HTML** o página, se puede definir la forma, en unas pocas líneas en la cabecera, a toda la página.

- Una porción del documento, aplicando estilos visibles a una parte de la página.

- Una etiqueta en concreto, llegando incluso a poder definir varios estilos diferentes para una sola etiqueta. Esto es muy importante ya que ofrece

potencia en nuestro diseño. Podemos definir, por ejemplo, varios tipos de párrafos: en rojo, en azul, con márgenes, sin ellos.

Cabe destacar que no todos los navegadores implementan las mismas funciones de hojas de estilos. Esto quiere decir que debemos de usar esta tecnología con cuidado, ya que muchos usuarios no podrán ver los formatos que apliquemos a las páginas con **CSS**. Un clásico es el Internet Explorer que no sigue los estándares de la **W3C** (World Wide Web Consortium). Por esto, es muy recomendable testear nuestro sitio en desarrollo en todos los exploradores importantes del mercado como: **Firefox** 2 y 3, **Explorer** 6 y 7, **Safari** y **Opera**.

La información de estilo puede ser adjuntada como un documento separado o en el mismo documento HTML. En este último caso podrían definirse estilos generales en la cabecera del documento o en cada etiqueta particular mediante el atributo "style".

Los tres tipos de estilos

CSS proporciona tres caminos diferentes para aplicar las reglas de estilo a una página Web:

Una hoja de estilo externa, es una hoja de estilo que está almacenada en un archivo diferente al archivo donde se almacena el código HTML de la página Web. Esta es la manera de programar más potente, porque separa completamente las reglas de formateo para la página HTML de la estructura básica de la página.

Una hoja de estilo interna, que es una hoja de estilo que está incrustada dentro de un documento HTML. (Va a la derecha dentro del elemento <head>.) De esta manera se obtiene el beneficio de separar la información del estilo del código HTML propiamente dicho. Se puede optar por copiar la hoja de estilo incrustada de una página a otra (esta posibilidad es difícil de ejecutar si se desea para guardar las copias sincronizadas). En general, la única vez que se usa una hoja de estilo interna, es cuando se quiere proporcionar alguna característica a una página Web en un simple fichero, por ejemplo, si se está enviando algo a la página Web.

Un estilo en línea, (inline) es un método para insertar el lenguaje de estilo de página directamente dentro de una etiqueta HTML. Esta manera de proceder no es totalmente adecuada. El incrustar la descripción del formateo dentro del documento

de la página Web, a nivel de código, se convierte en una manera larga, tediosa y poco elegante de resolver el problema de la programación de la página. Este modo de trabajo se podría usar de manera ocasional si se pretende aplicar un formateo con prisa, al vuelo. No es todo lo claro o estructurado que debería ser, pero funciona. Éste es el método recomendado para maquetar correos electrónicos en HTML.

Características: Principales características aportadas por CSS en contraposición a los elementos de visualización presentes en la especificación de HTML.

Estilo enriquecido

CSS permite la creación de documentos visualmente mucho más ricos que lo que HTML nunca permitirá. No en vano CSS está pensado única y exclusivamente para asistir al diseñador a la hora de dar estilo a un documento estructurado.

Fácil de utilizar

La utilización de hojas de estilo CSS hace que el diseñador pueda reducir sustancialmente su carga de trabajo al diseñar todo un site. Esto se debe a que CSS es capaz de centralizar ciertos efectos visuales que plasmemos en diversas secciones del site, en lugar de tenerlos diseminados por páginas y páginas del site.

Reutilización en múltiples páginas

Una hoja de estilo que recoja aspectos visuales comunes a varias páginas puede ser reutilizada en cualquier sección del site aprovechando dichos efectos ya definidos. De esta manera es sencillo generar un estilo general del web y mantenerlo así consistente para todas las páginas.

Así, si deseamos modificar un estilo que es común a todo el site, sólo necesitaríamos modificar una línea de nuestro fichero CSS (con la aproximación clásica que ofrece HTML, deberíamos modificar todas y cada una de las páginas).

Mantenibilidad

Los responsables de sitios en la Web pueden simplificar el mantenimiento y conservar un estilo y un efecto consistente a todo lo largo del sitio.

Ejemplo: si el color del fondo de las páginas de una organización cambia, sólo un archivo necesita ser modificado, ahorrando tiempo de edición y minimizando la posibilidad de errores.

Rendimiento de la red

CSS proporciona una compacta codificación para presentar los contenidos.
Generalmente disminuyen el tamaño del contenido.
Menos conexiones de la red tienen que ser abiertas.

Flexibilidad

Las CSS pueden ser aplicadas al contenido de varias maneras.

Característica clave: su capacidad de formar una cascada de estilos con la información de:

- la hoja de estilo predeterminada (aplicación del usuario)
- las hojas de estilo del usuario
- las hojas de estilo vinculadas
- el encabezamiento del documento los atributos de los elementos de la página

Ventajas de usar las hojas de estilo

Las ventajas de utilizar CSS (u otro lenguaje de estilo) son:

- Control centralizado de la presentación de un sitio web completo con lo que se agiliza de forma considerable la actualización del mismo.

- Los navegadores permiten a los usuarios especificar su propia hoja de estilo local, que será aplicada a un sitio web, con lo que aumenta considerablemente la accesibilidad. Por ejemplo, personas con deficiencias visuales pueden configurar su propia hoja de estilo para aumentar el tamaño del texto o remarcar más los enlaces.

- Una página puede disponer de diferentes hojas de estilo según el dispositivo que la muestre o, incluso, a elección del usuario. Por ejemplo, para ser

impresa, mostrada en un dispositivo móvil o ser "leída" por un sintetizador de voz.

- El documento HTML en sí mismo es más claro de entender y se consigue reducir considerablemente su tamaño (siempre y cuando no se utilice estilo en línea).

2.6 Metodología RUP

El RUP está basado en 6 principios clave que son los siguientes:

RUP, Proceso Unificado de Rational en ingles significa Rational Unified Process es un producto del proceso de ingeniería de software que proporciona un enfoque disciplinado para asignar tareas y responsabilidades dentro de una organización del desarrollo. Su meta es asegurar la producción del software de alta calidad que resuelve las necesidades de los usuarios dentro de un presupuesto y tiempo establecidos.

Adaptar el proceso

El proceso deberá adaptarse a las necesidades del cliente ya que es muy importante interactuar con él. Las características propias del proyecto u organización. El tamaño del mismo, así como su tipo o las regulaciones que lo condicionen, influirán en su diseño específico. También se deberá tener en cuenta el alcance del proyecto en un área subformal.

Equilibrar prioridades

Los requisitos de los diversos participantes pueden ser diferentes, contradictorios o disputarse recursos limitados. Debe encontrarse un equilibrio que satisfaga los deseos de todos. Gracias a este equilibrio se podrán corregir desacuerdos que surjan en el futuro.

Demostrar valor iterativamente

Los proyectos se entregan, aunque sea de un modo interno, en etapas iteradas. En cada iteración se analiza la opinión de los inversores, la estabilidad y calidad del

producto, y se refina la dirección del proyecto así como también los riesgos involucrados

Colaboración entre equipos

El desarrollo de software no lo hace una única persona sino múltiples equipos. Debe haber una comunicación fluida para coordinar requisitos, desarrollo, evaluaciones, planes, resultados, etc.

Elevar el nivel de abstracción

Este principio dominante motiva el uso de conceptos reutilizables tales como patrón del software, lenguajes 4GL o marcos de referencia (frameworks) por nombrar algunos. Esto evita que los ingenieros de software vayan directamente de los requisitos a la codificación de software a la medida del cliente, sin saber con certeza qué codificar para satisfacer de la mejor manera los requisitos y sin comenzar desde un principio pensando en la reutilización del código. Un alto nivel de abstracción también permite discusiones sobre diversos niveles y soluciones arquitectónicas. Éstas se pueden acompañar por las representaciones visuales de la arquitectura, por ejemplo con el lenguaje UML.

Enfocarse en la calidad

El control de calidad no debe realizarse al final de cada iteración, sino en todos los aspectos de la producción. El aseguramiento de la calidad forma parte del proceso de desarrollo y no de un grupo independiente.

Ciclo de vida

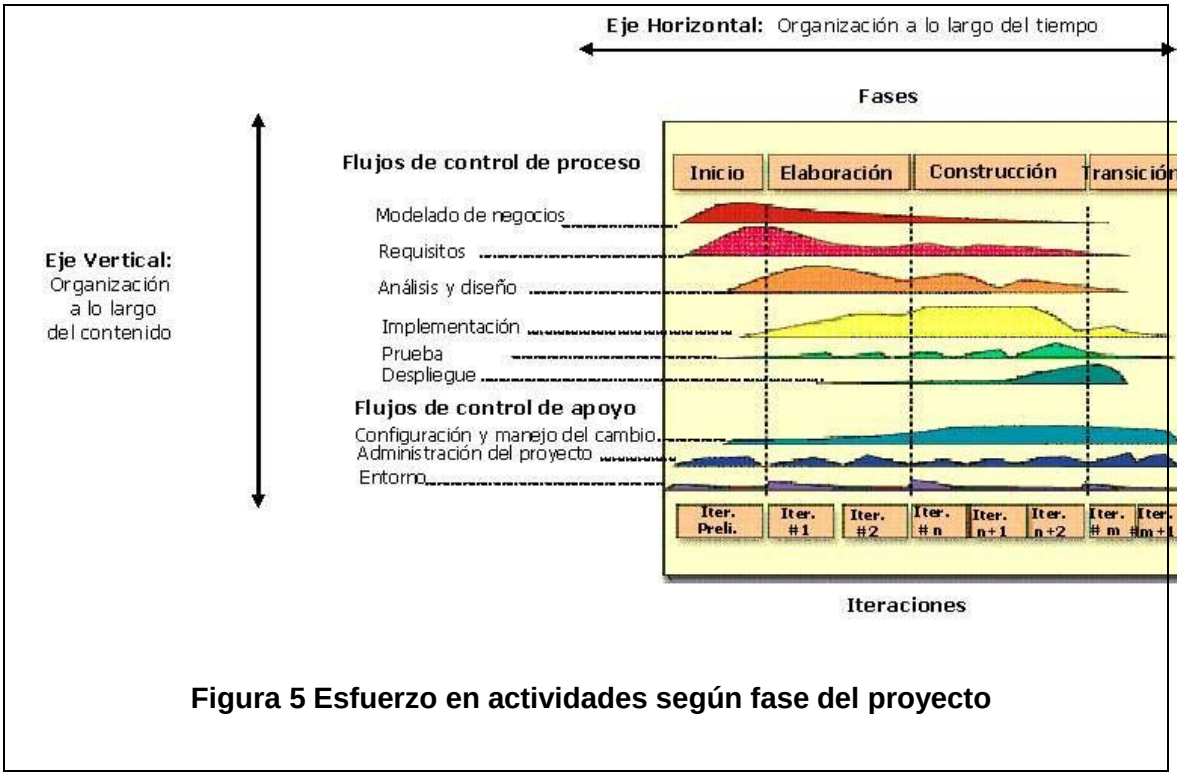


Figura 5 Esfuerzo en actividades según fase del proyecto

Fuente: [WEB04]

Esfuerzo en actividades según fase del proyecto

El ciclo de vida RUP es una implementación del Desarrollo en espiral. Fue creado ensamblando los elementos en secuencias semi-ordenadas. El ciclo de vida organiza las tareas en fases e iteraciones.

RUP divide el proceso en cuatro fases, dentro de las cuales se realizan varias iteraciones en número variable según el proyecto y en las que se hace un mayor o menor hincapié en las distintas actividades. En la Figura muestra cómo varía el esfuerzo asociado a las disciplinas según la fase en la que se encuentre el proyecto RUP.

Las primeras iteraciones (en las fases de Inicio y Elaboración) se enfocan hacia la comprensión del problema y la tecnología, la delimitación del ámbito del proyecto, la eliminación de los riesgos críticos, y al establecimiento de una baseline (Línea Base) de la arquitectura.

Durante la fase de inicio las iteraciones hacen mayor énfasis en actividades de modelado del negocio y de requisitos.

En la fase de elaboración, las iteraciones se orientan al desarrollo de la baseline de la arquitectura, abarcan más los flujos de trabajo de requisitos, modelo de negocios (refinamiento), análisis, diseño y una parte de implementación orientado a la baseline de la arquitectura.

En la fase de construcción, se lleva a cabo la construcción del producto por medio de una serie de iteraciones.

Para cada iteración se selecciona algunos Casos de Uso, se refina su análisis y diseño y se procede a su implementación y pruebas. Se realiza una pequeña cascada para cada ciclo. Se realizan tantas iteraciones hasta que se termine la implementación de la nueva versión del producto.

En la fase de transición se pretende garantizar que se tiene un producto preparado para su entrega a la comunidad de usuarios.

Como se puede observar en cada fase participan todas las disciplinas, pero que dependiendo de la fase el esfuerzo dedicado a una disciplina varía.

Características esenciales

Los autores de RUP destacan que el proceso de software propuesto por RUP tiene tres características esenciales: está dirigido por los Casos de Uso, está centrado en la arquitectura, y es iterativo e incremental.

- Forma disciplinada de asignar tareas y responsabilidades (quién hace qué, cuándo y cómo)
- Pretende implementar las mejores prácticas en Ingeniería de Software
- Desarrollo iterativo
- Administración de requisitos
- Uso de arquitectura basada en componentes
- Control de cambios
- Modelado visual del software
- Verificación de la calidad del software

Proceso dirigido por Casos de Uso

Se refiere a la utilización de los Casos de Uso para el desenvolvimiento y desarrollo de las disciplinas con los artefactos, roles y actividades necesarias. Los Casos de Uso son la base para la implementación de las fases y disciplinas del RUP.

Un Caso de Uso es una secuencia de pasos a seguir para la realización de un fin o propósito, y se relaciona directamente con los requerimientos, ya que un Caso de Uso es la secuencia de pasos que conlleva la realización e implementación de un Requerimiento planteado por el Cliente.

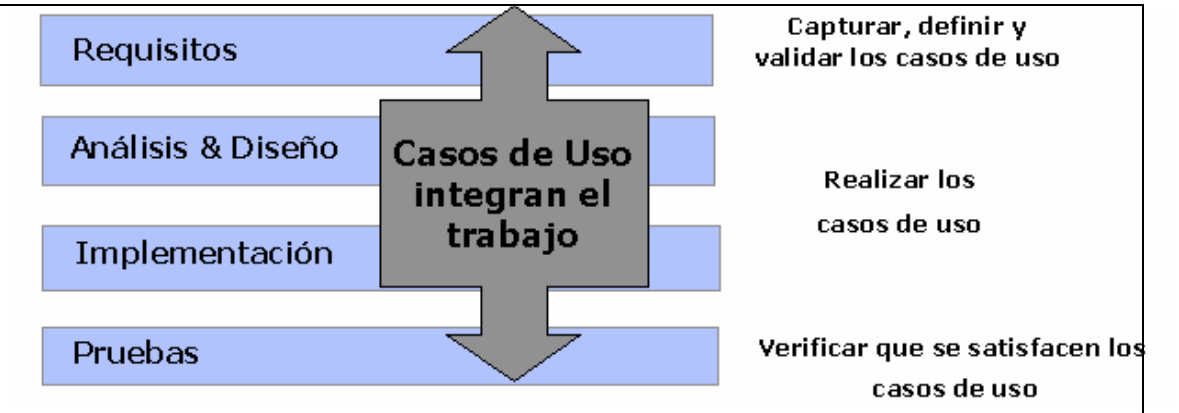


Figura 6 Los casos de integran el trabajo

Fuente: [WEB04]

Los Casos de Uso no sólo inician el proceso de desarrollo sino que proporcionan un hilo conductor, permitiendo establecer trazabilidad entre los artefactos que son generados en las diferentes actividades del proceso de desarrollo.

Basándose en los Casos de Uso se crean los modelos de análisis y diseño, luego la implementación que los lleva a cabo, y se verifica que efectivamente el producto implemente adecuadamente cada Caso de Uso. Todos los modelos deben estar sincronizados con el modelo de Casos de Uso.

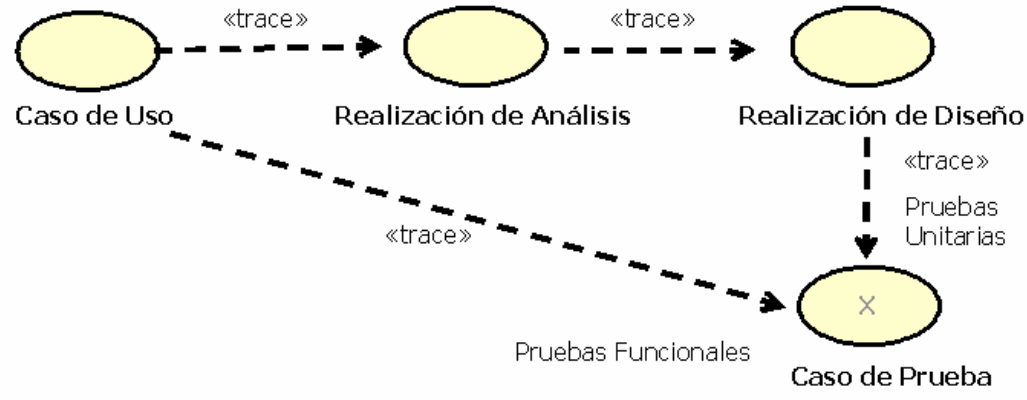


Figura 7 Trazabilidad a partir de los Casos de Uso

Fuente: [WEB04]

Proceso centrado en la arquitectura

La arquitectura de un sistema es la organización o estructura de sus partes más relevantes, lo que permite tener una visión común entre todos los involucrados (desarrolladores y usuarios) y una perspectiva clara del sistema completo, necesaria para controlar el desarrollo.

La arquitectura involucra los aspectos estáticos y dinámicos más significativos del sistema, está relacionada con la toma de decisiones que indican cómo tiene que ser construido el sistema y ayuda a determinar en qué orden. Además la definición de la arquitectura debe tomar en consideración elementos de calidad del sistema, rendimiento, reutilización y capacidad de evolución por lo que debe ser flexible durante todo el proceso de desarrollo. La arquitectura se ve influenciada por la plataforma software, sistema operativo, gestor de bases de datos, protocolos, consideraciones de desarrollo como sistemas heredados. Muchas de estas restricciones constituyen requisitos no funcionales del sistema.

En el caso de RUP además de utilizar los Casos de Uso para guiar el proceso se presta especial atención al establecimiento temprano de una buena arquitectura que no se vea fuertemente impactada ante cambios posteriores durante la construcción y el mantenimiento.

Cada producto tiene tanto una función como una forma. La función corresponde a la funcionalidad reflejada en los Casos de Uso y la forma la proporciona la arquitectura.

Existe una interacción entre los Casos de Uso y la arquitectura, los Casos de Uso deben encajar en la arquitectura cuando se llevan a cabo y la arquitectura debe permitir el desarrollo de todos los Casos de Uso requeridos, actualmente y en el futuro. Esto provoca que tanto arquitectura como Casos de Uso deban evolucionar en paralelo durante todo el proceso de desarrollo de software.

Se ilustra la evolución de la arquitectura durante las fases de RUP. Se tiene una arquitectura más robusta en las fases finales del proyecto. En las fases iniciales lo que se hace es ir consolidando la arquitectura por medio de baselines y se va modificando dependiendo de las necesidades del proyecto.

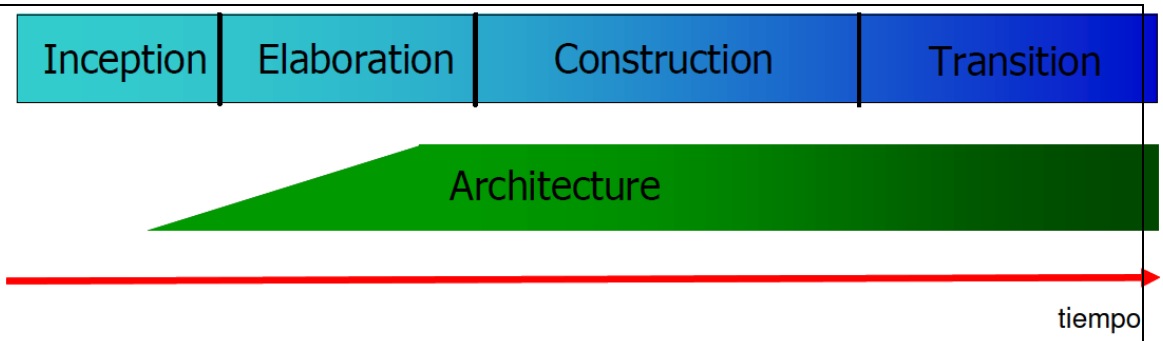


Figura 8 Evolución de la arquitectura del Sistema

Fuente: Propia

Es conveniente ver el sistema desde diferentes perspectivas para comprender mejor el diseño por lo que la arquitectura se representa mediante varias vistas que se centran en aspectos concretos del sistema, abstrayéndose de los demás.

Proceso iterativo e incremental

El equilibrio correcto entre los Casos de Uso y la arquitectura es algo muy parecido al equilibrio de la forma y la función en el desarrollo del producto, lo cual se consigue con el tiempo. Para esto, la estrategia que se propone en RUP es tener un proceso iterativo e incremental en donde el trabajo se divide en partes más pequeñas o mini proyectos. Permitiendo que el equilibrio entre Casos de Uso y arquitectura se vaya logrando durante cada mini proyecto, así durante todo el proceso de desarrollo. Cada mini proyecto se puede ver como una iteración (un recorrido más o menos completo a

lo largo de todos los flujos de trabajo fundamentales) del cual se obtiene un incremento que produce un crecimiento en el producto.

Una iteración puede realizarse por medio de una cascada como se muestra en la siguiente figura, se pasa por los flujos fundamentales (Requisitos, Análisis, Diseño, Implementación y Pruebas), también existe una planificación de la iteración, un análisis de la iteración y algunas actividades específicas de la iteración. Al finalizar se realiza una integración de los resultados con lo obtenido de las iteraciones anteriores.

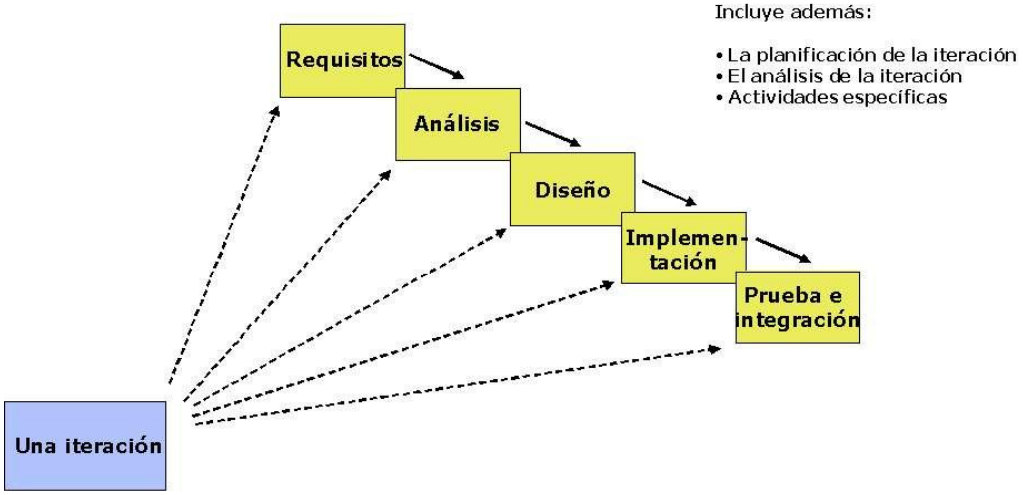


Figura 9 Una iteración RUP

Fuente: [WEB04]

El proceso iterativo e incremental consta de una secuencia de iteraciones. Cada iteración aborda una parte de la funcionalidad total, pasando por todos los flujos de trabajo relevantes y refinando la arquitectura. Cada iteración se analiza cuando termina. Se puede determinar si han aparecido nuevos requisitos o han cambiado los existentes, afectando a las iteraciones siguientes. Durante la planificación de los detalles de la siguiente iteración, el equipo también examina cómo afectarán los riesgos que aún quedan al trabajo en curso. Toda la retroalimentación de la iteración pasada permite reajustar los objetivos para las siguientes iteraciones. Se continúa con esta dinámica hasta que se haya finalizado por completo con la versión actual del producto.

Fases

El ciclo de vida consiste en una serie de ciclos, cada uno de los cuales produce una nueva versión del producto, cada ciclo está compuesto por fases y cada una de estas fases está compuesta por un número de iteraciones, estas fases son:

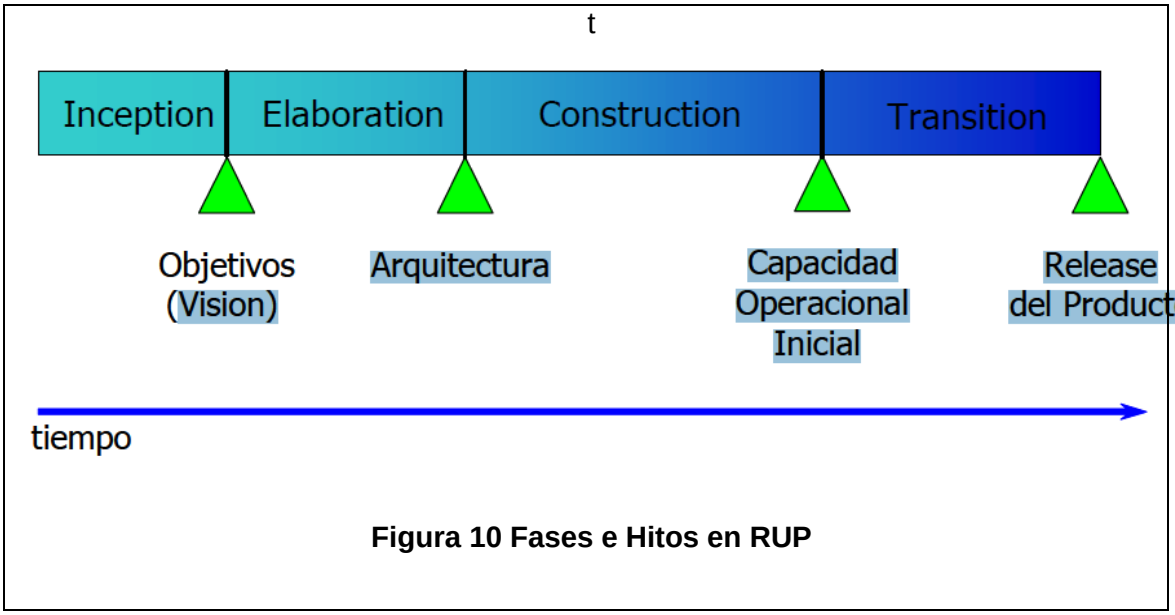


Figura 10 Fases e Hitos en RUP

Fuente: [WEB04]

Concepción, Inicio o Estudio de oportunidad

Define el ámbito y objetivos del proyecto
Se define la funcionalidad y capacidades del producto

Elaboración

Tanto la funcionalidad como el dominio del problema se estudian en profundidad
Se define una arquitectura básica
Se planifica el proyecto considerando recursos disponibles

Construcción

El producto se desarrolla a través de iteraciones donde cada iteración involucra tareas de análisis, diseño e implementación.

Las fases de estudio y análisis sólo dieron una arquitectura básica que es aquí refinada de manera incremental conforme se construye (se permiten cambios en la estructura).

Gran parte del trabajo es programación y pruebas
Se documenta tanto el sistema construido como el manejo del mismo
Esta fase proporciona un producto construido junto con la documentación

Transición

Se libera el producto y se entrega al usuario para un uso real
Se incluyen tareas de marketing, empaquetado atractivo, instalación, configuración, entrenamiento, soporte, mantenimiento, etc.
Los manuales de usuario se completan y refinan con la información anterior
Estas tareas se realizan también en iteraciones

Todas las fases no son idénticas en términos de tiempo y esfuerzo.
Aunque esto varía considerablemente dependiendo del proyecto, un ciclo de desarrollo inicial típico para un proyecto de tamaño mediano debe anticipar la distribución siguiente el esfuerzo y horario:

	Inicio	Elaboración	Construcción	Transición
Esfuerzo	5 %	20 %	65 %	10%
Tiempo Dedicado	10 %	30 %	50 %	10%

Figura 11 Distribución típicas de esfuerzo y tiempo

Fuente: [WEB04]

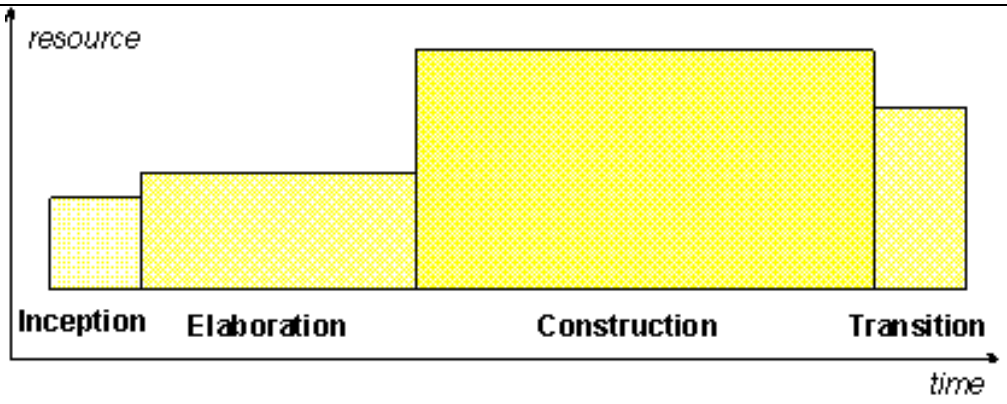


Figura 12 Distribución típicas de recursos humanos

Fuente: [WEB04]

En un ciclo evolutivo, las fases de concepción y elaboración serían considerablemente más pequeñas. Algunas herramientas que pueden automatizar una cierta porción del esfuerzo de la fase de Construcción pueden atenuar esto, haciendo que la fase de construcción sea mucho más pequeña que las fases de concepción y elaboración juntas. Este es precisamente el objetivo del trabajo.

Cada paso con las cuatro fases produce una generación del software. A menos que el producto "muera", se desarrollará nuevamente repitiendo la misma secuencia las fases de concepción, elaboración, construcción y transición, pero con diversos énfasis cada fase.

Disciplinas

Las disciplinas conllevan los flujos de trabajo, los cuales son una secuencia de pasos para la culminación de cada disciplina, estas disciplinas se dividen en dos grupos: las primarias y las de apoyo. Las primarias son las necesarias para la realización de un proyecto de software, aunque para proyectos no muy grandes se pueden omitir algunas; entre ellas se tienen:

Modelado del Negocio, Requerimientos, Análisis y Diseño, Implementación, Pruebas, Despliegue. Las de apoyo son las que como su nombre lo indica sirven de apoyo a las primarias y especifican otras características en la realización de un proyecto de software; entre estas se tienen: Entorno, Gestión del Proyecto, Gestión de Configuración y Cambios.

Modelado del negocio

Esta disciplina tiene como objetivos comprender la estructura y la dinámica de la organización, comprender problemas actuales e identificar posibles mejoras, comprender los procesos de negocio. Utiliza el Modelo de caso de uso del Negocio para describir los procesos del negocio y los clientes, el Modelo de Objetos del Negocio para describir cada caso de uso del Negocio con los Trabajadores, además utilizan los Diagramas de Actividad y de Clases.

Requerimientos

Esta disciplina tiene como objetivos establecer lo que el sistema debe hacer (Especificar Requisitos), definir los límites del sistema, y una interfaz de usuario, realizar una estimación del costo y tiempo de desarrollo. Utiliza el Modelo de caso de uso para modelar el Sistema que comprenden los casos de uso, Actores y Relaciones, además utiliza los diagramas de Estados de cada caso de uso y las especificaciones suplementarias.

Análisis y diseño

Esta disciplina define la arquitectura del sistema y tiene como objetivos trasladar requisitos en especificaciones de implementación, al decir análisis se refiere a transformar caso de uso en clases, y al decir diseño se refiere a refinar el análisis para poder implementar los diagramas de clases de análisis de cada caso de uso, los diagramas de colaboración de de cada caso de uso, el de clases de diseño de cada , el de secuencia de diseño de caso de uso, el de estados de las clases, el modelo de despliegue de la arquitectura.

Implementación

Esta disciplina tiene como objetivos implementar las clases de diseño como componentes, asignar los componentes a los nodos, probar los componentes individualmente, integrar los componentes en un sistema ejecutable (enfoque incremental). Utiliza el Modelo de Implementación, conjuntamente los Diagramas de Componentes para comprender cómo se organizan los Componentes y dependen unos de otros.

Pruebas

Esta disciplina tiene como objetivos verificar la integración de los componentes (prueba de integración), verificar que todos los requisitos han sido implementados (pruebas del sistema), asegurar que los defectos detectados han sido resueltos antes de la distribución

Despliegue

Esta disciplina tiene como objetivos asegurar que el producto está preparado para el cliente, proceder a su entrega y recepción por el cliente. En esta disciplina se realizan las actividades de probar el software en su entorno final (Prueba Beta), empaquetarlo, distribuirlo e instalarlo, así como la tarea de enseñar al usuario.

Gestión y configuración de cambios

Es esencial para controlar el número de artefactos producidos por la cantidad de personal que trabajan en un proyecto conjuntamente. Los controles sobre los cambios son de mucha ayuda ya que evitan confusiones costosas como la compostura de algo que ya se había arreglado etc., y aseguran que los resultados de los artefactos no entren en conflicto con algunos de los siguientes tipos de problemas:

- **Actualización simultánea: Es** la actualización de algo elaborado con anterioridad, sin saber que alguien más lo está actualizando.

- **Notificación limitada:** Al realizar alguna modificación, no se deja información sobre lo que se hizo, por lo tanto no se sabe quien, como, y cuando se hizo.

- **Versiones múltiples:** No saber con exactitud, cual es la última versión, y al final no se tiene un orden sobre que modificaciones se han realizado a las diversas versiones.

Gestión del proyecto

Su objetivo es equilibrar los objetivos competitivos, administrar el riesgo, y superar restricciones para entregar un producto que satisface las necesidades de ambos

clientes con éxito (los que pagan el dinero) y los usuarios. Con la Gestión del Proyecto se logra una mejoría en el manejo de una entrega exitoso de software. En resumen su propósito consiste en proveer pautas para:

- Administrar proyectos de software intensivos.
- Planear, dirigir personal, ejecutar acciones y supervisar proyectos.
- Administrar el riesgo.

Sin embargo, esta disciplina no intenta cubrir todos los aspectos de dirección del proyecto. Por ejemplo, no cubre problemas como:

- Administración de personal: contratando, entrenando, enseñando.
- Administración del presupuesto: definiendo, asignando.
- Administración de los contratos con proveedores y clientes.

Entorno

Esta disciplina se enfoca sobre las actividades necesarias para configurar el proceso que engloba el desarrollo de un proyecto y describe las actividades requeridas para el desarrollo de las pautas que apoyan un proyecto.

Su propósito es proveer a la organización que desarrollará el software, un ambiente en el cual basarse, el cual provee procesos y herramientas para poder desarrollar el software.

- **Organización y elementos en RUP**

Ya conociendo varias partes del RUP nos concentraremos ahora en los elementos que lo componen, entre estos se tienen: Flujos de Trabajo, Detalle de los Flujos de Trabajo**,** Actores, Actividades y Artefactos. En la siguiente figura se muestran más claramente cómo se representan gráficamente cada uno de estos elementos y la interrelación entre ellos. Se puede observar que el Flujo de Trabajo de Requerimientos conlleva varios pasos, cada uno de estos pasos tiene asociado uno o varios actores, los cuales a su vez son los encargados de la ejecución de varias

actividades, las cuales a la vez están definidas en artefactos o guías para su realización.

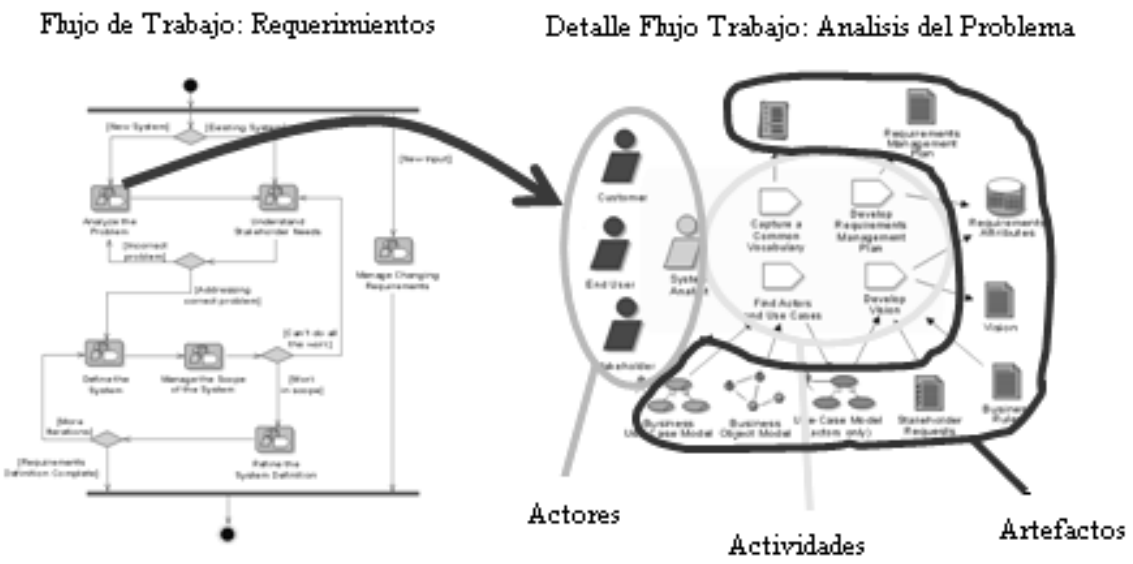


Figura 13 Elementos que conforman RUP

Fuente: [WEB04]

Actores o roles

Son los personajes encargados de la realización de las actividades definidas dentro de los flujos de trabajo de cada una de las disciplinas del RUP, estos actores se dividen en varias categorías: Analistas, Desarrolladores, Probadores, Encargados, Otros actores. A continuación se presenta una lista de actores de acorde a las categorías mencionadas con anterioridad:

Analistas

_ Analista del Proceso del Negocio
_ Diseñador del Negocio
_ Revisor del Modelo del Negocio
_ Revisor de Requerimientos
_ Analista del Sistema
_ Especificador de Casos de Uso
_ Diseñador de Interfaz del Usuario

Desarrolladores

_ Arquitecto
_ Revisor de la Arquitectura
_ Diseñador de Cápsulas
_ Revisor del Código y Revisor del Diseño
_ Diseñador de la Base de Datos
_ Diseñador
_ Implementador y un Integrador

Probadores Profesionales

_ Diseñador de Pruebas
_ Probador

Encargados

_ Encargado de Control del Cambio
_ Encargado de la Configuración
_ Encargado del Despliegue
_ Ingeniero de Procesos
_ Encargado de Proyecto
_ Revisor de Proyecto

Otros

_ Cualquier trabajador
_ Artista Gráfico
_ Stakeholder
_ Administrador del Sistema
_ Escritor técnico
_ Especialista de Herramientas

Artefactos

Los artefactos son el resultado parcial o final que es producido y usado por los actores durante el proyecto. Son las entradas y salidas de las actividades, realizadas por los actores, los cuales utilizan y van produciendo estos artefactos para tener guías. Un artefacto puede ser un documento, un modelo o un elemento de modelo.

Conjuntos de artefactos

Se tiene un conjunto de artefactos definidos en cada una de las disciplinas y utilizadas dentro de ellas por lo actores para la realización de las mismas, a continuación se definen cada una de estas categorías o grupos de artefactos dentro de las disciplinas del RUP:

Modelado del negocio

Los artefactos del modelado del negocio capturan y presentan el contexto del negocio del sistema. Los artefactos del modelado del negocio sirven como entrada y como referencia para los requisitos del sistema.

Requerimientos

Los artefactos de requerimientos del sistema capturan y presentan la información usada en definir las capacidades requeridas del sistema.

Análisis y diseño del sistema

Los artefactos para el análisis y del diseño capturan y presenta la información relacionada con la solución a los problemas se presentaron en los requisitos fijados.

Implementación

Los artefactos para la implementación capturan y presentan la realización de la solución presentada en el análisis y diseño del sistema.

Pruebas

Los artefactos desarrollados como productos de las actividades de prueba y de la evaluación son agrupadas por el actor responsable, con el cual se lleva un conjunto de documentos de información sobre las pruebas realizadas y las metodologías de pruebas aplicadas.

Despliegue

Los artefactos del despliegue capturan y presentan la información relacionada con la transitividad del sistema, presentada en la implementación en el ambiente de la producción.

Administración del proyecto

El conjunto de artefactos de la administración del proyecto capturan los artefactos asociados con el proyecto, el planeamiento y a la ejecución del proceso.

Administración de cambios y configuración

Los artefactos de la configuración y administración del cambio capturan y presentan la información relacionada con la disciplina de configuración y administración del cambio.

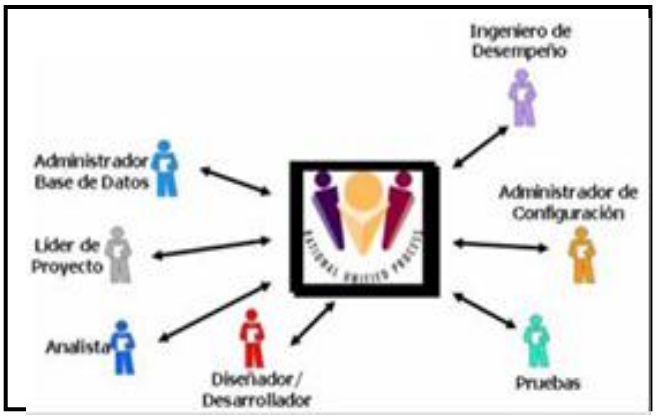
Entorno o ambiente

El conjunto de artefactos del ambiente presentan los artefactos que se utilizan como dirección a través del desarrollo del sistema para asegurar la consistencia de todos los artefactos producidos.

CAPITULO III

VISIÓN

FASE DE INICIO



3. VISIÓN

3.1 Introducción

3.1.1 Propósito

El propósito de este documento es definir los requerimientos de la aplicación “SISTEMA DE ADMINISTRACIÓN DE BRÓKER DE SEGUROS UTILIZANDO SOFTWARE LIBRE PARA LA INSTITUCIÓN “ASESORES PRODUCTORES DE SEGUROS ÁLVARO CAZAR”.

Este sistema está encargado de administrar la información del Bróker de manera integral y eficiente, utilizando los recursos con los que dispone.

Presenta las necesidades y características del Sistema al igual que se mostrarán los módulos roles y necesidades de los involucrados en el proyecto, así como las funcionalidades a ser implementadas las que se reflejan en la elaboración de los casos de uso.

Por otro lado, también es el propósito llevar a cabo una de las mejores prácticas en el desarrollo de un proyecto: la de mantener una documentación del proyecto desde el inicio de su ciclo de vida.

3.1.2 Alcance

Este documento de visión se aplica al “SISTEMA DE ADMINISTRACIÓN DE BRÓKER DE SEGUROS UTILIZANDO SOFTWARE LIBRE PARA LA INSTITUCIÓN “ASESORES PRODUCTORES DE SEGUROS ÁLVARO CAZAR” que es desarrollado por la egresada Villegas Estévez Jhoanna Paola de la Facultad de Ingeniería en Ciencias Aplicadas.

El alcance está orientado a las fases de diseño, desarrollo e implementación del sistema de administración del bróker de seguros Álvaro Cazar.

3.1.3 Definiciones, Acrónimos, y Abreviaciones

RUP: En ingles significa Rational Unified Process es un producto del proceso de ingeniería de software que proporciona un enfoque disciplinado para asignar tareas y responsabilidades dentro de una organización del desarrollo. Su meta es asegurar la producción del software de alta calidad que resuelve las necesidades de los usuarios dentro de un presupuesto y tiempo establecidos.

3.1.4 Referencias

- Glosario
- Plan de desarrollo de software
- RUP (Rational Unified Process)
- Diagrama de casos de uso

3.2 Posicionamiento

3.2.1 Oportunidad de Negocio

En la actualidad la mayoría de los negocios referentes a brókers no cuentan con sistemas para automatizar o mejorar los procesos, es por ello que un sistema de control desarrollado con nuevos métodos, con interfaces hará que se explote el negocio al máximo aumentando sus ventas y captando mayor cantidad de clientes; y en cuanto al negocio interno nos permita llevar un control personal con mayor eficiencia.

3.2.2 Definición del problema

El problema de	La complejidad en el control de ventas de seguros y la falta de organización de la información de los Brókers, así como el manejo manual de los documentos relacionados a sus clientes.
Afecta	Al administrador (bróker)
El impacto está	En que el bróker no cuenta con información disponible, para dar una atención diferencial a sus clientes y tomar decisiones a tiempo.
Una solución adecuada es	Una aplicación WEB que le permita: • Mantener una comunicación más cercana con sus clientes. • Automatizar los procesos vinculados con la cotización, seguimiento de pólizas y control de comisiones y siniestros. • Administrar su propia documentación manteniendo un archivo digitalizado.

Fuente: Propia

Tabla 3 Definición del Problema

3.2.3 Sentencia que define la posición del Producto

Para	Bróker Álvaro Cazar, que maneja un número considerables de documentación de sus clientes.
Quien	Tiene la necesidad de automatizar sus actividades operativas diarias y llevar un control automatizado de sus clientes y ventas.
El (Producto)	Sistema de Administración del Bróker de Seguros (ACBróker 1.0)
Que	Permite: • Mantener una comunicación más cercana con sus clientes. • Automatizar los procesos vinculados con la cotización, seguimiento de pólizas y control de comisiones y siniestros. • Administrar su propia documentación manteniendo un archivo digitalizado.
A diferencia de	Otros sistemas y aplicaciones WEB. El Software para el Bróker de Seguros (ACBróker 1.0) será de fácil manejo para el usuario y personalizado para sus necesidades.
Nuestro producto	Tendrá flexibilidad de implantación y configuración por parte del usuario al momento de instalarlo.

Fuente: Propia

Tabla 4 Sentencia que define la posición del producto

3.3 Descripción de Stakeholders (Participantes en el Proyecto) y Usuarios

Para proveer de productos y servicios de una manera efectiva y que se ajusten a las necesidades de los usuarios, es necesario identificar e involucrar a todos los participantes en el proyecto como parte del proceso de modelado de requerimientos.

Al igual que es necesario identificar a los usuarios del sistema y certificar de que el todos los participantes en el proyecto se los represente adecuadamente, para lo cual en este capítulo se muestra un perfil de los participantes y de los usuarios que se encuentran involucrados en el proyecto, al igual que los problemas más importantes que los mismos perciben para enfocar la solución propuesta hacia ellos.

Los interesados son todas aquellas personas directamente involucradas en la definición y alcance del proyecto.

3.3.1 Resumen de Usuarios

Son todos aquellos individuos que directamente se encuentran involucrados en lo que se refiere al uso del sistema, los mismos que serán detallados a continuación.

NOMBRE	DESCRIPCION	RESPONSABILIDADES
Usuario que navega por el sitio web	Cliente que se provee de la información sobre productos adquiridos, estado de sus cuentas y siniestros.	Representa a los clientes de la empresa. Navega en la web.
Ing. Álvaro Cazar	Gerente- Propietario del Bróker de Seguros Álvaro Cazar, "AC Bróker"	Responsable de tomar decisiones para poder potenciar la atención al cliente, ventas, y crecimiento de la empresa.
Personal administrativo	Usuario que administra la información del bróker	Responsable de administrar la información del Bróker: mantener actualizada la información de clientes, pólizas, compañías, siniestros, comisiones.

Fuente: Propia

Tabla 5 Resumen de Usuarios

3.3.2 Entorno de Usuario

No se tiene especificaciones de requerimientos de hardware ni software, para que los clientes puedan acceder al sitio web lo poder hacer bajo cualquier sistema operativo ya que al ser un sistema web se requiere solamente utilizar un navegador web como por ejemplo Mozila / Firefox versión 6 o superior.

3.3.3 Perfil de los Stakeholders

3.3.3.1 Representante del Bróker de Seguros Álvaro Cazar

Representante	Ing. Álvaro Cazar
Descripción	Gerente- Propietario del Bróker de Seguros Álvaro Cazar, "AC Bróker"
Tipo	Gerente de la empresa
Responsabilidades	Analizar y autorizar el manejo de la infraestructura tecnológica de AC Bróker Permitir el acceso a servidores para publicar la aplicación web.

Criterio de éxito	Analizar el costo para la publicación del sistema
Grado de participación	Revisión de requerimientos y estructura del sistema
Comentarios	Ninguno

Fuente: Propia

Tabla 6 Perfil del responsable de sistemas

Representante	Egda. Jhoanna Villegas Estévez
Descripción	Responsable del correcto funcionamiento del sistema del Bróker de Seguros Álvaro Cazar, "AC Bróker"
Tipo	Programadora de Sistemas
Responsabilidades	Gestionar el correcto funcionamiento del Sistema de AC Bróker
Criterio de éxito	Funcionamiento correcto del sistema
Grado de participación	Análisis, diseño y desarrollo del sistema web
Comentarios	Ninguno

Fuente: Propia

Tabla 7 Perfil de los programadores del sistema

3.3.4 Perfiles de Usuario

Usuario en internet

Representante	Navegante en Internet
Descripción	Usuario del sistema del Bróker de Seguros Álvaro Cazar, "AC Bróker"
Tipo	Usuario
Responsabilidades	Ninguna
Criterio de éxito	Sistema funcionando
Grado de participación	Activa
Comentarios	Ninguno

Fuente: Propia

Tabla 7 Perfil de Usuario

3.4 Descripción Global Del Proyecto

3.4.1 Perspectiva del Producto

Esta aplicación web ha sido desarrollada utilizando las tecnologías libres y de actualidad, para poder crear una herramienta que servirá al Bróker de Seguros "Álvaro Cazar", para automatizar las actividades operativas diarias que este tiene, llevando un

control automatizado de sus clientes y ventas, y de esta manera potenciar el crecimiento del mismo.

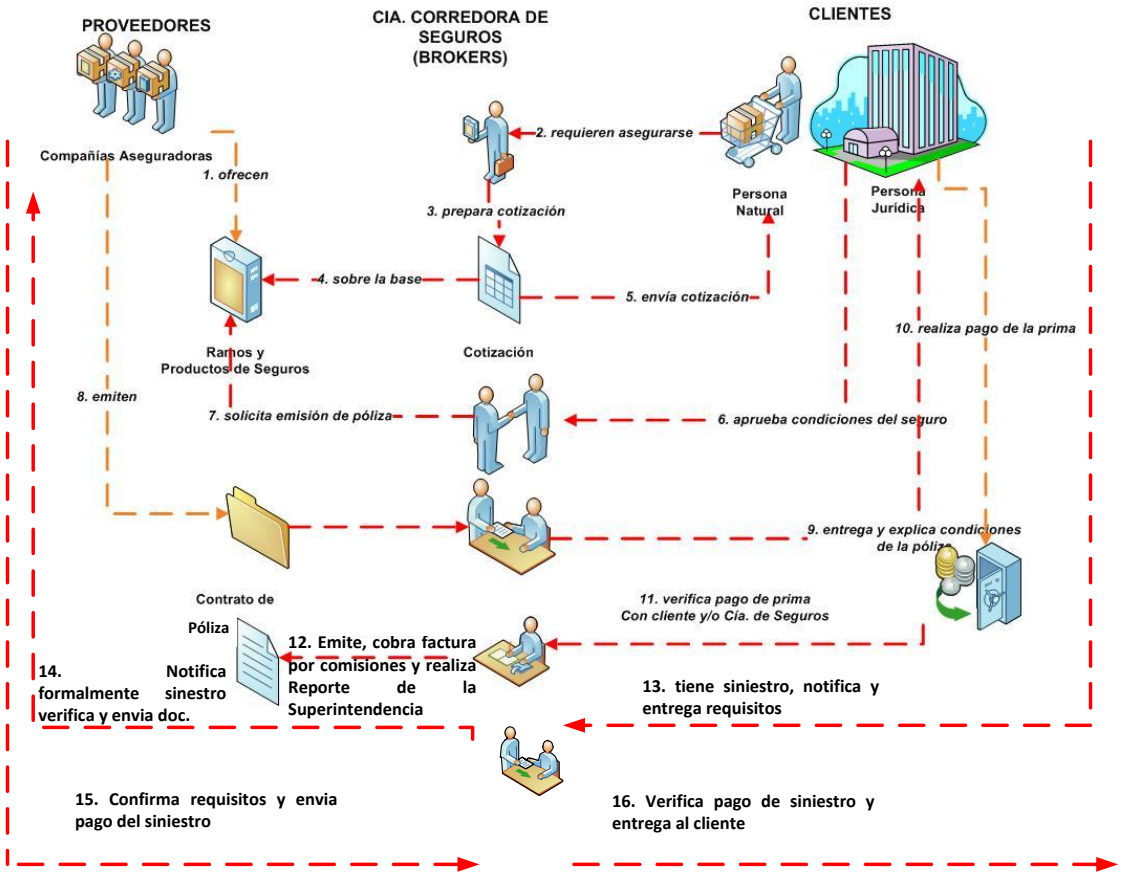


Figura 14 Perspectiva del Producto

Fuente: Propia

3.4.2 Resumen de características

Mostramos en el siguiente cuadro las características y beneficios que presta el producto tanto al Bróker como a los clientes del mismo:

BENEFICIOS DEL USUARIO	CARACTERÍSTICAS QUE LO APOYAN
Tener Información exacta y a tiempo de los clientes, pólizas, siniestros.	Se tiene módulos dentro del sistema en los cuales se administra los clientes, pólizas, y siniestros.
Poder dar adecuadamente y de forma ordenada el seguimiento de la producción de la empresa.	Se tiene un módulo de reporte donde se puede ver la producción realizada ya sea semanal, trimestral,

	semestral, como desee el usuario.
Obtener automáticamente el Reporte anual para la Superintendencia de Bancos y Seguros, ya que se lo realizaba de forma manual.	Cuenta con el módulo de reporte para la SUPERINTENDENCIA DE BANCOS Y SEGUROS, de acuerdo al formato exigido por esta institución.
Personal Administrativo de la empresa, puede obtener la información sobre el estado de sus comisiones.	El sistema cuenta con un módulo donde se lleva el control de las comisiones que se generan para el bróker por compañía aseguradora.

Fuente: Propia

Tabla 8 Resumen de características

3.4.3 Suposiciones y Dependencias

El cliente a través de internet podrá acceder a la aplicación web del Bróker de Seguros, mediante un navegador web para poder revisar la información sobre los productos adquiridos, y por su parte el personal administrativo podrá acceder de igual manera para poder obtener información de la empresa y si desea realizar ingresos, cambios o eliminaciones deberá contar con su respectiva clave y permisos, de acuerdo a los privilegios que estos tengan en el sistema.

3.4.4 Costo y Precio

ANALISIS FINANCIERO

Datos utilizados para el análisis financiero:

La inversión total del proyecto es de 3500 con la estrategia 14% INVERSIÓN PROPIA y 86% INVERSIÓN FINANCIADA.					86,00%	14,00%		3500
La inversión financiada es a 3 años con pagos mensuales al 14,3% anual.			3		14,30		Vf	3010
La inversión propia tiene una alternativa de rendimiento en el mercado de valores del 5.14%.						9,85	Vp	490
La inversión se estructura de la siguiente manera								
Inversión variable mensual	16269,76	**kt mensual**						
Inversión fija	3000,00			**vida útil**				
Equipo de Oficina		20,00		10	años			
Muebles y Enseres		400,00		10				
Equipo informático		2.580,00		3				
	19269,757	**3000,00**						INFLACION
								5,53%
Riesgo país						837	puntos	
Imprevistos 5% de los ingresos proyectados						3%		
Costos operativos son el 26.31% del precio de venta proyectado.								
Tasa de crecimiento anual 7.7%								7,7%
Valor hora por mantenimiento de sistema 20,00 dólares								

OFERTA Y DEMANDA:

OFERTA

	PRECIO
SIES - SISTEMA EXPERTO DE SEGUROS	2500
SISBS - SISTEMA DE CONTROL PARA BROKERS DE SEGUROS	1800
SIBS - SISTEMA INFORMATICO PARA BROKERS DE SEGUROS	1500

DEMANDA

VERTICAL BROKER DE SEGUROS
BYRON LOPEZ ASESOR PRODUCTOR
BROKER TORRES GUARIN Y ASOCIOADOS S.A.
TECNISEGUROS
NOVA
IMBASEGUROS
SEGUROS VASQUEZ Y VASQUEZ
EDUARDO VACAS ASESOR PRODUCTOR
FERNANDO ROSAS
MARIA DAVILA
MANCHENO
SAYO
KASEG ASESOR DE SEGUROS
CRISTIAN ALVARADO HERRERA
ALIADA 2000 AGENCIA ASESORA DE SEGUROS CÍA.LTDA
AVALTEK CÍA.LTDA.
ALAMO S.A. ASESORES DE SEGUROS
ARMUSA S.A. AGENCIA ASES.
CONFISEG C.LTDA.
ASTECSE S.A.
PRIVANZA CÍA.LTDA.
JUAN JESÚS SOCIAS SÁNCHEZ
SEGUR CUENCA, S.L.
SEGURMEDIA SEGUROS
NICOLÁS NAVARRO
ASEGUTE

SEGUR JUCAR
Santa lucia
SEGUROS DOMINGUEZ AVILA S.L.
MOTISEGUR DE MOTILLA AGENCIA S.L.
JESUS GARCIA NIEVES
JUAN ALMAZAN
JESUS ANTONIO GALLEGO AGENCIA DE SEGUROS
CASTILLA RED ASEGURADORA

MOTISEGUR DE MOTILLA AGENCIA S.L
MANJARREZ MANCHENO FABIAN VINICIO
MANCHENO VILLACRESES GALO PATRICIO
ALMOCOLS CIA. LTDA.
USINIA SANCHEZ DENIS PATRICIO
MOLINA VILLALBA GERMAN VINICIO
ESPINOZA ARELLANO RENAN EDMUNDO
SANCHEZ SANCHEZ RUBEN MAURICIO
CONFIA S.A.
SEGURA SOLIS CARLOS ENRIQUE
AVILES ALBAN CATALINA DE LAS MERCEDES
COINBROKER CIA. LTDA.
MOLINA VILLACIS GALO EDMUNO
ASERTEC CIA. LTDA.
UNISEGUROS C.A.

NACIONALES CIA. LTDA.
RAUL COKA BARRIGA CIA.
COLCORDES SOCIEDAD ANONIMA
ECUAPATRIA CIA. LTDA.
Z.H.M. ZULOAGA
HIDALGO & MAQUILON S.A.
ECUAPRIMAS CIA. LTDA.
EL SOL CIA. LTDA.
COLARI S. A.
INTERSEA CIA. LTDA.
XPRESSEG
AFFINITY ANGENCIA ASESORA PRODUCTORA DE SEGUROS
EMI ECUADOR
ALCONSA S.A.
COLNEXOS
OLIVOSEG
THOMAS COOPER ECUADOR
INFRASEG
SEMERCA S.A.

NICOLAS NAVARRO CAÑAS	GESTION SEGUROS
RAFAEL DIAZ CASTELLOTE	G-R CORPORATHION
SAIZ Y AROCA	SUAREZ Y SUAREZ
REDES Y SISTEMA CASTILLA S.L.	SML S.A.
PONCE Y SERRANO S.L.	SEGUROS DEL PICHINCHA
NAVAMO PRODUCES	SETEC S.A.
SERVISEGUROS	POPULAR
PROSEGUROS	ESMOBROKER
COLSEGUROS	PRIMACIA S.A.
FRAUDA	BROKER DE SEGUROS IZQUIERDO PENAHERRERA
COVERSA	BROKER CIA. LTDA
CIASEG	ZION SEGUROS
JOHNSON Y ASOCIADOS	DIRECT SEGUROS
COLCOLRDES	AGENCIA ASESORA PRODUCTORA DE SEGUROS DEL PACIFICO
PLURISEG	AR SEGUROS
SEGUROS DOMINGUEZ AVILA S.L.	BROKER SEGSEGUROS S.A.

Detalle de Ingresos y Egresos

Detalle		USD
Activos Fijos	2 Equipos de Computación PC De escritorio	1.500,00
	1 Laptop	1.000,00
	1 Muebles de Oficina	400,00
	1 Impresora	80,00
	1 Teléfono	20,00
	TOTAL	**3.000,00**
Software	PostgreSql	0,00
	Framework Spring	0,00
	Servidor de Aplicaciones ApacheTomcat	0,00
	TOTAL	**0,00**
Costo de Desarrollo	Análisis	50
	Diseño	125
	Programación	250
	Pruebas	75
	TOTAL	**500,00**

Gastos	Sueldo	7.200,00
	Beneficios Sociales	2.302,80
	Movilización	1.200,00
	Internet	180,00
	Servicios Básicos	240,00
	Suministros de Oficina	60,00
	Depreciaciones	902,00
	Costos Financieros	843,54
	Arriendo	2.400,00
	TOTAL	**15.328,34**
Subtotal	(Parcial)	18.828,34
5% Imprevistos		941,42
Total		**19.769,76**

Fuente: Propia

Tabla 9 Costos y Precios

Tasa de Rendimiento del Mercado

Resolución

TRM

Descripción	Valor	% composición	Tasa ponderada	Valor ponderado
Inversión propia	490	14,00%	6,00	84
Inversión financiada	3.010	86,00%	14,30	1.230
Inversión total	3.500	100%		1.314
			Rendimiento	13,14
			Riesgo país	8,37

Cálculo del TRM

TRM = (1+Ck)(1+Rp)-1

0,226076506	22,61%	Mínimo que debe rendir en condiciones reales de la economía,

Quiere decir que en nuestro país la tasa de rendimiento del proyecto no debe ser menor

Obligaciones Financieras

MONTO DEL CRÉDITO 3.000,00

TASA DE INTERÉS ANUAL 14,3 CUOTAS 36

Cuotas	Fech. Venc.	# Días	Valor Capital	Interés	Otros	total
1	2012-01-10	31	67,22	36,94	5,00	109,16
2	2012-02-10	31	68,02	36,11	5,00	109,13
3	2012-03-10	29	68,83	33,00	5,00	106,83
4	2012-04-10	31	69,65	34,43	5,00	109,08
5	2012-05-10	30	70,48	32,49	5,00	107,97
6	2012-06-10	31	71,32	32,70	5,00	109,02

7	2012-07-10	30	72,17	30,80	5,00	107,97
8	2012-08-10	31	73,03	30,94	5,00	108,97
9	2012-09-10	31	73,90	30,04	5,00	108,94
10	2012-10-10	30	74,78	28,19	5,00	107,97
11	2012-11-10	31	75,67	28,21	5,00	108,88
12	2012-12-10	30	76,58	26,39	5,00	107,97
13	2013-01-10	31	77,49	26,33	5,00	108,82
14	2013-02-10	31	78,41	25,38	5,00	108,79
15	2013-03-10	28	79,35	22,05	5,00	106,40
16	2013-04-10	31	80,29	23,43	5,00	108,72
17	2013-05-10	30	81,25	21,72	5,00	107,97
18	2013-06-10	31	82,22	21,45	5,00	108,67
19	2013-07-10	30	83,20	19,77	5,00	107,97
20	2013-08-10	31	84,19	19,41	5,00	108,60
21	2013-09-10	31	85,19	18,37	5,00	108,56
22	2013-10-10	30	86,21	16,76	5,00	107,97
23	2013-11-10	31	87,23	16,26	5,00	108,49
24	2013-12-10	30	88,27	14,70	5,00	107,97
25	2014-01-10	31	89,33	14,10	5,00	108,43
26	2014-02-10	31	90,39	13,00	5,00	108,39
27	2014-03-10	28	91,47	10,74	5,00	107,21
28	2014-04-10	31	92,56	10,76	5,00	108,32
29	2014-05-10	30	93,66	9,31	5,00	107,97
30	2014-06-10	31	94,78	8,47	5,00	108,25
31	2014-07-10	30	95,91	7,06	5,00	107,97
32	2014-08-10	31	97,05	6,12	5,00	108,17
33	2014-09-10	31	98,21	4,92	5,00	108,13
34	2014-10-10	30	99,38	3,60	5,00	107,98
35	2014-11-10	31	100,56	2,49	5,00	108,05
36	2014-12-10	30	101,75	1,21	5,00	107,96
TOTALES			3000,00	717,65	180,00	3897,65

Depreciaciones

Cuadro de Depreciación

Activo	Valor	Vida útil	Depreciación Anual
Equipo de Oficina	20,00	10	2
Muebles y Enseres	400,00	10	40
Equipo informático	2.580,00	3	860
	3000,00		**902,00**

Balance General de Situación Inicial

Balance de Arranque

Al 01 de Diciembre del 2011

ACTIVOS		**PASIVOS**	
Activo Circulante		Financiamiento	3.010
Software	500	**PATRIMONIO**	
Activos Diferidos		Inversión propia	490
Inversión diferida			
Activos Fijos			
Equipo de Oficina	20		
Muebles y Enseres	400		
Equipo informático	2.580		
TOTAL ACTIVOS	3.500	**TOTAL PASIVOS Y PATRIMONIO**	3.500

Estado de Resultados

BALANCE DE RESULTADOS PROYECTADO

			Año 1		**Año 2**		**Año 3**
	Ventas proyectadas		28800,00		28800,00		28800,00
+	Otros ingresos proyectados		1440		1440		1440,00
-	Costos proyectadas	-	12000,00	-	12000,00	-	12000,00
	Utilidad bruta proyectada		18240,00		18240,00		18240,00
	Gastos proyectados						
	Gastos administrativos proyectados	10702,80		10702,80		10702,80	
	Imprevistos	941,42		941,42		941,42	
	Depreciación	902,00		902,00		902,00	
	Servicios básicos	480,00		480,00		480,00	
	Otros	2880,00		2880,00		2880,00	
	Total gastos proyectados	-	15906,22	-	15906,22	-	15906,22
	Utilidad operacional proyectada		2333,78		2333,78		2333,78
-	Gastos financieros proyectados	-	440,24	-	305,63	-	151,78
	Utilidad neta proyectada		1893,54		2028,15		2182,00

Ventas Proyectadas

	Año 1
Ingresos mensuales	2400
Ventas efectivas en volumen	2400

Descripción	**Año 1**	**Año 2**	**Año 3**
Volumen proyectado	24	24	24
Ventas efectivas	24	24	24
Precio de venta proyectado sistema	1200	1200	1200
Ingreso total proyectado	**28800**	**28800**	**28800**

Presupuesto de Costos

Descripción	**Año 1**	**Año 2**	**Año 3**
Volumen proyectado	24	24	24

Costo unitario proyectado	500	500	500
Costo total proyectado	**12000**	**12000,00**	**12000,00**

Presupuesto de Gastos

Descripción	Año 1	Año 2	Año 3
Gastos administrativos	10.702,80	10702,8	10702,8
Imprevistos	941,42	941,417	941,417
Gastos financieros	440,24	305,63	151,78
Depreciación	902,00	902,00	902,00
Otros	2.880,00	2.880,00	2.880,00
Total gastos	**15866,46**	**15731,85**	**15578,00**

Flujo y Calculo del VAN y TIR

Flujo de caja libre

		Año 1	Año 2	Año 3
Ingresos				
Ingresos proyectados		28800,00	28800,00	28800,00
Otros ingresos		1440	1440	1440,00
Total de ingresos proyectados	-3500	30240,00	30240,00	30240,00
Egresos				
Costos proyectados		12000	12000	12000
Imprevistos		941,417	941,417	941,417
Gastos administrativos		10702,8	10702,8	10702,8
Depreciación		902,00	902,00	902,00
Otros		2880,00	2880,00	2880,00
Total de egresos proyectados		27426,22	27426,22	27426,22
Flujo neto (dife entre ing-gast)	**-3500**	**2813,78**	**2813,78**	**2813,78**
SALDO INICIAL EN CAJA/CTA.CORR		0	2373,54	4881,69
SALDO DE CAJA SIN FINANC.		**2813,78**	**5187,32**	**7695,47**
Obligaciones financieros		440,24	305,63	151,78
SALDO FINAL DE CAJA CON FINANC.	**-3500**	**2373,54**	**4881,69**	**7543,69**

TRM	22,61%
VAN	4710,69
TIR	95,53%

TRM	TASA DE RENDIMIENTO DE MERCADO
VAN	VALOR ACTUAL NETO
TIR	TASA INTERNA DE RETORNO

CONCLUSIONES:

- EL RESULTADO DE LA TIR ES DEL 95,53% , QUE ES MUY SUPERIOR AL TRM DE 22,61% QUE SIGNIFICA QUE EL PROYECTO PRESENTARA UN RETORNO (RENTABILIDAD) DE 95,53% POR TANTO , ES FACTIBLE EL PROYECTO.

- CON RESPECTO AL VAN SU VALOR ES DE 4710,69 QUE ES SUPERIOR A 0 POR LO TANTO EL PROYECTO ES VIABLE, YA QUE LOS FLUJOS DESCONTADOS SON SUPERIORES AL MONTO DE LA INVERSIÓN REALIZADA.
- EN BASE A LOS DATOS ANTERIORMENTE MENCIONADOS LA INVERSIÓN SE RECUPERARA EN EL SEGUNDO AÑO YA QUE SE OBSERVA EL AUMENTO EN LA UTILIDAD.

3.5 Características Del Producto

- Facilidad de acceso y uso

El Sistema de Administración del Bróker de Seguros (ACBróker 1.0), es desarrollado utilizando herramientas de Software Libre (Open Source), lo que permitirá fácil acceso y uso.

- Unificación de la información

Unos de los principales objetivos del sistema es determinar y presentar al usuario formatos unificados de ingreso y consulta de datos.

- Mejor control y validación de la información

Los clientes y personal del bróker contarán con facilidades para la verificación de la información consolidada.

3.6 Restricciones

3.6.1 Restricción de licencia

Todo el Sistema está realizado con software libre y por lo tanto no se debe preocupar por el pago de licencias, al igual que los usuarios pueden acceder al sistema desde cualquier navegador sin importar el sistema operativo que posea la máquina.

3.6.2 Restricciones de lugar

El sistema fue desarrollado para que pueda ser utilizado tanto para el personal que trabaja en el Bróker de Seguros “Álvaro Cazar”, como para los Clientes y posibles clientes del mismo.

El sistema está realizado desacuerdo a las necesidades del Bróker de Seguros “Álvaro Cazar”, motivo por el cual esta empresa solamente podrá dar los requisitos de la misma al igual que tendrá a su disposición el código fuente para si en el futuro necesitara realizar cambios.

3.6.3 Restricciones del Software

El sistema será dado por terminado siempre y cuando se encuentre funcionando de acuerdo a los requerimientos dados por el Gerente del Bróker, y a la constancia de la firma de aceptación del mismo.

3.6.4 Rangos de calidad

El desarrollo del sistema se ajusta a la metodología de desarrollo de software RUP, contemplando de esta manera los parámetros de calidad que la metodología define.

CAPITULO IV

PLAN DE DESARROLLO DEL SOFTWARE

FASE DE INICIO

4. PLAN DE DESARROLLO DE SOFTWARE

4.1 Introducción

Este Plan de Desarrollo del Software es una versión preliminar preparada para ser incluida en la propuesta elaborada como respuesta al proyecto: Sistema de Administración del Bróker de Seguros (ACBróker 1.0), para el Bróker de Seguros del Ing. Álvaro Cazar. Este documento provee una visión global del enfoque de desarrollo propuesto.

El proyecto ha sido basado en una metodología de Unificación de Procesos con el fin de implantar un esquema inicial de ésta metodología para futuros desarrollos.

El enfoque desarrollo propuesto constituye una configuración del proceso Unificación de Procesos de acuerdo a las características del proyecto, seleccionando las actividades a realizar y los artefactos (entregables) que serán generados. Este documento es a su vez uno de los artefactos de la Unificación de Procesos.

4.1.1 Propósito

El propósito del Plan de Desarrollo de Software es proporcionar la información necesaria para controlar el proyecto. En él se describe el enfoque de desarrollo del software.

Los usuarios del Plan de Desarrollo del Software son:

- El jefe del proyecto lo utiliza para organizar la agenda y necesidades de recursos, y para realizar su seguimiento.

- Los miembros del equipo de desarrollo lo usan para entender lo qué deben hacer, cuándo deben hacerlo y qué otras actividades dependen de ello.

- Personas responsables del análisis y revisión del sistema, en este caso intervienen el Gerente de AC Bróker.

4.1.2 Alcance

El Plan de Desarrollo del Software describe el plan global usado para el desarrollo del Sistema de Administración del Bróker de Seguros Álvaro Cazar (ACBróker 1.0).

Durante el proceso de desarrollo en el artefacto “Visión” se definen las características del producto a desarrollar, lo cual constituye la base de la planificación de las iteraciones.

Para el Plan de Desarrollo del Software, me he basado en la captura de requisitos por medio del Gerente – Propietario y la persona que se encarga del manejo operativo del Bróker de Seguros para hacer una estimación aproximada, una vez comenzado el proyecto y durante la fase de Inicio se generará la primera versión del artefacto “Visión”, el cual se utilizará para refinar este documento. Posteriormente, el avance del proyecto y el seguimiento en cada una de las iteraciones ocasionará el ajuste de este documento produciendo nuevas versiones actualizadas.

4.1.3 Resumen

Después de esta introducción, el resto del documento está organizado en las siguientes secciones:

Vista General del Proyecto.- Proporciona una descripción del propósito, alcance y objetivos del proyecto, estableciendo los artefactos que serán producidos y utilizados durante el proyecto.

Organización del Proyecto.- Describe la estructura organización del equipo de desarrollo.

Gestión del Proceso.- Explica los costos y planificación estimada, define las fases e hitos del proyecto y describe como se realizara su seguimiento.

Planes de guías de Aplicación.- Proporciona una vista global del proceso de desarrollo de software, incluyendo métodos, herramientas y técnicas que serán utilizadas.

4.2 Vista General del Proyecto

La información que a continuación se incluye ha sido extraída de las diferentes reuniones que se han celebrado con el stakeholder de la empresa desde el inicio del proyecto.

A partir de los procedimientos ya establecidos en el Bróker de Seguros Álvaro Cazar, y como parte del plan de automatización establecida, se determina la creación del Sistema de Administración del Bróker de Seguros Álvaro Cazar (ACBróker 1.0) que permita mejorar la gestión de las actividades relacionadas con los clientes y con la administración interna de la empresa.

El proyecto debe proporcionar una propuesta clara y sencilla de todos los módulos implicados en el desarrollo del Sistema de Administración del Bróker de Seguros Álvaro Cazar. Estos módulos se pueden diferenciar de la siguiente manera:

a) Clientes

- Gestionar a información personal del cliente
- Gestionar a la información sobre las pólizas que ha adquirido.
- Obtener un detalle de los vencimientos de sus pólizas.
- Gestionar la información sobre el estado de sus pólizas.
- Gestionar el estado de los siniestros.

b) Compañías Aseguradoras

- Gestionar a información de la compañía, direcciones, teléfonos, ruc, datos de cuenta bancaria.
- Acceder a productos que ofrecen

c) Pólizas

- Gestionar toda la información con respecto a pólizas vendidas
- Realizar ingreso, modificación, o eliminación de pólizas si se lo requiere
- Administrar los estados de las pólizas
- Definir comisión en las pólizas
- Renovación de pólizas

d) Liquidación de comisiones

- Definir las comisiones que percibe el bróker por compañía

e) Siniestros

- Administrar los siniestros suscitados, detalle de lo ocurrido
- Información requerida para hacer efectivo el seguros
- Documentación recibida y enviada a la compañía
- Fechas de notificaciones, envió de documentos y pago de siniestros

f) Reportes

- Presentar la información sobre clientes
- Presentar la información sobre pólizas
- Presentar la información sobre siniestros
- Presentar la información sobre liquidación de comisiones
- Reporte anual para la SUPERINTENDECIA DE COMPANIAS Y SEGUROS

4.2.1 Suposiciones y Restricciones del Sistema

Las suposiciones y restricciones respecto del sistema, y que se derivan directamente de las entrevistas y/o recomendaciones de los responsables del Bróker de Seguros Álvaro Cazar son:

- El sistema debe ser lo más sencillo de manejar por el usuario final

- Deben estar bien definidos los permisos de usuarios, para que no existan inconvenientes al momento de que es sistema sea utilizado ya sea por el personal del bróker o por los clientes del mismo.

- Debe estar tener facilidades de acceso a la información, bien organizado y distribuido para que el usuario tenga un ambiente amigable para navegar en el sitio.

- Debe tenerse en cuenta las limitaciones de los medios tecnológicos de los usuarios finales, es por eso que se decidió realizar una aplicación web y esta no requiere mucha tecnología, basta con tener una conexión a internet y un navegador web.

- La implementación de del proyecto se la realizara solamente con la información de los clientes que dispone el bróker así como de las pólizas que han sido vendidas por el mismo, y de las compañías que está trabajando al momento.
- El sistema estará publicado en la infraestructura tecnológica que tiene el Bróker de Seguros Álvaro Cazar.

Como es natural en el desarrollo de este tipo de sistemas, la lista de suposiciones y restricciones se incrementará durante el desarrollo del proyecto, particularmente una vez establecido el artefacto “Visión”.

4.2.2 Entregables del Proyecto

A continuación se indican y describen cada uno de los artefactos que serán generados y utilizados por el proyecto y que constituyen los entregables. Esta lista constituye la configuración de RUP desde la perspectiva de artefactos, y que propongo para este proyecto.

Es preciso destacar que de acuerdo a la filosofía de RUP (y de todo proceso iterativo e incremental), todos los artefactos son objeto de modificaciones a lo largo del proceso de desarrollo, con lo cual, sólo al término del proceso podríamos tener una versión definitiva y completa de cada uno de ellos. Sin embargo, el resultado de cada iteración y los hitos del proyecto están enfocados a conseguir un cierto grado de completitud y estabilidad de los artefactos.

- Plan de Desarrollo del Software

  Es el presente documento.

- Glosario de términos

  Es un documento que define los principales términos usados en el proyecto. Permite establecer una terminología consensuada.

- Especificaciones de Casos de Uso

  Para los casos de uso que lo requieran (cuya funcionalidad no sea evidente o que no baste con una simple descripción narrativa) se realiza una descripción detallada utilizando una plantilla de documento, donde se incluyen: precondiciones, post-condiciones, flujo de eventos, requisitos no-funcionales

asociados. También, para casos de uso cuyo flujo de eventos sea complejo podrá adjuntarse una representación gráfica mediante un Diagrama de Actividad.

- Prototipos de Interfaces de Usuario

  Se trata de prototipos que permiten al usuario hacerse una idea más o menos precisa de las interfaces que proveerá el sistema y así, conseguir retroalimentación de su parte respecto a los requisitos del sistema. Estos prototipos se realizarán como: dibujos a mano en papel, dibujos con alguna herramienta gráfica o prototipos ejecutables interactivos, siguiendo ese orden de acuerdo al avance del proyecto. Sólo los de este último tipo serán entregados al final de la fase de Elaboración, los otros serán desechados. Asimismo, este artefacto, será desechado en la fase de Construcción en la medida que el resultado de las iteraciones vayan desarrollando el producto final.

- Modelo de Implementación

  Este modelo es una colección de componentes y los subsistemas que los contienen. Estos componentes incluyen: ficheros ejecutables, ficheros de código fuente, y todo otro tipo de ficheros necesarios para la implantación y despliegue del sistema. (Este modelo es sólo una versión preliminar al final de la fase de Elaboración, posteriormente tiene bastante refinamiento).

- Material de Apoyo al Usuario Final

  Corresponde a un conjunto de documentos y facilidades de uso del sistema, incluyendo: Guías del Usuario, Guías de Operación, Guías de Mantenimiento.

4.3 Organización del Proyecto

4.3.1 Participantes en el Proyecto

Los participantes del proyecto son todos aquellos que intervienen en el mismo, tales como el director, los desarrolladores y todos aquellos que se estimen convenientes para proporcionar los requisitos y validar el sistema.

El personal del proyecto que actúa en las fases de Inicio, Elaboración e Iteraciones de la fase de construcción estará formado por los siguientes puestos de trabajo y personal asociado.

- Director del Proyecto

  Con una experiencia en metodología de desarrollo, herramientas case y notaciones en particular la notación UML y el proceso de desarrollo de RUP.
- Ingeniero de Software

  El perfil establecido es: Ingeniero en informática que participara realizando labores de gestión de requisitos, gestión de configuración, documentación, diseño de datos y desarrollo de la aplicación.

  Todo este trabajo ha sido encomendado a la Egda. Jhoanna Paola Villegas Estévez.

4.3.2 Interfaces Externas

Se define los participantes del proyecto que proporcionarán los requisitos del sistema, y entre ellos quiénes serán los encargados de evaluar los artefactos de acuerdo a cada módulo y según el plan establecido.

El equipo de desarrollo interactuará activamente con los participantes para especificación y validación de los artefactos generados.

4.3.3 Roles y Responsabilidades

A continuación se describen las principales responsabilidades de cada uno de los puestos en el equipo de desarrollo durante las fases de Inicio y Elaboración, de acuerdo con los roles que desempeñan en RUP.

Puesto	Responsabilidad
Jefe de Proyecto	El jefe de proyecto asigna los recursos, gestiona las prioridades, coordina las interacciones con los clientes y usuarios, y mantiene al equipo del proyecto enfocado en los objetivos. El jefe de proyecto también establece un conjunto de prácticas que aseguran la integridad y calidad de los artefactos del proyecto. Además, el jefe de proyecto se encargará de supervisar el establecimiento de la arquitectura del sistema. Gestión de riesgos. Planificación y control del proyecto. Responsable: Ing. Edgar Maya

Analista de Sistemas	Elaboración del modelo de análisis y diseño mediante entrevistas, captura y validación de requisitos, así como en la colaboración de las pruebas funcionales y el modelo de datos. Responsable: Jhoanna Villegas
Programador	Construcción de prototipos. Colaboración en la elaboración de las pruebas funcionales, modelo de datos y en las validaciones con el usuario. Responsable: Ing. Edgar Maya

Fuente: Propia

Tabla 10 Roles y Responsabilidades

4.4 Plan del Proyecto

En esta sección se presenta la organización en fases e iteraciones y el calendario del proyecto.

4.4.1 Plan de Fases

El desarrollo se llevará a cabo en base a fases con una o más iteraciones en cada una de ellas. La siguiente tabla muestra una la distribución de tiempos y el número de iteraciones de cada fase (para las fases de Construcción y Transición es sólo una aproximación muy preliminar).

Los hitos que marcan el final de cada fase se describen en la siguiente tabla.

	Hito
Fase de Inicio	En esta fase desarrollará los requisitos del producto desde la perspectiva del usuario, los cuales serán establecidos en el artefacto Visión. Los principales casos de uso serán identificados y se hará un refinamiento del Plan de Desarrollo del Proyecto. La aceptación del cliente / usuario del artefacto Visión y el Plan de Desarrollo marcan el final de esta fase.

Fase de Elaboración	En esta fase se analizan los requisitos y se desarrolla un prototipo de arquitectura (incluyendo las partes más relevantes y / o críticas del sistema). Al final de esta fase, todos los casos de uso correspondientes a requisitos que serán implementados en la primera release de la fase de Construcción deben estar analizados y diseñados (en el Modelo de Análisis / Diseño). La revisión y aceptación del prototipo de la arquitectura del sistema marca el final de esta fase. En nuestro caso particular, por no incluirse las fases siguientes, la revisión y entrega de todos los artefactos hasta este punto de desarrollo también se incluye como hito. La primera iteración tendrá como objetivo la identificación y especificación de los principales casos de uso, así como su realización preliminar en el Modelo de Análisis / Diseño, también permitirá hacer una revisión general del estado de los artefactos hasta este punto y ajustar si es necesario la planificación para asegurar el cumplimiento de los objetivos. Ambas iteraciones tendrán una duración de una semana.
Fase de Construcción	Durante la fase de construcción se terminan de analizar y diseñar todos los casos de uso, refinando el Modelo de Análisis / Diseño. El producto se construye en base a 2 iteraciones, cada una produciendo una release a la cual se le aplican las pruebas y se valida con el cliente / usuario. Se comienza la elaboración de material de apoyo al usuario. El hito que marca el fin de esta fase es la versión de la release 2.0, con la capacidad operacional parcial del producto que se haya considerado como crítica, lista para ser entregada.
Fase de Transición	En esta fase se prepararán se asegura la una implantación y cambio del sistema previo de manera adecuada, incluyendo el entrenamiento de los usuarios. El hito que marca el fin de esta fase incluye, la entrega de toda la documentación del proyecto con los manuales de instalación y todo el material de apoyo al usuario, la finalización del entrenamiento de los usuarios y el empaquetamiento del producto.

Fuente: Propia

Tabla 11 Plan de Fases

4.4.2 Calendario del Proyecto

A continuación se presenta un calendario de las principales tareas del proyecto incluyendo sólo las fases de Inicio y Elaboración. Como se ha comentado, el proceso iterativo e incremental de RUP está caracterizado por la realización en paralelo de todas las disciplinas de desarrollo a lo largo del proyecto, con lo cual la mayoría de los artefactos son generados muy tempranamente en el proyecto pero van desarrollándose en mayor o menor grado de acuerdo a la fase e iteración del proyecto.

La siguiente figura ilustra este enfoque, en ella lo ensombrecido marca el énfasis de cada disciplina (workflow) en un momento determinado del desarrollo.

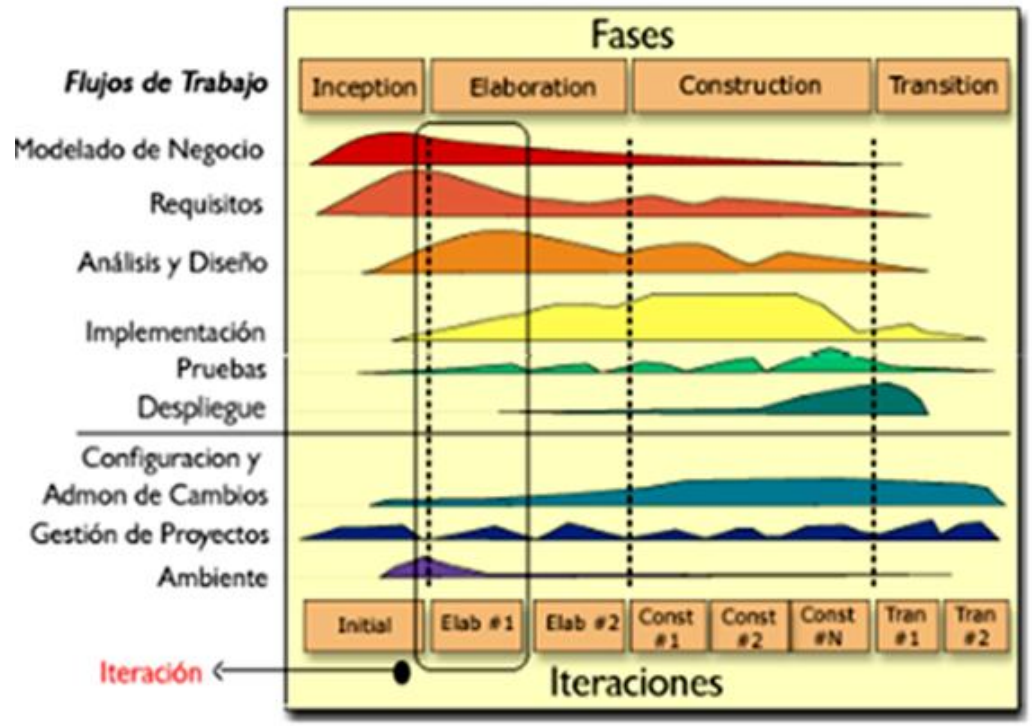


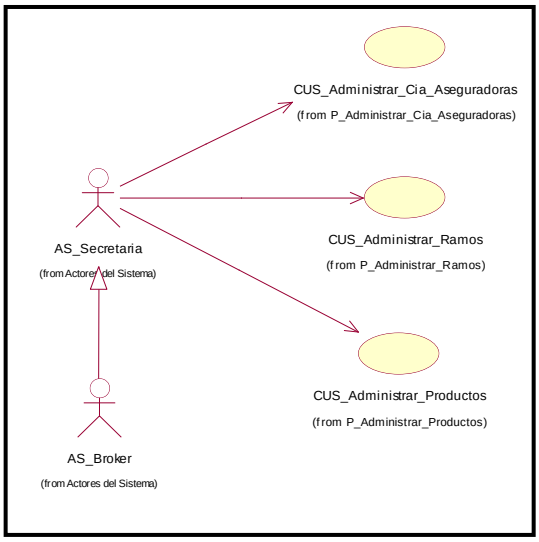
Figura 15 Calendario del proyecto

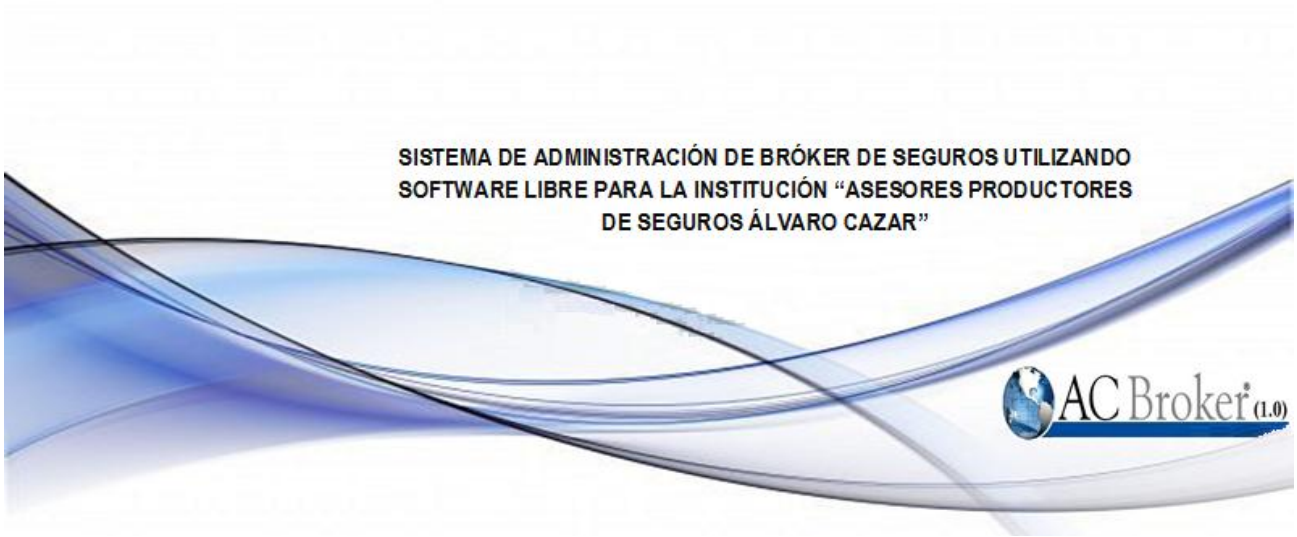
Fuente: Propia

CAPITULO V

FASE DE ELABORACIÓN

ESPECIFICACIÓN DE CASOS DE USO





5. ESPECIFICACIONES DE LOS CASOS DE USO

5.1 Escenario 1 (Usuario Navega en el Sistema)

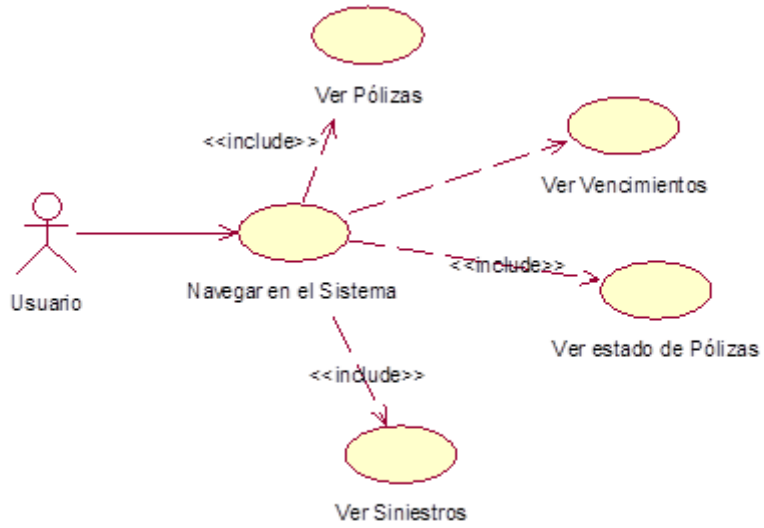


Figura 16 Usuario navegando en el sistema

Fuente: Propia

5.1.1 Caso de Uso (Navegar en el Sistema)

a) Breve Descripción

Este caso de uso describe el proceso que se genera al momento que el usuario ingresa a la web para obtener su información del Bróker de seguros.

b) Flujo básico de eventos

- El usuario debe ingresar a la web y poner la dirección web para poder acceder al sistema.
- Una vez ingresado la dirección, debemos esperar a que se cargue la página e ingresar su password y clave para escoger que tipo de información desea.

c) Flujo Alternativo

- Puede ingresar a la información escogiendo las opciones del menú.

- También puede seleccionar cualquier otra opción de la barra de herramientas.

5.1.2 Caso de Uso (Ver Pólizas)

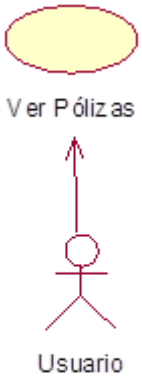


Figura 17 Ver Pólizas

Fuente: Propia

a) Breve Descripción

Este caso de uso describe el proceso del usuario, cuando ya se encuentra accediendo a la información sobre las pólizas ya adquiridas por él.

b) Flujo básico de eventos

- Para ver las pólizas que el usuario a adquirido deberá escoger en el menú la opción de “Pólizas Adquiridas”.
- Una vez que haya dado click en la opción puede ver todas las pólizas que ha comprado.
- Una vez que puede ver el listado de las pólizas puede escoger la opción de “detalle”, para poder ver todos los detalles de la póliza escogida.
- Una vez que haya hecho click en detalle podrá observar todos los detalles de la póliza seleccionada.

5.1.3 Caso de Uso (Ver Vencimientos)

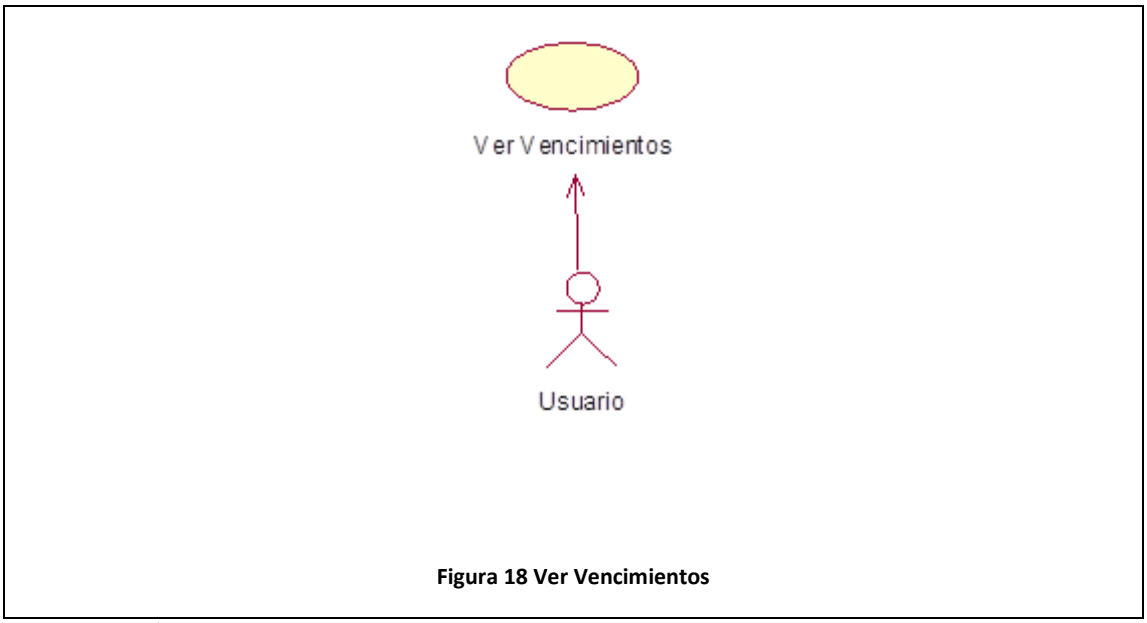


Figura 18 Ver Vencimientos

Fuente: Propia

a) Breve Descripción

Este caso de uso describe el proceso del usuario, cuando desea conocer los vencimientos sobre sus pólizas.

b) Flujo básico de eventos

- Para ver los vencimientos que el usuario tiene debe escoger la opción de vencimientos ya sea por número de póliza o por vencimientos entre un rango de fechas.
- Una vez que haya escogido la opción deseada el usuario podrá observar los vencimientos de las pólizas.

c) Precondiciones

- Haber escogido la opción por número de póliza, e ingresado el número de la misma.
- Haber escogido la opción por fechas, e ingresado la fecha inicial y final.

5.1.4 Caso de Uso (Ver Estado de Pólizas)

Figura 19 Ver estado de Pólizas

Fuente: Propia

a) Breve Descripción

Este caso de uso describe el proceso del usuario, cuando desea conocer los estados sobre sus pólizas.

b) Flujo básico de eventos

- Para ver los estados que el usuario tiene debe escoger la opción de estado de póliza.
- Una vez que haya escogido la opción deseada el usuario podrá observar los estados de sus pólizas.

c) Precondiciones

- Haber escogido la opción estado de póliza, e ingresado el número de la misma.

5.1.5 Caso de Uso(Ver Siniestros)

Figura 20 Ver Siniestros

Fuente: Propia

a) Breve Descripción

Este caso de uso describe el proceso del usuario, cuando desea conocer los siniestros sobre sus pólizas.

b) Flujo básico de eventos

- Para ver los siniestros que el usuario tiene debe escoger la opción de siniestros.
- Una vez que haya dado click en la opción puede ver todos los siniestros que tiene.
- Una vez que puede ver el listado de los siniestros puede escoger la opción de “detalle”, para poder ver todos los detalles del siniestro.

5.2 Escenario 2 (Administrador / Bróker de Seguros)

5.2.1 Caso de Uso (Autentificar / Logear)

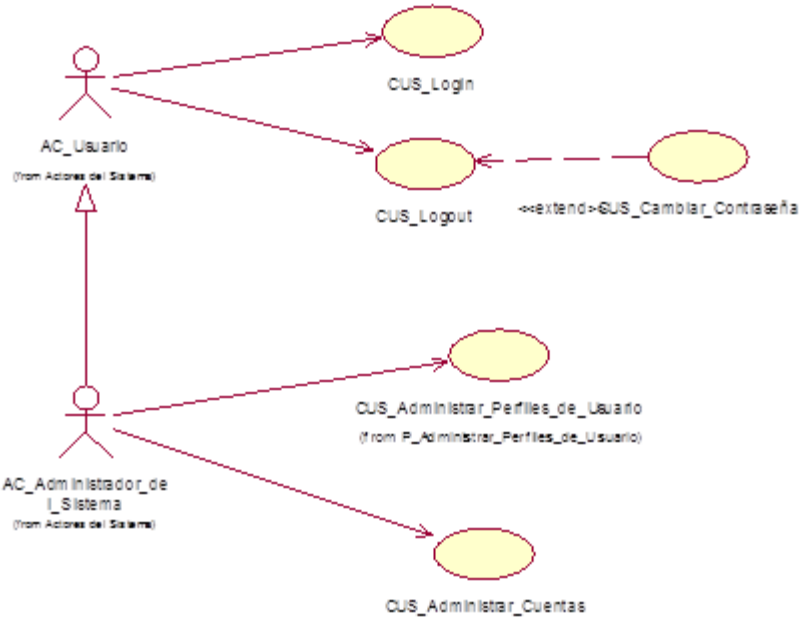


Figura 21 Autentificación y Seguridad del Sistema

Fuente: Propia

a) Breve Descripción

Este caso de uso describe el proceso de seguridad del sistema o logeo que el administrador del bróker debe realizar para que los usuarios del sistema puedan acceder y realizar las tareas de acuerdo a sus permisos o prioridades.

b) Flujo básico de eventos

- El usuario debe acceder al sistema mediante el ingreso a la página web.
- Luego si el usuario es cliente del bróker, podrá seleccionar la categoría que posee y finalmente ingresar la clave.
- Con esto ya podrá ingresar y ver la información que desea como cliente del bróker, como también a cambiar su clave si desea.
- Ahora si el usuario es empleado del bróker, podrá seleccionar su categoría y finalmente ingresar la clave.

- Con esto el empleado ya podrá acceder a realizar las operaciones a las que tiene acceso como ingresos, modificaciones, a cambiar su clave si desea, perfiles, cuentas, etc.

c) Flujo Alternativo

- El usuario puede cancelar la autentificación de ingreso al sistema
- Formulario de ingreso

d) Precondiciones

- Ser administrador o empleado del bróker.
- Ser cliente del bróker.

5.3 Escenario 3 (Administración de Compañías y Seguros)

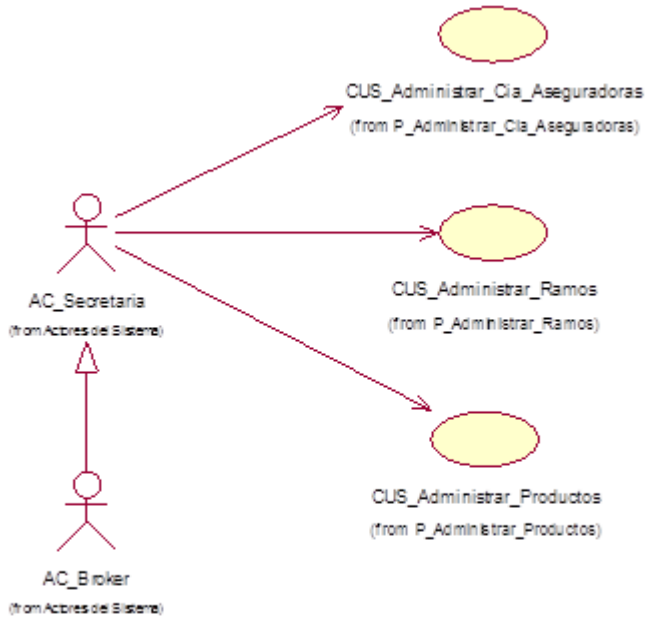


Figura 22 Administración de Compañías y Seguros

Fuente: Propia

5.3.1 Caso de Uso (Administrar Compañías Aseguradoras)

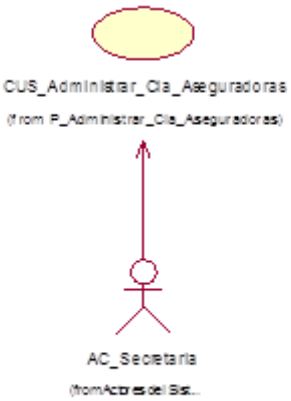


Figura 23 Administrar Compañías

Fuente: Propia

a) Breve Descripción

El caso de uso comienza cuando el usuario indica “Administrar Compañías”. Según su requerimiento el usuario puede agregar, modificar o eliminar la información de las compañías. El caso de uso termina cuando los datos de la compañía quedan actualizados.

b) Flujo básico de eventos

- El usuario una vez en el sistema selecciona la opción “Administrar Compañías”.
- El sistema carga la pantalla “Administrar Compañías” mostrando los criterios de búsqueda de nombre de compañía y RUC.
- El usuario selecciona la operación que desea realizar.
    - **Si elige “Buscar” revisar Subflujo “Buscar Compañía”.**
    - **Si elige “Seleccionar” revisar Subflujo “Modificar Compañía”.**
    - **Si elige “Eliminar” revisar Subflujo “Eliminar Compañía”.**
    - **Si elige “Nuevo” revisar Subflujo “Registrar Compañía”.**
- El usuario selecciona “Salir”.
- El sistema cierra la interface “Administrar Compañías” y el caso de uso termina.

- **Subflujos**
- Buscar Compañía

El usuario utiliza uno de los dos criterios de búsqueda nombre de compañía y/o RUC.
El sistema muestra una lista de las compañías coincidentes con los campos de: nombre, RUC y
teléfono.
El caso de uso continúa en el punto de seleccionar operaciones del Flujo Básico

- Modificar Compañía

El sistema muestra la ventana “Modificar Compañía” con todos los datos que se pueden modificar:
razón social, RUC, direccion1, direccion2, teléfono central y website.

El usuario modifica los datos de la compañía. Luego selecciona “Modificar”.

El sistema muestra un mensaje solicitando confirmación de modificación.
El usuario confirma modificación.
El sistema verifica que se hayan ingresado todos los datos y que estos

sean válidos.
El sistema modifica el registro de la compañía.
El sistema cierra la ventana “Modificar Compañía”, actualiza el listado de las compañías.

- Eliminar Compañía

El sistema muestra la ventana “Eliminar Compañía” con todos los datos como solo lectura: razón social, RUC, direccion1, direccion2, teléfono central y website.

El usuario selecciona “Eliminar”.
El sistema muestra un mensaje solicitando confirmación de eliminación.
El usuario confirma eliminación
El sistema elimina el registro de la compañía.
El sistema cierra la ventana “Eliminar Compañía”, actualiza el listado de las compañías.

- Registrar Producto

El sistema muestra la ventana “Registrar Compañía” con los campos en blanco para ingresar
los datos: razón social, RUC, direccion1, direccion2, direccion3, direccion4, teléfono central y
website.

El usuario ingresa los campos y selecciona “Registrar”.
El sistema muestra un mensaje solicitando confirmación de registro.
El usuario confirma el registro.
El sistema verifica que se hayan ingresado todos los datos y que estos sean válidos.
El sistema registra una nueva compañía.
El sistema cierra la ventana “Registrar Compañía”, actualiza el listado de las compañías.

c) Flujos Alternativos

- Lista vacía

El sistema no encuentra compañías coincidentes con los criterios de búsqueda y muestra mensaje
Indicándolo.
El usuario acepta el mensaje mostrado.
El sistema cierra el mensaje y el caso de uso continúa.

- Datos incompletos

Si los datos de la compañía ingresados al sistema están incompletos o son

inválidos.
El sistema detecta que no se ha registrado información en todos los campos obligatorios o que estos son inválidos y muestra mensaje indicando.
El usuario acepta el mensaje mostrado.
El sistema cierra el mensaje y el caso de uso continúa.

- Cancelar proceso

Si el usuario no desea guardar los cambios en la información de la compañía.
El usuario selecciona “Cancelar” en la ventana “Registrar Compañía” o “Modificar Compañía” o “Eliminar Compañía”.
El sistema cierra la ventana correspondiente y el caso de uso continúa.

- No es posible eliminación

Si la compañía a eliminar cuenta actualmente con póliza vigente.
El sistema detecta que la compañía a eliminar cuenta con póliza vigente y muestra mensaje indicándolo.
El usuario acepta el mensaje mostrado.
El sistema cierra el mensaje y el caso de uso continúa.

d) Precondiciones

Para poder ejecutar este caso de uso se requiere del acceso exitoso del usuario al sistema

e) Post condiciones

Al grabar los cambios, la información de la compañía queda actualizada, ya sea con un registro ingresado, modificado o eliminado.

5.3.2 Caso de Uso (Administrar Ramos)

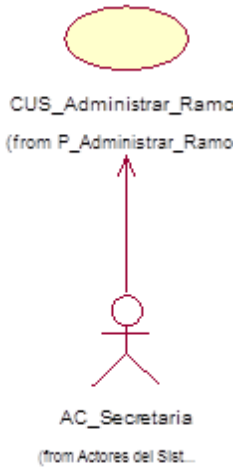


Figura 24 Administrar Ramos

Fuente: Propia

a) Breve Descripción

El caso de uso comienza cuando el usuario indica "Administrar Ramos Por Compañía". Según su requerimiento el usuario puede agregar, modificar o eliminar la información de los ramos por compañía. El caso de uso termina cuando los datos de los ramos por compañía quedan actualizados.

b) Flujo básico de eventos

El usuario una vez en el sistema selecciona la opción "Administrar Ramos por Compañía".
El sistema carga la pantalla "Administrar Ramos por Compañía" mostrando los criterios de búsqueda de compañía y ramo.

El usuario selecciona una operación que desea realizar.

- **Si elige "Buscar" revisar Subflujo "Buscar Ramo Por Compañía".**
- **Si elige "Seleccionar" revisar Subflujo "Modificar Ramo por Compañía".**
- **Si elige "Eliminar" revisar Subflujo "Eliminar Ramo por Compañía".**
- **Si elige "Nuevo" revisar Subflujo "Registrar Ramo por Compañía".**

El usuario selecciona "Salir".
El sistema cierra la interface "Administrar Ramo por Compañía" y el caso de uso termina.

- Subflujos

- Buscar Ramo por Compañía

  El usuario utiliza uno de los dos criterios de búsqueda compañía y/o ramo.
  El sistema muestra una lista de los ramos por compañía coincidentes con los campos de: Compañía, ramo y porcentaje de comisión.

- Modificar Ramo por Compañía

  El sistema muestra la ventana “Modificar Ramo por Compañía” con todos los datos que se pueden modificar: ramo, compañía y porcentaje de comisión.
  El usuario modifica los datos del ramo por compañía. Luego selecciona “Modificar”.

  El sistema muestra un mensaje solicitando confirmación de modificación.
  El usuario confirma modificación.
  El sistema verifica que se hayan ingresado todos los datos y que estos sean válidos.
  El sistema modifica el registro del ramo por compañía.
  El sistema cierra la ventana “Modificar Ramo por Compañía”, actualiza el listado de los ramos por compañía.

- Eliminar Ramo por Compañía

  El sistema muestra la ventana “Eliminar Ramo por Compañía” con todos los datos como solo lectura: ramo, compañía y porcentaje de comisión.
  El usuario selecciona “Eliminar”.
  El sistema muestra un mensaje solicitando confirmación de eliminación.
  El usuario confirma eliminación.
  El sistema elimina el registro del ramo por compañía.
  El sistema cierra la ventana “Eliminar Ramo por Compañía”, actualiza el listado de los ramos por compañía.

- Registrar Ramo por Compañía

  El sistema muestra la ventana “Registrar Ramo por Compañía” con los campos en blanco para ingresar los datos: compañía, ramo y porcentaje de comisión.
  El usuario ingresa los campos y selecciona “Registrar”.
  El sistema muestra un mensaje solicitando confirmación de registro.
  El usuario confirma el registro.
  El sistema verifica que se hayan ingresado todos los datos y que estos sean válidos.
  El sistema registra un nuevo ramo por compañía.
  El sistema cierra la ventana “Registrar Ramo por Compañía”, actualiza el listado de los ramos por compañía.

c) Flujos Alternativos

- Lista vacía

El sistema no encuentra ramos y/o compañía coincidentes con los criterios de búsqueda y muestra mensaje indicándolo.
El usuario acepta el mensaje mostrado.
El sistema cierra el mensaje y el caso de uso continúa.

- Datos incompletos

Si los datos del ramo por compañía ingresados al sistema están incompletos o son inválidos.
El sistema detecta que no se ha registrado información en todos los campos obligatorios o que estos son inválidos y muestra mensaje indicándolo.
El usuario acepta el mensaje mostrado.
El sistema cierra el mensaje y el caso de uso continúa.

- Cancelar proceso

Si el usuario no desea guardar los cambios de la información del ramo por compañía.
El usuario selecciona “Cancelar” en la ventana “Registrar Ramo por Compañía” o “Modificar
Ramo por Compañía” o “Eliminar Ramo por Compañía”.
El sistema cierra la ventana correspondiente y el caso de uso continúa.

- No es posible eliminación

Si el ramo por compañía a eliminar cuenta actualmente con pólizas vigentes.
El sistema detecta que el ramo por compañía a eliminar cuenta con póliza vigente y muestra
mensaje indicándolo.
El usuario acepta el mensaje mostrado.
El sistema cierra el mensaje y el caso de uso continúa.

d) Precondiciones

Para poder ejecutar este caso de uso se requiere del acceso exitoso del usuario al sistema

e) Post condiciones

Al grabar los cambios, la información del rapo por compañía queda actualizada, ya sea con un registro ingresado, modificado o eliminado.

5.3.3 Caso de Uso (Administrar Productos)

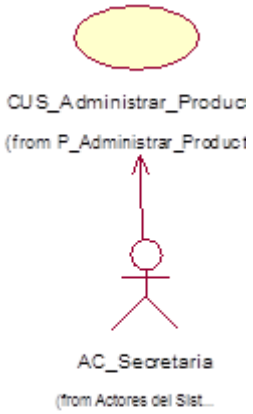


Figura 25 Administrar Productos

Fuente: Propia

a) Breve Descripción

El caso de uso comienza cuando el usuario indica "Administrar Productos". Según su requerimiento el usuario puede agregar, modificar o eliminar la información de los productos. El caso de uso termina cuando los datos del producto quedan actualizados.

b) Flujo básico de eventos

El usuario una vez en el sistema selecciona la opción "Administrar Productos".
El sistema carga la pantalla "Administrar Productos" mostrando los criterios de búsqueda de compañía y ramo y producto.

El usuario selecciona una operación que desea realizar.

- **Si elige "Buscar" revisar Subflujo "Buscar Producto".**
- **Si elige "Seleccionar" revisar Subflujo "Modificar Producto".**
- **Si elige "Eliminar" revisar Subflujo "Eliminar Producto".**
- **Si elige "Nuevo" revisar Subflujo "Registrar Producto".**

El usuario selecciona "Salir".
El sistema cierra la interface "Administrar Producto" y el caso de uso termina.

Subflujos

- Buscar Producto

El usuario utiliza uno de los dos criterios de búsqueda compañía, ramo y/o producto.
El sistema muestra una lista de los productos coincidentes con los campos de: Producto, compañía y ramo.

- Modificar Producto

El sistema muestra la ventana "Modificar Producto" con todos los datos que se pueden modificar: ramo, compañía y Producto.

El usuario modifica los datos del Producto. Luego selecciona “Modificar”.
El sistema muestra un mensaje solicitando confirmación de modificación.
El usuario confirma modificación.
El sistema verifica que se hayan ingresado todos los datos y que estos sean válidos.
El sistema modifica el registro del Producto.
El sistema cierra la ventana “Modificar Producto”, actualiza el listado de los ramos por compañía.

- Eliminar Producto

El sistema muestra la ventana “Eliminar Producto” con todos los datos como solo lectura: ramo, compañía y Producto.
El usuario selecciona “Eliminar”.
El sistema muestra un mensaje solicitando confirmación de eliminación.
El usuario confirma eliminación.
El sistema elimina el registro del producto.
El sistema cierra la ventana “Eliminar producto”, actualiza el listado de los productos.

- Registrar Producto

El sistema muestra la ventana “Registrar Producto” con los campos en blanco para ingresar los datos: compañía, ramo y producto.
El usuario ingresa los campos y selecciona “Registrar”.
El sistema muestra un mensaje solicitando confirmación de registro.
El usuario confirma el registro.
El sistema verifica que se hayan ingresado todos los datos y que estos sean válidos.
El sistema registra un nuevo producto
El sistema cierra la ventana “Registrar Producto”, actualiza el listado de los productos.

c) Flujos Alternativos

- Lista vacía

El sistema no encuentra productos coincidentes con los criterios de búsqueda y muestra mensaje indicándolo.
El usuario acepta el mensaje mostrado.
El sistema cierra el mensaje y el caso de uso continúa.

- Datos incompletos

Si los datos del producto ingresados al sistema están incompletos o son inválidos.
El sistema detecta que no se ha registrado información en todos los campos obligatorios o que estos son inválidos y muestra mensaje indicándolo.
El usuario acepta el mensaje mostrado.
El sistema cierra el mensaje y el caso de uso continúa.

- Cancelar proceso

  Si el usuario no desea guardar los cambios de la información del producto.
  El usuario selecciona “Cancelar” en la ventana “Registrar Producto” o “Modificar
  Producto” o “Eliminar Producto”.
  El sistema cierra la ventana correspondiente y el caso de uso continúa.

- No es posible eliminación

  Si el producto a eliminar cuenta actualmente con póliza vigente.
  El sistema detecta que el producto a eliminar cuenta con póliza vigente y muestra
  mensaje indicándolo.
  El usuario acepta el mensaje mostrado.
  El sistema cierra el mensaje y el caso de uso continúa.

d) Precondiciones

Para poder ejecutar este caso de uso se requiere del acceso exitoso del usuario al sistema

e) Post condiciones

Al grabar los cambios, la información del producto queda actualizada, ya sea con un registro ingresado, modificado o eliminado.

5.4 Escenario 4 (Administración de Clientes, Pólizas)

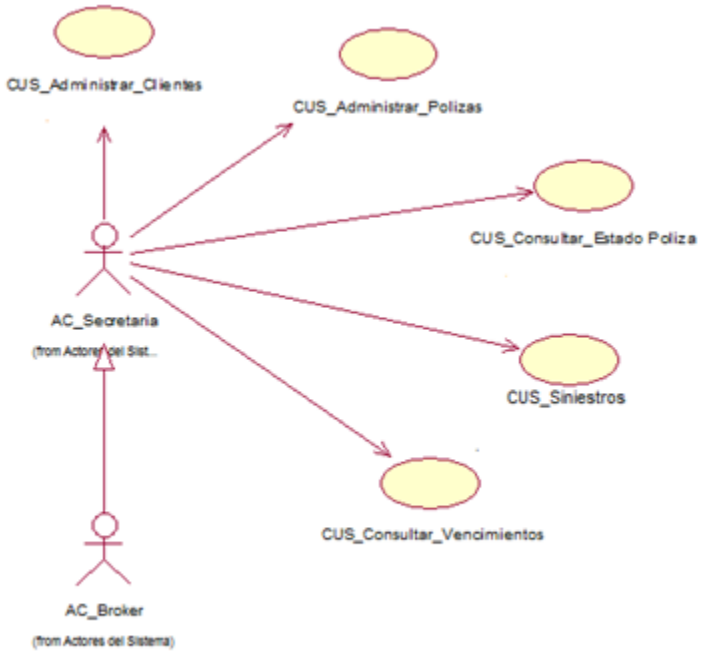


Figura 26 Administración de Clientes y Pólizas

Fuente: Propia

5.4.1 Caso de Uso (Administrar Clientes)

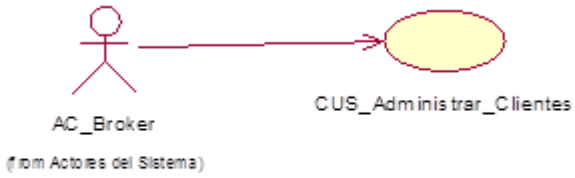


Figura 27 Administración de Clientes

Fuente: Propia

a) Breve Descripción

El caso de uso comienza cuando el usuario indica “Administrar Contratantes”. Según su requerimiento el usuario puede agregar, modificar o eliminar la información de los clientes. El caso de uso termina cuando los datos del cliente quedan actualizados.

b) Flujo básico de eventos

El usuario una vez en el sistema selecciona la opción “Administrar Contratantes”.
El sistema carga la pantalla “Administrar Contratantes” mostrando el criterio de búsqueda de nombre de contratante.
El usuario selecciona una operación que desea realizar.
Si elige “Buscar” revisar SubFlujo “Buscar Contratante”.

Si elige “Seleccionar” revisar SubFlujo “Modificar Contratante”.
Si elige “Eliminar” revisar Subflujo “Eliminar Contratante”.
Si elige “Nuevo” revisar Subflujo “Registrar Contratante”.

El usuario selecciona “Salir”.
El sistema cierra la interface “Administrar Contratantes” y el caso de uso termina.

- Subflujos

- Buscar Contratante

El usuario utiliza el criterio de búsqueda por nombre del contratante.
El sistema muestra una lista de los contratantes coincidentes con los campos: contratante, tipo de contratante, tipo de documento y numero de documento.

- Modificar Contratante

El sistema muestra la ventana “Modificar Contratante” con todos los datos que se pueden modificar: contratante, beneficiario, código, tipo de persona, tipo de documento, numero de documento, contacto, fecha de nacimiento/aniversario, direccion1, direccion2, telefono1, telefono2, celular1, celular2, email1, email2 y observaciones.
El usuario modifica los datos del contratante. Luego selecciona “Modificar”.
El sistema muestra un mensaje solicitando confirmación de modificación.
El usuario confirma modificación.
El sistema verifica que se hayan ingresado todos los datos y que estos sean válidos.
El sistema modifica el registro del contratante.
El sistema cierra la ventana “Modificar Contratante”, actualiza el listado de los contratantes y el caso de uso continúa.

- Eliminar Contratante

El sistema muestra la ventana “Eliminar Contratante” con todos los datos como solo lectura: contratante, beneficiario, código, tipo de persona, tipo de persona, tipo de documento, numero de documento, contacto, fecha de nacimiento/aniversario, direccion1, direccion2, telefono1, telefono2, celular1, celular2, email1, email2 y observaciones.
El usuario selecciona “Eliminar”.
El sistema muestra un mensaje solicitando confirmación de eliminación.
El usuario confirma eliminación.
El sistema elimina el registro del contratante.
El sistema cierra la ventana “Eliminar Contratante”, actualiza el listado de los contratantes y el caso de uso continúa.

- Registrar Contratante

El sistema muestra la ventana “Registrar Contratante” con los campos en blanco para ingresar los datos: contratante, beneficiario, código, tipo de persona, tipo de documento, numero de documento, contacto, fecha de nacimiento/aniversario, direccion1, direccion2, telefono1, telefono2, celular1, celular2, email1, email2 y observaciones.
El usuario ingresa los campos y selecciona “Registrar”.
El sistema muestra un mensaje solicitando confirmación de registro.
El usuario confirma el registro.
El sistema verifica que se hayan ingresado todos los datos y que estos sean válidos.
El sistema registra un nuevo contratante.
El sistema cierra la ventana “Registrar Contratante”, actualiza el listado de los contratantes y el caso de uso continúa.

c) Flujos Alternativos

- Lista vacía

El sistema no encuentra contratantes coincidentes con los criterios de búsqueda y muestra mensaje indicándolo.
El usuario acepta el mensaje mostrado.
El sistema cierra el mensaje y el caso de uso continúa.

- Datos incompletos

Si los datos del contratante ingresados al sistema están incompletos o son inválidos.
El sistema detecta que no se ha registrado información en todos los campos obligatorios o que estos son inválidos y muestra mensaje indicándolo.
El usuario acepta el mensaje mostrado.
El sistema cierra el mensaje y el caso de uso continúa.

- Cancelar proceso

Si el usuario no desea guardar los cambios de la información del contratante.

El usuario selecciona “Cancelar” en la ventana “Registrar contratante” o “Modificar
Producto” o “Eliminar contratante”.
El sistema cierra la ventana correspondiente y el caso de uso continúa.

- No es posible eliminación

Si el contratante a eliminar cuenta actualmente con póliza vigente.
El sistema detecta que el contratante a eliminar cuenta con póliza vigente y muestra
mensaje indicándolo.
El usuario acepta el mensaje mostrado.
El sistema cierra el mensaje y el caso de uso continúa.

d) Precondiciones

Para poder ejecutar este caso de uso se requiere del acceso exitoso del usuario al sistema

e) Post condiciones

Al grabar los cambios, la información del contratante queda actualizada, ya sea con un registro ingresado, modificado o eliminado.

5.4.2 Caso de Uso (Administrar Pólizas)

Figura 28 Administración de pólizas

Fuente: Propia

a) Breve Descripción

El caso de uso comienza cuando el usuario indica “Administrar Pólizas”. Según su requerimiento el usuario puede agregar, modificar o eliminar la información de las pólizas. El caso de uso termina cuando los datos de la póliza quedan actualizados.

b) Flujo básico de eventos

El usuario una vez en el sistema selecciona la opción “Administrar Póliza”.

El sistema carga la pantalla “Administrar Póliza” mostrando el criterio de búsqueda por compañía, ramo, plan y/o producto.
El usuario selecciona una operación que desea realizar.
Si elige “Buscar” revisar SubFlujo “Buscar Póliza”.

Si elige “Seleccionar” revisar SubFlujo “Modificar Póliza”.
Si elige “Eliminar” revisar Subflujo “Eliminar Póliza”.
Si elige “Nuevo” revisar Subflujo “Registrar Póliza”.

El usuario selecciona “Salir”.
El sistema cierra la interface “Administrar Póliza” y el caso de uso termina.

- Subflujos

- Buscar Póliza

El usuario utiliza uno de los cuatro criterios de búsqueda compañía, ramo, plan y/o producto.
El sistema muestra una lista de las pólizas coincidentes con los campos de: Código de póliza, contratante, plan, fecha emisión, inicio de vigencia y fin de vigencia.

- Modificar Póliza

El sistema muestra la ventana “Modificar Póliza” con todos los datos que se pueden modificar: Compañía, producto, ramo, plan, contratante, fecha de emisión, comisión ganada, inicio de vigencia, fin de vigencia, prima, suma aseguradora, estado de póliza, código póliza CIA, asegurado, documento01, documento02 y documento03 .
El usuario modifica los datos de la póliza. Luego selecciona “Modificar”.
El sistema muestra un mensaje solicitando confirmación de modificación.
El usuario confirma modificación.
El sistema verifica que se hayan ingresado todos los datos y que estos sean válidos.
El sistema modifica el registro de la póliza.
El sistema cierra la ventana “Modificar Póliza”, actualiza el listado de las pólizas y el caso de uso continúa.

- Eliminar Póliza

El sistema muestra la ventana “Eliminar Producto” con todos los datos como solo lectura.
El usuario selecciona “Eliminar”.
El sistema muestra un mensaje solicitando confirmación de eliminación.
El usuario confirma eliminación.
El sistema elimina el registro de la póliza.
El sistema cierra la ventana “Eliminar Póliza”, actualiza el listado de las pólizas y el caso de uso continúa.

- Registrar Póliza

El sistema muestra la ventana “Registrar Póliza” con los campos en blanco para ingresar los datos: Compañía, producto, ramo, plan, contratante, fecha de emisión, comisión ganada, inicio de vigencia, fin de vigencia, prima, suma aseguradora, estado de póliza, código póliza CIA, asegurado, documento01, documento02 y documento03.
El usuario ingresa los campos y selecciona “Registrar”.
El sistema muestra un mensaje solicitando confirmación de registro.
El usuario confirma el registro.
El sistema verifica que se hayan ingresado todos los datos y que estos sean válidos.
El sistema registra una nueva póliza.
El sistema cierra la ventana “Registrar Póliza”, actualiza el listado de las pólizas y el caso de uso continúa.

c) Flujos Alternativos

- Lista vacía

El sistema no encuentra pólizas coincidentes con los criterios de búsqueda y muestra mensaje indicándolo.
El usuario acepta el mensaje mostrado.
El sistema cierra el mensaje y el caso de uso continúa.

- Datos incompletos

Si los datos de la póliza ingresados al sistema están incompletos o son inválidos.
El sistema detecta que no se ha registrado información en todos los campos obligatorios o que estos son inválidos y muestra mensaje indicándolo.
El usuario acepta el mensaje mostrado.
El sistema cierra el mensaje y el caso de uso continúa.

- Cancelar proceso

Si el usuario no desea guardar los cambios de la información de la póliza.
El usuario selecciona “Cancelar” en la ventana “Registrar contratante” o “Modificar
Producto” o “Eliminar póliza”.
El sistema cierra la ventana correspondiente y el caso de uso continúa.

- No es posible eliminación

Si la póliza a eliminar cuenta actualmente con periodo vigente.
El sistema detecta que la póliza a eliminar cuenta con un periodo vigente y muestra mensaje indicándolo.
El usuario acepta el mensaje mostrado.
El sistema cierra el mensaje y el caso de uso continúa.

d) Precondiciones

Para poder ejecutar este caso de uso se requiere del acceso exitoso del usuario al sistema

e) Post condiciones

Al grabar los cambios, la información del contratante queda actualizada, ya sea con un registro ingresado, modificado o eliminado.

5.4.3 Caso de Uso (Consultar Estado de Póliza)

Figura 29 Estado de póliza

Fuente: Propia

a) Breve Descripción

El caso de uso comienza cuando el usuario indica “Consultar Estado de Pólizas”. Según su requerimiento el usuario puede consultar el estado de las pólizas por contratante, código de póliza y estado de póliza. El caso de uso termina cuando el sistema muestra el estado de la póliza solicitada.

b) Flujo básico de eventos

El usuario una vez en el sistema selecciona la opción “Consultar estado de Póliza”.
El sistema carga la pantalla “Consultar estado de Póliza” mostrando el criterio de búsqueda por contratante, código de póliza, y estado de póliza.
El usuario selecciona un criterio de búsqueda que desea realizar.

Si elige “Contratante” revisar SubFlujo “Buscar por Contratante”.
Si elige “Código Póliza” revisar Subflujo “Buscar por Código de Póliza”.
Si elige “Estado Póliza” revisar Subflujo “Buscar por Estado de Póliza”.

El usuario selecciona “Salir”.
El sistema cierra la interface “Consultar Estado de Pólizas” y el caso de uso termina.

- Subflujos

- Buscar por Contratante

El usuario ingresa el contratante y selecciona “Buscar”.
El sistema muestra una lista de pólizas coincidentes con los campos de:

código de póliza, prima, fecha de emisión, estado de póliza y contratante.

- Buscar por Código de Póliza

  El usuario ingresa el código de póliza y selecciona “Buscar”.
  El sistema muestra una lista de pólizas coincidentes con los campos de: código de póliza, prima, fecha de emisión, estado de póliza y contratante.

- Buscar por Estado de Póliza

  El usuario ingresa el estado de póliza y selecciona “Buscar”.
  El sistema muestra una lista de pólizas coincidentes con los campos de: código de póliza, prima, fecha de emisión, estado de póliza y contratante.

c) Flujos Alternativos

- Lista vacía

  El sistema no encuentra pólizas coincidentes con los criterios de búsqueda y muestra mensaje indicándolo.
  El usuario acepta el mensaje mostrado.
  El sistema cierra el mensaje y el caso de uso continúa.

d) Precondiciones

Para poder ejecutar este caso de uso se requiere del acceso exitoso del usuario al sistema

e) Post condiciones

El Sistema muestra los datos del estado de la póliza consultada.

5.4.4 Caso de Uso (Administrar Siniestros)

Figura 30 Consultar Siniestros

Fuente: Propia

a) Breve Descripción

El caso de uso comienza cuando el usuario indica “Administrar Siniestros”. Según su requerimiento el usuario puede agregar, modificar o eliminar la información de los Siniestros. El caso de uso termina cuando los datos del siniestro quedan actualizados.

b) Flujo básico de eventos

El usuario una vez en el sistema selecciona la opción “Administrar Siniestros”.
El sistema carga la pantalla “Administrar Siniestros” mostrando el criterio de búsqueda por compañía, ramo, plan y/o producto, código de póliza.
El usuario selecciona una operación que desea realizar.
Si elige “Buscar” revisar SubFlujo “Buscar Siniestro”.

Si elige “Seleccionar” revisar SubFlujo “Modificar Siniestro”.
Si elige “Eliminar” revisar Subflujo “Eliminar Siniestro”.
Si elige “Nuevo” revisar Subflujo “Registrar Siniestro”.

El usuario selecciona “Salir”.
El sistema cierra la interface “Administrar Siniestros” y el caso de uso termina.

- Subflujos

- Buscar Siniestro

El usuario utiliza uno de los cuatro criterios de búsqueda compañía, ramo, plan y/o producto, código de póliza.
El sistema muestra una lista de los siniestros coincidentes con los campos de: Código de Sinestro, Código de póliza, contratante, plan, fecha emisión, inicio de vigencia y fin de vigencia, fecha del siniestro, fecha de notificación, fecha de entrega de documentos, detalle del siniestro, estado del siniestro.

- Modificar Siniestro

El sistema muestra la ventana “Modificar Siniestro” con todos los datos que se pueden modificar: fecha del siniestro, fecha de notificación, fecha de entrega de documentos, detalle del siniestro, estado del siniestro.

El usuario modifica los datos del siniestro. Luego selecciona “Modificar”.

El sistema muestra un mensaje solicitando confirmación de modificación.
El usuario confirma modificación.
El sistema verifica que se hayan ingresado todos los datos y que estos sean válidos.
El sistema modifica el registro del siniestro.
El sistema cierra la ventana “Administrar Siniestros”, actualiza el listado de los siniestros y el caso de uso continúa.

- Eliminar Siniestro

El sistema muestra la ventana “Eliminar Siniestro” con todos los datos como solo lectura.
El usuario selecciona “Eliminar”.
El sistema muestra un mensaje solicitando confirmación de eliminación.
El usuario confirma eliminación.
El sistema elimina el registro del siniestro.
El sistema cierra la ventana “Eliminar Siniestro”, actualiza el listado de los siniestros y el caso de uso continúa.

- Registrar Siniestro

El sistema muestra la ventana “Registrar Siniestro” con los campos en blanco para ingresar los datos: Código de póliza, contratante, fecha del siniestro, fecha de notificación, fecha de entrega de documentos, detalle del siniestro, estado del siniestro.
El usuario ingresa los campos y selecciona “Registrar”.
El sistema muestra un mensaje solicitando confirmación de registro.
El usuario confirma el registro.
El sistema verifica que se hayan ingresado todos los datos y que estos sean válidos.
El sistema registra un nuevo siniestro.
El sistema cierra la ventana “Registrar Siniestro”, actualiza el listado de los siniestros y el caso de uso continúa.

c) Flujos Alternativos

- Lista vacía

El sistema no encuentra siniestros coincidentes con los criterios de búsqueda y muestra mensaje indicándolo.
El usuario acepta el mensaje mostrado.
El sistema cierra el mensaje y el caso de uso continúa.

- Datos incompletos

Si los datos del siniestro ingresados al sistema están incompletos o son inválidos.
El sistema detecta que no se ha registrado información en todos los campos obligatorios o que estos son inválidos y muestra mensaje indicándolo.
El usuario acepta el mensaje mostrado.

El sistema cierra el mensaje y el caso de uso continúa.

- Cancelar proceso

Si el usuario no desea guardar los cambios de la información de l siniestro.
El usuario selecciona “Cancelar” en la ventana “Registrar siniestro” o “Modificar
siniestro” o “Eliminar siniestro”.
El sistema cierra la ventana correspondiente y el caso de uso continúa.

- No es posible eliminación

Si el siniestro a eliminar cuenta actualmente con periodo vigente.
El sistema detecta que la póliza a eliminar cuenta con un periodo vigente y muestra
mensaje indicándolo.
El usuario acepta el mensaje mostrado.
El sistema cierra el mensaje y el caso de uso continúa.

d) Precondiciones

Para poder ejecutar este caso de uso se requiere del acceso exitoso del usuario al sistema

e) Post condiciones

Al grabar los cambios, la información del siniestro queda actualizada, ya sea con un registro ingresado, modificado o eliminado.

5.4.5 Caso de uso (Consultar Vencimientos)

Figura 31 Consultar Vencimientos

Fuente: Propia

a) Breve Descripción

El caso de uso comienza cuando el usuario indica “Consultar Vencimientos”. Según su requerimiento el usuario puede consultar los vencimientos de las pólizas por contratante, código de póliza. El caso de uso termina cuando el sistema muestra el vencimiento solicitado.

b) Flujo básico de eventos

El usuario una vez en el sistema selecciona la opción “Consultar Vencimientos”.
El sistema carga la pantalla “Consultar Vencimientos” mostrando el criterio de búsqueda por contratante, código de póliza.
El usuario selecciona un criterio de búsqueda que desea realizar.

Si elige “Contratante” revisar SubFlujo “Buscar por Contratante, y rango de fechas”.
Si elige “Código Póliza” revisar Subflujo “Buscar por Código de Póliza y rango de fechas”.
Si elige “Código Póliza” revisar Subflujo “Buscar por Rango de fechas”.

El usuario selecciona “Salir”.
El sistema cierra la interface “Consultar Vencimientos” y el caso de uso termina.

- Subflujos

- Buscar por Contratante y rango de fechas

El usuario ingresa el contratante, rango de fechas y selecciona “Buscar”.
El sistema muestra una lista de pólizas coincidentes con los campos de: código de póliza, prima, fecha de emisión, estado de póliza y contratante.

- Buscar por Código de Póliza y rango de fechas

El usuario ingresa el código de póliza, rango de fechas y selecciona “Buscar”.
El sistema muestra una lista de vencimientos coincidentes con los campos

de: código de póliza, prima, fecha de emisión, fecha de inicio, fecha de finalización, estado de póliza y contratante.

- Buscar por Rango de fechas

  El usuario ingresa el rango de fechas y selecciona “Buscar”.
  El sistema muestra una lista de vencimientos coincidentes con los campos de: código de póliza, prima, fecha de emisión, fecha de inicio, fecha de finalización, estado de póliza y contratante.

c) Flujos Alternativos

- Lista vacía

  El sistema no encuentra vencimientos coincidentes con los criterios de búsqueda y muestra mensaje indicándolo.
  El usuario acepta el mensaje mostrado.
  El sistema cierra el mensaje y el caso de uso continúa.

d) Precondiciones

Para poder ejecutar este caso de uso se requiere del acceso exitoso del usuario al sistema

e) Post condiciones

El Sistema muestra los datos del estado del vencimiento consultada.

5.5 Escanario5 (Control Comisiones)

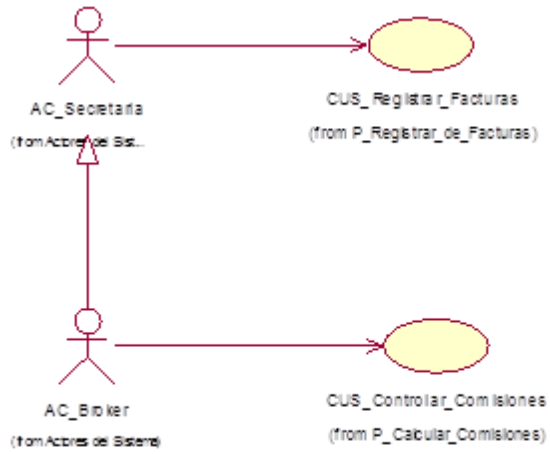


Figura 32 Control Comisiones

Fuente: Propia

5.5.1 Caso de Uso (Registrar Facturas)

Figura 33 Registrar Facturas

Fuente: Propia

a) Breve Descripción

El caso de uso comienza cuando el usuario indica “Administrar Facturas”. Según su requerimiento el usuario puede agregar, modificar o eliminar la información de las facturas. El caso de uso termina cuando los datos de la factura quedan actualizados.

b) Flujo básico de eventos

El usuario una vez en el sistema selecciona la opción “Administrar Facturas”.
El sistema carga la pantalla “Administrar Facturas” mostrando una lista de facturas.
El usuario selecciona una operación que desea realizar.

Si elige “Seleccionar” revisar Subflujo “Modificar Factura”.
Si elige “Eliminar” revisar Subflujo “Eliminar Factura”.
Si elige “Nuevo” revisar Subflujo “Registrar Factura”.

El usuario selecciona “Salir”.
El sistema cierra la interface “Administrar Facturas” y el caso de uso termina.

- Subflujos

- Modificar Factura

El sistema muestra la ventana “Modificar Factura” con todos los datos que se pueden modificar: Código de comprobante, número de comprobante, motivo, compañía, RUC, fecha emisión, fecha cobranza, monto total, monto calculado, monto pagado, impuesto, retención y observaciones.
El usuario modifica los datos de la factura. Luego selecciona “Modificar”.
El sistema muestra un mensaje solicitando confirmación de modificación.
El usuario confirma modificación.
El sistema verifica que se hayan ingresado todos los datos y que estos sean válidos.
El sistema modifica el registro de la factura.
El sistema cierra la ventana “Modificar Factura”, actualiza el listado de las facturas y el caso de uso continúa.

- Eliminar Factura

El sistema muestra la ventana “Eliminar Factura” con todos los datos como solo lectura: Código de comprobante, número de comprobante, motivo, compañía, RUC, fecha emisión, fecha cobranza, monto total, monto calculado, monto pagado, impuesto, retención y observaciones.
El usuario selecciona “Eliminar”.
El sistema muestra un mensaje solicitando confirmación eliminación.
El usuario confirma eliminación.
El sistema elimina el registro de la factura.
El sistema cierra la ventana “Eliminar Factura”, actualiza el listado de las facturas y el caso de uso continúa.

- Registrar Factura

El sistema muestra la ventana “Registrar Factura” con los campos en blanco para ingresar los datos: Código de comprobante, número de comprobante, motivo, compañía, RUC, fecha emisión, fecha cobranza, monto total, monto calculado, monto pagado, impuesto, retención y observaciones.
El usuario ingresa los datos y selecciona “Registrar”.
El sistema muestra un mensaje solicitando confirmación de registro.
El usuario confirma el registro.
El sistema verifica que se hayan ingresado todos los datos y que estos sean válidos.
El sistema registra una nueva factura.
El sistema cierra la ventana “Registrar Factura”, actualiza el listado de las

facturas y el caso de uso continúa.

c) Flujos Alternativos

- Datos incompletos

Si los datos de la factura ingresados al sistema están incompletos o son inválidos.

El sistema detecta que no se ha registrado información en todos los campos obligatorios o que estos son inválidos y muestra mensaje indicándolo.
El usuario acepta el mensaje mostrado.
El sistema cierra el mensaje y el caso de uso continúa.

- Cancelar proceso

Si el usuario no desea guardar los cambios en la información de la factura.
El usuario selecciona “Cancelar” en la ventana “Registrar Factura” o “Modificar Factura” o “Eliminar Factura”.
El sistema cierra la ventana correspondiente y el caso de uso continúa.

d) Precondiciones

Para poder ejecutar este caso de uso se requiere del acceso exitoso del usuario al sistema

e) Post condiciones

Al grabar los cambios, la información del siniestro queda actualizada, ya sea con un registro ingresado, modificado o eliminado.

5.5.2 Caso de Uso (Controlar comisiones)

Figura 34 Controlar Comisiones

Fuente: Propia

a) Breve Descripción

El caso de uso comienza cuando el usuario indica “Controlar Comisión”. La pantalla le muestra una búsqueda de fechas para determinar el periodo que desea consultar. Se muestra el listado de contratantes cuya fecha de inicio se

encuentra dentro del periodo consultado. De esta lista el bróker debe marcar cuales son los clientes que han realizado el primer pago de su prima y así se mostrará el monto calculado de la comisión.

b) Flujo Básico

El usuario una vez en el sistema selecciona la opción “Controlar Comisión”.
El sistema carga la pantalla “Controlar Comisión” con la grilla vacía. El sistema solicita el criterio de búsqueda por periodo que el usuario mismo ingresa.

El usuario ingresa el periodo que desea ver:
Elige una fecha para “Desde”
Elige una fecha para “Hasta”

El usuario selecciona “Buscar”.
El sistema muestra enlistado de contratantes cuya fecha de inicio se encuentra dentro del periodo consultado.
El usuario selecciona los contratantes que ya han realizado el primer pago de su prima.
El usuario selecciona “Calcular”.
Se muestra el monto calculado de la comisión.
El sistema cierra la interface “Controlar Comisión” y el caso de uso termina.

f) Precondiciones

Para poder ejecutar este caso de uso se requiere del acceso exitoso del usuario al sistema
Que existan pólizas registradas.

g) Post condiciones

El Sistema muestra el monto de la comisión calculada dentro del periodo seleccionado.

5.6 Escenario6 (Control Reporte Superintendencia De Bancos Y Seguros)

Figura 35 Control Reporte Superban

Fuente: Propia

a) Breve Descripción

El caso de uso comienza cuando el usuario indica “Controlar Reporte SuperBan”. La pantalla le muestra una búsqueda de fechas para determinar el periodo que desea consultar. Se muestra el reporte según el formato que

solicita la SUPERINTENDENCIA DE COMPANIAS Y SEGUROS cuya fecha de inicio se encuentra dentro del periodo consultado. De este se mostrará el monto calculado de la comisión percibida por Compañía y Ramo.

b) Flujo Básico

El usuario una vez en el sistema selecciona la opción “Controlar Reporte SuperBan”.
El sistema carga la pantalla “Controlar Reporte SuperBan” con la grilla vacía. El sistema solicita el criterio de búsqueda por periodo que el usuario mismo ingresa.

El usuario ingresa el periodo que desea ver:
Elige una fecha para “Desde”
Elige una fecha para “Hasta”

El usuario selecciona “Buscar”.
El sistema muestra el reporte cuya fecha de inicio se encuentra dentro del periodo consultado.
El usuario selecciona “aceptar”.
Se muestra el reporte con el monto calculado de la comisión por compañía y ramo.
El sistema cierra la interface “Controlar Reporte SuperBan” y el caso de uso termina.

h) Precondiciones

Para poder ejecutar este caso de uso se requiere del acceso exitoso del usuario al sistema
Que existan comisiones cobradas a las aseguradoras.

i) Post condiciones

El Sistema muestra el reporte con el monto calculado de la comisión por compañía y ramo dentro del periodo seleccionado.

CAPITULO VI

FASE DE CONSTRUCCIÓN

VISTA DE LA IMPLEMENTACIÓN DEL SISTEMA

6. VISTA E IMPLEMENTACIÓN DEL SISTEMA

En este capítulo se describen los principales diagramas que se generan de la aplicación web AC BROKER (1.0).

6.1 Modelo Dinámico del Sistema

El modelo dinámico del sistema consiste de dos diagramas principales que son:

- Diagramas de secuencia
- Diagrama de colaboración

6.1.1 Diagramas De Secuencia

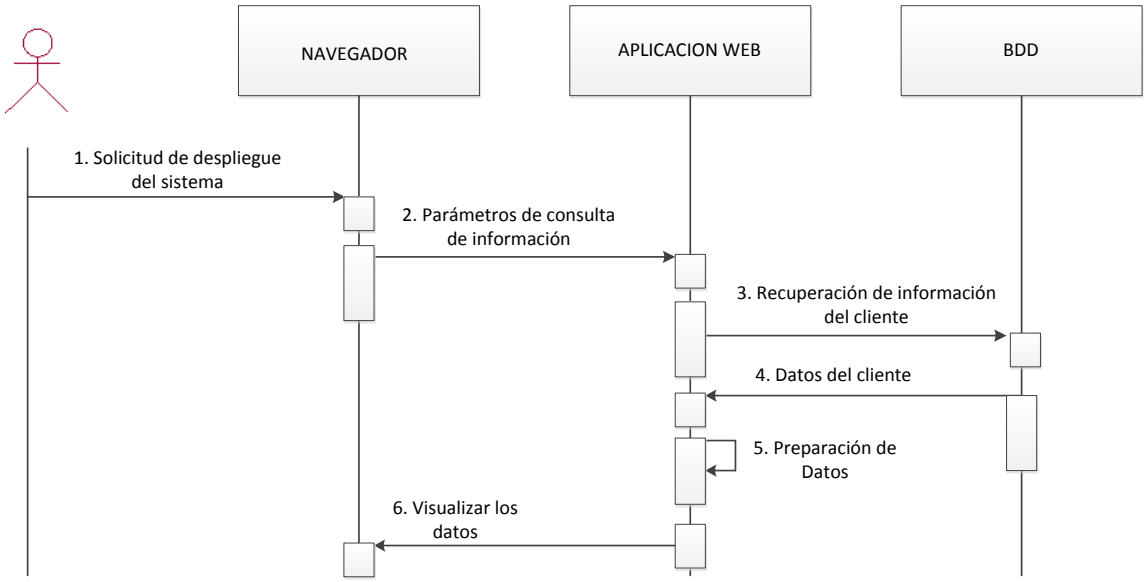
Los diagramas de secuencia describen a cada proceso (caso de uso) de la aplicación y su funcionalidad. Mientras que el diagrama de colaboración tiene la función de mostrar los elementos que interactúan en el sistema y la relación que sostienen entre ellos.

A continuación se listan los procesos que cubren los diagramas de secuencia:

- Despliegue de Información para los clientes del Bróker de Seguros
- Administración de Compañías y Seguros
- Administración de Clientes y Pólizas
- Control de Comisiones
- Control Reporte Superintendencia De Bancos y Seguros

6.1.1.1 Despliegue de Información para los clientes del Bróker de Seguros

Este módulo tiene la función de efectuar la visualización de la información del cliente del Bróker de Seguros a través del sistema. En este proceso existe una relación directa con la base de datos ya que a partir de estos se obtiene la información requerida por el cliente, tales como información sobre las pólizas adquiridas por el mismo, estado, vencimientos y siniestros de las mismas.

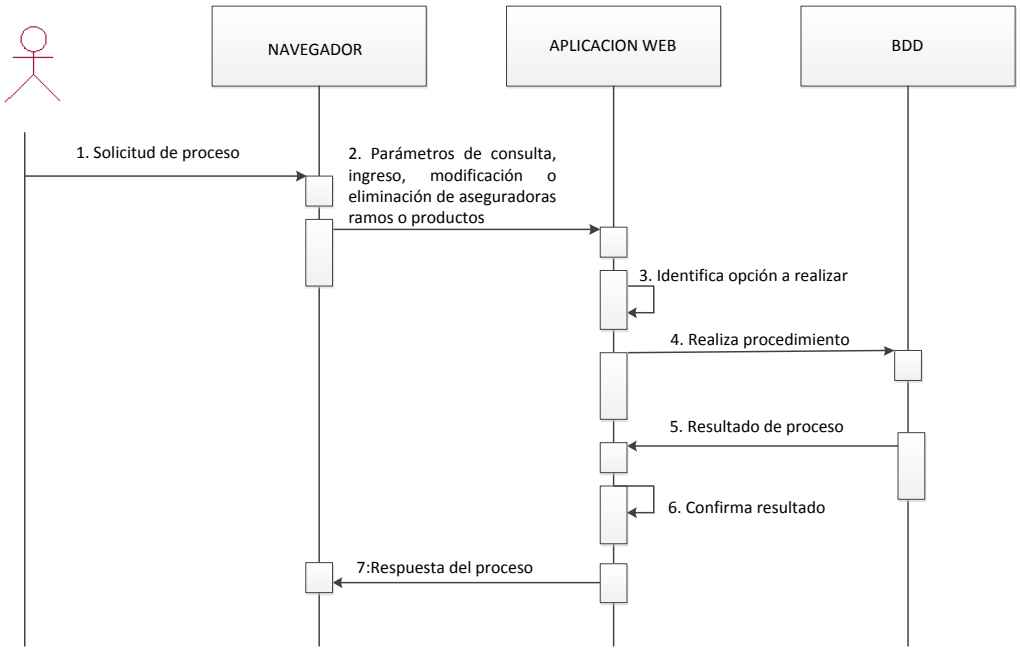


Fuente: Propia

Figura 36 Diagrama de secuencia del Cliente accediendo a su información

6.1.1.2 Administración de Compañías y Seguros

Este módulo tiene la función de efectuar la visualización, modificación, eliminar o registrar información del Bróker de Seguros a través del sistema. En este proceso existe una relación directa con la base de datos ya que a partir de estos se administra la información requerida por el usuario del bróker, tales como información sobre las Compañías y ramos con las que se trabaja al igual que los productos que las mismas ofrecen para ofertar a los clientes.

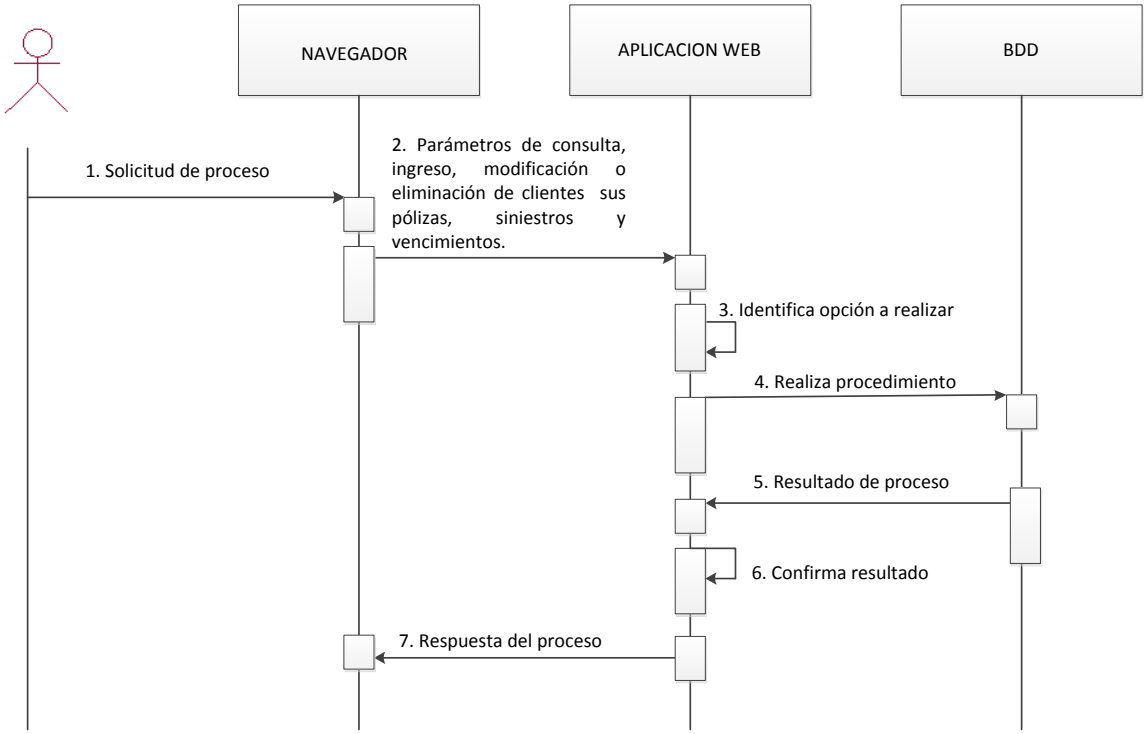


Fuente: Propia

Figura 37 Diagrama de Secuencia de Administración de Compañías y Seguros

6.1.1.3 Administración de Clientes y Pólizas

Este módulo tiene la función de efectuar la visualización, modificación, eliminar o registrar información del Bróker de Seguros a través del sistema. En este proceso existe una relación directa con la base de datos ya que a partir de estos se administra la información requerida por el usuario del bróker, tales como información sobre los Clientes del mismo así como la información sobre las pólizas, estado de pólizas, siniestros y vencimientos que los clientes adquirieron al Bróker de Seguros.

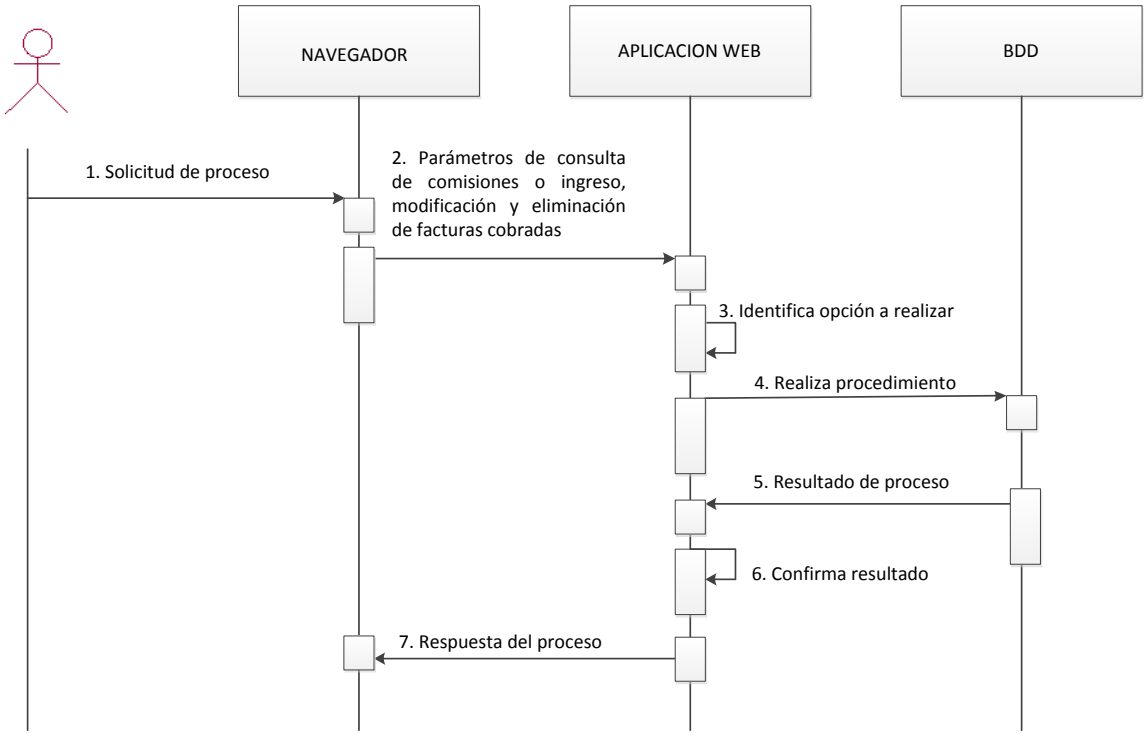


Fuente: Propia

Figura 38 Diagrama de secuencia de Administración de Clientes y Pólizas

6.1.1.4 Control de Comisiones

Este módulo tiene la función de efectuar la visualización y administración de las comisiones que genera el bróker de seguros con las Compañías que trabaja, administrando facturas cobradas y visualizando las comisiones generadas por el mismo.



Fuente: Propia

Figura 39 Diagrama de secuencia de Administrar Comisiones

6.1.1.5 Control Reporte Superintendencia De Bancos y Seguros

Este módulo tiene la función de efectuar la visualización de la información sobre el Reporte Anual para la Superintendencia de Bancos y Seguros que todos los brókers deben presentar obligatoriamente, permite la visualización del reporte ingresando una fecha de inicio y fin las cuales son definidas por el usuario.

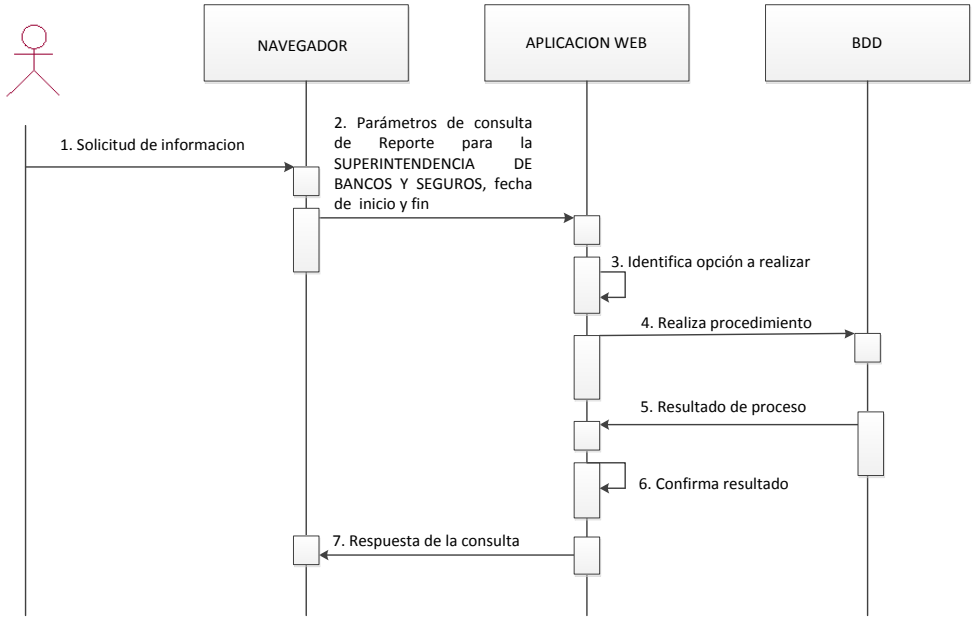


Figura 40 Diagrama de secuencia de Reporte Superintendencia de Bancos y Seguros

6.1.2 Diagramas de Colaboración

A continuación se presenta el diagrama de colaboración que se define para el sistema.

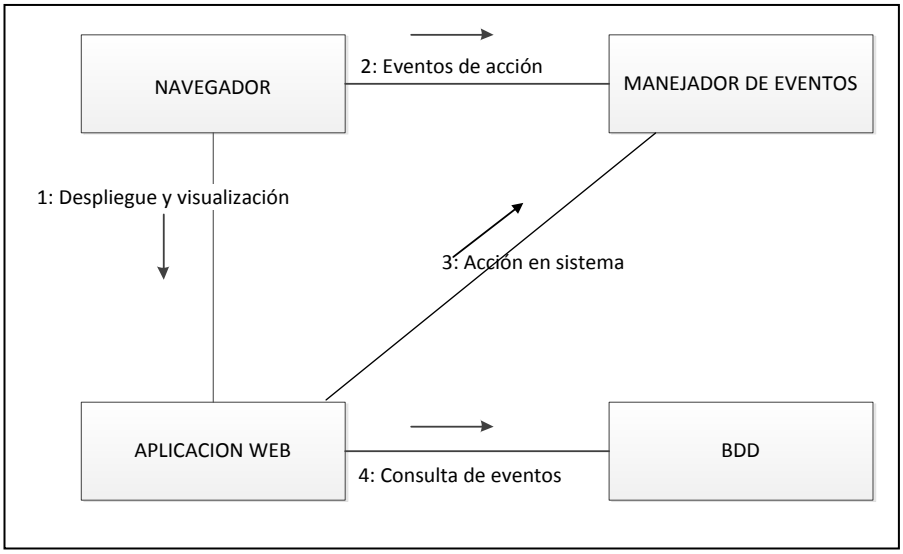


Figura 41 Diagrama de colaboración del sistema

Fuente: Propia

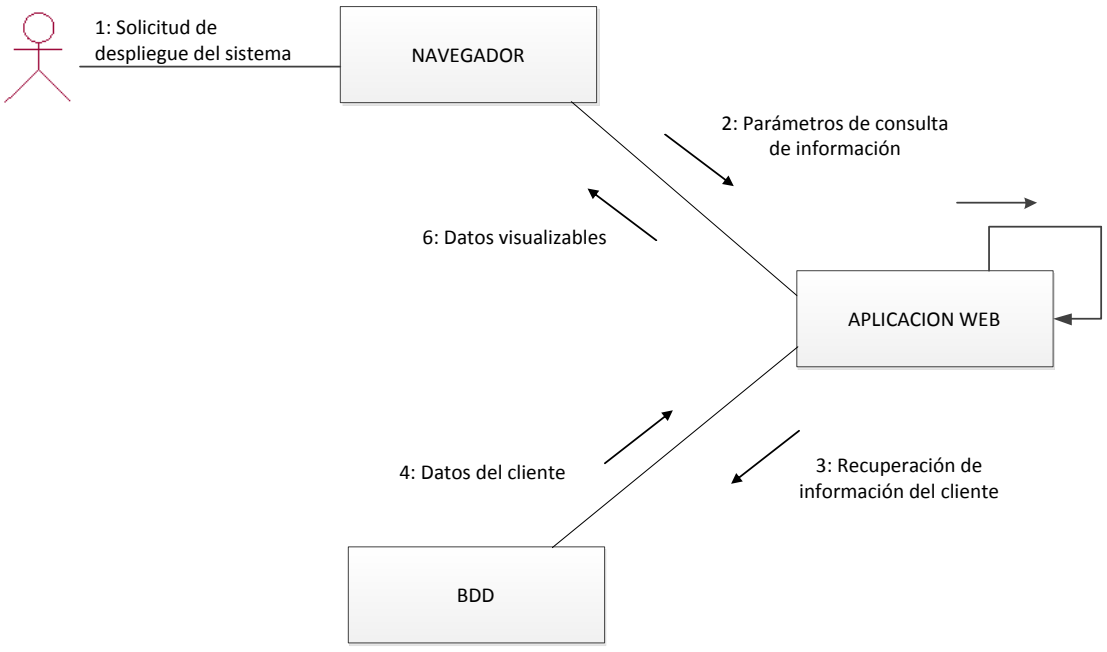


Figura 42 Diagrama de colaboración del Cliente navegando en el sistema

Fuente: Propia

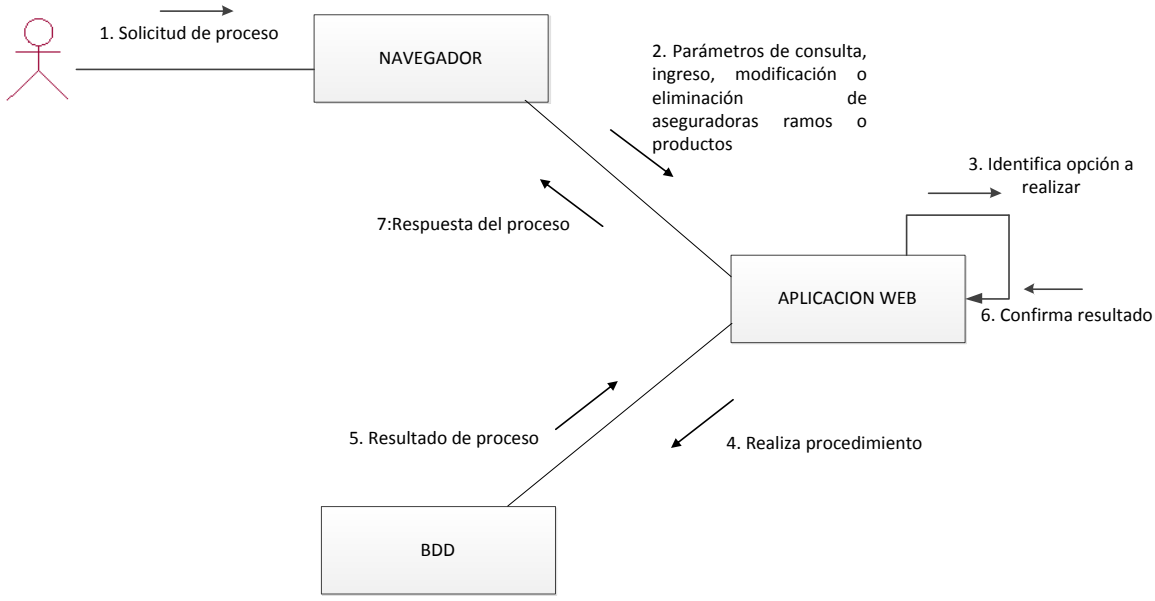


Figura 43 Diagrama de secuencia Administrando Compañías y Seguros

Fuente: Propia

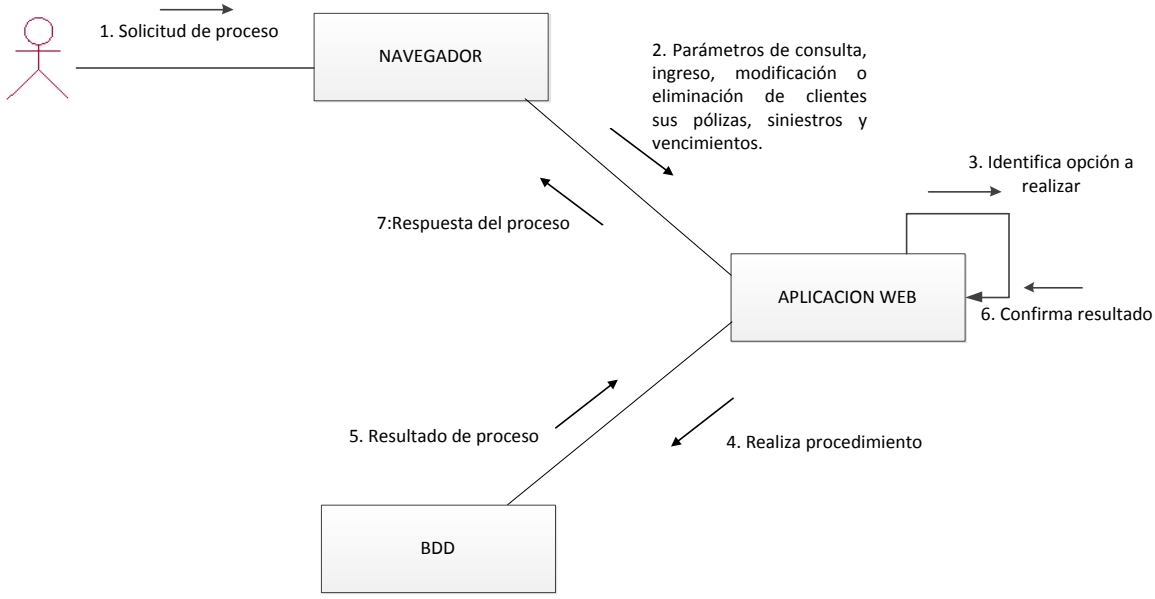


Figura 44 Diagrama de colaboración Administrar Clientes y Pólizas

Fuente: Propia

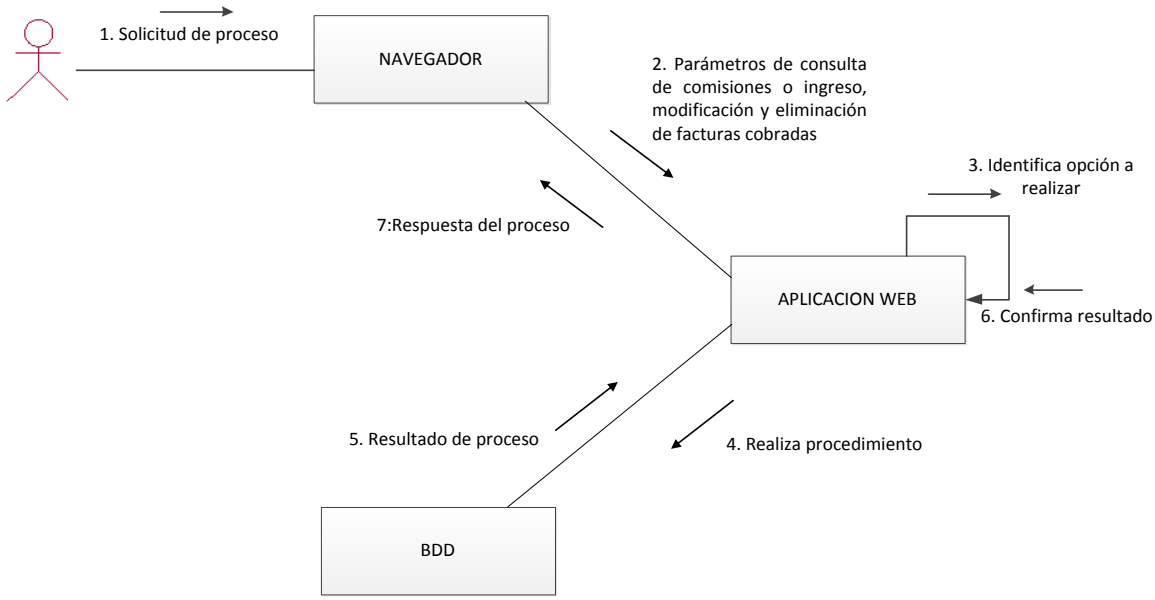


Figura 45 Diagrama de colaboración Administrar Comisiones

Fuente: Propia

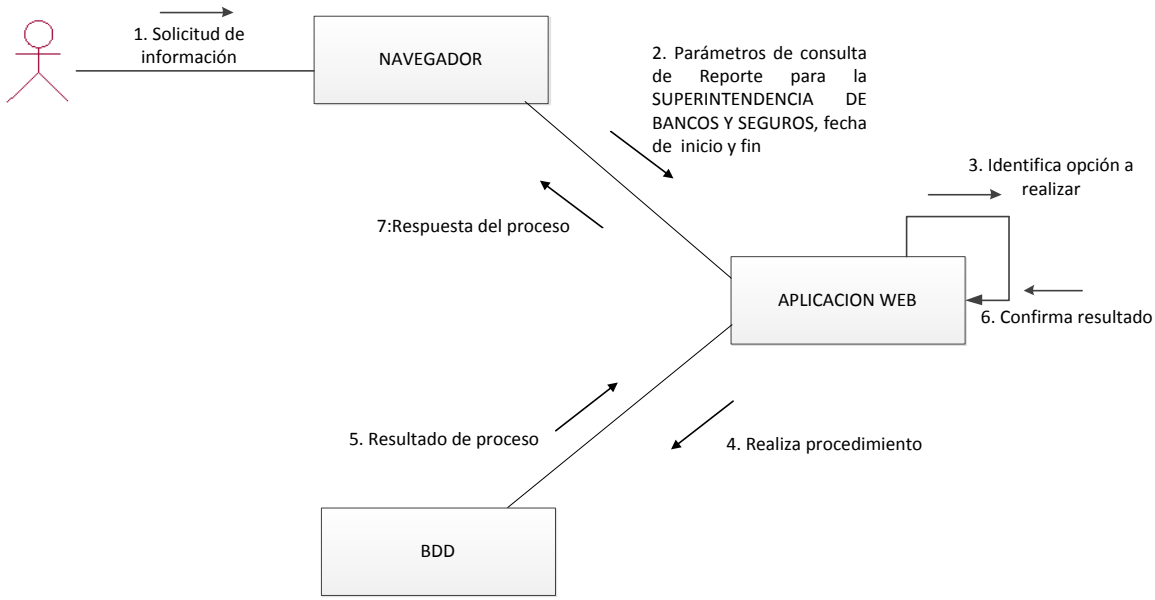


Figura 46 Diagrama de colaboración Reporte Superintendencia de Bancos y seguros

Fuente: Propia

6.2 Modelo Estático del Sistema

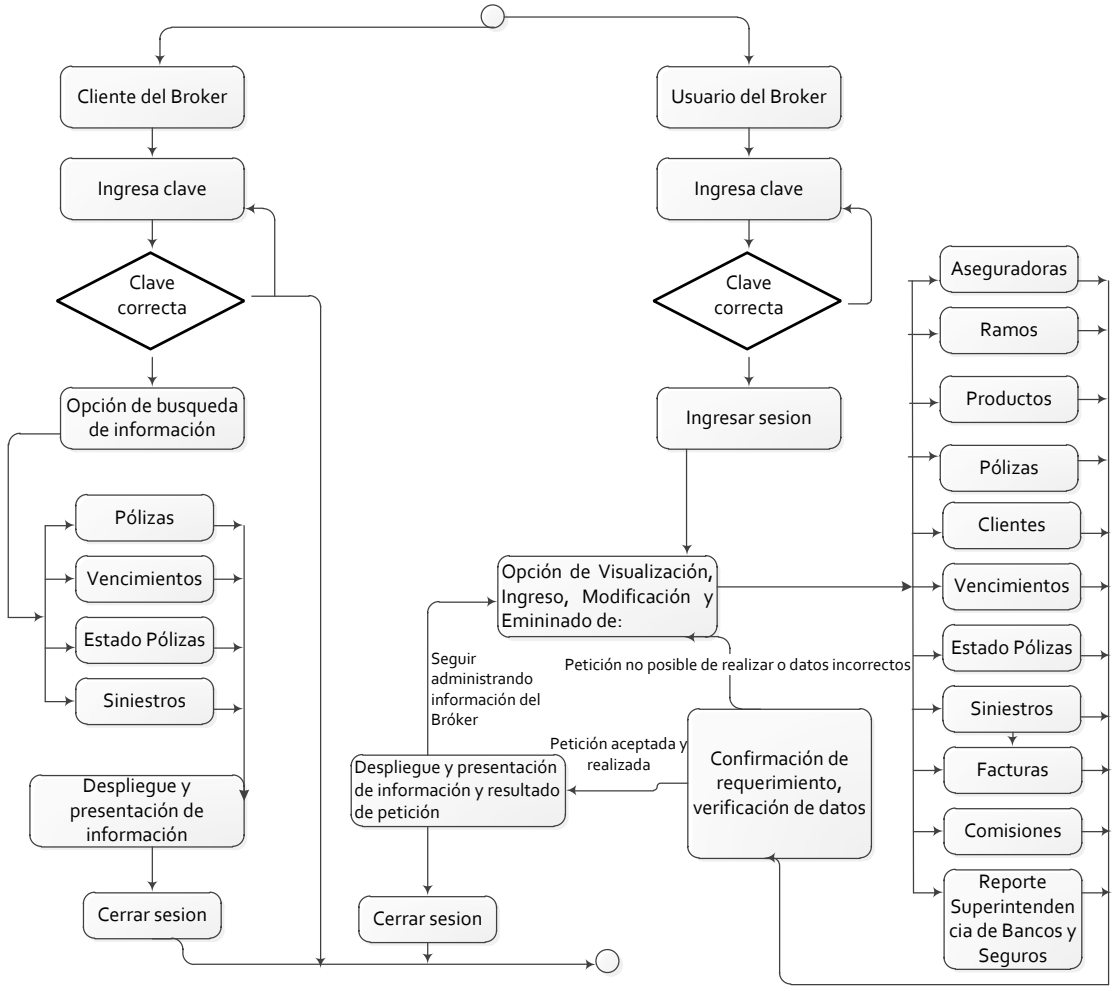


Figura 47 Diagrama de actividad del sistema

Fuente: Propia

CONCLUSIONES Y RECOMENDACIONES

7. CONCLUSIONES Y RECOMENDACIONES

7.1 Conclusiones

- La implementación del Sistema de Administración del Bróker de Seguros utilizando Software Libre para la institución “Asesores de seguros Álvaro Cazar” permitió un cambio organizacional previo y durante la implementación, atendiendo los requerimientos en los procesos de los servicios y administración del Bróker, tales como el control de la información de los clientes, pólizas, aseguradoras, productos, siniestros, vencimientos, comisiones y reporte de la Superintendencia de bancos y seguros.

- Al implementar el Sistema, permitió contar con un sistema de calidad para el desempeño de sus empleados, facilidad de acceso a información por parte de los clientes, y se simplificó el control de acceso a la información de los empleados y administrativos del Bróker, contando con información organizada, centralizada y de fácil acceso y poder explotar los conocimientos de los empleados de manera adecuada. Se eliminó la duplicidad de información, teniendo la información del Bróker en un solo repositorio.

- Al utilizar PostgreSQL para la elaboración de mi proyecto de tesis, confirmé que es una herramienta excelente para el almacenamiento masivo de información, poseen una serie de características que la hacen sólidas frente al manejo de gran volumen de datos, lo que a mi criterio hace que la elección realizada sea acertada por el tipo de información que se maneja en el Bróker.

- El uso de herramientas Open Source, en conjunto constituyen una completa plataforma de desarrollo de Sistemas de Administración de Brókers de Seguros, debido a los diferentes componentes que permiten manipular y construir aplicaciones a la medida, ya que son herramientas seguras, gratuitas y compatibles con los distintos sistemas operativos, ya sea que tengan licencias o no.

- El manejo de la metodología RUP basada en UML proporciona guías para conocer el camino a recorrer antes de empezar la implementación con lo cual asegura la calidad del producto final.

7.2 Recomendaciones

Al realizar la implementación del Sistema de Administración del Bróker de Seguros utilizando Software Libre para la institución “Asesores de seguros Álvaro Cazar”, se debe tener en consideración las siguientes recomendaciones que ayudará al buen funcionamiento y desempeño del Sistema:

- Antes de desarrollar cualquier proyecto informático es necesario empaparse bien con toda la información posible del tema al cual se va a hacer referencia en el desarrollo de la aplicación del software.

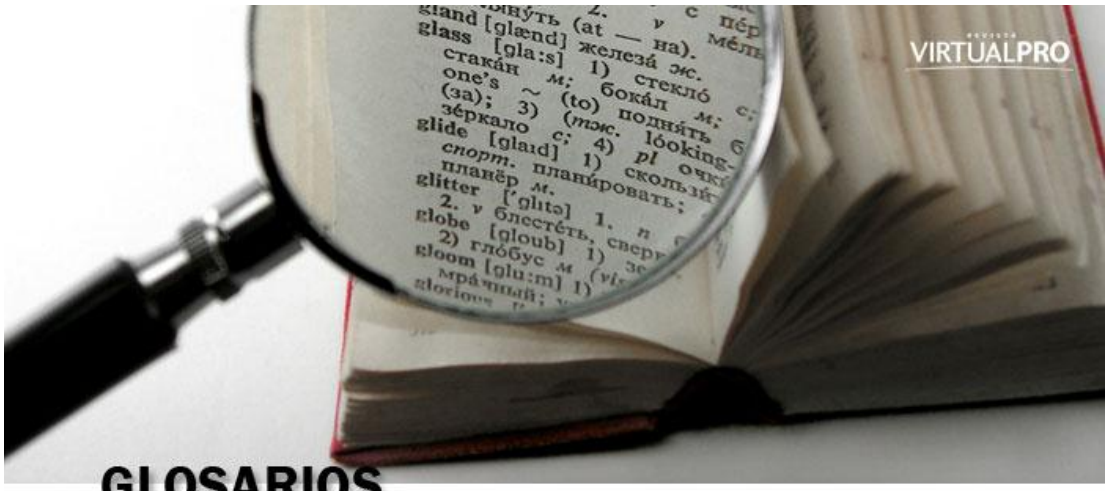
- Para cumplir los objetivos planteados, es fundamental la participación activa de los usuarios.

- Se requiere la coordinación entre los departamentos del Bróker, para organizar e ingresar de manera adecuada la información que se maneja, de esta manera se tendrá la información de la institución de una manera completa y actual.

- El sistema deberá funcionar permanentemente y de una manera adecuada en la Institución para que no existan falencias en los datos que se manejan y presentar a los usuarios, y así contar con información real para el Bróker.

- Se recomienda realizar en lo posible respaldos diarios de la base de datos para no perder la información del bróker, y si existe riesgo de pérdida de información se puede subir el último respaldo.

GLOSARIO DE TÉRMINOS

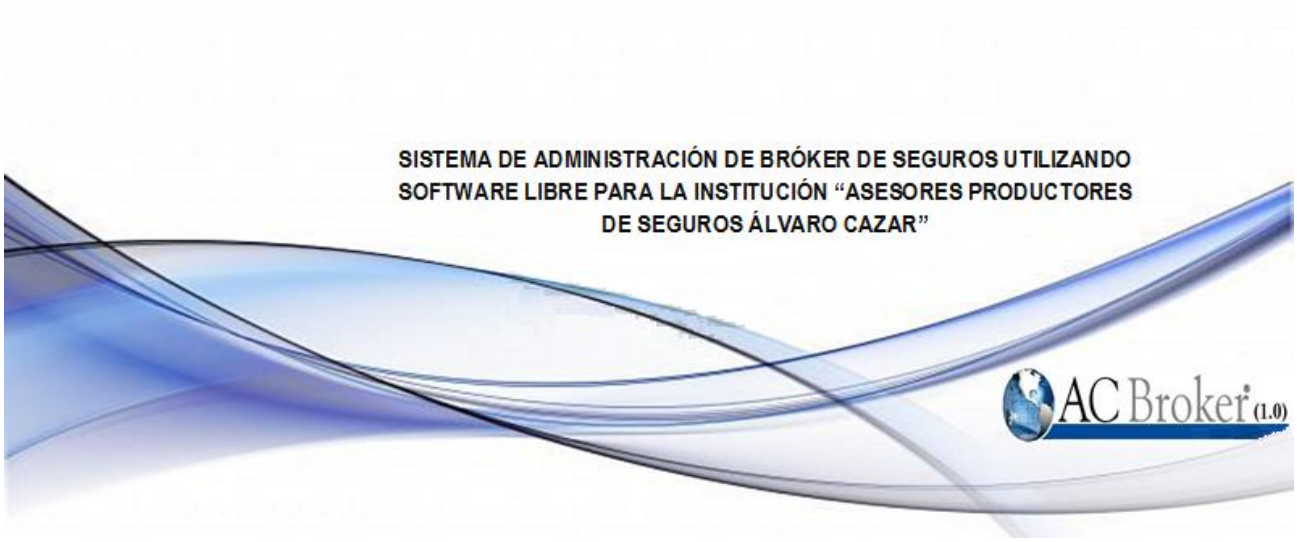


8. GLOSARIO DE TÉRMINOS

- **API** Aplication Programming Interface
- **HTML** Hyper Text Markup Language
- **CSS** Cascading Style Sheets, Hojas de Estilo en Cascada
- **RUP:** Rational Unified Process.
- **UP:** Unificación de Procesos
- **UML** Lenguaje Unificado de Modelado (UML, por sus siglas en inglés, Unified Modeling Language) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad.
- **Framework** Es una estructura conceptual y tecnológica de soporte definida, normalmente con módulos de software concretos.
- **Vulnerable:** Debilidad en la seguridad de la información de una organización
- **Redundancia:** Repetición de una información ya dada en el mensaje
- **Inconsistencia:** Falta de consistencia en la estructura de un lenguaje documental
- **Automatización:** Son acuerdos documentados que contienen especificaciones técnicas u otros criterios específicos para ser usados como referentes, guías o definiciones.
- **PostgreSQL** Repositorio de Información.

- **Stakeholder:** Cualquier persona interesada en, afectada por y/o implicada con el funcionamiento del sistema o software.
- **Project Manager:** Jefe de Proyecto
- **Release:** Nueva versión de una aplicación informática.
- **Workflow:** Flujo de Trabajo
- **Artefactos:** Elementos materiales que los humanos han construido o modificado.
- **Bróker de Seguros** Un corredor de seguros es una persona que actúa como intermediario de varias compañías aseguradoras, sin estar vinculado en exclusiva a ninguna de ellas.
- **Suma Asegurada** Es el valor que fija el asegurado sobre su persona o sus bienes, y que es determinante para que la aseguradora cobre la prima o haga una indemnización en caso de siniestro.
- **Vigencia** Es el período durante el cual la aseguradora se compromete, a proteger mediante el pago de una prima, a cubrir un bien o una persona.
- **Siniestro** Es el acontecimiento o hecho previsto en el contrato, cuyo acaecimiento genera la obligación de indemnizar al asegurado
- **Comisiones** Cantidad de dinero que percibe el Bróker de seguros por ingresar producción a las Compañías Aseguradoras

REFERENCIAS Y BIBLIOGRAFÍA



9. BIBLIOGRAFÍA Y REFERENCIAS

9.1 Libros

[LIB01]. Rich Bowen, Ken Coar (2007) Apache Administrators. Apache Cookbook: Solutions and Examples, 2da Edición

[LIB02]. Andrea Steelman and Joel Murach (2010). Murach's Java Servlets and JSP, 2da Edición

[LIB03]. Nicklas Persson and Christopher Murphy(2008), HTML and CSS Web Standards Solutions: A Web Standardistas' Approach

[LIB04]. Sergio Guardiola Herrador (2010) HTML & CSS Fácil y sencillo

[LIB05]. Per Kroll Philippe Kruchten, Grady Booch(2003). The Rational Unified Process Made Easy: A Practitioner's Guide to the RUP

[LIB06]. Ahmad K. Shuja, Jochen Krebs(2008), IBM Rational Unified Process Reference and Certification Guide: Solution Designer (RUP)

[LIB07]. “Manual del usuario de PostgreSQL” , El Equipo de Desarrollo de PostgreSQL – Editado por Thomas Lockhart. 1996, 497 pp.

[LIB08]. Philippe Kruchten, The Rational Unified Process: An Introduction (3rd Edition)

[LIB09]. Martin Aguero , Introduccion a Spring Framework, version 1.0

[LIB10]. FITSE, Curso Broker de Seguros – Certificacion SUPERINTENDENCIA DE BANCOS Y SEGUROS, version 1.0

[LIB11]. Jochen Kerbs (2009), Rational Unified Process (RUP), version 2.0

9.2 Páginas Web

[WEB01]. ”Risk Seguros”. http://riskseguros.wordpress.com/2010/02/02/diferencia-entre-corredores-de-seguros-y-agentes-de-seguros

[WEB02]. ” Grupo mayo”.

http://www.grupomayo.com/mapaweb.php

[WEB03]. ” Que es un Agente de Seguros ”.
http://www.mediaseguros.es

[WEB04]. Wikepedia, de
http://es.wikipedia.org

[WEB05]. Apache Software Foundation, de
http://es.wikipedia.org/wiki/Apache_Software_Foundation

[WEB06]. JavaServer Pages, de
http://es.wikipedia.org/wiki/JSP

[WEB07]. Hojas de estilo en cascada, de
http://es.wikipedia.org/wiki/CSS

[WEB08]. Proceso Unificado de Rational, de
http://es.wikipedia.org/wiki/RUP

[WEB09]. FRAMEWORK SPRING, de
http://sentidoweb.com/2006/12/26/spring-framework-de-java.php

[WEB10]. Actualización Spring 2.0
http://www.infoq.com/articles/spring-2-intro

[WEB11]. Simplifying Enterprise Applications with Spring 2.0 and AspectJ
http://www.infoq.com/articles/Simplifying-Enterprise-Apps

[WEB12]. Manual Practico de HTML
http://www-app.etsit.upm.es/~alvaro/manual/manual.html

[WEB13]. Pagina Oficial de PostgreSQL
http://www.postgresql.org

[WEB14]. Framework Spring, principales características
http://www.sicuma.uma.es/sicuma/independientes/argentina08/Badaracco/spprincar.ht

[WEB15]. Documentacion de PostgreSQL
http://en.wikipedia.org/wiki/PostgreSQL

ANEXOS

SISTEMA DE ADMINISTRACIÓN DE BRÓKER DE SEGUROS UTILIZANDO SOFTWARE LIBRE PARA LA INSTITUCIÓN "ASESORES PRODUCTORES DE SEGUROS ÁLVARO CAZAR"

AC Broker (1.0)

10. ANEXOS

10.1 Anexo A Manual de Instalación

10.2 Anexo B Manual de Usuario

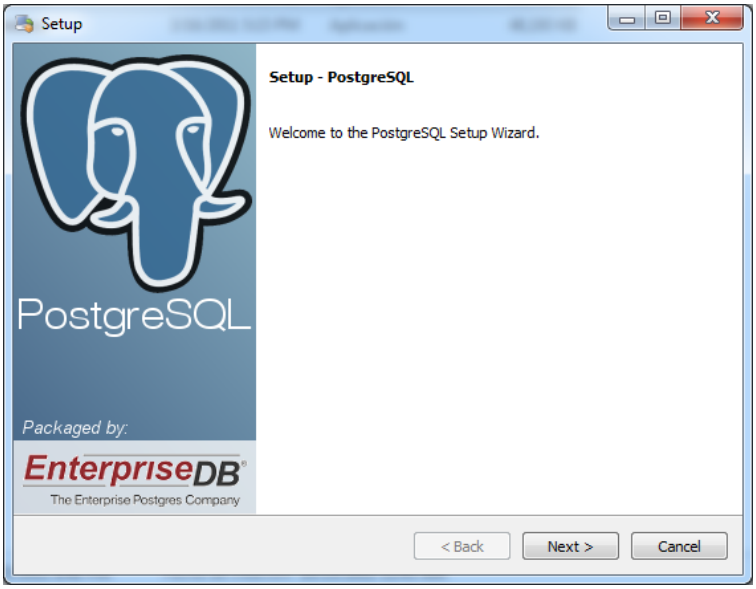
10.3 Requerimientos del Spring Roo

10.4 Anexo C Anteproyecto

ANEXO A: MANUAL DE INSTALACIÓN
Adjuntando la Base de Datos al PostgreSQL

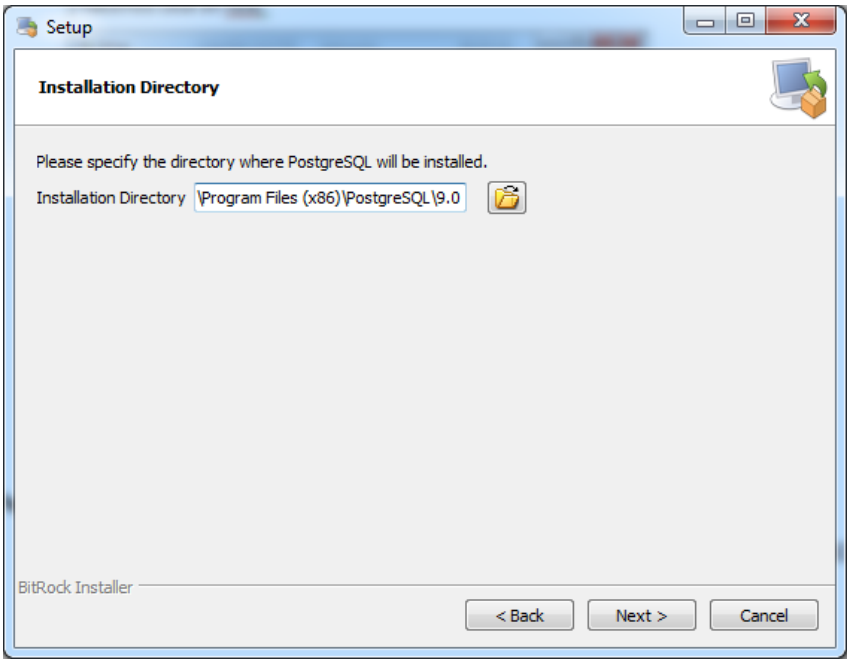
Pasó uno:

El cliente hace click en el archivo de instalación de PostgreSQL, y le aparecerá la siguiente pantalla y hacemos click en next.:



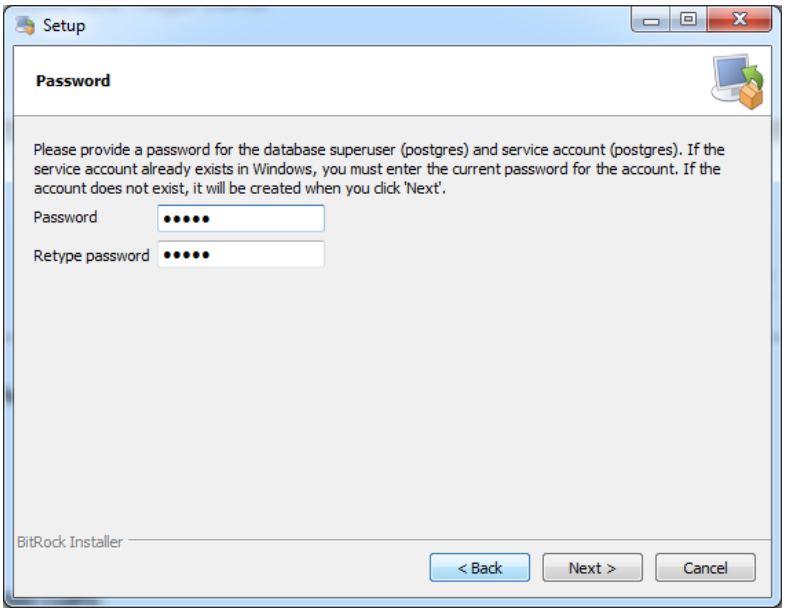
Paso dos:

Luego nos aparece una ventana donde nos indica el directorio donde se va a instalar la BDD, dejamos el mismo y presionamos en siguiente.



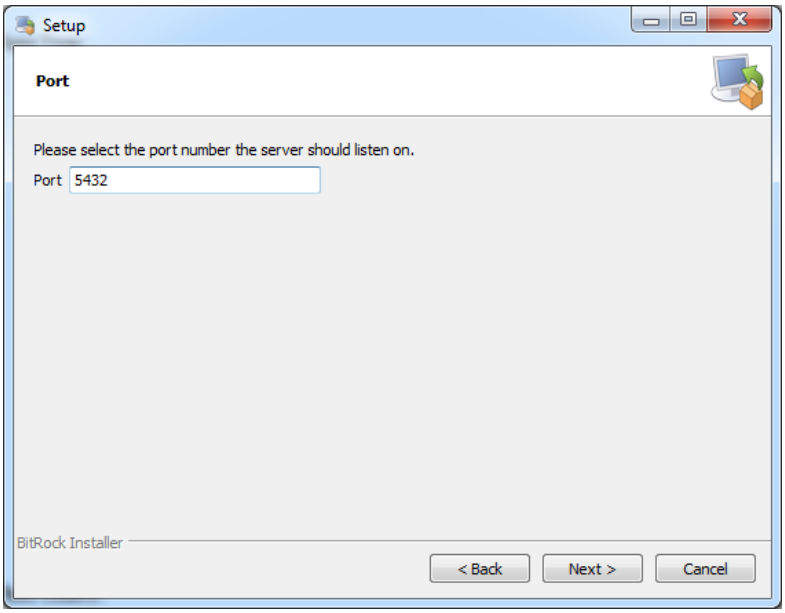
Paso Tres:

En la siguiente pantalla el cliente debe ingresar el password y presionar siguiente.



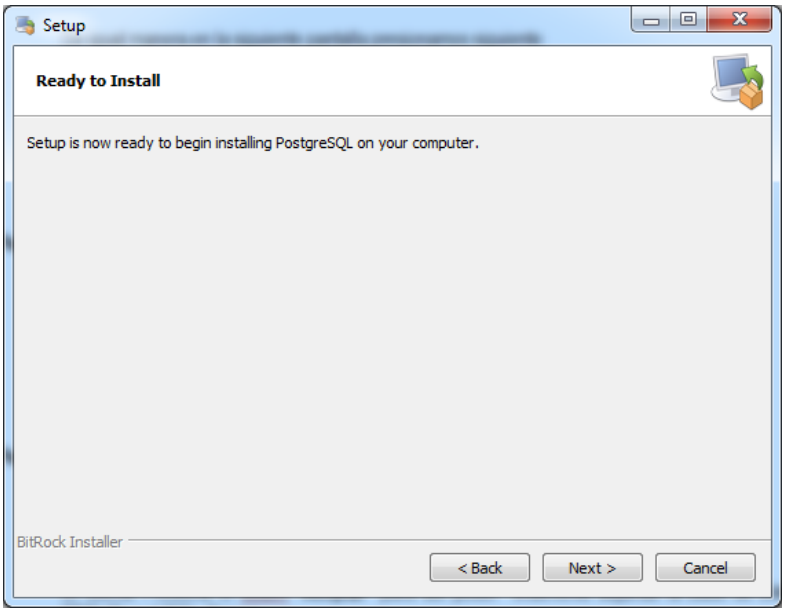
Paso Cuatro:

En la siguiente pantalla se muestra puerto donde funcionara la BDD, no cambiamos nada y hacemos click en siguiente.



Paso Cinco:

De igual manera en la siguiente pantalla presionamos siguiente y se procede con la instalación.



Paso Seis:

Luego que ya se competa la instalación damos click en Finalizar.

ANEXO B: Manual de Usuario del Sistema

Objetivo

El Sistema de Administración de Bróker de Seguros utilizando Software Libre para la Institución “Asesores Productores De Seguros Álvaro Cazar”, tiene como objetivo satisfacer las necesidades del Corredor de seguros mediante una Aplicación Web.

Requerimientos del sistema

- Requerimentos de Hardware
  - Processador: Pentium IV 2.6 GHz
  - Memória RAM: 256 MB.
  - Espacio libre en disco de: 20MB
  - Tarjeta de Video: 8 MB

- Requerimientos de Software:
  - Sistema operativo Windows XP
  - Java SDK 1.5 o superior
  - PostgreSQL 2.8 o superior
  - Apache Tomcat

Características Técnicas.

- Ambiente Transaccional.
- Ambiente Cliente/Servidor.
- Diseñado para Plataformas Windows XP, 2008, Linux
- Versión del Sistema 1.0

Instalación del AC Broker 1.0

Antes de instalar el Sistema, debe asegurarse de Contar con el equipo hardware y software mínimo requerido.

Aplicación del Sistema AC Bróker 1.0.

Para comenzar con la explicación de las funciones con las que cuenta el sistema **h**aremos referencia al menú que lo contiene:

- Seguridad
  - Iniciar sesión

- Menú

Clientes

- Visualizar Pólizas
- Visualizar vencimientos
- Visualizar Estado Pólizas
- Visualizar Siniestros

Usuarios bróker

- Administrar Compañía
- Administrar Ramo de Seguro
- Administrar Producto

- Administrar Clientes
- Administrar Póliza
- Consultar Estado de Póliza

- Administrar Siniestros
- Consultar vencimientos
- Administrar Facturas

-Administrar Comisiones

-Consultar Reporte Superintendencia de Bancos y Seguros

Pantallas de Acceso al Sistema

Sirve como medida de seguridad y confidencialidad de su información.

Basta colocar el nombre de usuario (login) y la clave (password) que se le configuró y suministró de antemano.

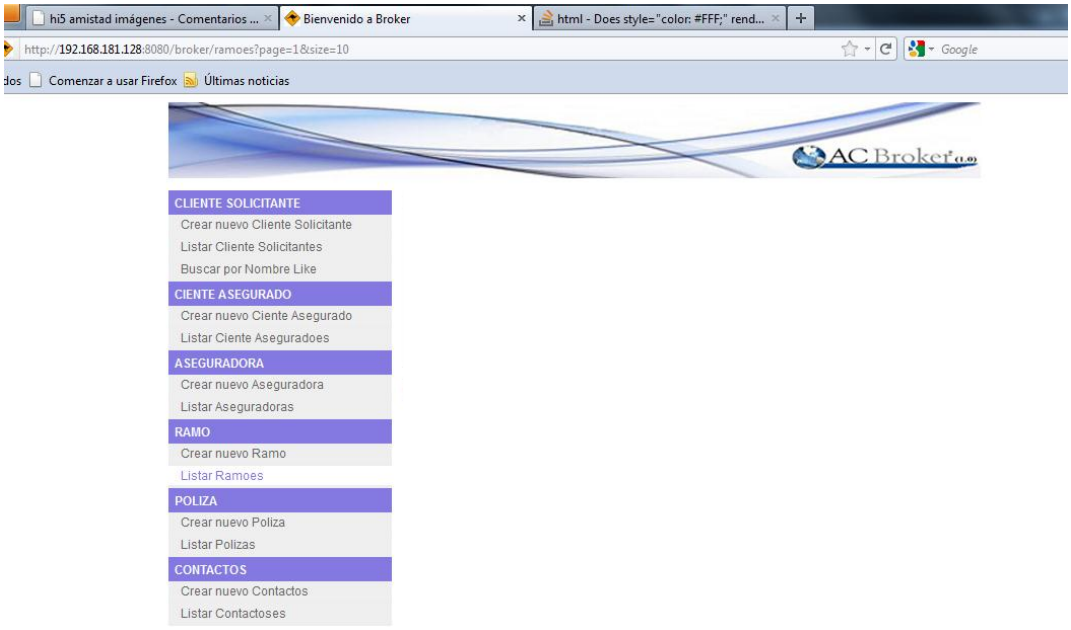
Usted puede ingresar como:

Cliente.

Usuario Bróker.

PANTALLA PRINCIPAL

A continuación se muestra la pantalla principal que contiene, los siguientes interfaces:



Clientes

En el menú de “Clientes” del paquete de Clientes, se puede acceder a información sobre:
Pólizas
Vencimientos
Estado de Pólizas
Siniestros

Pólizas

Se ingresa a la interface Cliente al dar click en la opción de Pólizas se muestra una pantalla, donde se escoge si se desea ver la información de todas las pólizas que mantiene el cliente ya sea por aseguradora, número de póliza o todas, y dar click en el botón buscar:

Vencimientos

Se ingresa a la interface Cliente, al dar click en la opción de Vencimientos se muestra una pantalla, donde se escoge si se desea ver la información de todas las pólizas que mantiene el cliente ya sea por aseguradora, número de póliza o todas, ingresando el rango de fechas que se desee y luego dar click en el botón buscar.

Estado de pólizas

En la interface Cliente al dar click en la opción de Estado de Pólizas se muestra una pantalla, donde se escoge si se desea ver la información de todas las pólizas que mantiene el cliente ya sea por aseguradora, número de póliza o todas, y dar click en el botón buscar.

Siniestros

En la interface Cliente al dar click en la opción de Siniestros se muestra una pantalla, donde se escoge si se desea ver la información de todos los siniestros de las pólizas que mantiene el cliente ya sea por aseguradora, número de póliza o todas, al dar click en el botón buscar.

Usuarios bróker

En la pantalla de "Usuarios Bróker", el usuario del bróker puede acceder a administrar información sobre:

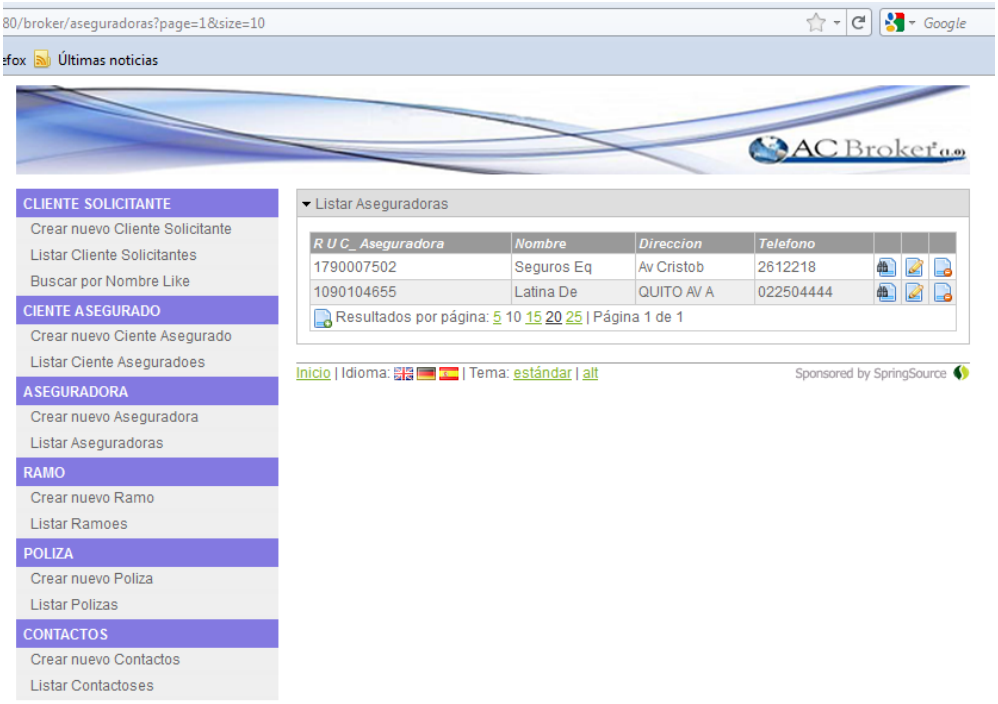
- Administrar Compañía
- Administrar Ramo de Seguro
- Administrar Producto
- Administrar Clientes
- Administrar Póliza
- Consultar Estado de Póliza
- Administrar Siniestros
- Consultar vencimientos
- Administrar Facturas
-Administrar Comisiones
-Consultar Reporte Superintendencia de Bancos y Seguros

Administrar Compañías

En la pantalla tenemos las opciones de modificar, eliminar, agregar compañía y buscar compañía por nombre o RUC.

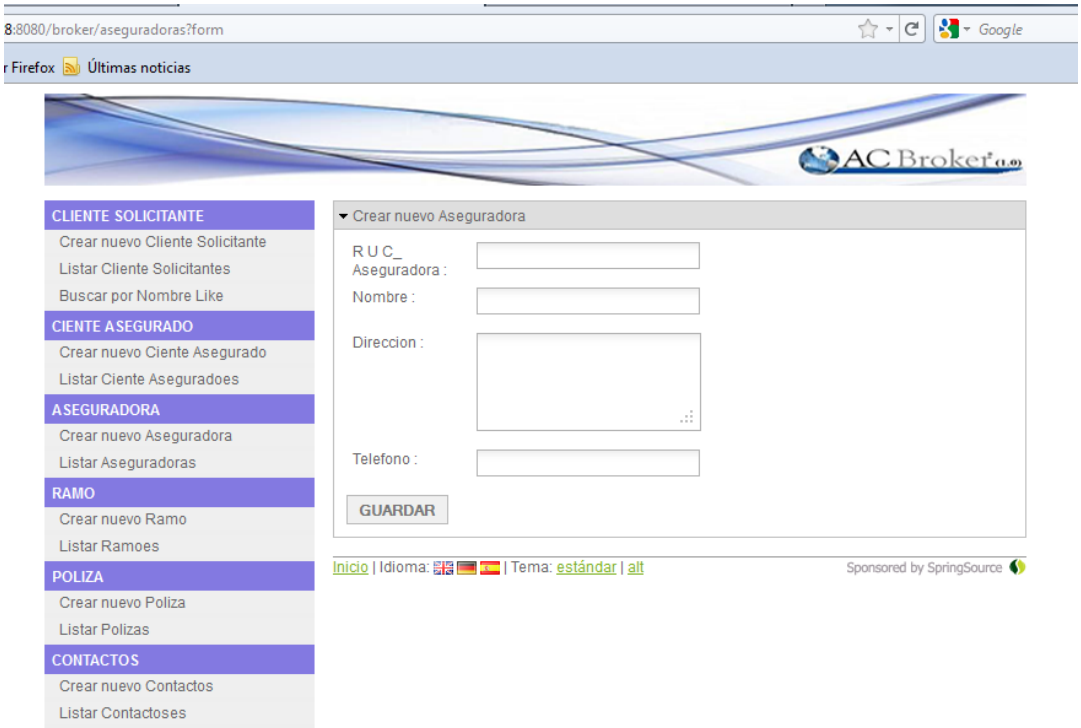
Buscar compañía

En la pantalla se puede obtener un listado de las compañías con las que se trabaja, ingresando nombre o ruc y realizando click en el botón listar aseguradoras y se muestra una ventana con la compañía buscada, en esta misma pantalla aparecen los íconos de modificar, ingresar o eliminar compañía.



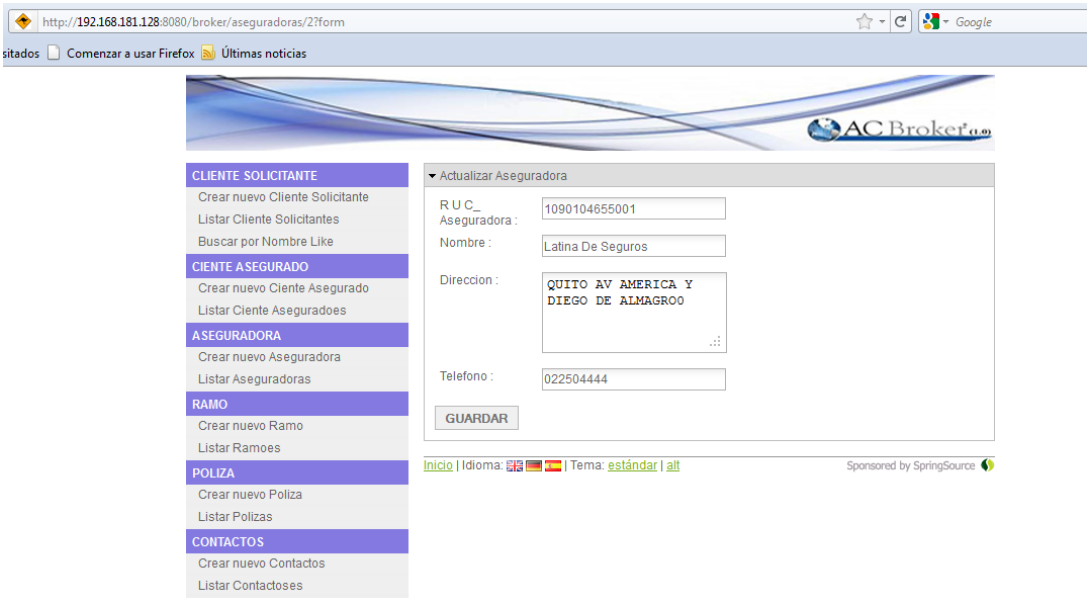
Registrar Compañía

En crear nueva aseguradora hacemos click y se llenan todos los campos en blancos de la compañía y se finaliza haciendo click en el botón “Guardar”.



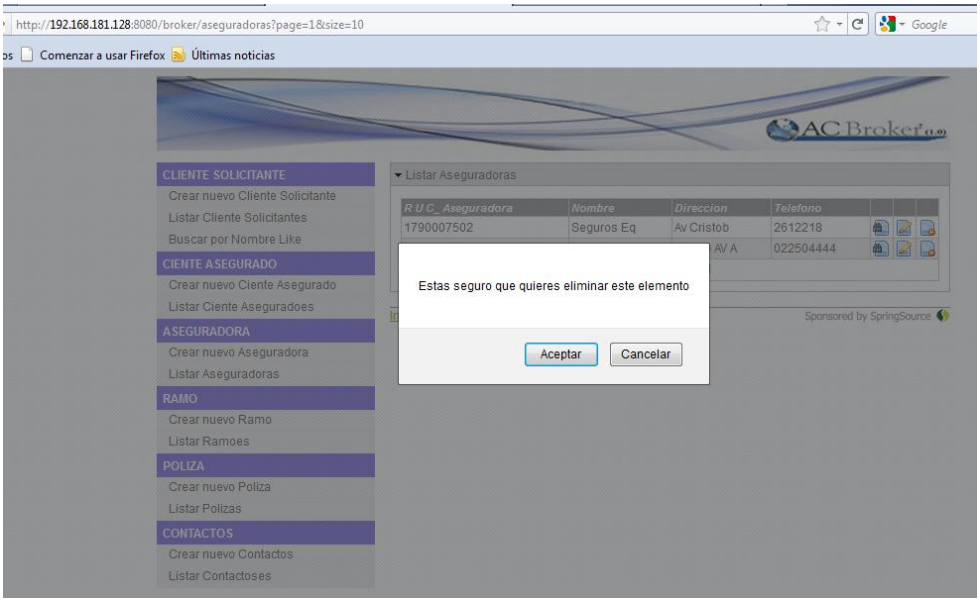
Modificar Compañía

Se busca la Compañía por nombre o ruc, damos click en modificar luego se verifican todos los campos de la compañía y se finaliza haciendo click en el botón “Guardar”.



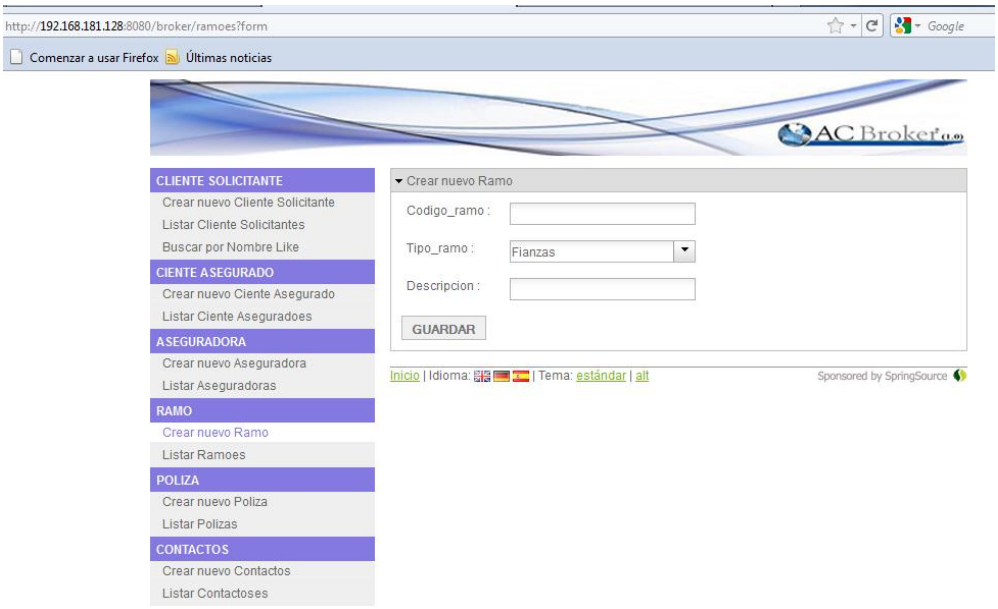
Eliminar Compañía

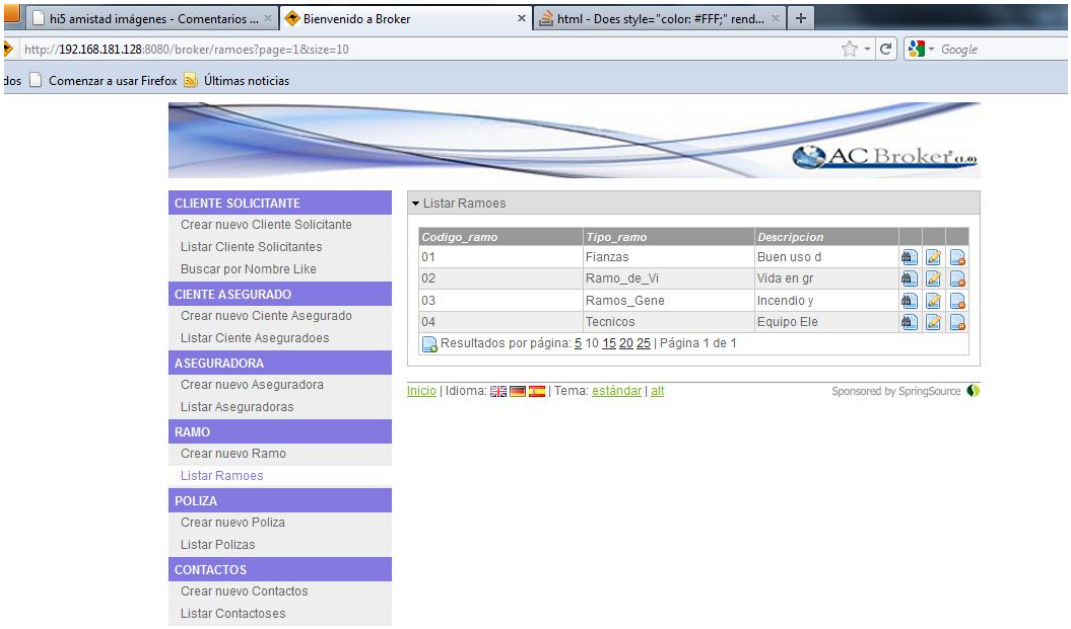
Se busca la Compañía por nombre o ruc, luego si se quiere eliminar hacer click en el botón “eliminar”, y proceder a aceptar o cancelar la operación.



Administrar Ramo de seguro

Esta es la pantalla de “Ramo de seguro”, el bróker puede listar agregar, modificar y eliminar un Ramo de la misma manera como se maneja el menú de aseguradoras.

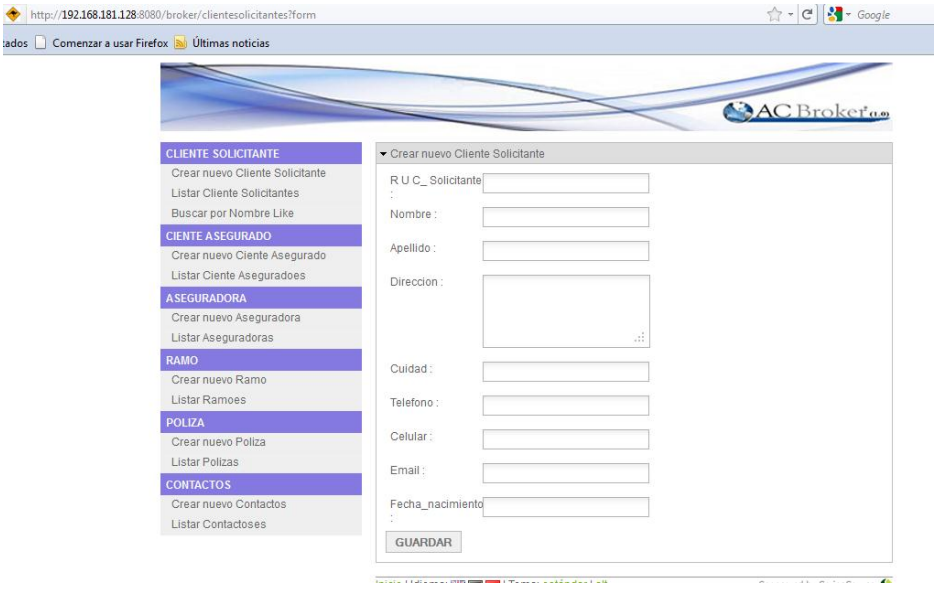


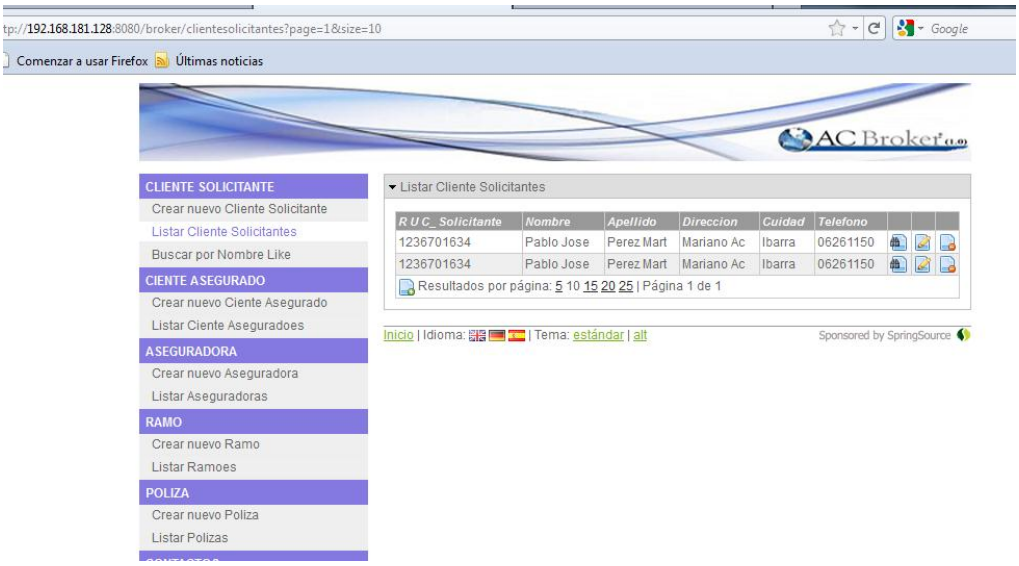


Esta es la pantalla podemos elegir la opción que necesitemos realizar ya sea ingreso, modificación o eliminación de ramo

Administrar Clientes

Esta es la pantalla de “clientes” del paquete Administración de Clientes y Pólizas de Seguro; el usuario del bróker, puede agregar, modificar y eliminar un cliente también buscar por contratante.
En ingresar cliente, se llenan los campos respectivos, luego hacer click en el botón “Guardar”.

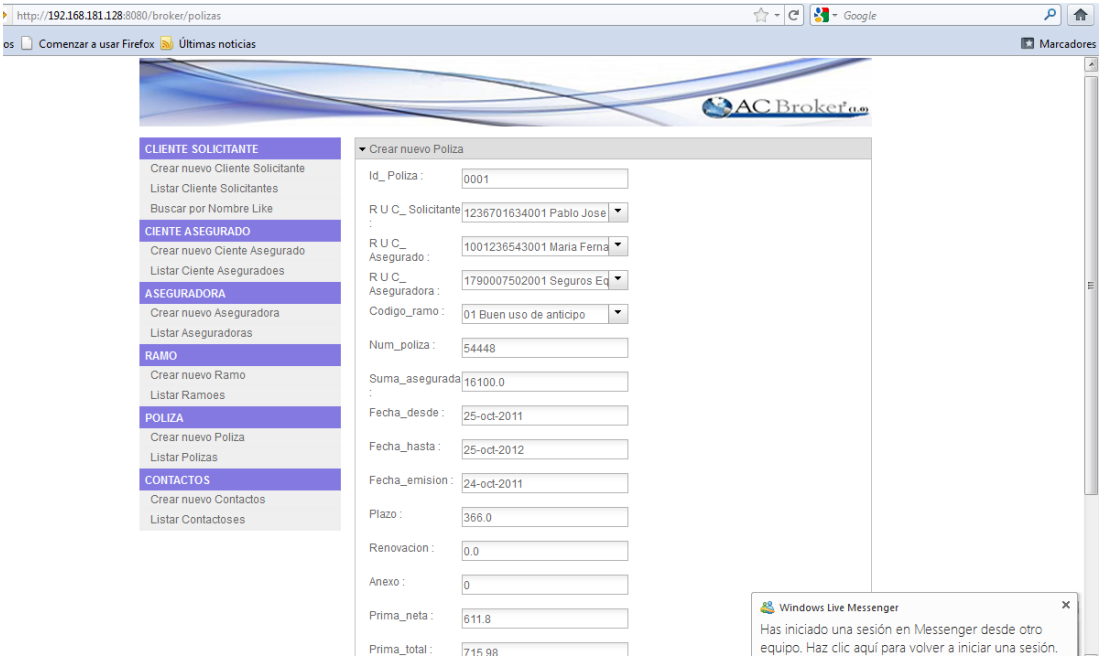




De igual manera después de listar se puede modificar, eliminar o ingresar nuevos cliente.

Administrar Pólizas

Se elige la opción de ADMINISTRAR PÓLIZAS; esta pantalla se muestra una opción donde se puede ingresar nuevas pólizas, luego a las cuales en el listado de pólizas se las puede modificar, o eliminar según sea la necesidad.



Se pone el código o nombre que identifiquen a los clientes que tengan una o más de una póliza registrada

En la pantalla principal de administrar póliza podemos realizar una búsqueda por Aseguradora y contratante, también se puede modificar, eliminar y registrar póliza.

Estado de pólizas

Se ingresa a la interface de Administrar Clientes y Pólizas, al dar click en la opción de Estado de Póliza se muestra una pantalla, donde se escoge si se desea ver la información de pólizas que mantienen con las aseguradoras ya sea por cliente o por aseguradora, número de póliza o todas, y dar click en el botón buscar.

Siniestros

Se ingresa a la interface de Administrar Clientes y Pólizas al dar click en la opción de Siniestros nos muestra una pantalla, donde se escoge si se desea ver la información de siniestros que se han suscitado con las aseguradoras ya sea por cliente o por aseguradora, número de póliza o todas, y dar click en el botón buscar.

Vencimientos

Entramos a la interface de Administrar Clientes y Pólizas al dar click en la opción de Vencimientos nos muestra una pantalla, donde debemos escoger si se desea ver la

información de vencimientos de las pólizas de las aseguradoras ya sea por cliente o por aseguradora, número de póliza o todas, e ingresar el rango de fechas de inicio y fin luego dar click en el botón buscar.

Administrar Facturas

En la ventana se observa una lista de facturas en la cual podemos eliminar, modificar y agregar comprobante.

Modificar Factura

Se puede modificar los campos necesarios y luego presionar el botón “Modificar” sino “cancelar”.

Eliminar Factura

Se muestra en la ventana la factura a la cual se desea eliminar, y luego se da click en el botón “eliminar” sino “cancelar”**.**

Agregar nueva Factura

En la ventana se muestra todos los campos vacíos en la cual rellenaremos y finalizaremos al hacer click en el botón “registrar” sino en “cancelar”

Controlar Comisiones

Entramos a la interface de Control comisiones y al dar click en la opción de Controlar Comisiones nos muestra una pantalla, donde debemos escoger si se desea ver la información de comisiones que se han cobrado a las aseguradoras por número de factura o por aseguradora, y dar click en el botón buscar.

Control Reporte Superintendencia de bancos y Seguros

Entramos a la interface de Reporte Superintendencia de bancos y seguros, donde se desea ver la información de comisiones que se han cobrado a las aseguradoras según el formato de la Superintendencia de bancos y Seguros, se debe ingresar el rango de fecha inicio y fin, luego dar click en el botón buscar.

Requerimientos Spring ROO

Spring Roo es una herramienta RAD extensible para Java basada en Spring Framework.

Básicamente es un generador de código avanzado, que se utiliza desde la línea de comandos invocando sentencias.

La idea detrás de **Spring Roo** es incrementar la productividad del desarrollador Java sin comprometer la integridad estructural o la flexibilidad de la solución. No contiene un componente Runtime. Esto es muy importante, no solo porque no ata la solución al framework, sino porque no genera overhead. Se puede eliminar fácilmente.

Está construido con una serie de plugins (al estilo de Maven). Fue diseñado pensando en la usabilidad. Los comandos son contextuales; tiene los comandos "hint" y "help" para consultar el uso rápidamente y la tecla TAB para autocompletar.

Está basado en scripts, por lo que en caso que se cometer un error, se puede hacer rollback. Todos los comandos escritos se van guardando automáticamente (log.roo).

Características

- Permite desarrollar una aplicación Web en minutos. Genera un war. Construye dos capas: la de persistencia y la de presentación. Para agregar la capa de negocio, se pueden agregar las clases manualmente a los controladores generados con Roo.
- Permite generar un modelo de datos complejo, incluyendo validaciones.
- La aplicación Web es RESTful y tiene soporte para internacionalización en varios idiomas.
- Tiene soporte de seguridad out-of-the-box.
- La presentación usa Dojo (incorporar otro, por ejemplo JQuery, se hace a mano y es costoso).

Generación de código

Es un generador de código *hibrido*, utilizando generación activa y pasiva.

Utiliza generación pasiva para generación de archivos java y xml. Utiliza generación activa para la metadata con la ayuda de los tags @Roo* y modifica gradualmente .aj y los jsp.

Sincroniza los cambios entre Roo y las modificaciones realizadas al código (roundtrip).

Spring Roo no interviene en runtime, solo lo hace en tiempo de desarrollo. Una vez creado el proyecto con Spring Roo, podría eliminarse y seguir trabajando con un IDE.

Librerías/Frameworks que usa

- Java
- AspectJ
- Eclipse
- Java Persistence API: Hibernate, OpenJPA, EclipseLink
- Java Server Pages
- JUnit
- Log4J
- Maven
- Selenium
- Spring Framework
- Spring MVC
- Spring Security
- Spring Web Flow

Requisitos

Tener instalado Java y Maven, las últimas versiones estables.

Configuración

Descargar la última versión de Roo del sitio oficial. Descomprimir en una carpeta en el disco local. Ej: c:\roo-1.2 Incluir la carpeta con los ejecutables dentro de la variable de entorno PATH. Ej.: c:\roo-1.2\bin

Uso

Para la ejecución de comandos podemos utilizar:

- el shell de roo mediante línea de comando
- el shell que viene embebido en el SpringSource Tool Suite
- o el IDE que utilizas habitualmente, por ejemplo Eclipse. En este caso deberías instalar el plugin para trabajar con aspectos y maven.

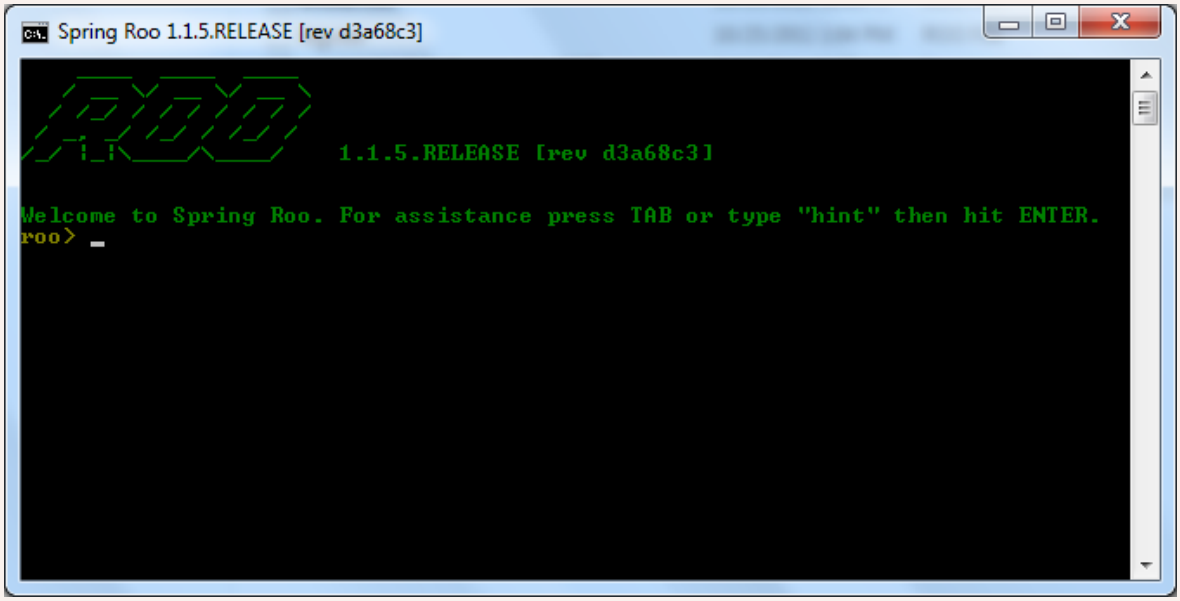
Para empezar a usar los comandos Roo mediante el shell, crear una carpeta e invocar a **Spring Roo** Ej.:

```
mkdir helloworld

cd helloworld

roo
```

Al hacer eso se iniciará la interfaz y desplegará el siguiente mensaje:

```
    / __ V __ V __ \
    / / _ / / / / / / /
    / _ , _ / / _ / / / _ / /
    / _ / |_|\ ____ /\ ____ /    1.0.2.RELEASE [rev 638]
    Welcome to Spring Roo. For assistance press TAB or type "hint" then hit ENTER.
roo>
```

La tecla TAB despliega las opciones:



La sentencia help lista todos los comandos con descripciones breves.

Pasos para crear una aplicación mínima

- Crear el proyecto
- Elegir el framework de persistencia
- Crear el modelo de dominio
- Generar de la capa de presentación

Manejo de propiedades

Roo genera archivos de propiedades dentro del proyecto, por ejemplo: database.properties. Dichos archivos pueden ser modificados automáticamente por los comandos de roo. Se debe tener en cuenta entonces que al ejecutar un comando que impacte valores configurados manualmente, será necesario volver a configurarlos. Las claves definidas manualmente por el usuario no son alteradas por ningún comando de roo.

Maven

Desde Roo se pueden ejecutar varias tareas de Maven sin tener que salir del entorno. Por ejemplo:

- 'perform tests' equivale a 'mvn test'
- 'perform eclipse' equivale a 'mvn eclipse:eclipse'

Seguridad

El módulo de seguridad que viene predeterminado necesita unos pequeños cambios de configuración. De forma predeterminada viene configurado un path segurizado que no corresponde con la aplicación en desarrollo. Se invoca mediante la sentencia:

```
security setup
```

Ejecutar un script

Todo lo que se va ejecutando por consola, se va grabando en un archivo de log (log.roo). Es posible re-ejecutar las acciones, copiando el archivo con otro nombre y utilizando la sentencia 'script <nombre de archivo>' Ej.:

```
roo>script helloworld.roo
```

EJEMPLO PARA CREAR UNA APLICACION EN SPRING ROO

BASE DE DATOS FACTURAS

El presente artículo muestra una previa introducción a Spring Roo y a Jasper Roo, muestra un nuevo ejemplo de aplicación desarrollada con estas dos herramientas. En este caso una Aplicación de facturación.

Spring Roo es una herramienta de desarrollo rápido de aplicaciones o RAD, que permite el desarrollo de aplicaciones Java EE de forma muy productiva y cómoda para el desarrollador.

Las aplicaciones resultantes utilizan tecnologías Java conocidas como Spring Framework, Java Persistence API, Java Server Pages, Apache Maven, AspectJ.

Jasper Roo generara informes en formato de documento portátil (PDF), MS Excel (xls), Valores separados por comas (CSV), Hypertext Markup Language (HTML), Texto de Documento Abierto (odt), Extensible Markup Language (XML), y formato de texto enriquecido (rtf) formatos. Para cada entidad especificada, jasperoo creará un informe de lista y un informe detallado.

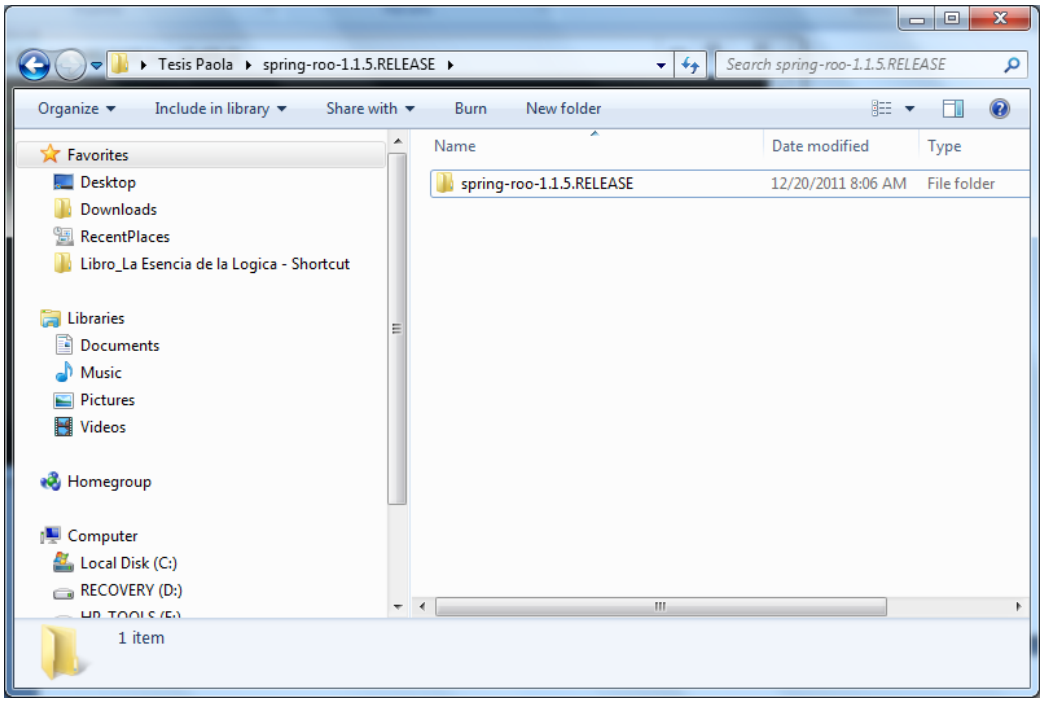
Jasperoo configura el proyecto para apoyar a Jasper Reports en la creación de informes genéricos de las entidades especificadas. En la actualidad, la versión 3.5.3 de Jasper Reports es compatible.

1. Requerimientos

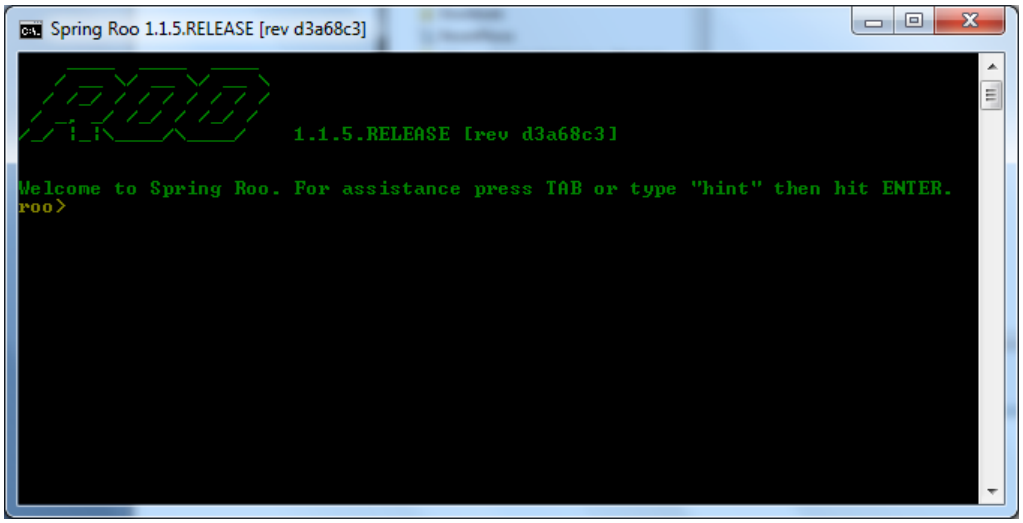
a) Java JDK
b) Maven
c) Base de datos Postgres 9.x
d) Spring Roo

Configurando Spring Roo 1.1.x

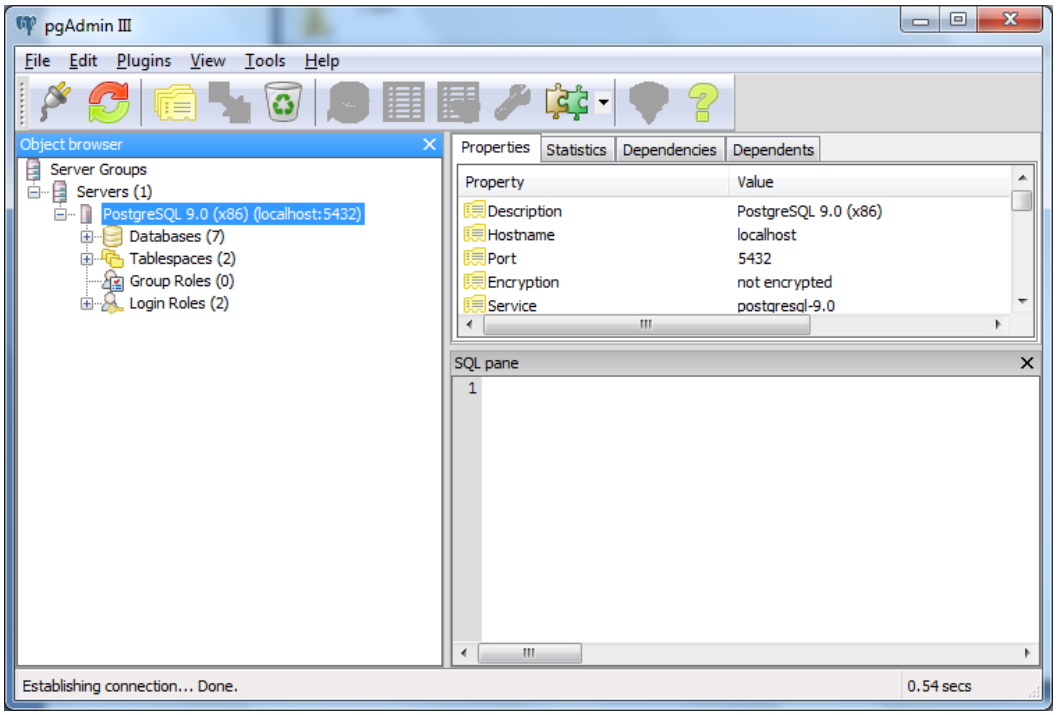
El proceso es fácil. Para utilizar Spring Roo primero debemos de descargarlo, por defecto al instalar STS (Springsource Tool Suite) llega la versión del Spring Roo 1.1.x RELEASE, pero nosotros utilizaremos la 1.1.5, la cual debemos de descargarla desde el proveedor de Spring roo. (Figura 1)



La interfaz de Spring Roo trabaja en modo de comandos:

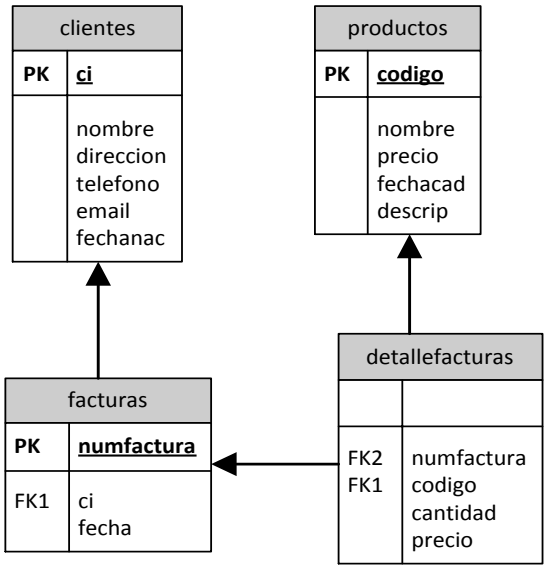


La utilización de la Base de datos será con **Postgres 9.x**, para el manejo con otra base de datos debemos especificar la cadena de conexión.



2. Creación de proyecto en Spring Roo

a) Establecer relaciones



b) Creación del proyecto

project --topLevelPackage com.app.facturas

c) Creación de la persistencia a la base de datos

persistence setup --provider HIBERNATE --database HYPERSONIC_IN_MEMORY

Para la configuración para una base de datos postgres

persistence setup --provider HIBERNATE --database POSTGRES --databaseName <nombrebdd> --userName <user> --password <pas>

Para la configuración para una base de datos mysql

persistence setup --provider HIBERNATE --database MYSQL databaseName <nombrebdd> --userName <user> --password <pas>

d) Creación de tablas

1. Crea una nueva tabla que se llame **clientes** con los siguientes campos:

Nombre	tipo de datos	tamaño /formato	Otras
Cedula	Numérico	entero largo	Clave principal
Nombre	Texto	50	
dirección	Texto	200	
Teléfono	Texto	9	
Fecha Nac.	Date	fecha corta	máscara de entrada fecha corta
e-mail	Texto	25	Validación

Spring Roo – Creación de la Entidad Cliente:

entity --class ~.domain.cliente --testAutomatically

field string --fieldName cedula --notNull --sizeMin 9 --sizeMax 10

field string --fieldName nombre –notNull --min 1 --max 50

field string --fieldName direccion –notNull --min 1 --max 200

field string --fieldName telefono –notNull --min 1 --max 9

field date --fieldName fechanac --type java.util.Date --notNull

field string --fieldName email --notNull --regexp ^([0-9a-zA-Z]([-.\w]*[0-9a-zA-Z])*@([0-9a-zA-Z][-\w]*[0-9a-zA-Z]\.)+[a-zA-Z]{2,9})$

2. Crea una nueva tabla que se llame "**Productos",** con los siguientes campos

Nombre	tipo de datos	tamaño /formato	otras
Código	Texto	6	
Nombre	Texto	50	
Precio	Numérico	simple /estándar	dos decimales
Fecha Caducidad	Date		
Descripción	Texto	100	

Spring Roo – Creación de la Entidad Productos:

entity --class ~.domain.producto --testAutomatically

field string --fieldName codigo --notNull --sizeMax 10

field string --fieldName nombre –-notNull –-max 100

field number --fieldName precio --type java.lang.Float --notNull --min 0

field date --fieldName fechacad --type java.util.Date –notNull

field string --fieldName descripcion --notNull

3. Crea una tabla que se llame factura con los siguientes datos

Nombre	tipo de datos	tamaño /formato	Otras
numfactura	Texto	6	
Cedula	Numérico	entero largo	Clave foránea
Fecha	fecha/hora	fecha corta	máscara de entrada fecha corta

Spring Roo – Creación de la Entidad Facturas:

entity --class ~.domain.factura --testAutomatically

field string --fieldName numfactura --notNull --sizeMax 10

field date --fieldName fechacad --type java.util.Date –notNull

Cardinalidad uno a muchos

field reference --fieldName clientes --type ~.domain.cliente --class ~.domain.factura

field set --fieldName facturas --type ~.domain.factura --mappedBy clientes --notNull false --cardinality ONE_TO_MANY --class ~.domain.cliente

4. crea una tabla que se llame detalles facturas con los siguientes campos

Nombre	tipo de datos	tamaño /formato	Otras
numfactura	texto	6	Clave foránea
Código	texto	6	Clave foránea
Cantidad	numérico	entero largo	

Spring Roo – Creación de la Entidad detallefacturas:

entity --class ~.domain. detallefactura –-testAutomatically

field number --fieldName cantidad --type java.lang.Integer --notNull --min 0

Cardinalidad uno a mucho

field reference --fieldName facturas --type ~.domain.factura --class ~.domain.detallefactura
field set --fieldName detallefacturas --type ~.domain. detallefactura --mappedBy facturas --notNull false --cardinality ONE_TO_MANY --class ~.domain.factura

field reference --fieldName productos --type ~.domain.producto --class ~.domain.detallefactura
field set --fieldName detallefacturas --type ~.domain. detallefactura --mappedBy productos --notNull false --cardinality ONE_TO_MANY --class ~.domain.producto

Spring Roo – Creación del proyecto Spring MVC con las entidades especificadas:

web mvc setup

web mvc all --package ~.web

Spring Roo – Creacion de lenguajes soportados en el proyecto

web mvc language --code de //Aleman

web mvc language --code es //Español

web mvc language --code en //Ingles

Spring Roo – Búsquedas

- Lista los posibles búsquedas que se puede realizar con las tablas
    - **finder list --class ~.domain.cliente**
    - finder add --finderName findClientesByNombreLike
    - finder add --finderName findClientesByTelefonoEquals

    - **finder list --class ~.domain.producto**

    - finder add --finderName findProductoesByNombreLike
    - finder add --finderName findProductoesByFechacadBetween
    - finder add --finderName findProductoesByPrecioBetween

    - **finder list --class ~.domain.factura**
    - finder add --finderName findFacturasByClientes
    - finder add --finderName findFacturasByFechacadBetween
    - finder add --finderName findFacturasByDetallefacturas

    - **finder list --class ~.domain.detallefactura**

- finder add --finderName findDetallefacturasByFacturas
- finder add --finderName findDetallefacturasByProductos

Jasper Roo - Reportes

Descargar el archivo ca**.digitalface.jasperoo-0.1.5-RELEASE.jar** dentro de la carpeta bin de spring roo

Instalacion de jasper

osgi start --url http://s.digitalface.ca/jasperoo-latest

jasperoo setup

jasperoo setup --formats "pdf, odt, csv,xls"

jasperoo all

Spring Roo – Seguridad en la apicacion

security setup

Spring Roo - Definir IDE

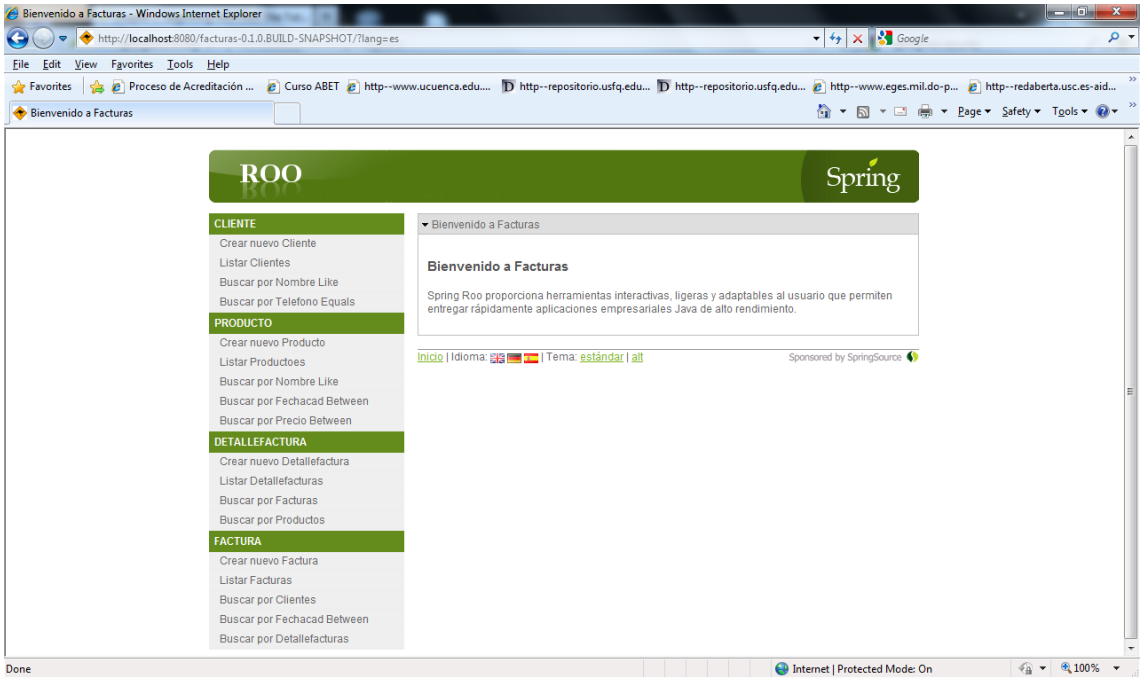
perform eclipse

Spring Roo - Compilación del proyecto

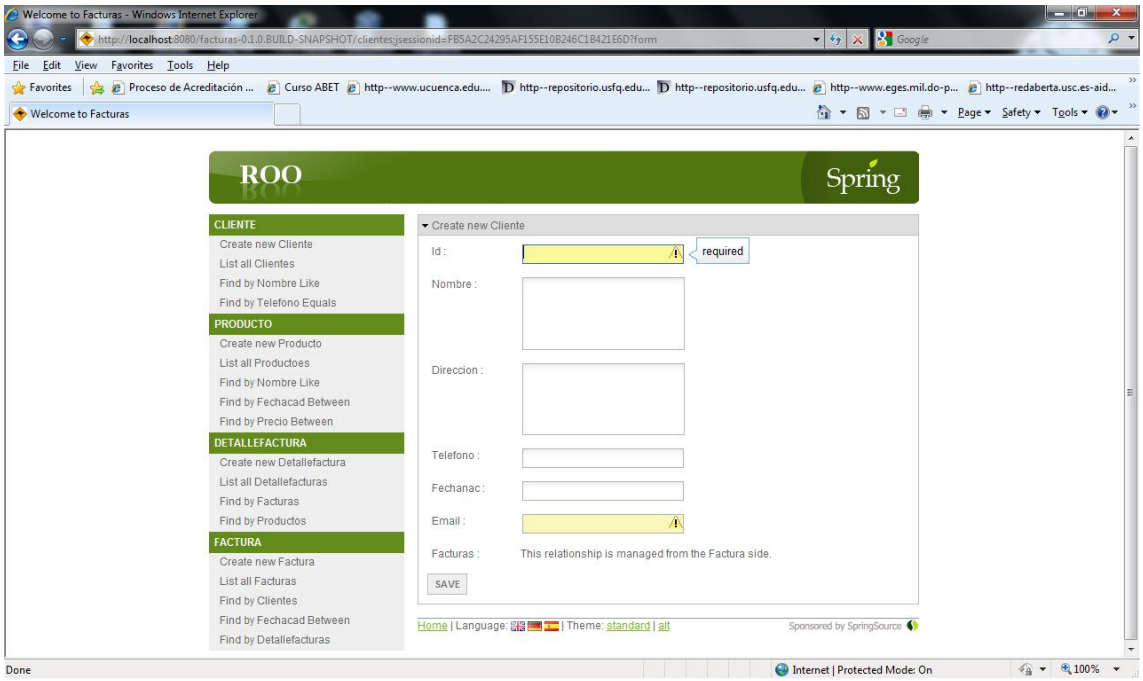
perform package

Aplicación resultante con su seguridad de ingreso:

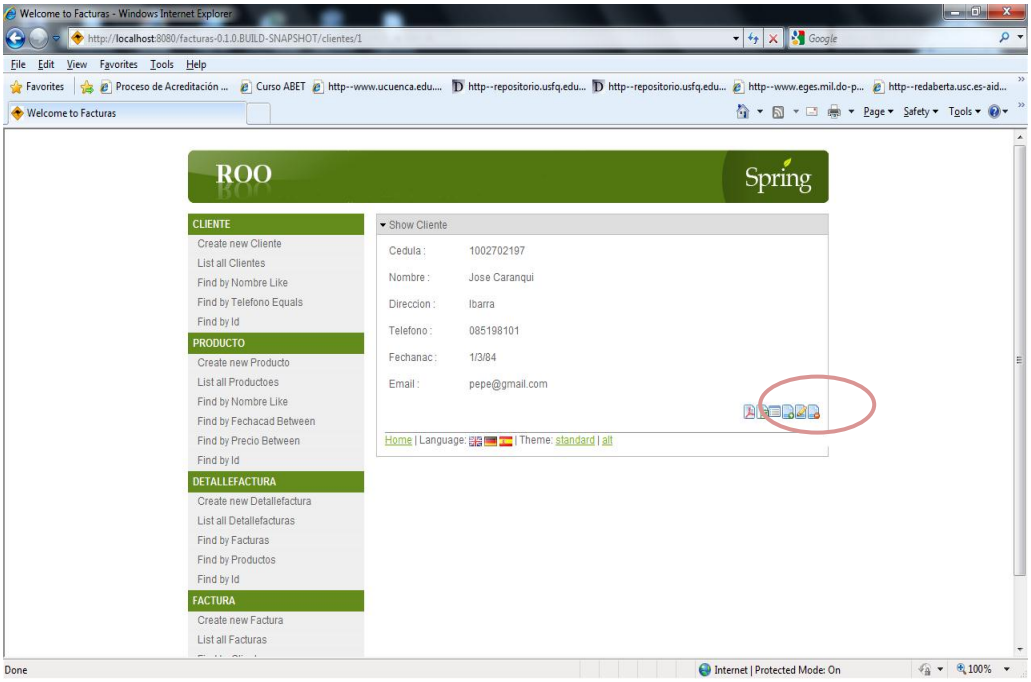
Una captura de pantalla de la aplicación Facturación:



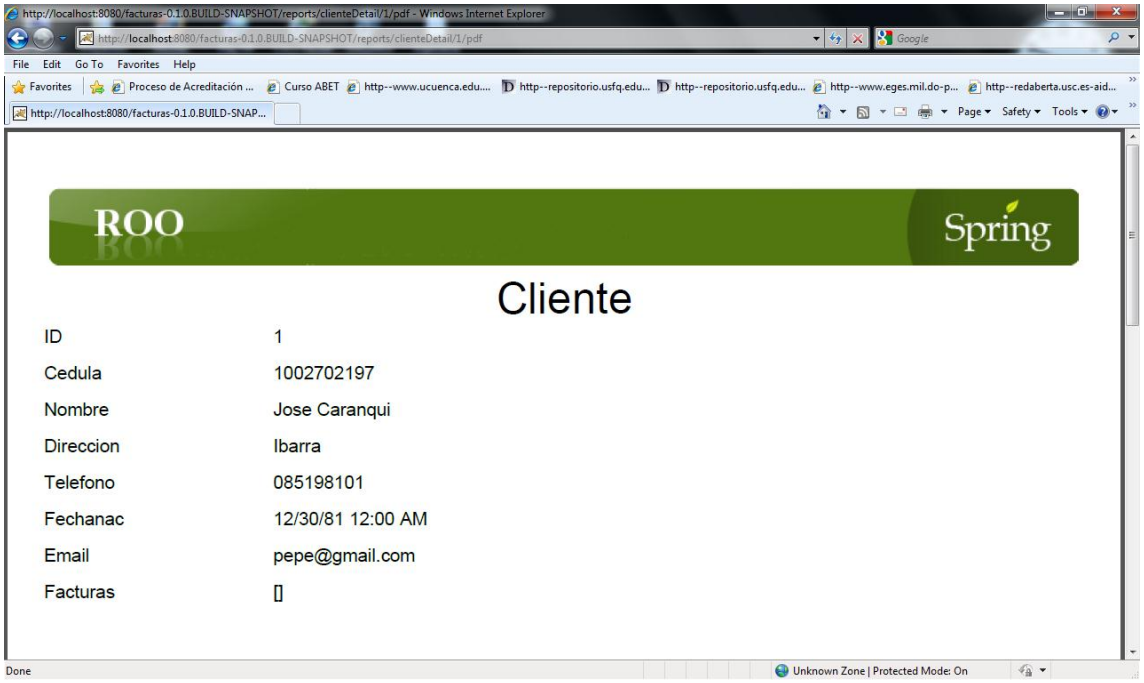
Insertar de datos con validaciones:



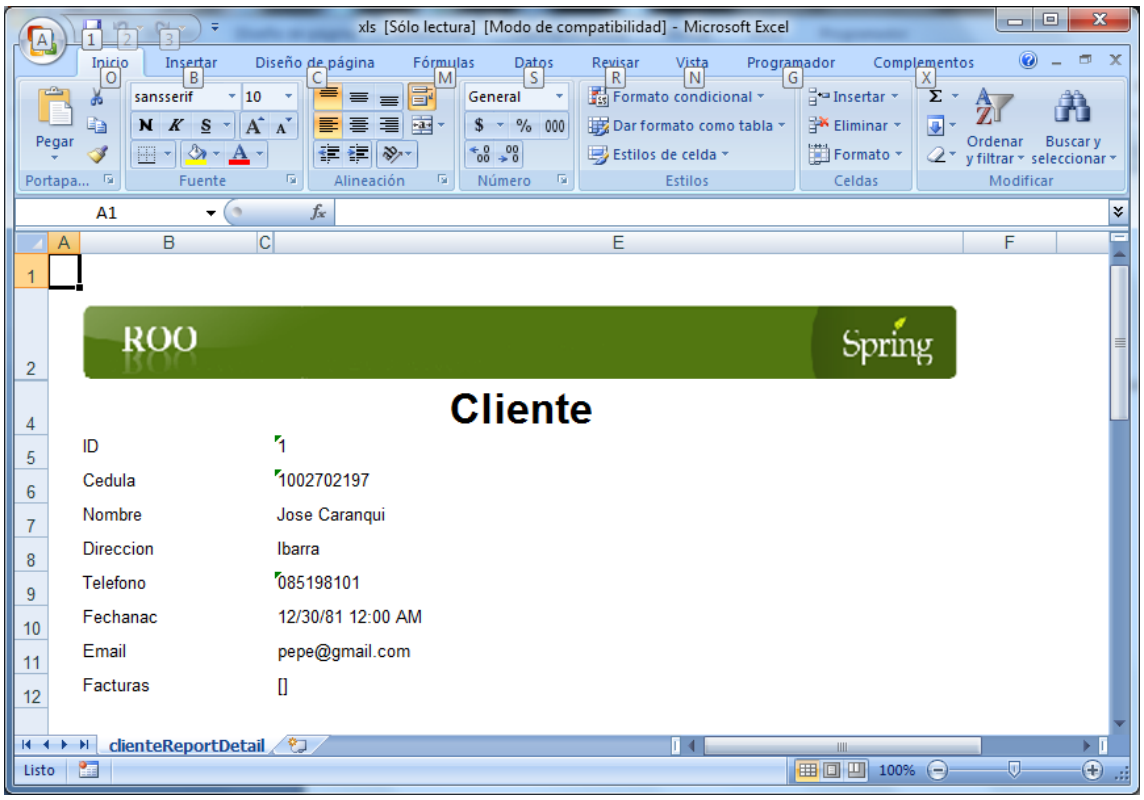
Reportes en formatos pdf, xls, entre otros:



- **Reporte en formato PDF**



- **Reporte en formato XLS**



Capa de Presentación de Spring Roo

Spring Roo genera la interfaz gráfica de usuario que puede personalizarse, teniendo en cuenta los siguientes componentes:

Directorio	Propósito
/styles	para las hojas de estilo del proyecto (*.css), puede haber más de una
/images	para las imágenes del proyecto
/WEB-INF/classes/*.properties	para la configuración de las páginas de estilo
/WEB-INF/Spring/*.xml	para la configuración de los controladores
/WEB-INF/i18n/*.properties	para la internacionalizacion de los mensajes
/WEB-INF/layauts/layouts.xml	para la configuración de las páginas maestras de la aplicación (home, menu, default)
/WEB-INF/tags/*.tagx	para los tags de declaración de paginado, lenguajes y estilos
/WEB-INF/views/**/*	para las vistas
/WEB-INF/web.xml	para configurar el contexto web de la aplicacion
/WEB-INF/urlrewrite.xml	para configurar la re-escritura de urls de la aplicación

Con la generación de una aplicación web con Spring Roo y algunos de los siguientes cambios podemos lograr una personalización de su apariencia, sin necesidad de modificar cada una de las pantallas.

ANEXO C: ANTEPROYECTO

UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS
ESCUELA DE INGENIERÍA DE SISTEMAS COMPUTACIONALES

ANTEPROYECTO DE TESIS

DATOS GENERALES

1. TEMA: **Sistema de Administración de Bróker de Seguros utilizando Software Libre para la Institución "Asesores Productores De Seguros Álvaro Cazar"**	
2. APLICATIVO: **Sistema de Administración del Bróker de Seguros (ACBróker 1.0)**	
3. DISCIPLINA / AREA: **Informática / Bróker de Seguros**	
4. ENTIDAD QUE AUSPICIA: **"Asesores Productores De Seguros Álvaro Cazar"**	
5. DIRECTOR DEL PROYECTO: Ing. Edgar Maya	
6. AUTOR: **Jhoanna Paola Villegas Estévez**	
7. TELÉFONO: **087416034**	8. FAX: 062601945
9. CORREO ELECTRÓNICO: sukave84@hotmail.es	
10. DURACIÓN DEL PROYECTO: **8 meses**	
11. ESTADO DEL PROYECTO: Nuevo [**x**] De Continuación [ ]	
12. PRESUPUESTO: $ 1.500,00	
PARA USO DEL CONSEJO ACADÉMICO	
FECHA DE ENTREGA:	FECHA DE REVISIÓN:
APROBADO: Si [ ] No [ ]	FECHA APROBACIÓN:
OBSERVACIONES:	

Universidad Técnica del Norte

FACULTAD DE INGENIERIA EN CIENCIAS APLICADAS
ESCUELA DE INGENIERIA EN SISTEMAS COMPUTACIONALES

PLAN DEL PROYECTO DE TITULACION

Propuesto por: Jhoanna Paola Villegas Estévez	**Áreas Técnicas del Tema:** Investigación Ingeniería de Software Ingeniería Web
Director del Proyecto: Edgar Alberto Maya Olalla	**Fecha:** 7 de febrero 2011
Teléfonos: (06)2959653 – 087416034	**Correo electrónico:** sukave84@hotmail.es

1. Tema o Título del proyecto

Sistema de Administración de Bróker de Seguros utilizando Software Libre para la Institución "Asesores Productores De Seguros Álvaro Cazar".

2. Problema

2.1. Antecedentes

El Bróker de Seguros actúa como intermediario entre Clientes y Aseguradoras, es responsable de conjuntar a dichas partes y su objetivo es proporcionar al cliente un producto (Programa de Seguros) que se ajuste a sus necesidades.

La principal función de un Bróker de Seguros es el de velar por los interés de los asegurados / clientes frente a las Aseguradoras, para lo cual desde el inicio de la relación el bróker es el encargado de realizar la inspección del riesgo para con esta información entregar al cliente el producto que requiere, el mismo que deberá contemplar las cláusulas y coberturas que cubran los riesgos potenciales.

Como también es el responsable de verificar que las liquidaciones de los siniestros sean acorde a las condiciones establecidas en los contratos de seguros a fin de que el cliente este satisfecho con la liquidación recibida.

El compromiso del bróker "Asesores Productores De Seguros Álvaro Cazar", es brindar un asesoramiento integral y personalizado a los clientes, con la precisión y la dinámica de un Bróker de sólida experiencia en la administración y elaboración de Programas de Seguros a la medida de cada cliente, basándonos primordialmente en la transparencia, la ética y la permanente innovación del mercado asegurador.

2.2. Situación Actual

La administración de la Institución ASESORES PRODUCTORES DE SEGUROS ÁLVARO CAZAR", la realiza utilizando registros repetitivos en varios archivos de Excel, desaprovechando tiempo y recursos que posee, por lo que cuando el Gerente o los Empleados de la Institución necesitan obtener reportes de Pólizas o información del estado de los siniestros ocurridos o de igual manera para realizar renovaciones de pólizas o dar seguimiento a las mismas, se encuentran con los siguientes inconvenientes:

- Administración incorrecta de la información, obteniendo Reportes a destiempo.
- En ocasiones información errónea o pérdida de la misma.
- Desactualización de datos.
- Inoportuna atención a los Clientes, ya que para brindar una atención de calidad es necesario contar con toda la información en tiempo real.

Por consecuencia la Institución no cuenta con un Sistema de Administración que permita tener un correcto manejo de la información.

2.3. Prospectiva

La Institución ASESORES PRODUCTORES DE SEGUROS ÁLVARO CAZAR, requiere de un Sistema que le permita la Administración de la Información de una manera integral y eficiente, utilizando los recursos con los que dispone.

2.4. Formulación del Problema.

El compromiso del bróker "Asesores Productores De Seguros Álvaro Cazar", es brindar un asesoramiento integral y personalizado a los clientes, con la precisión y la dinámica basada en la experiencia en la administración y elaboración de Programas de Seguros.

La falta de administración de la información de la Institución al no contar un Sistema que le permita una solución integral con la posibilidad de acceder a ésta de manera ágil.

3. Objetivos

3.1. Objetivo General

- Desarrollar e Implementar un Sistema de Administración de Bróker de Seguros, para la Institución Asesores Productores de Seguros Álvaro Cazar utilizando Software Libre, el cual permitirá brindar una mejor atención a sus Clientes por medio de la adecuada gestión de la información que posee.

3.2. Objetivos Específicos

- Describir las soluciones integrales que presta la Institución Asesores Productores de Seguros Álvaro Cazar, para desarrollar un sistema abierto y parametrizable.
- Estudiar las tecnologías de desarrollo de software libre aplicando las últimas tecnologías para obtener un producto de calidad.
- Diseñar la aplicación basada en la metodología de desarrollo de Software RUP.
- Diseñar un Interface amigable e intuitivo para mejorar la usabilidad y disminuir la curva de aprendizaje del sistema.

3.3. Objetivos De Estudio

- Estudiar las herramientas libres de desarrollo del Sistema de Administración de Bróker de Seguros.
- Analizar las ventajas y desventaja de utilizar herramientas libres.
- Estudiar y aplicar la metodología de desarrollo de software RUP.

4. Alcance y Limitaciones.

4.1. Alcance

En la Implementación del Sistema de Administración de Bróker de Seguros para la Institución ASESORES PRODUCTORES DE SEGUROS ÁLVARO CAZAR, se conseguirá que la gestión de la información que posee se la realice de la manera adecuada para poder obtener reportes de la información que se necesite en el momento indicado para poder cubrir con las necesidades de la misma y por ende poder brindar un eficiente seguimiento a los trámites y requerimientos del Cliente para de esta manera brindar una atención de la calidad a los mismos y también a los Usuarios del sistema.

El desarrollo del Sistema de Administración de Bróker de Seguros será escalable y flexible de acuerdo al crecimiento de la misma, sumando tecnología y nuevas herramientas.

Módulos

El Sistema a desarrollar será orientado a la web y será alojado en un servidor web de la Institución, todos los usuarios se conectaran a él desde la Intranet o Internet.

El Sistema constara de los siguientes módulos:

Modulo Seguridad

Gestión de Usuarios

Gestión de Roles de Usuarios

Clientes

Gestionar la base de asegurados con vistas personalizadas, contactos, actividades diarias, y documentación electrónica.

Compañías

Definición los contactos en cada compañía y registro las actividades realizadas. Planes, coberturas, planes comisiónales para la institución, productor y vendedores.

Gestión de Pólizas

Permitirá trabajar con distintos riesgos, definir el plan comisional, imprimir los recibos de pagos, renovar pólizas, anulación de pólizas y monto de las cuotas, administra estados de la póliza.

Liquidación de comisiones

Controlar y generar tanto las comisiones que recibe de parte de la compañía para el Bróker, como las que defina para los vendedores.

Siniestros y Trámites

Administrar al detalle cada uno de los siniestros que ocurran como así también trámites externos que se realice. Actividades, documentación electrónica, en fin, todas las herramientas para estar en todos los detalles.

Administración

Organizar y planificar la gestión administrativa con los módulos de compañías, asegurados, pólizas. Herramientas necesarias para la correcta gestión administrativa de la Organización.

Módulos de Reportes

Clientes

Pólizas

Siniestros

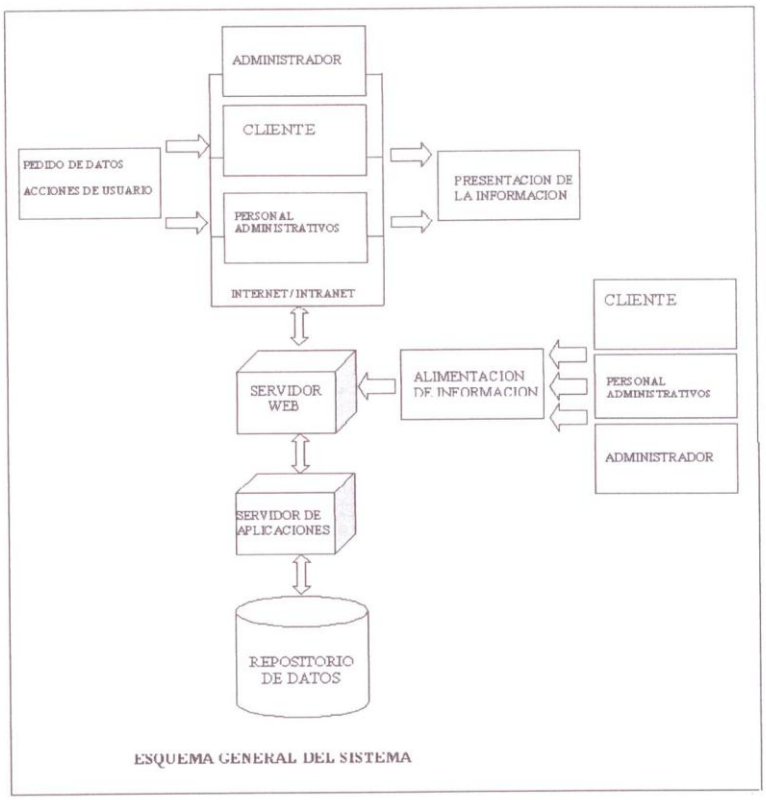
Liquidaciones

Reporte para la Superintendencia de Bancos y Seguros.

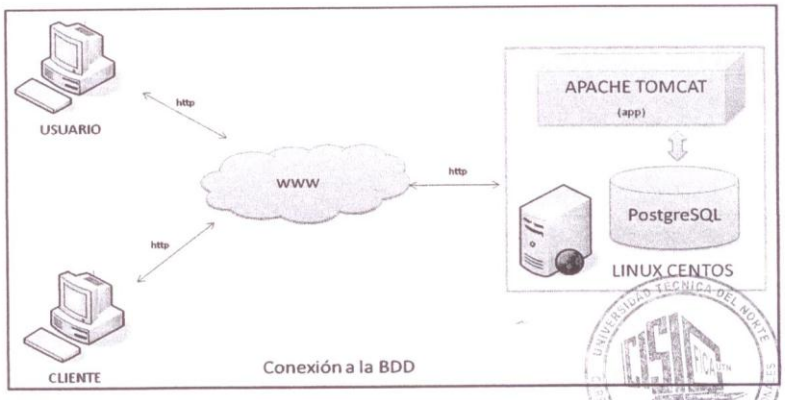
4.2. Limitaciones

- El Sistema será implementado en la Institución ASESORES PRODUCTORES DE SEGUROS ÁLVARO CAZAR.
- El desarrollo del Sistema de Administración será basada a la información proporcionada por la Institución.

5. ESQUEMA



ESQUEMA GENERAL DEL SISTEMA



Conexión a la BDD

Herramientas

Las Herramientas a utilizarse serán libres por lo que no requerirán costos en la adquisición de licencias pero que cumplan con los requerimientos necesarios del Sistema a desarrollar, se utilizaran las siguientes:

Software Libre

- Sistema Operativo Linux
- Servidor Web Apache Tomcat.
- Framework Spring
- Base de Datos Postgres
- Arquitectura MODELO – VISTA – CONTROLADOR RUP

Metodología de Desarrollo

Se usará la metodología de desarrollo RUP (Proceso Unificado de Rational) además del Lenguaje Unificado de Modelado UML para el análisis, implementación y documentación de sistema a desarrollar. Tiene cuatro etapas:

- **Planeación:** comprensión del problema y la tecnología. Delimitación del ámbito o alcance del sistema. Establecimiento de una línea base de la arquitectura.
- **Elaboración:** Construir una versión ejecutable de la arquitectura de la aplicación. Trabajo en requerimientos, modelo de negocios, análisis y diseño.
- **Construcción:** Completar el esqueleto de la aplicación con la funcionalidad. Construir una versión beta.
- **Transición:** Hacer disponible a la aplicación para los usuarios finales. Construir la versión Final.

6. JUSTIFICACIÓN

Como en toda Institución ya sea esta Pública o Privada, se necesita brindar una atención de calidad a los Clientes al momento de ofrecerles un bien o servicio, ya que ellos son vitales para el crecimiento, con relación a lo económico o al prestigio de las mismas, es por eso que siempre se debe estar implementando técnicas para alcanzar estos objetivos y estar siempre optimizando los recursos que se poseen, más ahora que se tiene al alcance un gran avance de la Ciencia y la

tecnología los cuales permiten implementar sistemas que gestionen la información de una manera actualizada que vaya a la par con el crecimiento tecnológico que se da hoy en día.

El "Bróker de Seguros Álvaro Cazar", a lo largo de su presencia en el medio, ha ido creciendo, motivo por el cual ha incrementado su registro de Clientes, Proveedores y por ende la información, y al no contar con un Sistema de Automatización, necesita obligatoriamente automatizar sus procesos por medio de una técnica actualizada que permita gestionar toda su información de una manera adecuada, para brindar un servicio eficaz a todos su actuales y futuros Clientes.

La implementación de estas modernas herramientas en la realización del Sistema aportará las siguientes ventajas.

- Adecuada gestión de la información que se tiene en la Institución
- Se obtendrá reportes, consultas en tiempo real, para las necesidades de los usuarios.
- Un entorno amigable hacia el usuario para que pueda acceder a la información desde cualquier ubicación que se encuentre.

7. Temas Afines

AUTOR	ANTEPROYECTO	DIFERENCIA
JACOME OROZCO ANA GABRIELA	ESTUDIO DE LAS METODOLOGIAS RAPIDAS DE DESARROLLO DE SOFTWARE CON EL APLICATIVO, PROTOTIPO SISTEMA DE INFORMACION DE UN BROKER DE SEGUROS	**Este aplicativo está dirigido específicamente al Bróker de Seguros NOVA ECUADOR, y se basara en la administración de la información para uso exclusivo de los empleados del bróker mediante metodologías rápidas de desarrollo de software, para poder obtener la información de las pólizas, sus clientes e indemnizaciones.** **La diferencia es que el Sistema de Administración ACBróker, será exclusivo para el Bróker Asesores de Seguros Álvaro Cazar basado en herramientas libres, administrara información de las aseguradoras, pólizas, sus clientes, siniestros, liquidación de comisiones, reporte para la Superintendencia de Bancos y Seguros, a la cual tendrán acceso, según corresponda a los empleados del Bróker ya sean de la matriz o sucursal y clientes del bróker, desde cualquier lugar.**

8. Temario

Capitulo 1. Asesores Productores de Seguros ALVARO CAZAR

1.1 Antecedentes de la Institución

1.2 Misión

9. Cronograma de Actividades

ID	NOMBRE DE LA TAREA	DURACIÓN							
		MES 1	MES 2	MES 3	MES 4	MES 5	MES 6	MES 7	MES 8
	Fase Inicial								
1	Reunión con el Gerente del Bróker								
2	Planificación del Proyecto								
3	Análisis de Requerimientos								
4	Recolección de Información Necesaria								
5	Desarrollo del Documento de Visión								
	Fase de Elaboración								
6	Investigación Bibliográfica								
7	Estudio de la Herramienta de Desarrollo								
8	Definición de Casos de Uso								
9	Diseño de Interfaces de Usuario								
10	Estructuración de la Base de Datos								
11	Elaboración del Sistema								
12	Validación y Pruebas								
	Fase de construcción								
13	Implementación de Casos de Uso fase 1								
14	Validación y Pruebas								
15	Implementación de Casos de Uso fase 2								
16	Validación y Pruebas								
17	Desarrollo de Manuales (Usuario y Técnico)								
	Entrega								
18	Desarrollo del Documento Final de Tesis								

10. Costos

COSTOS HARDWARE		
DESCRIPCIÓN	**COSTO REAL**	**COSTO FAVORECIDO**
Computador	1000	0
• Procesador Intel Core 2 Duo 2.2 GHz.		
• 4 GB de Memoria RAM		
• Disco de 120 GB.		
• Monitor 17.1"		
• Impresora Canon BJC-1000		
• Teclado y Mouse PS/2		
COSTOS SOFTWARE		
Linux	0	0
JDeveloper 11	0	0
Postgresql Data Base	0	0
BIBLIOGRAFIA		
Libros	200	200
Internet (7 meses)	150	150
Revistas y Folletos	80	80
Copias	100	100
OTROS MATERIALES		
Cartuchos de Impresión	100	100
Útiles de Oficina	200	200
Luz Eléctrica	200	200
Transportes	100	100
SUBTOTAL	**2130**	**1130**
Imprevistos 10%	213	113
TOTAL	**2343**	**1243**

11. Bibliografía

Web

- Wikepedia, de
  http://es.wikipedia.org
- Apache Software Foundation, de
  http://es.wikipedia.org/wiki/Apache_Software_Foundation
- JavaServer Pages, de
  http://es.wikipedia.org/wiki/JSP
- HTML, de
  http://es.wikipedia.org/wiki/HTML
- Hojas de estilo en cascada, de
  http://es.wikipedia.org/wiki/CSS
- Proceso Unificado de Rational, de
  http://es.wikipedia.org/wiki/RUP

Libros

- Rich Bowen, Ken Coar (2007) Apache Administrators. Apache Cookbook: Solutions and Examples, 2da Edición
- Ivan Ristic (2005). Apache Security, 1era Edición
- Andrew Ford (2008). Apache 2 Pocket Reference: For Apache Programmers & Administrators, 1era Edición
- The Apache Software Foundation (2010). Apache HTTP Server 2.2 Official Documentation - Volume I. Server Administration
- Andrea Steelman and Joel Murach (2010). Murach's Java Servlets and JSP, 2da Edición
- Vivek Chopra, Jon Eaves, Rupert Jones, Sing Li (2005), Beginning JavaServer Pages
- Elizabeth Castro, HTML, XHTML, and CSS, 6ta edición

- Nicklas Persson and Christopher Murphy(2008), HTML and CSS Web Standards Solutions: A Web Standardistas' Approach
- Steven M. Schafer (2010), HTML, XHTML, and CSS Bible
- Christopher Murphy and Nicklas Persson (2009), Html y CSS (Programacion/ Programming) (Spanish Edition)
- Sergio Guardiola Herrador (2010) HTML & CSS Fácil y sencillo
- Steven M. Schafer. HTML, XHTML y CSS (Spanish Edition)
- Per Kroll Philippe Kruchten, Grady Booch(2003). The Rational Unified Process Made Easy: A Practitioner's Guide to the RUP
- Ahmad K. Shuja, Jochen Krebs(2008), IBM Rational Unified Process Reference and Certification Guide: Solution Designer (RUP)
- Philippe Kruchten, The Rational Unified Process: An Introduction (3rd Edition)