

Trabajo para Optar por el Título de Ingeniero en Informática

Universidad de Cienfuegos “Carlos Rafael Rodríguez”

Facultad de Informática

Curso 2007-2008

Carisure

Aplicación web para vincular clientes potenciales con vendedores de productos de seguros

Autor: Felix St. Bernard

Tutor: MSc. Oscar Luis Muñoz

Declaración de Autoría

Yo, Felix St. Bernard, declaro que soy el único autor del trabajo de diploma titulado: Carisure – Sistema para vincular dientes potenciales con vendedores de productos de seguros y autorizo a la empresa Spicelander Financial de Granada y al Departamento de Informática de la Facultad de Informática en la Universidad de Cienfuegos “Carlos Rafael Rodríguez”, para que hagan el uso que estimen pertinente con el trabajo de diploma.

Para que así conste firmo la presente a los 27 días del mes de Junio del 2008.

X

Felix St. Bernard

Los abajo firmantes certificamos que el presente trabajo ha sido revisado según acuerdo de la dirección de nuestro centro y el mismo cumple los requisitos que debe tener un trabajo de esta envergadura.

X

Firma Tutor

X

Firma ICT

X

Firma Vicedecana

Dedicatoria

A mis padres...

Resumen

Este presente trabajo tiene como objetivo el desarrollo de una aplicación web para vincular clientes potenciales con vendedores de productos de seguros. Con tal sistema, se pretende resolver dos fuentes de frustración – por la parte del cliente, la búsqueda de una póliza de seguros y por la parte del agente, encontrar más gente que a lo mejor ya están en la búsqueda de seguros.

Se nota en el Caribe que la industria de seguros no aprovecha de los beneficios de Internet. La audiencia de objetivo será la región del Caribe donde no existe ninguna aplicación como la que se trata este trabajo.

Para elaborar el sistema propuesto por este trabajo, fueron analizados el uso de patrones de arquitectura de diseño y desarrollo, y también el uso del proceso ICONIX como la metodología elegida para el análisis e implementación de este trabajo.

Índice

Declaración de Autoría.....	ii
Dedicatoria.....	iii
Resumen.....	iv
Índice.....	v
Tabla de Figuras.....	viii
Introducción.....	1
1.1 Descripción del objeto de estudio.....	3
1.1.1 Situación Problemática.....	3
1.1.2 Problema a Resolver.....	4
1.1.3 Objeto de estudio y campo de acción.....	4
1.1.4 Idea a defender.....	4
1.1.5 Objetivo General.....	5
1.1.6 Valor Práctico.....	5
1.1.7 Estructuración del contenido del trabajo de diploma.....	5
Capítulo 1 Fundamentación Teórica.....	7
1.1 Introducción.....	7
1.2 Descripción del dominio del problema.....	7
1.2.1 Principios de seguros.....	7
1.2.2 Sistemas existentes.....	11
1.2.3 Tendencias, Metodologías y Tecnologías a considerar.....	11
1.3 Conclusión.....	21
Capítulo 2 Análisis y Modelación de Requerimientos.....	22
2.1 Introducción.....	22
2.2 Requerimientos Funcionales.....	22

2.2.1	Perspectiva del Agente	22
2.2.2	Perspectiva del diente	23
2.3	Requerimientos No-funcionales.....	23
2.3.1	Requisitos de Interfaz Externa	24
2.3.2	Requisitos de Rendimiento.....	24
2.3.3	Requerimientos de Soporte.....	24
2.3.4	Requerimientos de Usabilidad.....	24
2.3.5	Requerimientos de Portabilidad	24
2.3.6	Requerimientos de Seguridad.....	24
2.3.7	Requerimiento del Software.....	24
2.3.8	Requerimientos Básicos del Hardware	25
2.3.9	Requisitos Legales	25
2.4	Modelo del Dominio	25
2.5	Modelo de Casos de Uso	26
2.5.1	Actores del Sistema	27
2.5.2	Casos de uso del sistema.....	27
2.5.3	Diagramas de Robustez.....	41
2.6	Conclusiones	41
Capítulo 3	Modelación del Sistema Propuesto.....	43
3.1	Introducción.....	43
3.2	Diagramas de clases.....	43
3.3	Diagrama del modelo físico de datos.....	44
3.4	Diagramas de despliegue.....	46
3.5	Principio de Diseño del sistema	47
3.5.1	Estándares en la interfaz de la aplicación	47
3.5.2	Tratamiento de errores.....	48

3.5.3	Estándares de codificación.....	48
3.6	Conclusiones	49
Capítulo 4	Estudio de factibilidad.....	50
4.1	Introducción.....	50
4.2	Cálculos.....	51
4.3	Beneficio tangibles e intangibles.....	54
4.4	Conclusiones	54
Conclusiones.....		56
Recomendaciones		57
Bibliografía.....		58
Anexo A	Diagramas de Robustez.....	60

Tabla de Figuras

Figura 1-1 Diagrama de los características del RUP.....	13
Figura 1-2 El proceso ICONIX.....	15
Figura 1-3 Patrón MVC.....	18
Figura 1-4 Biblioteca de Codeigniter	19
Figura 2-1 Diagrama del modelo del dominio del sistema propuesto	26
Figura 2-2 Diagrama de actores del sistema	27
Figura 2-3 Diagrama de Paquetes de Casos de Uso	28
Figura 2-4 Diagrama de Casos de Uso del Paquete Agentes.....	28
Figura 2-5 Diagrama de Casos de Uso del Paquete Clientes	36
Figura 2-6 Análisis de Robustez.....	41
Figura 3-1 - Diagrama de Paquete del las clases del sistema propuesto.....	43
Figura 3-2 Diagrama de clase del paquete controladores	44
Figura 3-3 Diagrama de clase del paquete modelos	44
Figura 3-4 Modelo físico.....	45
Figura 3-5 Diagrama de Despliegue(desarrollo)	46
Figura 3-6 Diagrama de Despliegue(Propueta para la implementación).....	47
Figura 4-1 Factores de esfuerzo	53
Figura A-1 Diagrama de Robustez "Autenticar Agente"	60
Figura A-2 Diagrama de Robustez "Adquirir Créditos"	60
Figura A-3 Diagrama de Robustez "Borrar cuenta de Agente"	61
Figura A-4 Diagrama de Robustez "Crear cuenta de agente".....	62
Figura A-5 Descripción de caso de uso "Desuscribirse de una clasificación de leads"	63
Figura A-6 Diagrama de Robustez de caso de uso "Mostrar Panel de Control"	63
Figura A-7 Diagrama de robustez del caso de uso "Rechazar Lead"	64
Figura A-8 Diagrama de robustez del caso de uso "Suscribir a clasificaciones leads"	64
Figura A-9 Diagrama de robustez "Buscar Contraparte"	65
Figura A-10 Descripción de caso de uso "Buscar agentes por el directorio"	65
Figura A-11 Diagrama de Robustez del caso de uso "Refinar Búsqueda".....	66
Figura A-12 Diagrama de Robustez del Caso de Uso "Ver perfil del Agente"	66
Figura A-13 Diagrama de Robustez del caso de uso "Verificar Correo"	67

Introducción

La industria de seguros tiene un papel extremadamente importante en el mundo actual. Básicamente, el propósito de todos los tipos de seguros es proteger una persona y/o su familia, una empresa, etc., del impacto monetario de una tragedia. El propósito de seguros no es ayudar a presupuestar gastos de tamaño módico, sino proteger de un evento verdaderamente catastrófico. Por ejemplo, en la vasta mayoría de los países, no puede manejarse un auto sin la cobertura de una póliza de seguro de autos.

El seguro es casi tan antiguo como la sociedad humana. A lo largo de la historia han existido dos tipos de economías: Las economías de dinero (con mercados, dinero, instrumentos financieros, etcétera) y las economías naturales (sin dinero, los mercados, los instrumentos financieros, etcétera). Este segundo tipo es una forma más antigua. En tal economía y comunidad, el seguro se manifiesta en forma de personas ayudando a sus semejantes. Por ejemplo, si una casa es destruida por el fuego, los miembros de la comunidad ayudan a construir una nueva. Si le ocurre algo a un ciudadano, los vecinos deben ayudarlo. De otra manera, los vecinos no recibirán ayuda en el futuro. Este tipo de seguro ha sobrevivido hasta el presente en algunos países donde la economía moderna de dinero, con sus instrumentos financieros, no está muy extendida (por ejemplo en algunas zonas de países que formaban parte de la extinta Unión Soviética).

Las raíces del seguro podrían ser rastreadas desde Babilonia, donde los comerciantes fueron promovidos para asumir los riesgos del comercio de caravanas a través de los préstamos, que fueron recompensados (con interés) sólo después de que los bienes habían llegado en forma segura — una costumbre parecida a la de “barco en garantía” y con fuerza legal en el Código de Hammurabi (c.2100 AC) donde los fenicios y los griegos aplicaron un sistema similar a su comercio transportado por mar. Los romanos utilizaron asociaciones de entierro como una forma de seguro de vida, proveyendo gastos funerales para miembros y los posteriores pagos para los sobrevivientes.

Los monarcas Achaemenian fueron los primeros en asegurar a sus personas y le hicieron oficial registrando el proceso asegurador en oficinas notariales gubernamentales. La tradición de seguro se mantuvo cada año en Norouz (comenzando el Año Nuevo Iraní); Los superiores de grupos étnicos diferentes, así como también otros dispuestos a participar, presentaban regalos al monarca. El regalo más importante se entregaba durante una ceremonia especial. Cuando un regalo valía más que 10,000 Derrik (la moneda de oro Achaemenian) el detalle se registraba en una oficina especial. Esto fue

Carisure – Aplicación web para vincular clientes potenciales con vendedores de productos de seguros

ventajoso para aquellos que presentaron tales regalos especiales. Para otros, los regalos eran evaluados por los confidentes de la corte. También, esta valoración estaba registrada en oficinas(1).

El propósito del monarca y la corte, al registrar los regalos, era el de ayudar a la persona registrada, en caso de que esta tuviera problemas. Jahez, historiador y escritor, escribió en uno de sus libros en el antiguo Irán: " En cualquier momento que el dueño del regalo esté en problemas o quiera construir un edificio, hacer una fiesta, o realizar la boda de algún descendiente, etcétera. La persona que esté al frente de esos asuntos en la corte, revisaba la inscripción. Si la cantidad registrada excedía 10,000 Derrik, la persona recibiría un importe equivalente al doble de lo registrado".

Antes de que el seguro fuese establecido a finales de los el siglo 17, "las sociedades de socorro mutuo" ya existían en Inglaterra. En ellas las personas donaron cantidades de dinero a una suma general que pudo servir para emergencias. Los contratos separados de seguro (por ejemplo, las pólizas de seguros no atadas en bultos con préstamos u otras dases de contratos) fueron inventados en Génova en el siglo XIV.

Hacia fines del el siglo XVII, la importancia creciente de Londres como un centro para el comercio aumentó la demanda para el seguro marino. A finales de los 1680s, el Señor Edward Lloyd abrió una cafetería que se convirtió en un lugar muy frecuentado de dueños de barcos, comerciantes, y capitanes de barcos, y por lo tanto, consecuentemente, se tenía una fuente confiable de las últimas noticias navieras. Se convirtió en el lugar de reunión para grupos que deseaban comprar un seguro para cargamentos y barcos. Estos grupos estaban dispuestos a asumir responsabilidad financiera por tales riesgos(1).

El seguro, como se conoce hoy en día, puede ser rastreado por el 1666, cuando los acontecimientos del Gran Fuego de Londres, el cuál devoró 13,200 casas. Como consecuencias de este desastre, Nicholas Barbon abrió una oficina para garantizar edificios. En 1680, él estableció la primera compañía de seguro contra incendios de Inglaterra, "la Oficina de Fuego," para garantizar casas del ladrillo(2).

En la actualidad, casi todas las formas de hacer negocios han encontrado una forma de hacer lo mismo utilizando Internet como un medio. La industria de seguros no es una excepción de esta enunciación. Ya se puede ver manifestaciones de estas aplicaciones en formas de sitios web que hacen publicidad a sus productos. También, hay aplicaciones web que intentan vender o hacer publicidad a los productos ajenos, ganando una comisión de la venta (actúan como "corredores de seguros" en línea). Estas manifestaciones también han producido como respuesta un nivel de competencia muy alto entre los

Carisure – Aplicación web para vincular clientes potenciales con vendedores de productos de seguros

varios vendedores de productos seguros. Lo que significa esto, es que para los consumidores, hay más valor para su dinero.

Antes de la existencia de la web, una persona tenía tres opciones generales para comprar seguros:

1. Buscar un agente que venda seguros (esperando que su oferta sea la de mejor valor para su dinero)
2. Investigar por sus propios medios los diferentes tipos de pólizas disponibles e ir comparando las ofertas de una empresa a otra (a menudo, virtualmente imposible dado el número inmenso de empresas de seguros en una localidad dada).
3. Esperar que un agente le encuentre y le ofrezca una oferta que sea buena para su interés.

El nivel de incertidumbre, que hay aparente en las opciones disponibles, no es aceptable dado los avances de tecnología e información que ya existen. Reconociendo esto, una nueva clase de negocio ha evolucionado para dar respuesta. Aunque pertenece a la industria de seguros, no tiene ningún producto para vender. Lo que hace es eliminar el factor común del problema entre las opciones que tenía el consumidor. La idea es, a través de Internet, vincular el consumidor (cliente potencial) con el mejor producto ofrecido por los vendedores de productos de seguros. Y hacerlo casi sin esfuerzo. Además, como efecto secundario hacerlo más fácil desde la perspectiva del vendedor, vender sus productos con más eficiencia.

Desafortunadamente, la región del Caribe no tiene ninguna forma de aprovecharse de los beneficios de este servicio. Este proyecto, Carisure, servirá como alternativa en la región.

1.1 Descripción del objeto de estudio

1.1.1 Situación Problemática

Las tendencias sociales actuales demuestran que Internet se ha establecido como el ambiente dominante de interacción social. Para recreación o negocio, se está convirtiendo rápidamente en el medio preferido. Internet ha dado lugar a nuevas formas de negocio.

Ya existen muchas aplicaciones que son similares o igual al servicio que propone este proyecto. Entre las más establecidas están “Progressive Insurance” y “NetQuote.com”. Sin embargo ninguna de ellas brindan servicios en la región del Caribe - todos son de Estados Unidos o de otras regiones desarrolladas como ciertos países en Europa, por ejemplo.

Carisure – Aplicación web para vincular clientes potenciales con vendedores de productos de seguros

En realidad no hay ninguna aplicación existente en la región del Caribe que utilice Internet para vincular clientes potenciales con vendedores de seguros. Además, el proceso seguido por un cliente potencial en la búsqueda de seguros no es muy eficiente y los resultados a menudo no son la mejor opción para él. En caso del vendedor, resulta muy difícil conocer clientes potenciales ideales a quienes se pueda presentar sus productos. Su dificultad aumenta cuando se considera, la confiabilidad de los posibles clientes ante el vendedor en el momento de solicitar sus datos personales, la renuencia del cliente potencial para aceptar una oferta de una asesoría y el costo de transportación para llegar a esta misma asesoría.

Al no utilizarse las posibilidades brindadas por la Internet para asociar clientes potenciales con vendedores de servicios de seguros se desestima una importante brecha en los negocios de la industria de seguros.

Con una aplicación como la que propone este trabajo, Internet puede utilizarse como un medio que brinde la posibilidad para que una empresa, en este caso, Spicelander Financiera de Granada, actúe como intermediario entre clientes potenciales y vendedores de productos de seguros, aumentando las posibilidades de interacción de ambas partes.

1.1.2 Problema a Resolver

La ausencia de una aplicación informática que facilite la gestión de información necesaria para la vinculación de clientes potenciales con vendedores de productos de seguros, y la promoción del intercambio entre ambas partes, en la región del Caribe.

1.1.3 Objeto de estudio y campo de acción

Como objeto de estudio de esta investigación se define la industria de seguros y como campo de acción la gestión de la información asociada a los procesos de vinculación de clientes potenciales y vendedores y la promoción del intercambio entre ellos.

1.1.4 Idea a defender

Con una aplicación informática se puede facilitar la gestión de información necesaria para la vinculación de clientes potenciales con vendedores de productos de seguros, y la promoción del intercambio entre ambas partes, en la región del Caribe.

1.1.5 Objetivo General

Elaborar una aplicación informática que facilite la gestión de información necesaria para la vinculación de clientes potenciales con vendedores de productos de seguros, y la promoción del intercambio entre ambas partes, en la región del Caribe.

Las facilidades en la gestión de la información estarán dadas, por parte del cliente, en el modo en que podrá realizar la compra de seguros. Por la parte del vendedor, en el modo en que podrá aumentar sus oportunidades de conocer nuevos clientes potenciales y aumentar su potencial para vender sus productos.

Para el cumplimiento del objetivo propuesto se definieron un grupo de tareas a cumplir:

1.1.5.1 Tareas

- Estudio del dominio del problema a resolver, incluyendo los procesos relacionados con la compra y venta de productos de seguros en la región del Caribe
- Estudio de la viabilidad del proceso ICONIX para el desarrollo de software
- Análisis de los requerimientos del software
- Diseño e implementación del software.
- Estudio de la factibilidad del sistema a desarrollar

1.1.6 Valor Práctico

Una aplicación como la que propone este trabajo, brindará un servicio al consumidor del Caribe, que ya disfrutaban muchas personas en regiones más desarrolladas. Por otra parte, introducirá una nueva alternativa de la industria de seguros que todavía no ha sido aplicada en el Caribe.

1.1.7 Estructuración del contenido del trabajo de diploma

Este documento está estructurado en cuatro capítulos:

Capítulo 1: Fundamentación Teórica

Se describe el entorno o dominio del problema, dando definiciones y explicando conceptos asociados al objeto de estudio. Se hace referencia también a las diferentes tendencias y metodologías existentes en la actualidad para el desarrollo de aplicaciones informáticas. De ellas se describen sus características esenciales y se fundamentan los motivos por los cuales que fueron seleccionadas para la modelación e implementación del sistema propuesto.

Capítulo 2: Análisis y Modelación de Requerimientos

En este capítulo se modela la solución propuesta, partiendo de un análisis de los requerimientos funcionales y no funcionales recopilados durante la tarea del estudio del dominio del problema a resolver. Se identifican los actores del sistema propuesto y los casos de usos en que participan, los cuales, de acuerdo con el proceso ICONIX, conducen el desarrollo del sistema propuesto.

Capítulo 3: Modelación del Sistema Propuesto

A partir de un análisis de robustez de la modelación de los requerimientos, se describe la implementación del sistema propuesto. Están inducidos en esta descripción, varios diagramas que sirven como guías para la implementación del sistema propuesto.

Capítulo 4: Estudio de factibilidad

En este capítulo se hace un estudio de la factibilidad del sistema propuesto. Se utiliza el método COCOMO II para realizar este estudio.

Capítulo 1 Fundamentación Teórica

1.1 Introducción

En este capítulo se describe el entorno o dominio del problema, dando definiciones y explicando conceptos asociados. Se hace referencia también a las diferentes tendencias y metodologías existentes en la actualidad. De ellas se describen sus características esenciales y se justifica el por qué fueron seleccionadas para la modelación e implementación del sistema propuesto.

1.2 Descripción del dominio del problema

Una definición para “seguros” puede ser expuesta en las palabras de Ignacio Hernando de Larramendi¹:

“...Desde un punto de vista general, (el concepto de seguros) puede entenderse como una actividad económica-financiera que presta el servicio de transformación de los riesgos de diversa naturaleza a que están sometidos los patrimonios, en un gasto periódico (la prima) presupuestable, que puede ser soportado fácilmente por cada unidad patrimonial.”

1.2.1 Principios de seguros

Son los fundamentos doctrinarios en que se basa la actividad aseguradora y son las normas que rigen las relaciones entre Asegurador y Asegurado. Son los siguientes(3):

Principio de Buena Fe: Es la obligación de revelar todo hecho material relacionado con el riesgo a ser asegurado. Un hecho material es un hecho que influenciará la mente de un asegurador prudente al momento de decidir si aceptar un riesgo, y bajo qué condiciones.

Principio de Indemnización: Ante la ocurrencia de un acontecimiento riesgoso previsto en la póliza, el asegurado será puesto por el asegurador en la misma posición monetaria que él se encontraba justo antes que el acontecimiento ocurriera.

Principio de Interés Asegurable: El asegurado debe tener un interés asegurable en el objeto materia del seguro, es decir, debe beneficiarse de su seguridad o ser perjudicado por su pérdida.

¹ Hernando de Larramendi (Madrid, 1921-2001); Abogado y escritor, reconocido empresario cuya labor transformó MAPFRE en una de las mayores empresas españolas.

Principio de Subrogación: Es el derecho del asegurador que ha otorgado una indemnización a adoptar o subrogarse, luego del pago de un siniestro, el beneficio de todo derecho del asegurado sobre el bien asegurado.

Principio de Contribución: Aunque el asegurado pueda comprar más de una póliza para tener cobertura sobre la misma propiedad o interés, no puede recuperar en total más de una indemnización completa.

1.2.1.1 La prima

Es el importe que determina la aseguradora, como contraprestación o pago, por la protección que otorga en los términos del contrato de seguros ó póliza.

La aseguradora, para saber cuánto debe cobrar y qué cantidad de riesgo va a correr, recurre a las estadísticas. Con ellas detecta con qué frecuencia estadística ocurre cada incidencia cubierta y con ella hace los cálculos de lo que tendría que cobrar para cubrir ese gasto. Después, se añade el beneficio y otros gastos, como la gestión, de la aseguradora. Basándose en esa estadística, se aplican diferentes tarifas a las pólizas y en algunos casos, incluso se evita hacerles el contrato de seguro.

1.2.1.2 El riesgo

La base del negocio de los seguros es juntar todos los riesgos de los asegurados y repartirlos entre ellos. Por estadística, el riesgo global de asegurar a una gran cantidad de gente es mucho menor al que asumiría una sola de esas personas.

Los riesgos comercialmente asegurables típicamente comparten siete características comunes:

1. Un gran número de unidades homogéneas de exposición

La vasta mayoría de pólizas de seguros está provista para miembros individuales de clases muy grandes. La existencia de un gran número de unidades homogéneas de exposición deja a los aseguradores beneficiarse de la así llamada "ley de números grandes", lo cual en efecto indica de tal manera como el número de exposición las unidades aumentan, los resultados reales tienen progresivamente probabilidad de quedar cerca de resultados esperados.

2. La Pérdida Definitiva

El caso que da motivo a la pérdida que está sujeto al seguro, al menos en principio, debería llevarse a cabo en un tiempo sabido, en un lugar conocido, y de una causa sabida. El ejemplo clásico es la muerte de un asegurado tiene una póliza de seguro de vida. El incendio, los accidentes del automóvil, y las

Carisure – Aplicación web para vincular clientes potenciales con vendedores de productos de seguros

lesiones del trabajador todo fácilmente pueden cumplir con este criterio. Otros tipos de pérdidas sólo pueden ser definitivos en teoría. La enfermedad ocupacional, por ejemplo, puede implicar exposición prolongada a las condiciones dañosas donde ningún tiempo específico, puede colocar una causa identificable. Idealmente, el tiempo, lugar y causa de una pérdida deberían estar lo suficientemente evidentes que una persona razonable, con suficiente información, objetivamente podría verificar los tres elementos.

3. La Pérdida Accidental

El caso que constituye el factor provocante de un reclamo debería ser fortuito, o al menos fuera del control del beneficiario del seguro. Los casos que contienen elementos especulativos, como los riesgos comerciales comunes, no son generalmente considerados asegurables.

4. La Pérdida de Gran Capital

El tamaño de la pérdida debe ser significativo desde la perspectiva del asegurado. Las primas de seguro necesitan cubrir el costo esperado de pérdidas y el costo de expedir y administrar la póliza. Además, para el abastecimiento del capital necesitado para razonablemente afirmar que el asegurador podrá pagar reclamos.

5. La Prima Costeable

Si la probabilidad de un fenómeno asegurado es tan alta, o el costo del fenómeno tan grande, que la prima resultante es grande relativo a la cantidad de protección ofrecida, no es probable que alguien comprará el seguro, aunque esté disponible.

6. La Pérdida Calculable

Hay dos elementos que deben ser, por lo menos, estimables, si no formalmente calculable: La probabilidad de pérdida, y el costo acompañante.

La probabilidad de pérdida es generalmente un ejercicio empírico mientras que atribuir un costo a la pérdida actual tiene más que ver con la capacidad de una persona razonable para hacer una evaluación razonable definitiva y objetiva del costo de la pérdida, dada una copia de la póliza y una confirmación de pérdida asociada con un reclamo que está cubierto por la póliza.

7. El riesgo limitado de pérdidas catastróficamente grandes

El mayor riesgo para un asegurador es el fenómeno que afecta una gran cantidad de sus asegurados. Si el mismo fenómeno les puede causar pérdidas a numerosos asegurados del mismo asegurador, la habilidad de ese asegurador para expedir pólizas se vuelve constreñida, no por factores rodeando las características individuales de un asegurado dado, sino por los factores rodeando la suma de todos los asegurados expuestos. Típicamente, los aseguradores prefieren limitar su exposición a una pérdida de un fenómeno solo para alguna porción pequeña de su base capital, alrededor de 5 por ciento.

1.2.1.3 Tipos de seguros

El seguro generalmente se subdivide en una de las siguientes categorías: Propiedad, adeudo, salud, incapacidad, y seguro de vida.

Seguro de propiedad

Asegura contra la pérdida de propiedad (hogar, negocio, automóvil, u otras posesiones).

Seguro de responsabilidad

Alguien que conduce un coche, es dueño de una casa u otros bienes raíces, dirige un negocio, o emplea otros corre el riesgo de responsabilidad para un accidente. Y ese accidente puede dar como resultado daño de la propiedad o herida o muerte para otra persona. Un chofer o un dueño de una propiedad o un negocio puede incurrir la responsabilidad de pagar por reparación de la propiedad o para el tratamiento médico o hasta el dolor y el sufrimiento de otro. En muchos países los empleadores y los conductores son requeridos por ley para llevar seguro de responsabilidad para ayudar a pagar estos gastos. Aun donde el seguro no es un requisito legal, un conductor, un dueño de propiedad, o un empleador puede estar sujeto legalmente o moralmente responsable para ayudar a las víctimas de un accidente o sus familias.

Seguro de salud

Muchos países tienen alguna forma de seguro de garante estatal que provee tales beneficios como pensiones de jubilación para ancianos y la atención médica. Aun donde éste es el caso, sin embargo, tal seguro puede pagar sólo una porción de gastos médicos. Algunos individuos, por consiguiente, obtienen seguro privado adicional para ayudarles a pagar el resto. En muchos lugares los trabajadores pueden recibir seguro de salud como una condición de su empleo.

Seguro de incapacidad y seguro de vida

El seguro de incapacidad provee algún ingreso si una persona es herida y no puede trabajar. Seguro de vida les provee la asistencia financiera a los dependientes de una persona en caso de su muerte.

Carisure – Aplicación web para vincular clientes potenciales con vendedores de productos de seguros

1.2.1.4 Diferencia entre Agente de Seguros y Corredor de Seguros

Los Agentes son intermediarios que sólo trabajan para una compañía y sólo pueden recomendar productos de ésta. Los Corredores son profesionales cualificados que tienen una titulación especial y que trabajan con todas las aseguradoras por lo que pueden comparar distintos seguros y ofrecerle al consumidor el producto de la compañía más adecuada para sus necesidades(4).

1.2.2 Sistemas existentes

Ya existen muchos ejemplos de implementaciones que son similares o igual al servicio que propone Carisure. Entre las más establecidas están “Progressive Insurance” y “NetQuote.com”. Sin embargo ninguna de ellas brinda servicios en el Caribe.

1.2.2.1 Análisis comparativo con sistemas existentes

La diferencia fundamental entre Carisure y “Progressive Insurance” es que las ofertas de “Progressive Insurance” son propias y Carisure no venderá pólizas. “Progressive Insurance” es un Corredor de seguros que vende sus productos en Internet. A la hora de hacer la póliza actual, el cliente interactuará con un agente que trabaja con “Progressive Insurance”. La compra actual no está realizada en línea.

NetQuote.com está en las ramas altas de proveer información para un cliente buscando seguros. Además su modelo de negocio es igual a lo que aspira Carisure. NetQuote.com sirve como un medio de interacción entre clientes potenciales y los vendedores de pólizas de seguros. NetQuote.com categoriza los clientes y vende su información a vendedores con precios diferentes dependiendo de la categoría. Por ejemplo la información (lead) de un candidato para “Juvenile Auto” (seguro de auto para candidatos jóvenes) será vendida por US\$1.00, mientras que un candidato para “Preferred Plus Auto” (seguro de autos para candidatos preferidos) venderá para US\$11.00.

No obstante es preciso destacar que ni NetQuote.com ni otra compañía tiene ofertas en la región del Caribe.

1.2.3 Tendencias, Metodologías y Tecnologías a considerar

1.2.3.1 Metodologías de ingeniería de software

Según la definición dada por IEEE IEE93, la Ingeniería del Software es la aplicación de una metodología sistemática, disciplinada, cuantificable para el desarrollo, la operación, y el mantenimiento de software; Es decir, la aplicación de ingeniería a software, y el estudio de maneras para cómo se puede aplicar estas metodologías.

Carisure – Aplicación web para vincular clientes potenciales con vendedores de productos de seguros

Que es una metodología

Una metodología puede ser definida como un conjunto de métodos empleados para el desarrollo de sistemas automatizados.

Una metodología completa es algo más que una notación, un proceso, y herramientas. Además de una "notación, de un proceso, y de herramientas," estas "metodologías completas" proporcionan:

- Guías para estimar costos,
- Manejo del proyecto en las tareas y entregas,
- Medidas y métricas,
- Formas definidas y dirección en las entregas de la construcción,
- Políticas y procedimientos para garantizar la calidad del software,
- Descripciones de los roles y programas de entrenamiento detallados,
- Ejemplos totalmente trabajados,
- Ejercicios de entrenamiento,
- Técnicas para adaptar el método, y
- Técnicas definidas

Para este trabajo se consideraron dos métodos de desarrollo: el Proceso Unificado de Rational (RUP) y el proceso ICONIX. Debido a que hasta la fecha ninguna de las metodologías que se van a comparar cumplen completamente con las características mencionadas anteriormente, ya que falta mucho trabajo por hacer para que sean unas metodologías completas, la comparación se hace con ese conocimiento y entonces "metodología" se considera también como un método o proceso.

Apuntaciones sobre RUP

- Es un proceso formal
 - Provee un método disciplinado para asignar tareas y responsabilidades dentro de una organización de desarrollo
 - Su meta es asegurar la producción de software de alta calidad que satisfaga los requisitos de los usuarios finales (respetando la fecha límite y el presupuesto).
- Fue desarrollado y es mantenido por Rational Software (ahora una subsidiaria de IBM), y está integrado con toda la suite Rational de herramientas
 - Puede ser adaptado y extendido para satisfacer las necesidades de la organización que lo adopte

- Es dirigido por casos de usos y céntrico arquitectural, y utiliza UML como lenguaje de notación.
- Captura muchas de las “prácticas mejores” del desarrollo de software moderno.
 - Ej. Gestión de Proyectos, Gestión de Riesgos, Aseguramiento de Calidad, prototipos, etc.
- RUP repite una serie de ciclos compuestos por iteraciones a lo largo de la vida del sistema. Es decir, es incremental e iterativo.
 - Cada ciclo consiste de 4 fases: incepción, elaboración, construcción y transición.
 - Cada fase se descompone en iteraciones.
 - Cada fase termina en un hito, que consiste en un conjunto de artefactos.

En la Figura 1-1, se puede ver un diagrama que ilustra las características del RUP anteriormente señaladas.

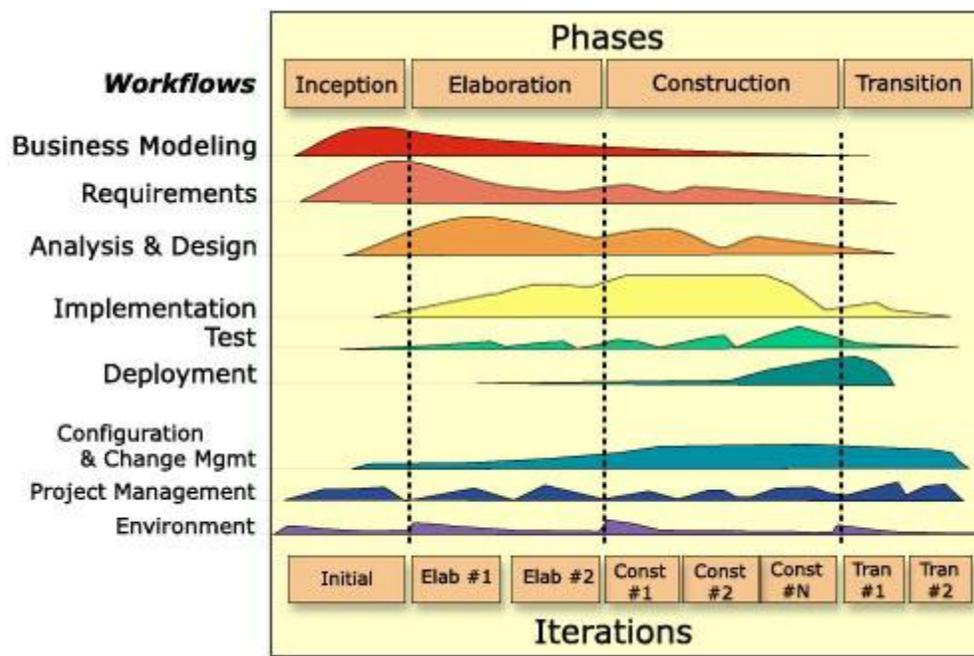


Figura 1-1 Diagrama de los características del RUP

Apuntaciones sobre el proceso ICONIX

- Es un modelo simple que puede ser considerado como una reducción funcional de RUP
 - El proceso ICONIX está entre RUP y el método XP
 - Es dirigido por casos de usos, como RUP, pero no posee todos los costos operativos de RUP.

Carisure – Aplicación web para vincular clientes potenciales con vendedores de productos de seguros

- Es relativamente pequeño como XP, pero no descarta las notaciones de análisis y diseño como lo hace XP
- Es minimalista y se focaliza en el área que queda entre los casos de uso y el código
 - Es decir que se focaliza en el “cómo”, en la transición desde requerimientos hacia implementación.
- Los tres elementos fundamentales de ICONIX son:
 1. Es iterativo e incremental
 2. Rastreabilidad de requerimientos
 3. Hace un uso estilizado de UML

En un estudio realizado por la empresa Pragma Consultores de España, se comparaban entre otros, el RUP e ICONIX. Señalaban en su conclusión que:

“Un modelo de proceso se selecciona usando escenarios de proyectos basados en el criterio de selección... Puntos críticos como el compromiso gerencial y resistencia cultural... El criterio de selección debe considerar también las características de la organización en la que se va a implementar el modelo.”(5)

Dado ciertos factores como el hecho de que el proyecto estaría elaborado por un solo desarrollador, el tiempo disponible y el tamaño del proyecto propuesto fue elegido el proceso ICONIX.

El Proceso ICONIX

Una distinción de ICONIX del RUP es su uso del análisis de robustez, un método para disminuir la brecha entre análisis y diseño. El análisis de robustez reduce la ambigüedad en descripciones de casos de uso, asegurando que están escritos en el contexto del modelo de dominio. Este proceso hace los casos de uso más fácil para diseñar, probar y estimar. Además, los creadores del UML (Grady Booch, James Rumbaugh, y Ivar Jacobson) señalaban que “uno puede hacer ochenta por ciento de la modelación con veinte por ciento del UML”.

El proceso ICONIX, como se muestra en la Figura 1-2, es un método minimalista y perfilado, que se focaliza en el área que queda dentro de casos de uso y código. El proceso es dividido en flujos de trabajo dinámico y estático, que son altamente iterativos.

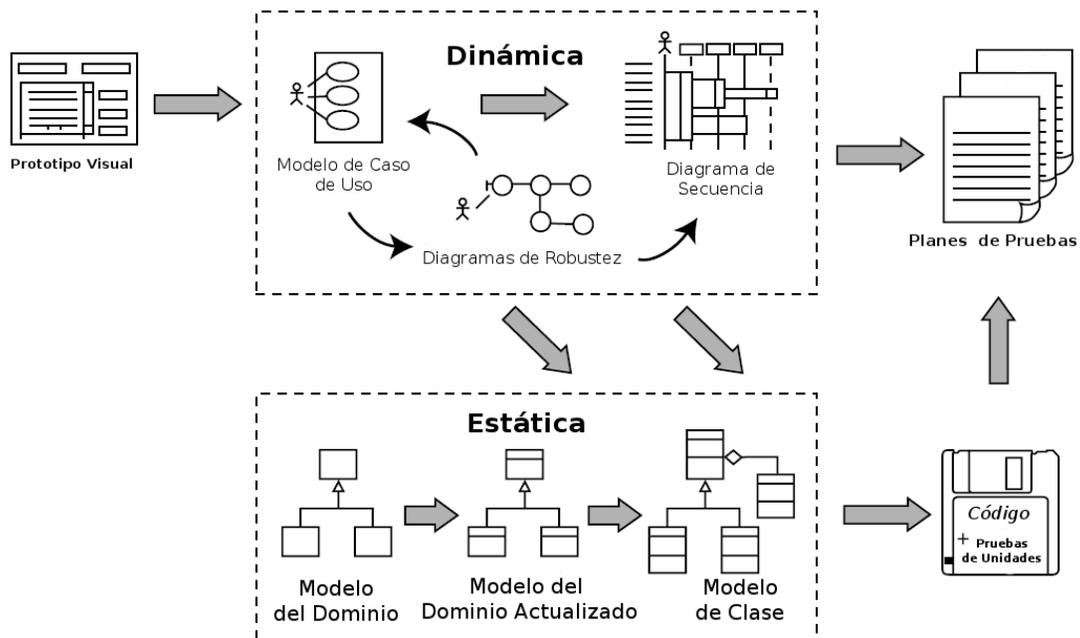


Figura 1-2 El proceso ICONIX

El proceso tiene seis fases(6):

1. Requerimientos
 - a. Requerimientos Funcionales
 - b. Modelación del Dominio
 - c. Requerimientos de comportamiento (hecho por casos de uso preliminares, representaciones gráficas preliminares, ...)
 - d. Revisión de Requerimientos
1. Y después, en cada iteración (i.e., para lotes pequeños de casos de uso), se hace las siguientes fases:
2. Análisis/Diseño Preliminar
 - a. Análisis de Robustez
 - b. Actualizar el modelo del dominio
 - c. Nombrar todos las funciones lógicas de software (controladores)
 - d. Editar los casos de uso preliminares
3. Revisión Preliminar del Diseño
4. Diseño Detallado

- a. Dibujar gráficas de secuencia (una para cada diagrama de robustez)
 - b. Actualizar el modelo del dominio (por esta etapa, se vea una transformación del modelo de dominio en modelo estática o modelo de clases que contiene clases del dominio (entidades))
5. Revisión Crítica del Diseño
 6. Implementación
 - a. Codificar/Pruebas de Unidades
 - b. Integración y Pruebas de escenarios
 - c. Realizar una revisión del código y actualizar el modelo para preparar para la próxima iteración.

1.2.3.2 Lenguajes de desarrollo para aplicaciones web

Existen un vasto número de lenguajes de programación. Cada uno con sus propias potencialidades y debilidades. En la rama de programación Web, hay algunos que sobresalen – como ASP.Net, PHP, Javascript y Java entre otros.

Lenguajes como Javascript y XHTML operan en el lado del cliente. Es decir que todo el procesamiento está hecho por el navegador. El código de PHP, ASP.Net y Java, está ejecutado en el lado del servidor.

Para elegir cuáles serían utilizadas para el desarrollo del proyecto propuesto, se hizo una evaluación cualitativa de sus cualidades y también de las características del sistema propuesto. Se consideró el ambiente de ejecución del sistema propuesto, la capacidad para utilizar el lenguaje y las características de cada lenguaje considerado. El sistema operativo de desarrollo fue principalmente Linux. Aunque en Linux sea posible el desarrollo en ASP.Net, la única vía de hacerlo en Linux es por el uso de Mono. Este autor ha intentado de utilizar software libre para el desarrollo de este proyecto cuando sea posible, por lo que el uso de ASP.Net como opción fue desestimado.

Otra opción en el lado del servidor fue Java. Java es poderoso, rápido y multiplataforma. Sin embargo, el autor seleccionó PHP 5 para el desarrollo del sistema. Esta decisión de utilizar PHP 5 en vez de Java, fue principalmente basada en el nivel de conocimiento ya poseído por el autor. También PHP 5 tiene otras cualidades que lo hace útil para el desarrollo de aplicaciones web(7):

- Es rápido
- Es software libre
- Es multiplataforma

Carisure – Aplicación web para vincular clientes potenciales con vendedores de productos de seguros

- Es seguro
- Soporte una amplia selección de bases de datos
- Permite el desarrollo orientado a objetos

Javascript y XHTML son lenguajes ubicuos en la programación web en el lado del cliente. Hay otras opciones (ej., Adobe Flash y Microsoft Silverlight) para el desarrollo en el lado del cliente, pero Javascript y XHTML consumen menos recursos en el navegador y están soportados hasta un cierto nivel estándar en todos los navegadores principalmente usados. Entonces, Javascript y XHTML fueron elegidos para el desarrollo en el lado del cliente.

1.2.3.3 Patrones

Los patrones ayudan a robustecer en la experiencia colectiva de ingenieros expertos del software. Capturan experiencia existente, bien probada en el desarrollo del software y ayudan a promover buenas prácticas de diseño. Cada patrón se ocupa de un problema específico y recurrente en el diseño o la implementación de un sistema informático. Se puede usar los patrones para construir arquitecturas del software con propiedades específicas.

Podemos derivar varias propiedades de patrones para la arquitectura del software:

- Los patrones identifican y especifican abstracciones que están por encima del nivel de instancias y clases solas, o de componentes.
- Los patrones proveen una comprensión y vocabulario común para los principios del diseño.
- Los patrones es una manera de documentar arquitecturas del software.
- Los patrones soportan la construcción de software con propiedades definidas.
- Los patrones le ayudan a construir arquitecturas complicadas y heterogéneas del software.
- Los patrones le ayudan a manejar complejidad del software.

Por lo tanto una definición formal de un patrón puede ser dada como:

“Un patrón para la arquitectura del software describe un problema recurrente particular del diseño que surge en los contextos específicos del diseño, y presenta un esquema bien genérico probado para su solución. El esquema de solución es especificado describiendo sus componentes constituyentes, sus responsabilidades y sus relaciones, y las formas en las cuales colaboran.”(8)

MVC

Para la elaboración de este proyecto se decidió utilizar el patrón de modelo-vista-controlador (MVC). El patrón MVC fue desarrollado en los años 70 por Trygve Reenskaug para el lenguaje Smalltalk.

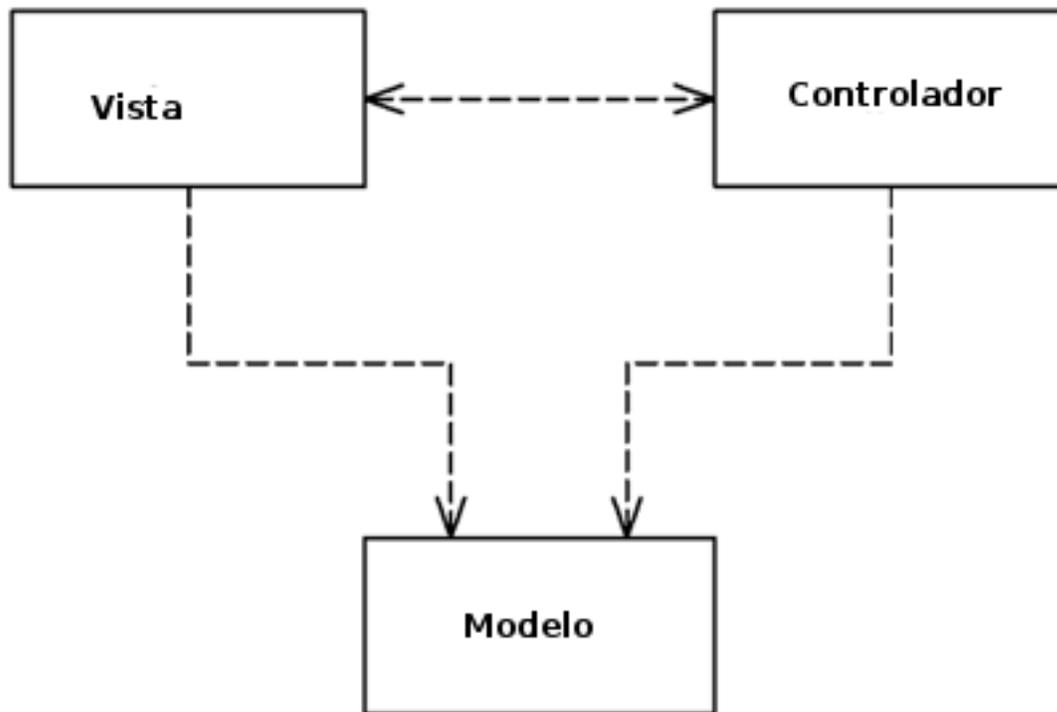


Figura 1-3 Patrón MVC

El MVC se considera tres papeles:

1. El modelo es un objeto que representa información del dominio. Es un objeto no-visual que contiene todos los datos y comportamiento que no sea usado para la interfaz. En su forma más pura de OO el modelo es un objeto por dentro del modelo del dominio.
2. La vista representa la visualización del modelo en la IU.
3. La vista solamente tiene la responsabilidad de mostrar información; los cambios de información son manejados por el controlador. El controlador acepta entradas del usuario, manipula el modelo, y causa la actualización apropiada en la vista. En esa manera la IU es una combinación de la vista y controlador. Ver la Figura 1-3.

1.2.3.4 Codeigniter

Codeigniter es un grupo de programas y rutinas que se utilizan como base para la programación de un nuevo sistema en el lenguaje PHP. Su meta es facilitar el desarrollo de proyectos más rápido comparado al desarrollo de software desde principio. Para lograr esta meta provee un conjunto de bibliotecas de código para tareas comunes, y también, una interfaz simple y estructura lógica para acceder estas bibliotecas de código(9).

Características de Codeigniter

- Es software libre y tiene cobertura bajo una licencia estilizada de Apache/BSD que permita el uso libre del código.
- Suporta PHP 4 y PHP 5
- Es “ligero”. El núcleo requiere un conjunto pequeño de bibliotecas. Bibliotecas adicionales pueden ser cargados si son necesarias
- Es rápido (dado a la característica de ser “ligero”)
- Utiliza el patrón MVC
- Genera URL limpias. Esta característica hace que la aplicación web sea fácilmente catalogado por motores de búsqueda
- Es extensible (por herencia de las clases bases de las bibliotecas)
- Es ampliamente documentado



Figura 1-4 Biblioteca de Codeigniter

1.2.3.5 Sistema de Gestión de Bases de Datos

Una base de datos es una colección ordenada de datos, usualmente guardados en uno o más ficheros asociados. Los datos son estructurados en forma de “tablas”, donde referencias cruzadas entre tablas

son posibles. MySQL, Oracle, Microsoft SQL e IBM DB2 son ejemplos de sistemas de gestión de base de datos (SGBD). Tales sistemas incluyen programas para la gestión de bases de datos relacionales(10).

MySQL

Para el desarrollo del sistema propuesto, el único SGBD considerado de manera seria fue MySQL. La familiaridad con este SGBD fue una de los factores para su selección pero no fue el factor principal para la decisión. MySQL tiene muchas características deseables para la implementación del proyecto propuesto. Las características expuestas en la siguiente lista hicieron que MySQL fuera una buena elección(11):

- Arquitectura cliente/servidor
- Compatibilidad con SQL estándar (SQL:2003 aunque todavía falta algunas cosas para implementar)
- Posibilidades de Vistas
- Soporte para Unicode
- Tiene interfaz nativa con PHP
- Es Multiplataforma
- Rapidez en el procesamiento

1.2.3.6 Herramientas utilizadas en la elaboración del proyecto

A continuación se muestra una tabla de otras herramientas y tecnologías utilizadas en la elaboración del proyecto propuesto:

Nombre	Propósito
Zend Development Environment	Programación (Versión de prueba)
SQLyog	Cliente para la manipulación del base de Datos(software libre)
Enterprise Architect	Ingeniera de Software (Versión de prueba)
Mozilla Firefox(Windows y Linux), Opera, Microsoft Internet Explorer	Navegadores utilizados para pruebas
Apache	Servidor Web
Zend Debugger	Depurador para PHP
Firebug	Depurador para Javascript y CSS
Microsoft Word	Elaboración de Documentos
CodeIgniter	Framework MVC para aplicaciones de PHP
jQuery	Biblioteca de Código para Javascript y AJAX
Subversion	Control de Versiones (sistema para la gestión de versiones)

Tabla 1-1 Tabla de Herramientas y Tecnologías Utilizadas

1.3 Conclusión

En este capítulo se describió el dominio del problema con respecto a la industria de seguros. Se hizo ejemplos de sistemas semejantes ya existentes, apuntando a la ausencia de una implementación de tales sistemas en la región del Caribe. También, se señaló las metodologías y tecnologías y herramientas utilizados para el desarrollo del sistema propuesto, profundizando en las justificaciones para su selección.

Capítulo 2 Análisis y Modelación de Requerimientos

2.1 Introducción

En este capítulo se modela la solución propuesta, partiendo de un análisis de los requerimientos funcionales y no-funcionales recopilados durante la tarea del estudio del dominio del problema a resolver. Se identifican los actores del sistema propuesto y los casos de usos en que participan, los cuales, de acuerdo con el proceso ICONIX, conducen el desarrollo del sistema propuesto.

Un requerimiento describe una propiedad a que debe tener el sistema o un servicio que debe tener el sistema. El IEEE 610.12 define un requerimiento como:

1. Una condición o capacidad requerido por un usuario para resolver un problema o lograr un objetivo;
2. Una condición o capacidad que debe ser logrado o poseído por un sistema o un componente de un sistema para satisfacer un contrato, un estándar, una especificación, u otro documento formalmente impuesto;
3. Una representación documentada de una condición o capacidad como en 1 o 2.

Típicamente, se pueden clasificar los requerimientos como requerimientos funcionales o no-funcionales. Los requerimientos funcionales definen las capacidades y servicios de un sistema, mientras que los requerimientos no-funcionales describen los niveles deseados de calidad (¿Es seguro?, ¿Fácil de utilizar?,...).

2.2 Requerimientos Funcionales

Para la definición de requerimientos funcionales del sistema propuesta los mismos se dividieron en dos áreas:

- Requerimientos desde la perspectiva del cliente y
- Requerimientos desde la perspectiva del agente

2.2.1 Perspectiva del Agente

Requerimiento	Descripción
REQ01	El agente debe ser capaz de crear una cuenta
REQ02	El agente debe ser capaz de borrar su cuenta

REQ03	El agente debe ser capaz de cancelar su suscripción a una o más clasificaciones de leads
REQ04	El agente debe ser capaz de autenticarse si tiene cuenta
REQ05	El agente debe ser capaz de editar los detalles de su cuenta
REQ06	El agente debe ser capaz de adquirir más créditos
REQ07	El agente debe ser capaz de suscribirse a diferentes clasificaciones de leads
REQ08	El sistema debe mostrar un reporte que incluye: <ul style="list-style-type: none"> ▪ La cantidad de leads activos ▪ Cantidad total de leads recibidos ▪ Cantidad de leads rechazados esperando atención ▪ Cantidad actual de créditos en la cuenta
REQ09	El sistema debe mostrar los precios de cada suscripción al agente a la hora de suscribirse
REQ10	El sistema debe proveer un panel de control donde el agente puede gestionar su cuenta
REQ11	El agente debe ser capaz de rechazar un lead si por alguna razón desea hacerlo

Tabla 2-1 Requerimientos funcionales - perspectiva del agente

2.2.2 Perspectiva del cliente

Requerimiento	Descripción
REQ12	El cliente debe ser capaz de crear una cuenta
REQ13	El cliente debe ser capaz de borrar su cuenta del sistema si desea hacerlo
REQ14	El cliente debe ser capaz de autenticarse si ya tiene una cuenta en el sistema
REQ15	El cliente debe ser capaz de editar sus detalles en su cuenta
REQ16	El cliente debe ser capaz de buscar agentes contrapartes que han suscrito a su clasificación
REQ17	El sistema debe tener un directorio de agentes
REQ18	El cliente debe ser capaz de refinar la lista de agentes en el directorio de agentes
REQ19	El cliente debe ser capaz de ver el perfil de cualquier agente listado

Tabla 2-2 Requerimientos funcionales - perspectiva del cliente

2.3 Requerimientos No-funcionales

A continuación se definen los requerimientos no-funcionales:

Carisure – Aplicación web para vincular clientes potenciales con vendedores de productos de seguros

2.3.1 Requisitos de Interfaz Externa

El sistema propuesto será utilizado por usuarios que no necesariamente son expertos en el uso de una computadora y que tienen entendimiento básico de navegar por Internet. Por lo tanto el sistema debe poseer una interfaz amigable e intuitiva.

2.3.2 Requisitos de Rendimiento

El sistema propuesto debe responder rápidamente a las peticiones del usuario. El usuario no debe percibir una demora en ningún momento que el sistema está procesando un gran cantidad de información. La arquitectura cliente/servidor con separación de propósitos (por ejemplo, un servidor dedicado a los servicios de la base de datos que no provee otro servicio como web, ftp, etc.) apoyará la eficiencia de operación.

2.3.3 Requerimientos de Soporte

La instalación del software tiene que ser hecha por un administrador con experiencia intermedia en la instalación de bases de datos de MySQL, servidor web Apache y PHP. Para el mantenimiento de la aplicación hace falta un administrador con experiencia básica para el cambio de contenido.

El diseño del sistema soportará la adición de nuevos módulos para ofertar futuros servicios.

2.3.4 Requerimientos de Usabilidad

Es importante que el sistema sea funcional para los navegadores más utilizados en el mundo actual.

2.3.5 Requerimientos de Portabilidad

Por el uso de tecnologías de código abierto que también sean multiplataforma, el sistema será implementable en varios entornos de ejecución. El sistema operativo escogido para el desarrollo del sistema fue Linux.

2.3.6 Requerimientos de Seguridad

El sistema propuesto tratará con mucha información personal y privada. Por esto, el sistema debe garantizar la seguridad de estos datos durante la recogida de los mismos y durante todo el tiempo que existan en la aplicación. Para la seguridad durante el intercambio de información, el sistema necesitará una conexión cifrada (SSL) para la transferencia de datos entre el usuario y el sistema.

2.3.7 Requerimiento del Software

Para la implementación del software se debe disponer de por lo menos:

- MySQL 5.1

Carisure – Aplicación web para vincular clientes potenciales con vendedores de productos de seguros

- PHP 4.2
- Apache 2.2

2.3.8 Requerimientos Básicos del Hardware

- Procesador Pentium
- 128mb RAM
- 40 GB de HDD Libre
- UPS o fuente de corriente ininterrumpida

2.3.9 Requisitos Legales

La aplicación propuesta por este trabajo y el código fuente serán propiedades de Spicelander Financial de Granada. Será comercializada.

2.4 Modelo del Dominio

De acuerdo con el proceso ICONIX, el modelo del dominio es un artefacto vivo y colaborativo. Está refinado y actualizado durante todo el proyecto, para que refleje el entendimiento actual del espacio del problema. Esencialmente, el modelo del dominio actúa como una mejora de un glosario de los términos utilizados en el proyecto. En la práctica es un diagrama de clase simplificado con líneas trazadas entre clases (objetos del dominio) para ilustrar como se relacionan entre si.

A continuación se muestra el modelo del dominio de sistema propuesto:

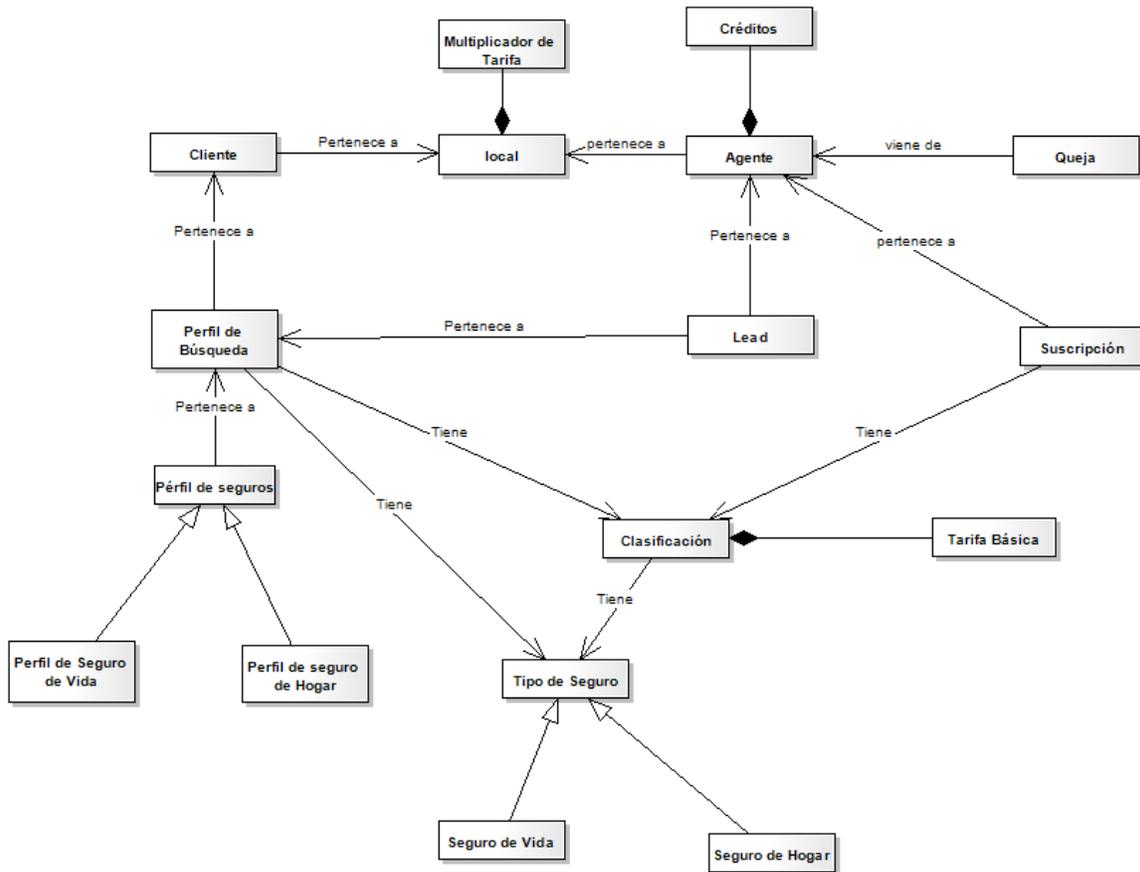


Figura 2-1 Diagrama del modelo del dominio del sistema propuesto

2.5 Modelo de Casos de Uso

Como fue mencionado anteriormente, el modelo del dominio ilustra cuáles son los objetos del dominio que interactúan en el sistema. Pero la respuesta al “cómo” interactúan es responsabilidad del modelo de caso de uso.

Un caso de uso captura un contrato entre los stakeholders de un sistema y su comportamiento. Un caso de uso define una manera de utilizar el sistema modelado. Todo el comportamiento dentro del sistema, o sea, todas las maneras en que se puede utilizar el sistema están capturadas por el caso de uso. Los usuarios del sistema (personas, máquinas y sistemas externos que interactúan con el sistema) están modelados como “Actores.” Juntos, los casos de uso y los actores forman un modelo de caso de uso. Este modelo describe todas las maneras en que el sistema puede ser utilizado por su entorno de operación(12).

2.5.1 Actores del Sistema

A continuación se identifican los actores encontrados en el proceso del análisis del sistema.

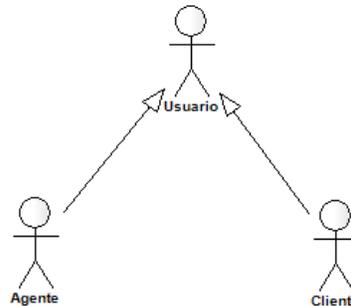


Figura 2-2 Diagrama de actores del sistema

Actor	Descripción Breve
Usuario	Este actor es un usuario general que interactúa con el sistema.
Agente	Este actor es un usuario que interactúa con el sistema exclusivamente desde la perspectiva del agente. Su propósito principal gestionar su suscripción al servicio de leads.
Cliente	Este actor es un usuario que interactúa con el sistema solamente desde la perspectiva del cliente. Su propósito es buscar uno o más agentes.

Tabla 2-3 Descripción de los actores del sistema

2.5.2 Casos de uso del sistema

Como se muestra en la figura más adelante, el sistema propuesto fue dividido en tres partes generales: agentes, clientes y general.

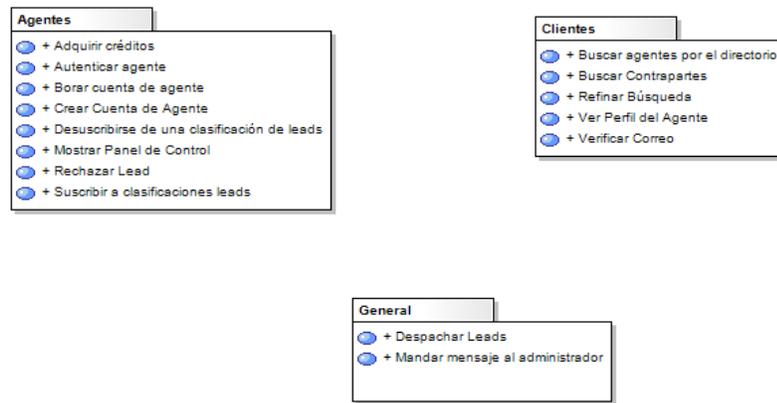


Figura 2-3 Diagrama de Paquetes de Casos de Uso

2.5.2.1 Casos de uso del paquete "Agentes"

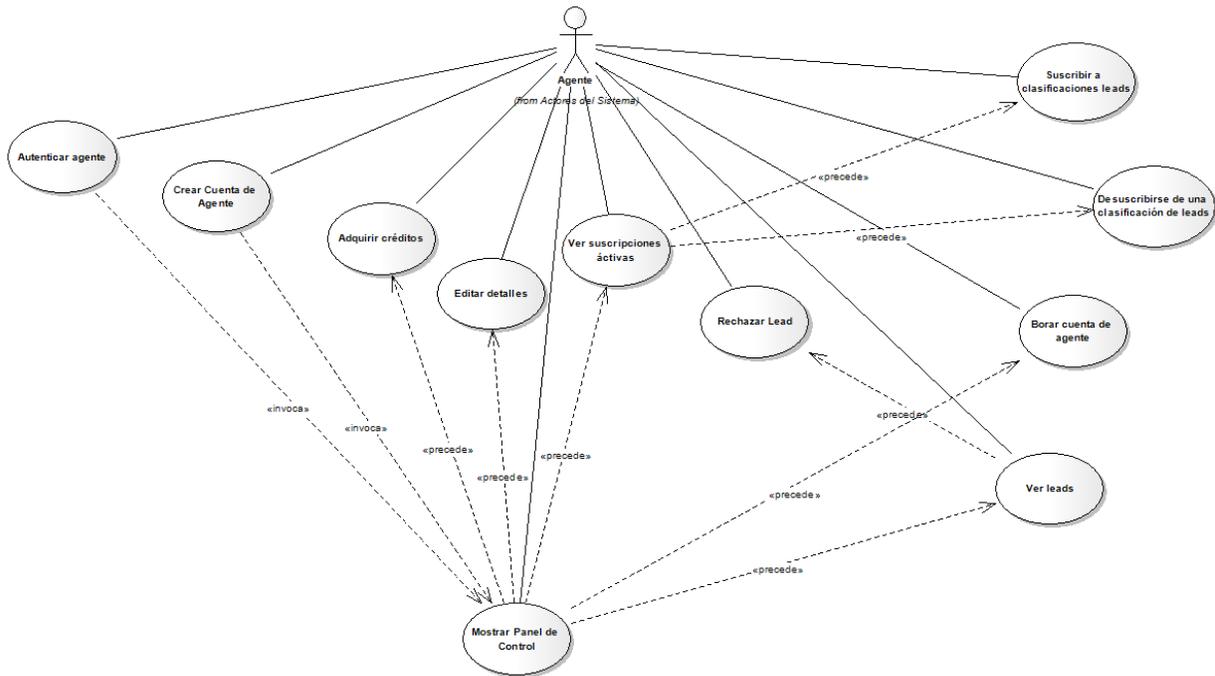


Figura 2-4 Diagrama de Casos de Uso del Paquete Agentes

Como se muestra en la Figura 2-4 los siguientes casos de uso fueron identificados en este paquete:

1. Autenticar agente
2. Suscribirse a clasificaciones leads
3. Desuscribirse de una clasificación de leads
4. Crear cuenta de agente

5. Mostrar panel de control
6. Rechazar lead
7. Adquirir créditos
8. Borrar cuenta de agente
9. Ver suscripciones activas
10. Ver leads activas
11. Editar detalles

Para cada caso de uso, el único actor involucrado es el Agente.

Nombre de caso de uso:	Autenticar agente	
Curso Básico:	Autenticar agente	El sistema muestra la vista "Página Inicial". El cliente hace clic en el enlace para autenticarse. El sistema muestra la vista "Autenticación". El agente entra su dirección de correo y su contraseña y hace clic en login. El sistema consulta la base de datos para verificar los datos entrados. Si los datos son válidos, el sistema cambia la sesión actual a autenticada y muestra la vista "Panel de Control".
Curso Alternativo:	Cancelar	El agente hace clic en el enlace "Cancelar". El sistema muestra la vista "Página Principal".
Curso Alternativo:	Información incorrecta	El cliente entró datos erróneos para la autenticación. El sistema muestra un mensaje que dice que los datos entrados eran erróneos y muestra la vista "Autenticación".

Tabla 2-4 Descripción de caso de uso "Autenticar Agente"

Nombre de caso de uso:	Adquirir Créditos	
Curso Básico:	Adquirir Créditos	En la vista "Panel de Control" el agente hace clic en el enlace "Adquirir Créditos". El sistema consulta la base de datos para la cantidad actual de créditos que tiene este agente y muestra la vista "Créditos". El agente hace clic en el enlace "Adquirir Más". El sistema muestra la vista "Entrar Cantidad de Créditos". El agente entra la cantidad de créditos que desea y hace clic en el botón "Adquirir". El sistema actualiza la cuenta del agente con los nuevos créditos y muestra la vista "Créditos".

Tabla 2-5 Descripción de caso de uso "Adquirir Créditos"

Nombre de caso de uso:	Borrar cuenta de agente
------------------------	-------------------------

Curso Básico:	Borrar cuenta de agente	En la vista "Panel de Control", el agente hace clic en el enlace "borrar cuenta". El sistema muestra un diálogo de confirmación. El agente decide seguir con su decisión. El sistema cambia el estado actual de la sesión a no-autenticado y borra los registros pertinentes de la base de datos. Pues, el sistema muestra la vista "Página Inicial de Agentes".
Curso Alternativo:	Agente hizo clic en "No" en el diálogo de confirmación	El sistema muestra la vista "Panel de Control".

Tabla 2-6 Descripción de caso de uso "Borrar Cuenta de Agente"

Nombre de caso de uso:	Crear Cuenta de Agente
------------------------	------------------------

Curso Básico:	Crear Cuenta de Agente	<p>En la vista "Pagina Inicial de Agentes", el agente hace clic en el enlace para mostrar la vista "Registro". El sistema muestra la vista "Registro". El Agente llena todos los campos necesarios para crear una nueva cuenta y hace clic en "Crear Cuenta". El sistema valida los datos y también, determina si la dirección de correo electrónico ya existe en el sistema.</p> <p>Si los datos son válidos y la dirección de correo electrónico no existe en el sistema, el sistema guarda la información del agente y manda un correo que contiene un código de confirmación al agente. Entonces el sistema muestra la vista "Correo de Confirmación Mandado". El agente hace clic en el enlace del código de confirmación. Al recibir el código de confirmación del agente, el sistema valida el código. Si el código es válido, el sistema muestra la vista "Entra Contraseña". El agente entra una contraseña y convierte la información guardada del agente en una cuenta real. Pues el sistema cambia el estado de la sesión a autenticada y muestra la vista "Panel de Control".</p>
Curso Alternativo:	Datos Inválidos o Dirección de Correo ya existe	El sistema muestra la vista "Registro" con indicadores en los campos donde había errores.
Curso Alternativo:	Código de confirmación inválido	El sistema muestra la vista "Código Inválido".

Tabla 2-7 Descripción de caso de uso "Crear Cuenta de Agente"

Nombre de caso de uso:	Desuscribirse de una clasificación de leads
------------------------	---

Curso Básico:	Desuscribirse de una clasificación de leads	En la vista "Suscripciones Activas", el agente selecciona una clasificación de leads suscrita y hace clic en el enlace "Desuscribir". El sistema actualiza la base de datos con respecto a los cambios hechos. El sistema actualiza y muestra la vista "Suscripciones Activas".
---------------	---	---

Tabla 2-8 Descripción de caso de uso "Desuscribirse de una Clasificación de Leads"

Nombre de caso de uso:	Mostrar Panel de Control	
Curso Básico:	Mostrar Panel de Control	<p>Primero el sistema consulta la base de datos para los siguientes datos:</p> <ul style="list-style-type: none"> ▪ Cantidad total de suscripciones activas ▪ La cantidad total de leads activas ▪ Cantidad total de leads nuevas ▪ Cantidad de leads rechazados esperando atención ▪ Cantidad actual de créditos en la cuenta <p>Después, el sistema muestra la vista "Panel de Control"</p>

Tabla 2-9 Descripción de caso de uso "Mostrar Panel de Gestión"

Nombre de caso de uso:	Rechazar Lead
------------------------	---------------

Curso Básico:	Rechazar Lead	En la vista de "leads activas", el agente selecciona un lead y hace clic en el enlace "Ver Detalles del Lead". El sistema consulta la base de datos para el lead y muestra la vista "Detalles del Lead". El cliente hace clic en el enlace "Rechazar Lead". El sistema muestra la vista "Razón de Rechazo". El cliente escribe su razón y hace clic en "Rechazar". El sistema crea una queja con los detalles del rechazo. Pues el sistema actualiza y muestra la vista "Detalles del Lead".
---------------	---------------	--

Tabla 2-10 Descripción de caso de uso "Rechazar Lead"

Nombre de caso de uso:	Suscribir a clasificaciones leads	
Curso Básico:	Suscribir a clasificaciones leads	En la vista "Panel de Control", el agente hace clic en el enlace "Suscribir a otros leads". El sistema muestra la vista "Suscripciones Disponibles". El agente selecciona uno o más clasificaciones de leads disponibles y hace clic en el botón "Suscribir". El sistema actualiza la base de datos con respecto a los cambios hechos. El sistema muestra la vista "Suscripciones Activas"

Tabla 2-11 Descripción de caso de uso "Suscribirse a una Clasificación Leads"

Nombre de caso de uso:	Ver leads	
Curso Básico:	Ver leads	En la vista "Panel de control" el cliente hace clic en el enlace "Ver leads". El sistema consulta la base de datos para los leads que ha recibido el agente. Pues el sistema muestra la vista "Leads Activas" con la lista de leads.
Curso Alternativo:	No encontró leads	El sistema muestra la vista "Leads Activas" con un mensaje que el agente no tiene leads activas.

Tabla 2-12 Descripción de caso de uso "Ver Leads"

Nombre de caso de uso:	Ver suscripciones activas
------------------------	---------------------------

Curso Básico:	Ver suscripciones activas	En la vista "Panel de Control", el agente hace clic en el enlace "Ver suscripciones activas". El sistema consulta la base de datos para buscar todas las suscripciones del agente. El sistema muestra la vista "Suscripciones Activas" con la lista de de suscripciones del agente.
Curso Alternativo:	Agente no tiene suscripciones	El sistema muestra la vista "Suscripciones Activas", con un mensaje que el agente no ha suscrito a ningún clasificación de leads.

Tabla 2-13 Descripción de caso de uso "Ver Suscripciones Activas"

Nombre de caso de uso:	Editar detalles	
Curso Básico:	Editar detalles	En la vista "Panel de control", el agente hace clic en el enlace "Editar datos personales". El sistema muestra la vista "Datos personales". El agente hace lo cambios deseados y hace clic en "Actualizar". El sistema valida los datos y si están válidos, actualiza la cuenta del agente con respeto a los cambios hechos.
Curso Alternativo:	Agente no tiene suscripciones	El sistema muestra la vista "Datos personales" indicando los errores

Tabla 2-14 Descripción de caso de uso "Editar Detalles"

2.5.2.2 Casos de uso del paquete "Clientes"

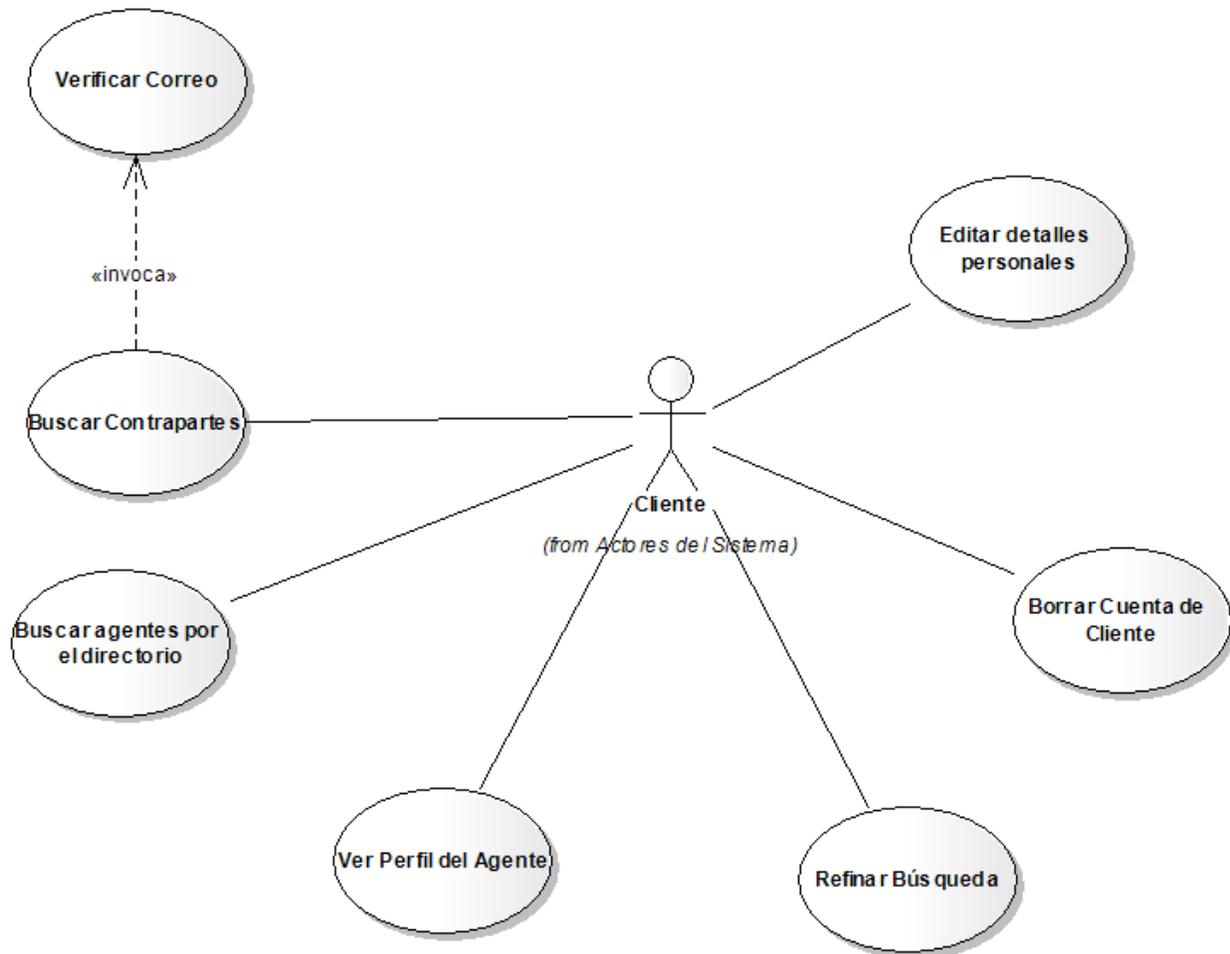


Figura 2-5 Diagrama de Casos de Uso del Paquete Clientes

Como se muestra en la Figura 2-5, los siguientes casos de uso fueron identificados en este paquete:

1. Verificar Correo
2. Buscar Contraparte
3. Buscar agentes por el directorio
4. Ver perfil del Agente
5. Refinar Búsqueda
6. Editar datos personales
7. Borrar cuenta de cliente

Nombre de caso de uso:	Buscar Contrapartes	
Curso Básico:	Buscar Contrapartes	<p>En la vista "Página Inicial (de clientes)", el cliente selecciona su país, su provincia y el tipo de seguros que está buscando, y hace clic en el enlace "Comenzar". El sistema muestra la vista "Entrar Detalles". El cliente llena el formulario y hace clic en el enlace "Buscar Contrapartes". El sistema guarda esta información en la sesión actual.</p> <p>Si el sistema encontró agentes emparejados, el sistema muestra la vista "Emparejada". El cliente selecciona los agentes que quiere y cuando termina, hace clic en el enlace "Confirmar". Si el cliente ya está autenticado el sistema crea un lead para cada agente e invoca el caso de uso "despachar leads". Si no está autenticado, el sistema guarda los detalles de la búsqueda e invoca el caso de uso "Verificar Correo". Cuando la dirección de correo está verificada, el sistema pasa por el caso de uso "despachar leads".</p>
Curso Alternativo:	No encontró contrapartes	El sistema muestra la vista "Agentes No encontrados"

Tabla 2-15 Descripción de caso de uso "Buscar Contrapartes"

Nombre de caso de uso:	Buscar agentes por el directorio
------------------------	----------------------------------

Curso Básico:	Buscar agentes por el directorio	En la vista "Página Inicial", el cliente hace clic en el enlace "Navegar Directorio de Agentes". El sistema consulta la base de datos para todos los agentes registrados. Después, el sistema muestra la vista "Directorio de Agentes" con la lista de agentes.
---------------	----------------------------------	---

Tabla 2-16 Descripción de caso de uso "Buscar Agentes por el Directorio"

Nombre de caso de uso:	Refinar Búsqueda	
Curso Básico:	Refinar Búsqueda	En la vista "Criterios de la Búsqueda", el cliente hace clic en una de las enlaces para refinar su búsqueda. El sistema hace una búsqueda utilizando los nuevos criterios y muestra la vista "Directorio de Agentes" con la nueva selección de agentes.

Tabla 2-17 Descripción de caso de uso "Refinar Búsqueda"

Nombre de caso de uso:	Ver Perfil del Agente	
Curso Básico:	Ver Perfil del Agente	En las vistas "Directorio de Agentes" y "Emparejada", o sea, donde hay una lista de agentes, el cliente hace clic en el enlace "Ver perfil" de un agente. El sistema hace una consulta en la base de datos para el agente especificado y muestra la vista "Perfil del Agente".

Tabla 2-18 Descripción de caso de uso "Ver Perfil de Agente"

Nombre de caso de uso:	Verificar Correo
------------------------	------------------

Curso Básico:	Verificar Correo	El sistema manda un correo de confirmación al cliente con el código único. El cliente hace clic en el enlace del código. El sistema verifica el código. Si el código es válido, muestra la vista, "Datos finales". El cliente llena los campos de este formulario y hace clic en "Mandar Info". El sistema verifica los datos. Si los datos son válidos el sistema convierte los datos del cliente guardados en una cuenta real. El sistema muestra la vista "Gracias" con una lista de los agentes elegidos y pasa por el caso de uso "despachar leads".
Curso Alternativo:	Datos inválidos	El sistema muestra la vista "Datos finales" con indicadores donde habían errores.
Curso Alternativo:	Código inválido	El sistema muestra la vista "Código Inválido"

Tabla 2-19 Descripción de caso de uso "Verificar Correo"

Nombre de caso de uso:	Editar detalles personales	
Curso Básico:	Editar detalles personales	En la vista "Panel de Control" el cliente hace clic en el enlace "cambiar datos personales". El sistema muestra la vista "Datos personales". El cliente hace sus cambios y hace clic en "Guardar Cambios". El sistema verifica si los datos son válidos y si son, el sistema actualiza los datos en la base de datos. Después, el sistema muestra la vista "Panel de Control".
Curso Alternativo:	Datos Inválido	El sistema muestra la vista "Datos personales" indicando que habían errores

Tabla 2-20 Descripción de caso de uso "Editar Detalles Personales"

Nombre de caso de uso:	Borrar cuenta de cliente
------------------------	--------------------------

Curso Básico:	Borrar cuenta de cliente	En la vista "Panel de Control", el cliente hace clic en el enlace "borrar cuenta". El sistema muestra un diálogo de confirmación. El cliente decide seguir con su decisión. El sistema cambia el estado actual de la sesión a no-autenticado y borra los registros pertinentes de la base de datos. Pues, el sistema muestra la vista "Página Inicial".
Curso Alternativo:	Cliente hizo clic en "No" en el diálogo de confirmación	El sistema muestra la vista "Panel de Control"

Tabla 2-21 Descripción de caso de uso "Borrar Cuenta de Cliente"

2.5.2.3 Matriz de responsabilidad

A continuación se muestran en las Tabla 2-22 y Tabla 2-23, las matrices de responsabilidad para los paquetes de Agente y Cliente respectivamente:

	REQ01	REQ02	REQ03	REQ04	REQ05	REQ06	REQ07	REQ08	REQ09	REQ10	REQ11
Adquirir créditos						X					
Autenticar agente				X							
Borrar cuenta de agente		X									
Crear Cuenta de Agente	X										
Desuscribirse de una clasificación de leads			X								
Editar detalles					X						
Mostrar Panel de Control								X		X	
Rechazar Lead											X
Suscribir a clasificaciones leads							X		X		
Ver leads								X		X	
Ver suscripciones activas								X		X	

Tabla 2-22 Matriz de Responsabilidad del Paquete Agentes

	REQ12	REQ13	REQ14	REQ15	REQ16	REQ17	REQ18	REQ19
Borrar Cuenta de Cliente		X	X					
Buscar agentes por el directorio						X		
Buscar Contrapartes	X				X			
Editar detalles personales			X	X				
Refinar Búsqueda							X	
Ver Perfil del Agente								X
Verificar Correo	X							

Tabla 2-23 Matriz de Responsabilidad del Paquete Cliente

2.5.3 Diagramas de Robustez

Para disminuir la brecha entre análisis y diseño, el ICONIX propone el análisis de robustez (ver Figura 2 - Análisis de Robustez). El resultado del análisis de robustez es un diagrama de robustez para cada caso de uso. Un diagrama de robustez es un “imagen” de un caso de uso, cuyo propósito es obligar el refinamiento del texto de los casos de uso y el modelo del objeto. Es decir, los diagramas de robustez pegan los casos de uso a los objetos(6).

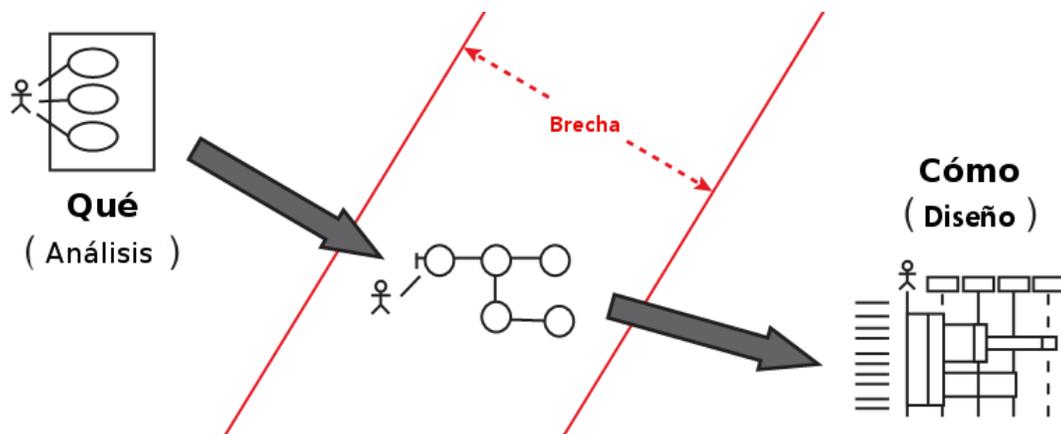


Figura 2-6 Análisis de Robustez

Ver el anexo 1 para los diagramas de robustez

2.6 Conclusiones

En este capítulo se realizó el modelo de casos de uso del sistema, partiendo del estudio y definición de los requerimientos funcionales y no-funcionales del sistema propuesto. Se realizó también el análisis de robustez del sistema. Se mostraron, como artefactos resultantes del proceso, diagramas de casos de

uso y diagramas de robustez. Queda claro que el proceso ICONIX coloca mucha importancia en el concepto de trazabilidad, se muestra también en este capítulo la matriz de requerimientos.

Capítulo 3 Modelación del Sistema Propuesto

3.1 Introducción

En este capítulo se describe la modelación del sistema propuesto. Utilizando artefactos como diagramas de clases del sistema, el diagrama del modelo físico de la base de datos y el diagrama de implementación, se ilustra cómo fue implementado los descubrimientos del análisis de requerimientos.

3.2 Diagramas de clases

Un diagrama de clase es una presentación gráfica de la vista estática que muestra una colección de elementos declarativos del modelo, tal como, clases, tipos, y su contenido y relaciones(13).

Para el sistema propuesto, como fue mencionado anteriormente, se utiliza una manifestación del patrón MVC. Las clases del sistema se dividen en tres paquetes, los cuales son clases controladores, clases de tipo modelo y clases que pertenecen a la biblioteca de código, CodeIgniter. A continuación se muestra el diagrama de paquetes de clases del sistema propuesta y los diagramas de clases del sistema propuesto:



Figura 3-1 - Diagrama de Paquete del las clases del sistema propuesto

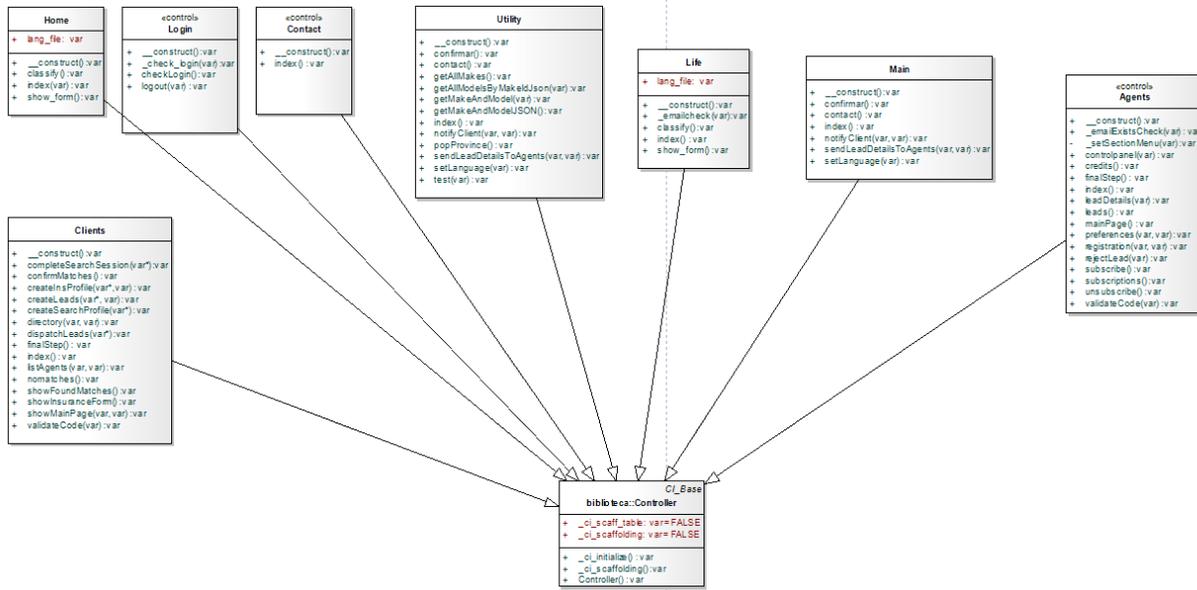


Figura 3-2 Diagrama de clase del paquete controladores

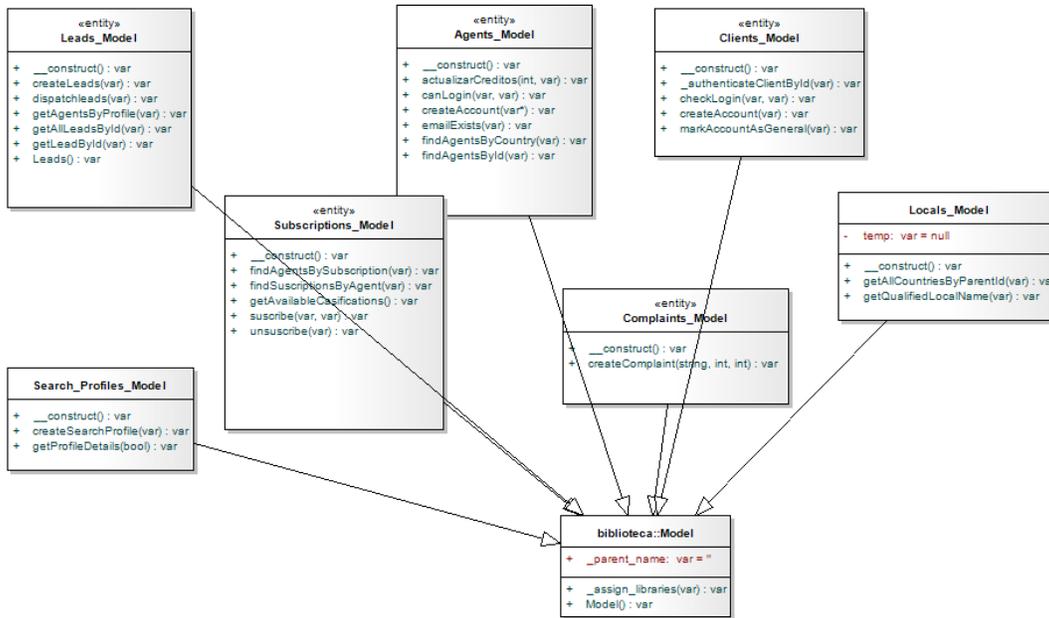
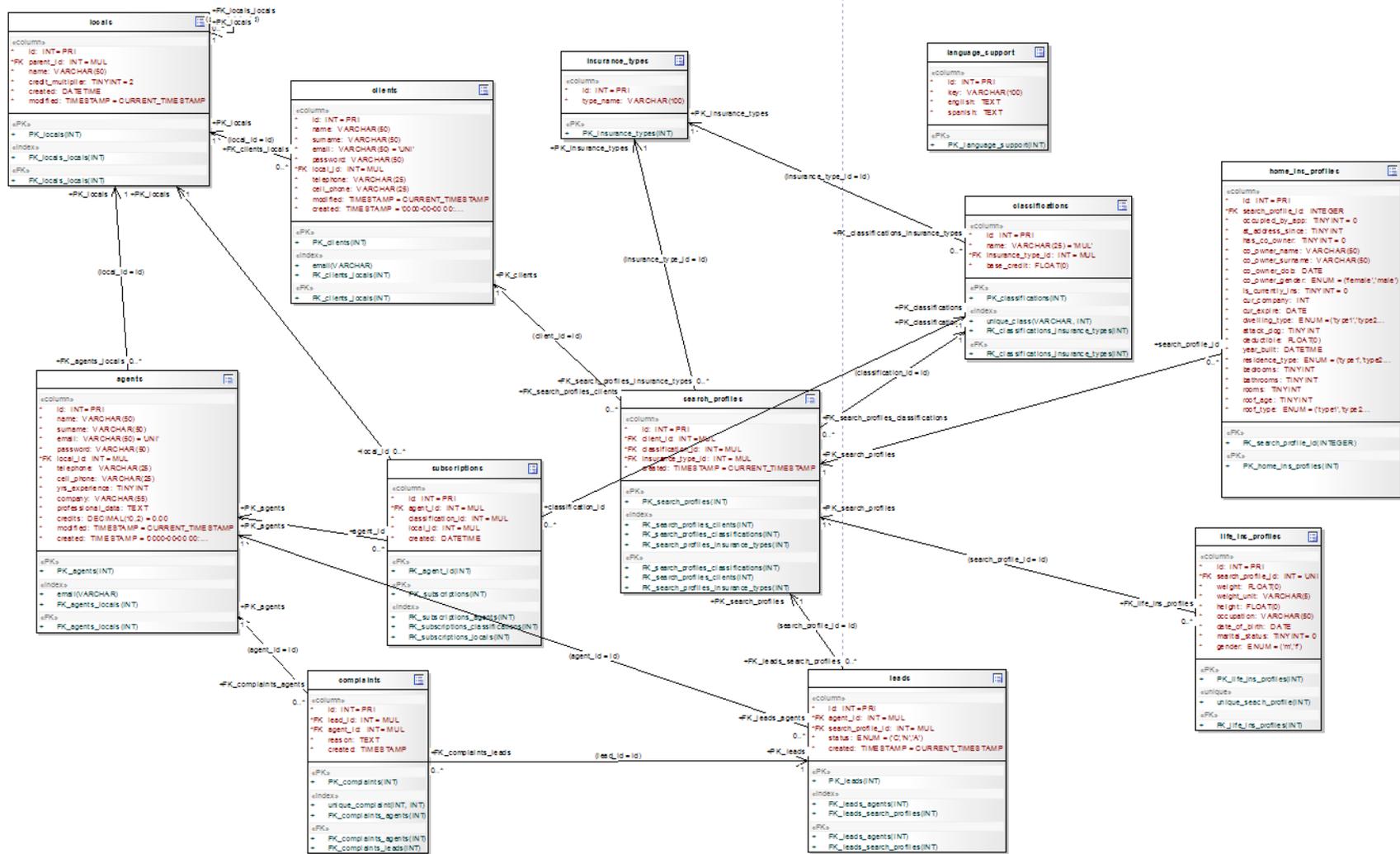


Figura 3-3 Diagrama de clase del paquete modelos

3.3 Diagrama del modelo físico de datos

El modelo físico de datos ilustra el diseño de las tablas de la base de datos y las relaciones entre si. En la Figura 3-4 se muestra el diagrama del modelo físico de datos del sistema propuesto.

Figura 3-4 Modelo físico



3.4 Diagramas de despliegue

El despliegue es la asignación de artefactos (ficheros, scripts, tablas de la base de datos, etc.) a nodos físicos durante la ejecución(13). Un diagrama de despliegue es una implementación visual del despliegue del sistema. En la Figura 3-5, se muestra el diagrama de despliegue del sistema propuesto durante la fase de desarrollo. En la Figura 3-6, se muestra el diagrama de despliegue propuesto para la implementación viva. La diferencia principal es que el despliegue del sistema durante el desarrollo tiene solamente un punto de fallo para la base de datos. Tener esta implementación en un despliegue vivo es riesgoso dado la importancia de la base de datos en el sistema propuesto. La arquitectura sugerida en la figura 6 tiene redundancia con respecto a la base de datos, la cual disminuye la probabilidad de un fallo crítico. También, este despliegue aumenta el rendimiento de la aplicación.

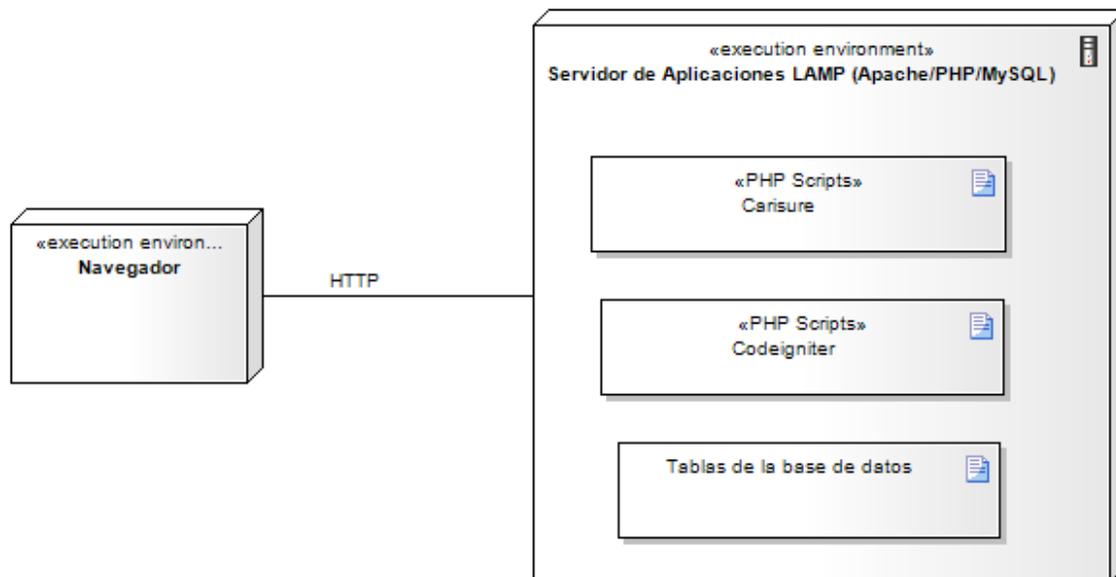


Figura 3-5 Diagrama de Despliegue(desarrollo)

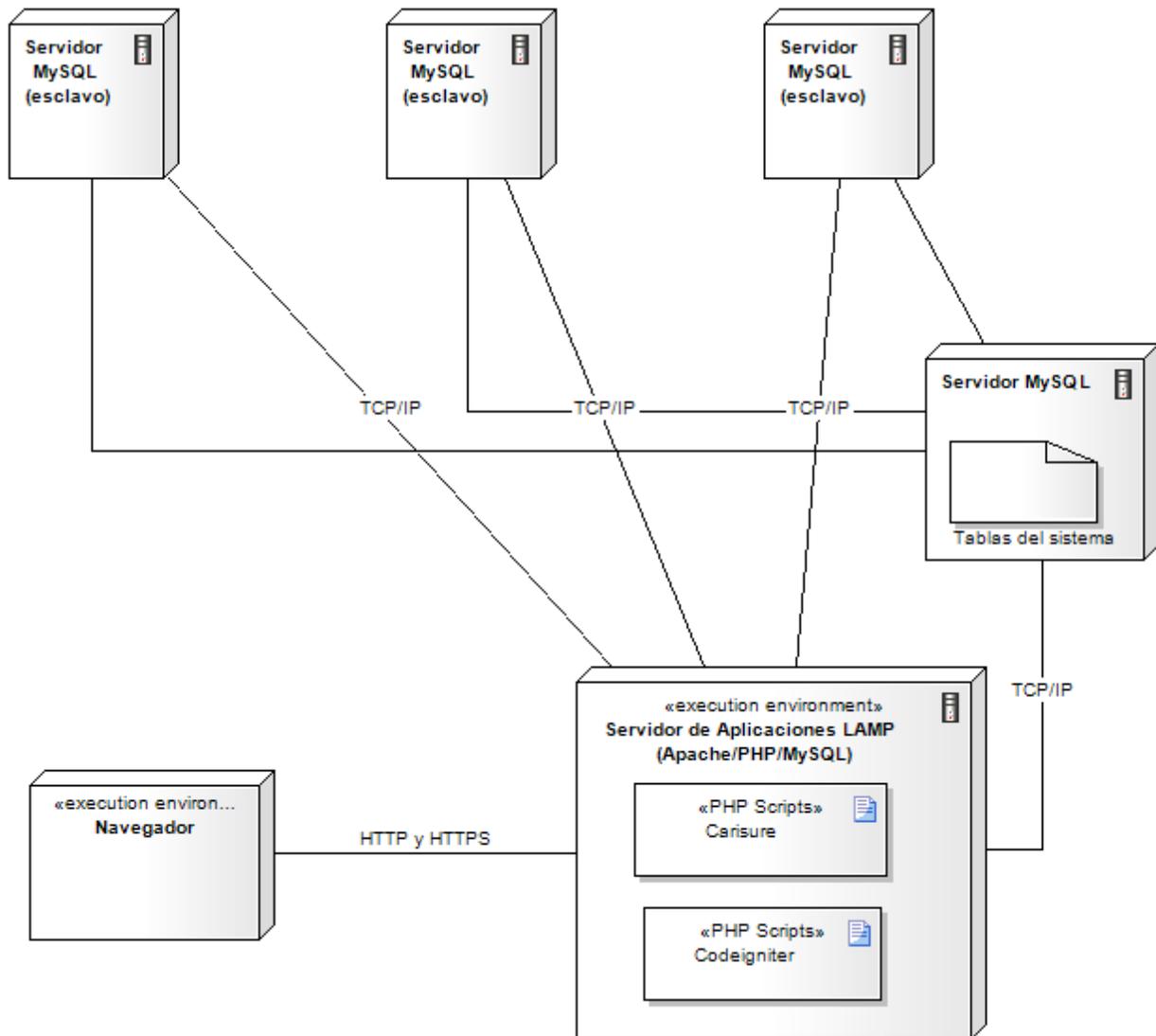


Figura 3-6 Diagrama de Despliegue(Propuesta para la implementación)

3.5 Principio de Diseño del sistema

3.5.1 Estándares en la interfaz de la aplicación

Un usuario utilizando un sitio web no es igual a un usuario utilizando una aplicación ejecutada en un ordenador personal. Normalmente un sitio web no tiene disponible un manual de uso, o si existe, el usuario no consulta un manual de uso. Por esto el sitio web tiene que ser explicativo, es decir su interfaz tiene que ser intuitiva. El usuario no debe quedarse en pensar a dónde ir.

3.5.2 Tratamiento de errores

Primeramente, el sistema está implementado en una manera que la interfaz guía el usuario. Es decir controla toda la navegación del usuario. Pero en los casos de excepciones, por ejemplo fallos en la conexión a la base de datos, el sistema cuenta con mensajes cordiales en lugar de mensajes crudos del servidor. Por ejemplo un mensaje que dice “Hubo un error el administrador fue notificado...” en vez de “MySQL Failure #129865 Connection with ODBC....”.

3.5.3 Estándares de codificación

Para el caso de un programador igual que múltiples programadores, la comprensión de código y la productividad general se aumentan al utilizar un estándar de codificación. A continuación se describen algunas convenciones tomadas durante el desarrollo del sistema propuesto con relación a estos aspectos.

3.5.3.1 Convención de nombres

En el desarrollo del sistema, los nombres de identificadores (variables, funciones, constantes, clases etc.) fueron descriptivas de su propósito pero no largos. Además, dependiendo del tipo de identificador, un patrón de nombre fue seguido.

Tipo de identificador	Descripción de la convención	Ejemplo
Variable	Palabras separadas con una “_” con letras minúsculas	\$leads_actual
Clase	Empieza con mayúscula	Controler
Constante	Todas las letras mayúsculas	MYSQLDATE
Nombre de una función	Estilo de camello; empieza con minúscula	buscarContraparte
Nombre de una tabla de la BD	Palabras separadas con una “_”; letras minúsculas; plural	Life_ins_profiles

3.5.3.2 Indentación

En el caso de la indentación, se establece que todas las líneas dentro de un procedimiento o función, estarán indentadas con respecto a la instrucción que encabeza a este, lo mismo ocurre con todas las líneas que conformen el cuerpo de un ciclo estructural condicional.

3.5.3.3 Comentarios

Los comentarios siguen el formato de PHPDoc. PHPDoc es una aplicación para automáticamente generar documentación de código. Para lograr esto, los comentarios tienen que tener un formato específico con el uso de palabras daves. Por ejemplo para comentar una función la se utiliza el siguiente formato:

```
/**
 * Descripción breve de la función
 * Descripción más descriptiva de la función
 * @author Felis St. Bernard
 * @var tipo_variable $nombre_variable
 */
```

3.6 Conclusiones

En este capítulo se muestran los diagramas de clases del sistema físico, el modelo físico de los datos y los diagramas de despliegue. Se definieron además los principios de diseño que gobernaron el desarrollo de la aplicación.

Capítulo 4 Estudio de factibilidad

4.1 Introducción

Una tarea muy importante, pero a la vez, difícil de hacer en la gestión de un proyecto de desarrollo de software es la estimación de costos. Debido a factores como la insuficiencia de recursos y fechas límites – que pueden ser pocos realistas, es preciso evaluar la viabilidad de abordar un nuevo proyecto antes de empezar de desarrollarlo.

En este capítulo se realiza un estudio que conduce si es viable o no, el desarrollo del sistema propuesto. La decisión se basa en los descubrimientos del estudio de costos y de beneficios. Las estimaciones buscadas son el costo, el tiempo esperado y la cantidad de desarrolladores requeridos. Para realizar el análisis de factibilidad se utiliza el Cocomo II.

4.2 Cálculos

Tipo de función	Nivel de complejidad									Total
	Simple			Medio			Complejo			
	Cuenta	Peso	Puntos de función	Cuenta	Peso	Puntos de función	Cuenta	Peso	Puntos de función	
Entradas	4	3	12	0	4	0	9	6	54	66
Salidas	5	4	20	3	5	15	5	7	35	70
Ficheros lógicos internos	11	7	77	7	10	70	2	15	30	177
Ficheros externos	0	5	0	0	7	0	0	10	0	0
Consultas	19	3	57	4	4	16	2	6	12	85
Total Puntos función desajustados (CF)										398

Tabla 4-1 Total Puntos Funcion

Características	Valor		
Puntos de Función	398		
Lenguaje	PHP	SQL	Javascript
Instrucciones fuentes por puntos de función	44	37	63
Por ciento de la aplicación en cuanto a requerimientos funcionales	70%	20%	10%
Instrucciones fuentes	12258.4	2945.2	2507.4
Total de instrucciones fuentes (MF)	17711		

Tabla 4-2 Total Instrucciones Fuentes

Factor	Valor	Wi	Justificación
PREC	4.96	4	Así que es la primera vez enfrentado con esta tipa de aplicación, se asuma nivel 4 (Bajo)
FLEX	3.04	3	El sistema cuenta con alguna flexibilidad en relación con las especificaciones de los requerimientos preestablecidos y a las especificaciones de interfaz externa(Nominal)

RESL	1.41	1	El desarrollador tiene bastante experiencia con el lenguaje principal de programación y además, el paquete principal de desarrollo cuenta con unas herramientas muy buenas para la depuración
TEAM	0	0	Hay solo un desarrollador entonces
PMAT	6.24	4	Así que es la primera vez enfrentado con esta tipa de aplicación, se asuma un nivel 1 alto

Tabla

4-3

Factores

de

Escala

Factor	Valor	Descripción
RELY	1.91	Muy alto
DATA	1.33	Alto
DOCU	1.91	Muy alto
CPLX	1.33	Alto
RUSE	1.15	Muy alto
TIME	0.87	Bajo
STOR	1	Nominal
PVOL	1	Nominal
ACAP	0.5	Extra alto
PCAP	0.5	Extra alto
AEXP	1.33	5 meses
LTEX	0.62	Extra alto
PEXP	0.62	Extra alto
TOOL	0.62	Extra alto
SITE	1	Nominal
SCED	1	Nominal

Tabla 4-4 Factores de Esfuerzo

Figura 4-1 Factores de esfuerzo

Cálculos

Valores de los coeficientes

A 2.94

B 0.91

C 3.67

D 0.24

CHM \$225 peso/mes

Variable	Formula	Valor Calculado
Multiplicador de esfuerzo	$EM = \prod_{i=0}^m EM_i$	0.511627438
Factor de escala	$SF = \sum_{j=0}^n SF_j$	15.65
Coefficiente	$E = B + 0.01(SF)$	1.0665
Coefficiente	$F = D + 0.2(E - B)$	0.2713
Miles de instrucciones fuentes	MF	17.711
Esfuerzo	$PM = A(MF)^E EM$	32.25168475 (hombre × mes)
Tiempo de desarrollo	$TDEV = C(PM)$	9.417515247 (mes)
Personal necesario	$CH = PM/TDEV$	3.424649061 (hombre)
Costo	$Costo = CHM \times PM$	\$7256.629068

Tabla 4-5 Sumario de los cálculos

4.3 Beneficio tangibles e intangibles

Ocupándonos primeramente con los beneficios tangibles, es preciso notar que el sistema propuesto será comercializado. Por lo tanto el beneficio es los ingresos que aportará esta aplicación.

Hablando de beneficios intangibles, Carisure no será el único servicio brindado por Spicelander Financial. Y con los altos gastos pagados por empresas para publicidad, ellas acogen cualquiera forma de publicidad gratis y positiva. La firma beneficiará de la exposición brindada por Carisure para sus otros servicios.

4.4 Conclusiones

Se estimó que el proyecto necesitará aproximadamente cuatro hombres para su realización. Sin embargo el proyecto será realizado por un hombre (ver la siguiente tabla). Entonces, calculando de nuevo el tiempo de desarrollo, se da un valor de 32.25168475 meses \approx 32 meses con un costo esperado de \$7256.63.

El análisis de factibilidad nos dio los siguientes valores:

Cálculo de:	Valor
Esfuerzo(PM)	32.25168475
Tiempo de desarrollo	≈ 32 meses
Cantidad de hombres	1
Costo	≈ \$7256.63
Salario medio	\$225.00

Tabla 4-6 Resulto del Análisis

Conclusiones

Se realizó un estudio de la disposición actual de la industria de seguros en la región del Caribe. Este estudio demostró que una aplicación tal como Carisure sería útil para clientes y así como también para agentes de esta región. Para clientes, les ayuda con su decisión a la hora de buscar un agente y para los agentes, lo hace más fácil conocer nuevas personas que ya están predispuestos a la compra de seguros – no hace falta convencerlas.

Se utilizó, para el desarrollo del trabajo, el proceso ICONIX. Quedó demostrado que este proceso es adecuado para el desarrollo de software pequeños y medios.

Se diseñó e implementó una aplicación que facilite la gestión de información necesaria para la vinculación de clientes potenciales con vendedores de productos de seguros, y la promoción del intercambio entre ambas partes, en la región del Caribe.

Se realizó un estudio que permitió demostrar la factibilidad del desarrollo de la aplicación.

Recomendaciones

La palabra “sistema” puede ser recursiva. Es decir, un sistema puede ser parte de un “sistema”. Y Carisure, el “sistema” presentado por este trabajo de diploma, tiene que ser parte de otro sistema. La versión presentada en este trabajo es básica. Debido al propósito comercial de esta aplicación hay muchas implicaciones a considerar – no solo su función fundamental de vincular clientes potenciales con vendedores de productos de seguros. Debido a este hecho y considerando otros se recomienda:

- Continuar el desarrollo de Carisure
 - Implementando nuevos módulos como cotizaciones instantes
 - Integración con un marchante de tarjetas de crédito (como PayPal, Visa Online, etc.) para la cobra en línea
- Implementar Carisure como un módulo externo de un sistema de gestión de contenido (CMS²). Por ejemplo, Joomla!³, Mambo CMS⁴ y Expression Engine⁵

² CMS – sigla en ingles para “Content Management System”

³ Joomla – <http://www.joomla.org>

⁴ Mambo CMS – <http://www.mambocms.com>

⁵ Expression Engine – www.expressionengine.com

Bibliografía

1. Insurance - Information from Reference.com. *Reference.com*. [En línea] [Citado el: 07 de June de 2008.] <http://www.reference.com/search?q=insurance>.
2. *Fundación Ignacio Larramendi*. [En línea] [Citado el: 27 de March de 2008.] <http://www.larramendi.es/inicio.htm>.
3. El Seguro - Monografias.com. [En línea] [Citado el: 7 de May de 2008.] <http://www.monografias.com/trabajos40/el-seguro/el-seguro.shtml>.
4. **Cicci, Sara Stefania Rizzi**. Concepto de Seguros. [En línea] 20 de May de 2008. http://www.elprisma.com/apuntes/administracion_de_empresas/seguroconcepto.
5. **Pragma Consultores**. *¿Agile o Unified? "Fanatismo" o desarrollo de software*. [Document] España : s.n., 2007.
6. **Rosenberg, Doug y Stephens, Matt**. *Use Case Driven Object Modeling with UML: Theory and Practice*. s.l. : Apress, 2007.
7. **Lerdorf, Rasmus, MacIntyre, Peter y Tatroe, Kevin**. *Programming PHP*. 2006.
8. **Martin Fowler, y otros**. *Patterns of Enterprise Application Architecture*. s.l. : Addison Wesley, 2002. 0-321-12742-0.
9. **Ellislab, Inc**. Manual de Codeigniter. *Codeigniter Manual*. [En línea] [Citado el: 23 de January de 2008.] <http://www.codeigniter.com/manual>.
10. **Kofler, Michael**. *The Definitive Guide to MySQL5*. s.l. : Apress, 2005.
11. **MySQL AB**. *MySQL Reference Manual*. 2006.
12. **Cockburn, Alistair**. *Writing Effective Use Cases*. s.l. : Addison-Wesley, 2000.
13. **Rumbaugh, James, Jacobson, Ivor y Booch, Grady**. *The Unified Modelling Language Reference Manual*. s.l. : Addison-Weseley, 2004. pág. 217.
14. Metodologías modernas de desarrollo de Sistemas de Información. [En línea] [Citado el: 23 de Mayo de 2008.] <http://www.monografias.com/trabajos12/documento/documento.shtml>.

15. **Buschman, Frank, y otros.** *Pattern-Oriented Software Architecture: A system of Patterns*. s.l. : John Wiley & Sons Ltd, 1996.
16. **Kappel, Gerti, et al.** *Web Engineering - The Discipline of Systematic Development of Web Applications*. s.l. : John Wiley and sons, p. 25.
17. **Gunnar Övergaard, Karin Palmkvist.** *Use Cases Patterns and Blueprints*. s.l. : Addison Wesley Professional, 2004.
18. **Kruchten, Philippe.** *The Rational Unified Process An Introduction, Second Edition*. s.l. : Addison Wesley, 2000.
19. **REIERSØL, DAGFINN, BAKER, MARCUS and SHIFLETT, CHRIS.** *PHP in Action: Obejets, Design, Agility*. s.l. : Manning Publications Co., 2007.
20. **Hall, Stephen.** Instant Underwriting. [En línea] 7 de June de 2008. http://www.loma.org/res-02-07-instant_underwriting.asp.
21. **McInturff, Tammy J.** insurance service. *www.loma.org*. [En línea] 07 de Jun de 2008. <http://www.loma.org/res-11-06-business-IT.asp>.
22. Manual de PHP.
23. **Larman, Craig.** *Applying UML and Patterns*. s.l. : Prentice-Hall, 200?
24. **Pressman, Roger S.** *Software Engineering A PRACTITIONER'S APPROACH*. s.l. : McGraw-Hill Higher Education, 2001.
25. **GradyBooch, y otros.** *Object-Oriented Analisis and Design with Applications*. s.l. : Pearson Education, 2007.

Anexo A Diagramas de Robustez

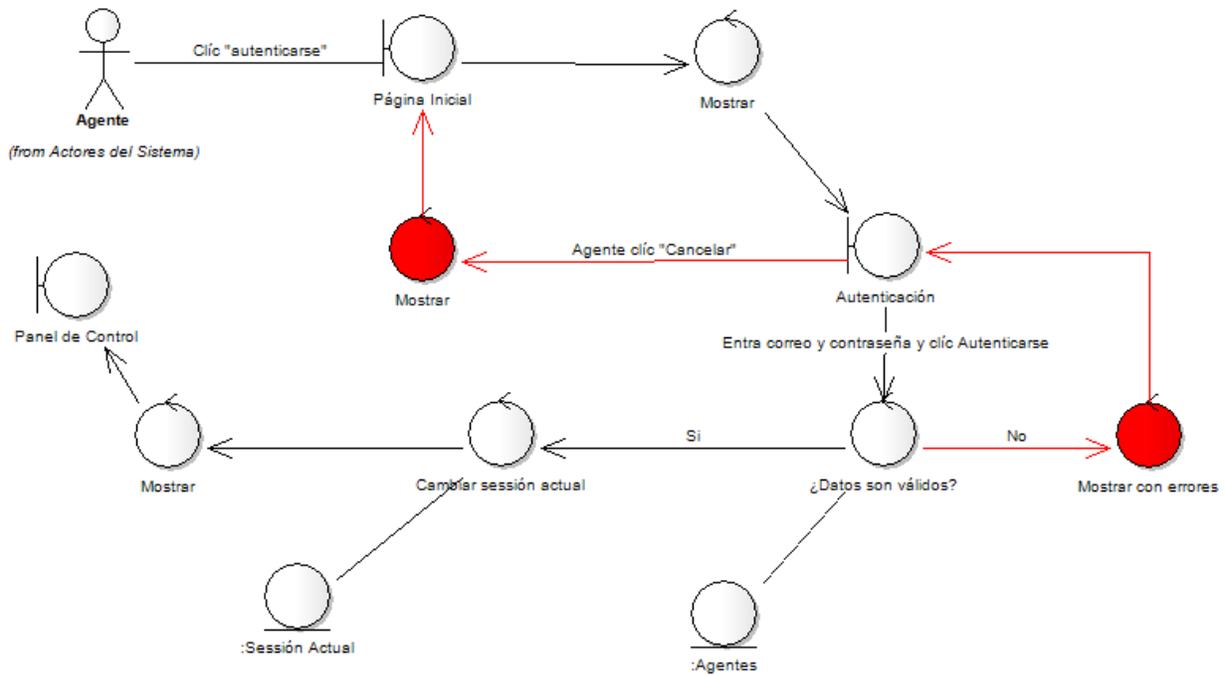


Figura A-1 Diagrama de Robustez "Autenticar Agente"

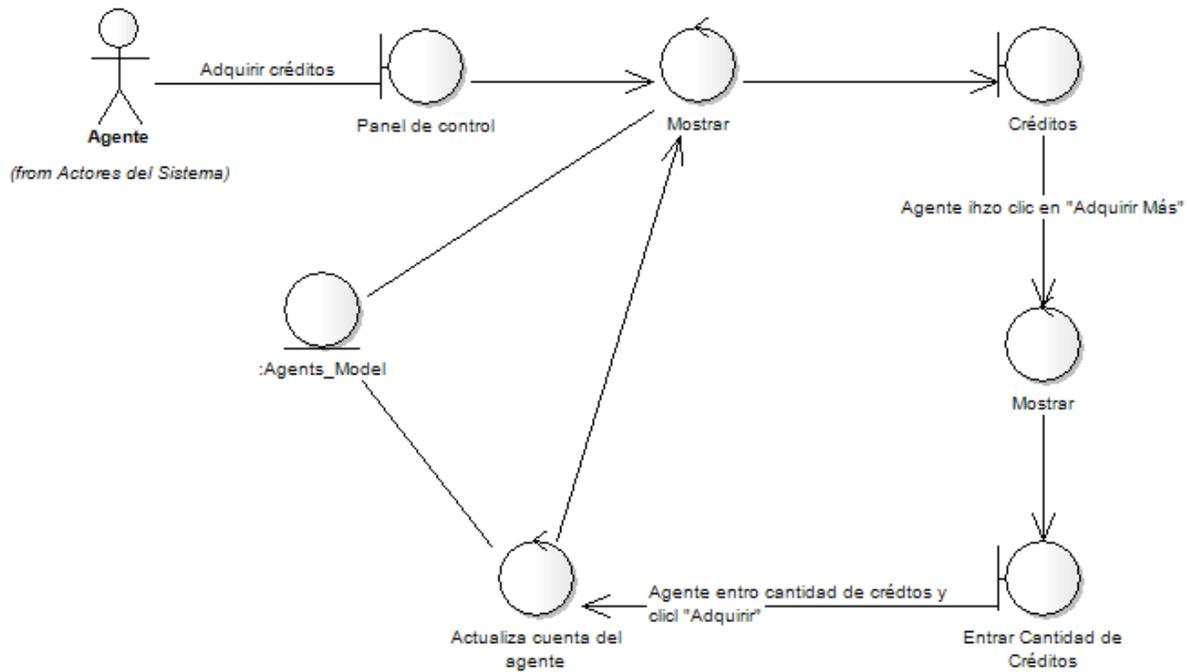


Figura A-2 Diagrama de Robustez "Adquirir Créditos"

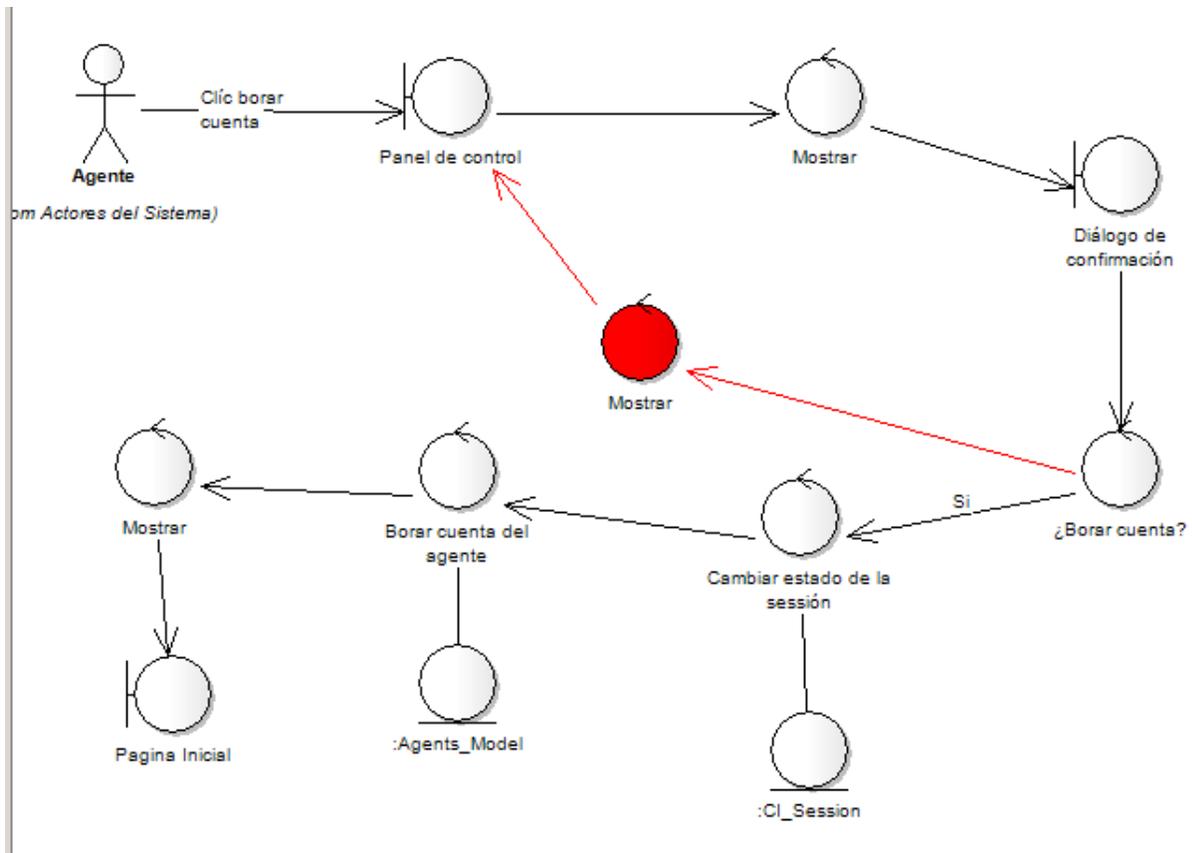


Figura A-3 Diagrama de Robustez "Borrar cuenta de Agente"

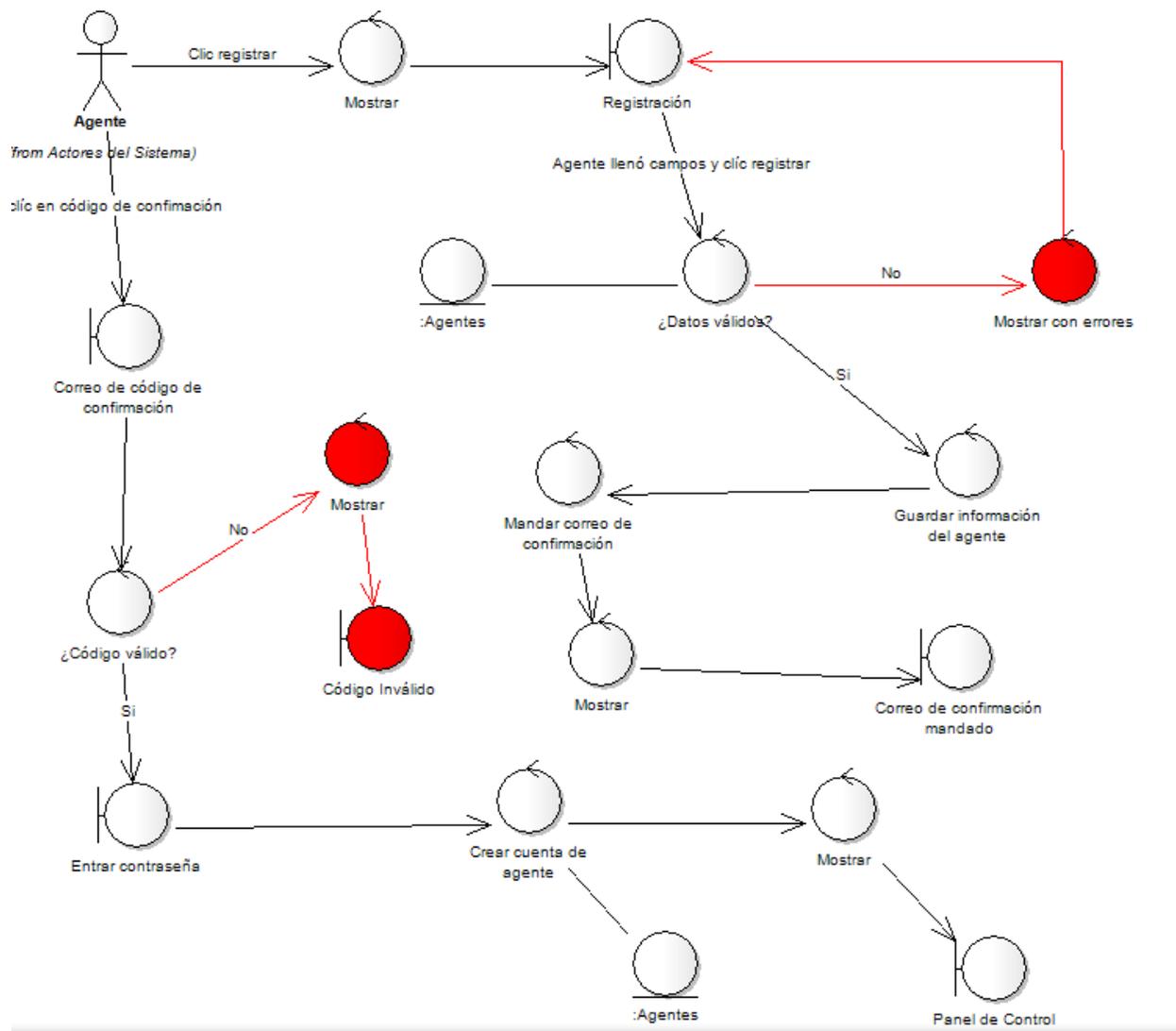


Figura A-4 Diagrama de Robustez "Crear cuenta de agente"

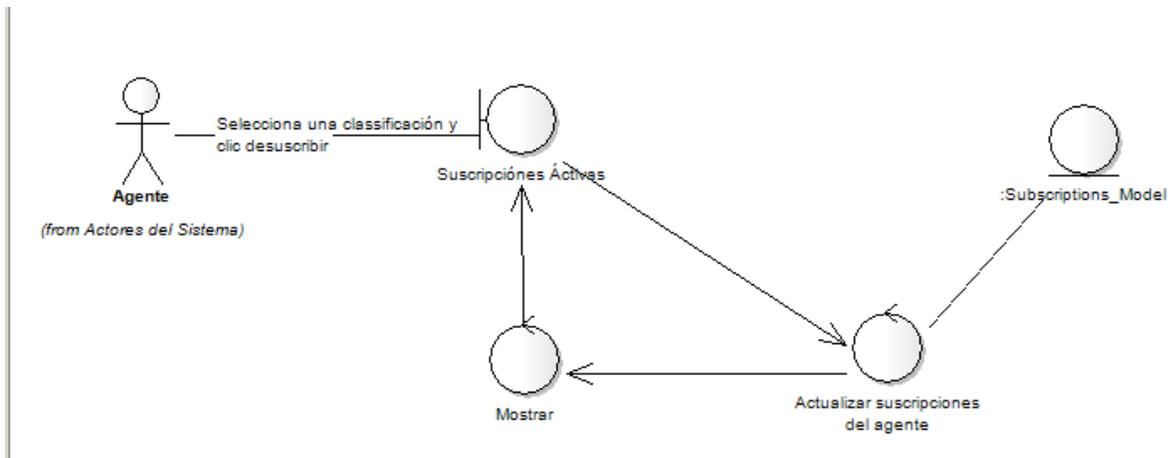


Figura A-5 Descripción de caso de uso "Describirse de una clasificación de leads"

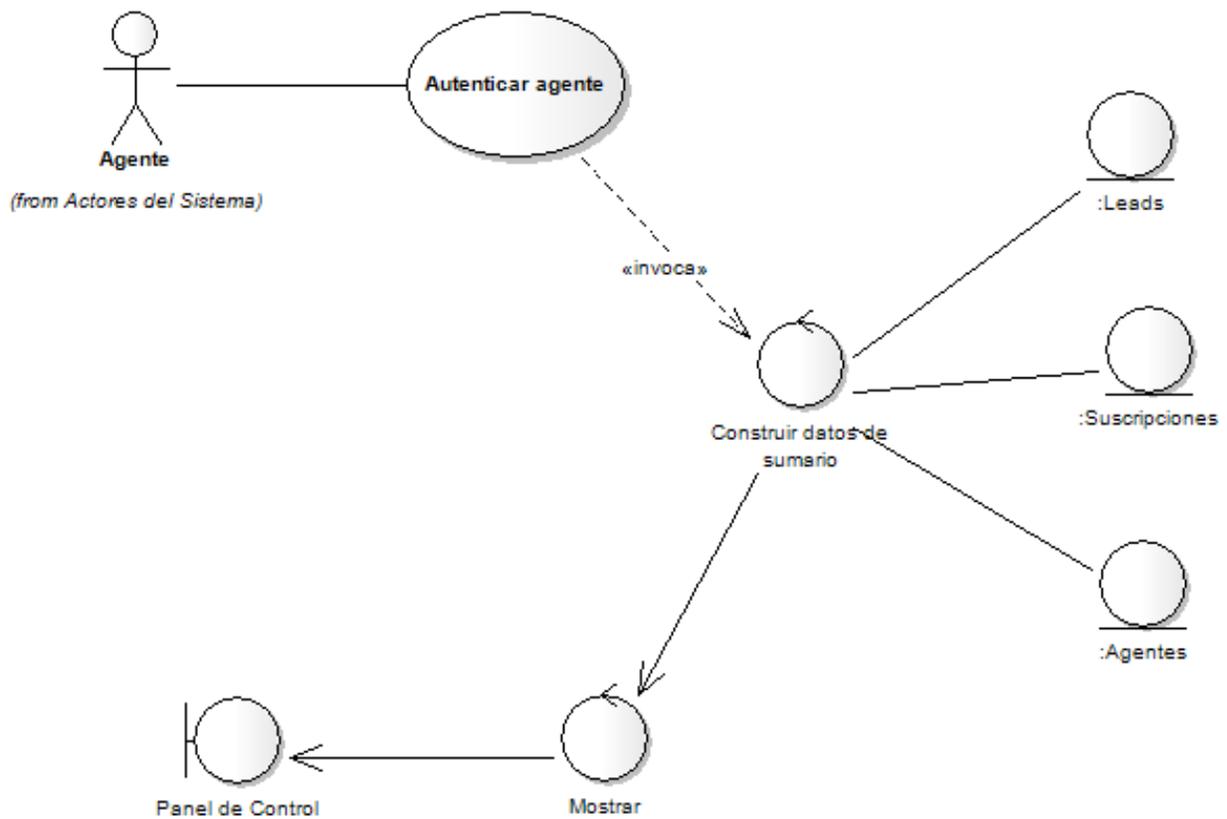


Figura A-6 Diagrama de Robustez de caso de uso "Mostrar Panel de Control"

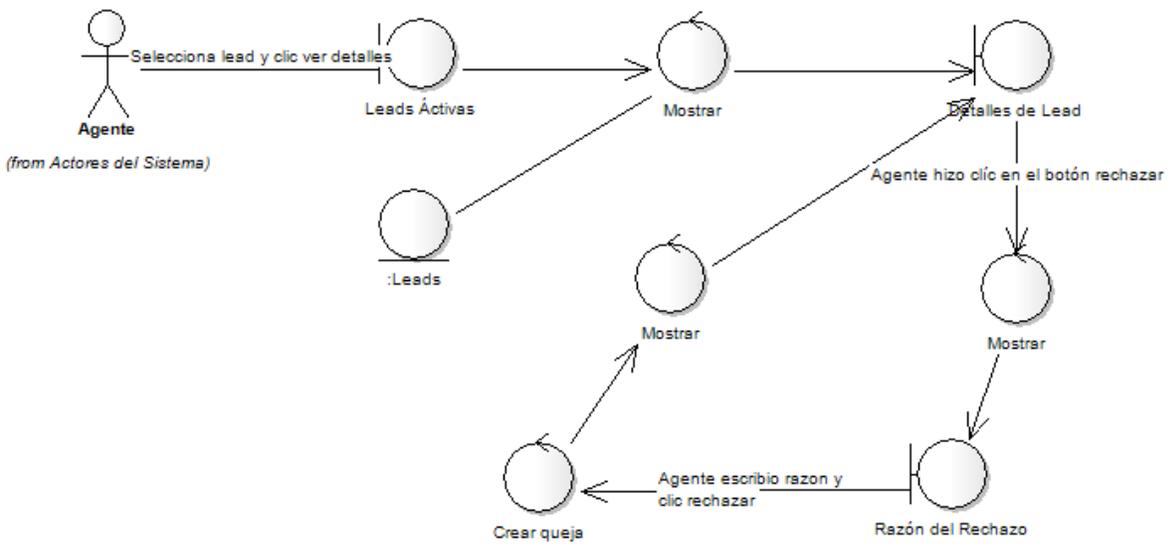


Figura A-7 Diagrama de robustez del caso de uso "Rechazar Lead"

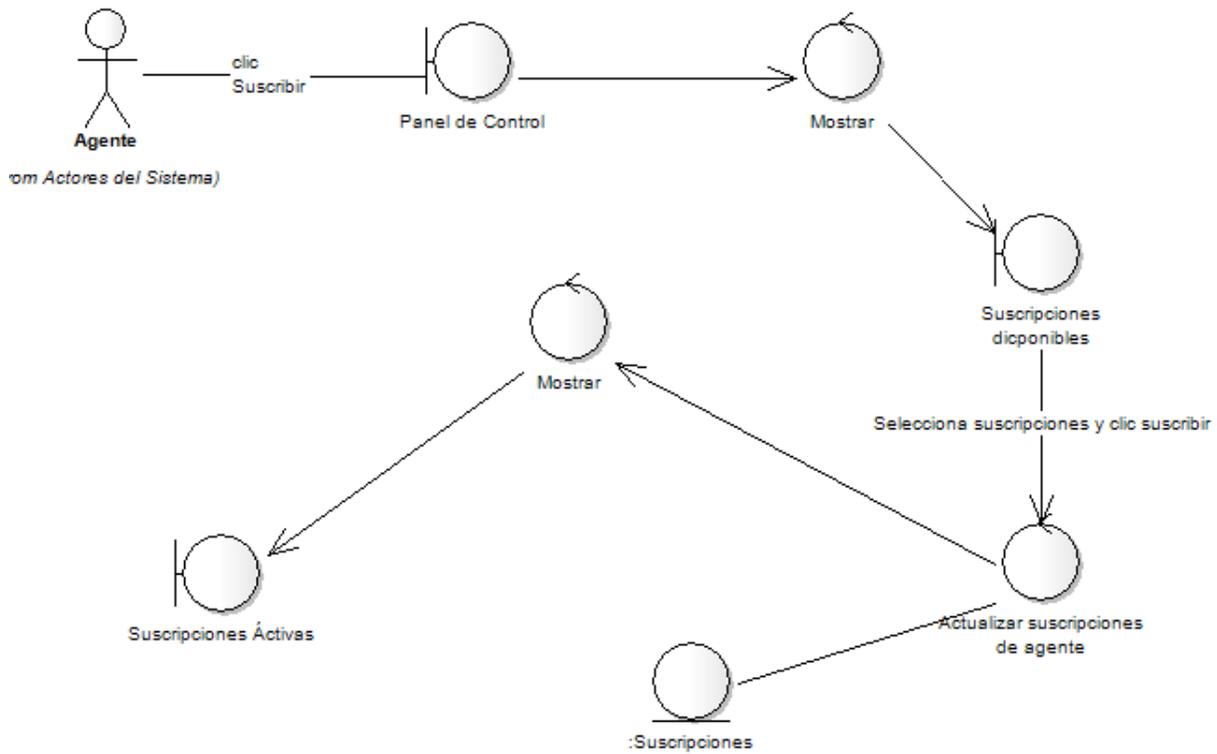


Figura A-8 Diagrama de robustez del caso de uso "Suscribir a clasificaciones leads"

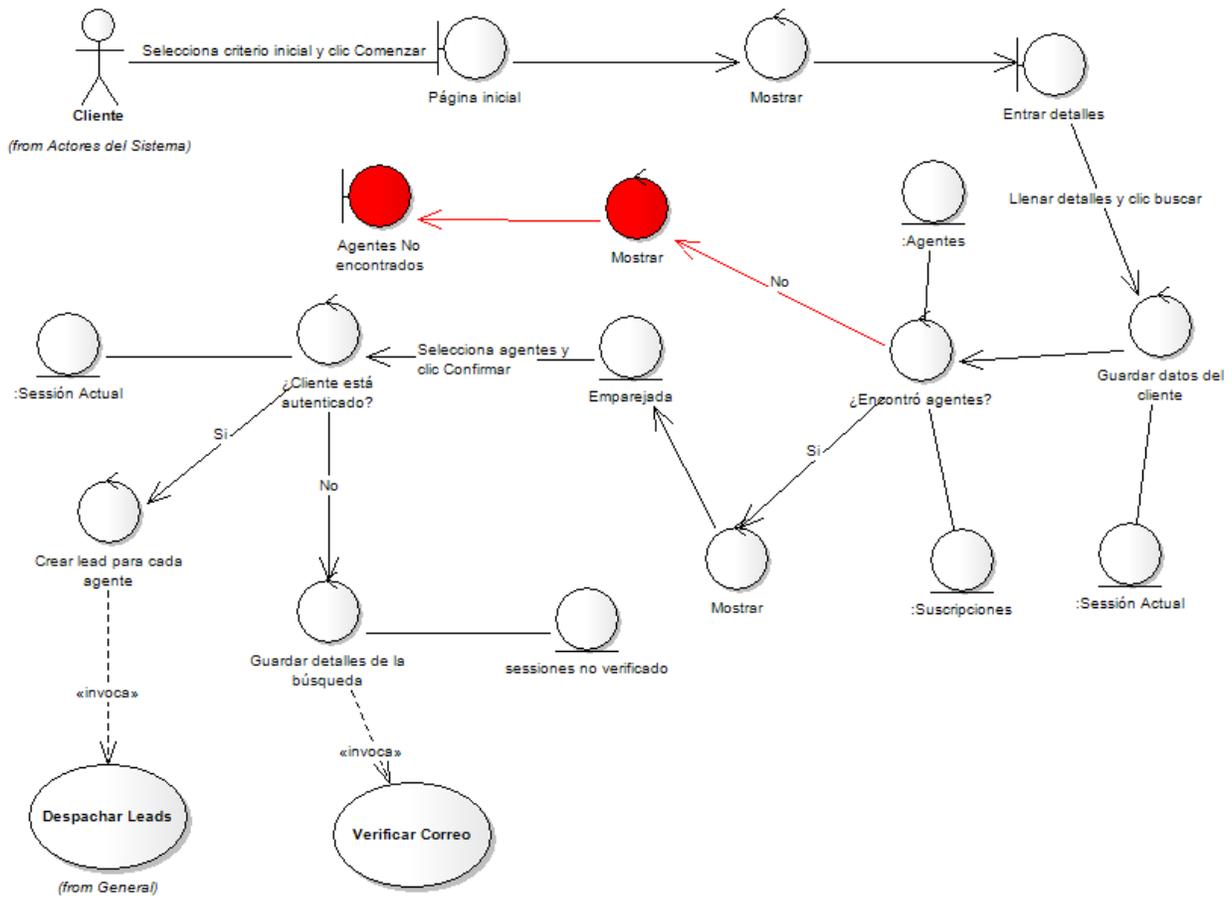


Figura A-9 Diagrama de robustez "Buscar Contraparte"

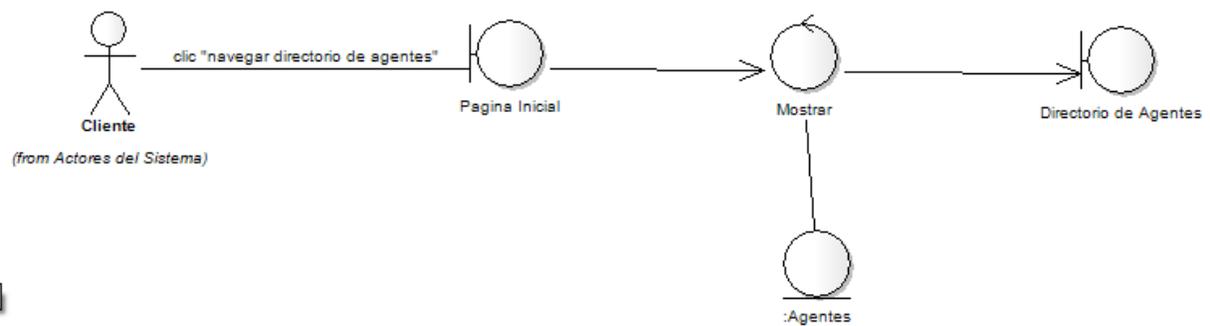


Figura A-10 Descripción de caso de uso "Buscar agentes por el directorio"

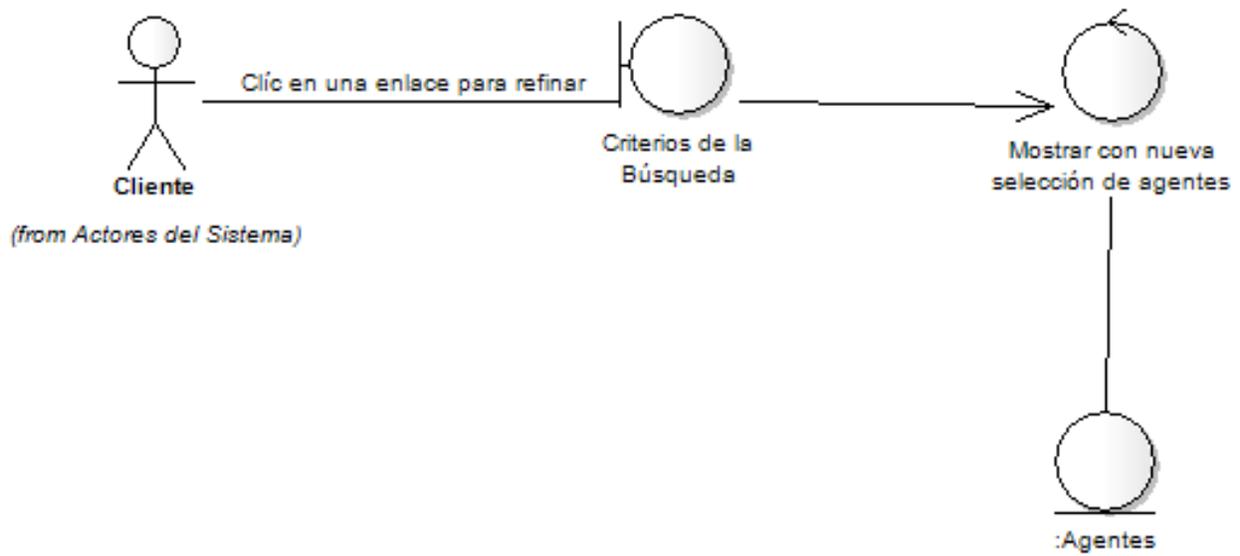


Figura A-11 Diagrama de Robustez del caso de uso "Refinar Búsqueda"

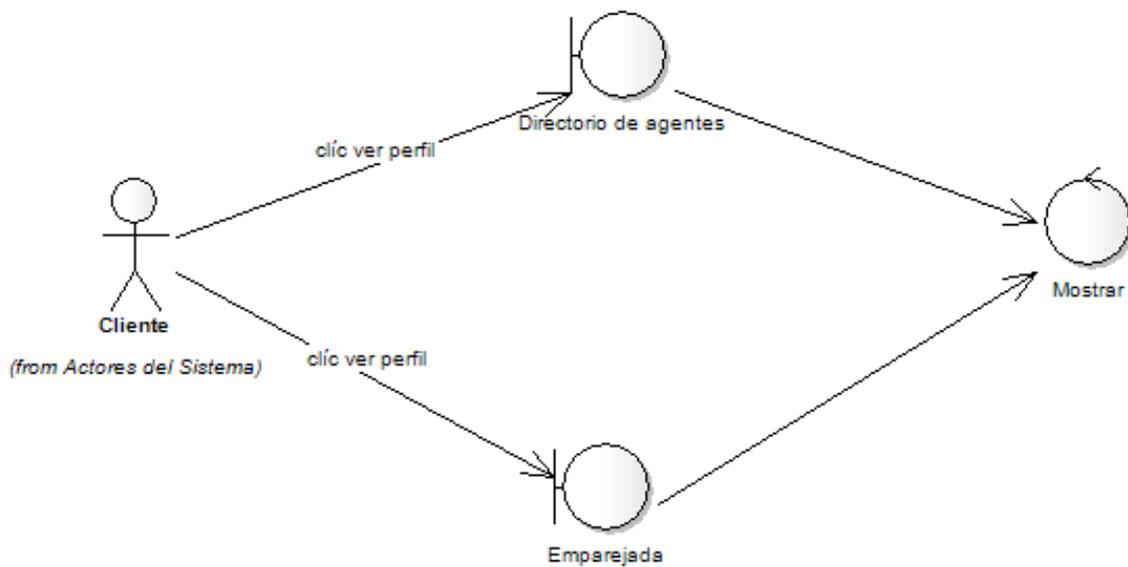


Figura A-12 Diagrama de Robustez del Caso de Uso "Ver perfil del Agente"

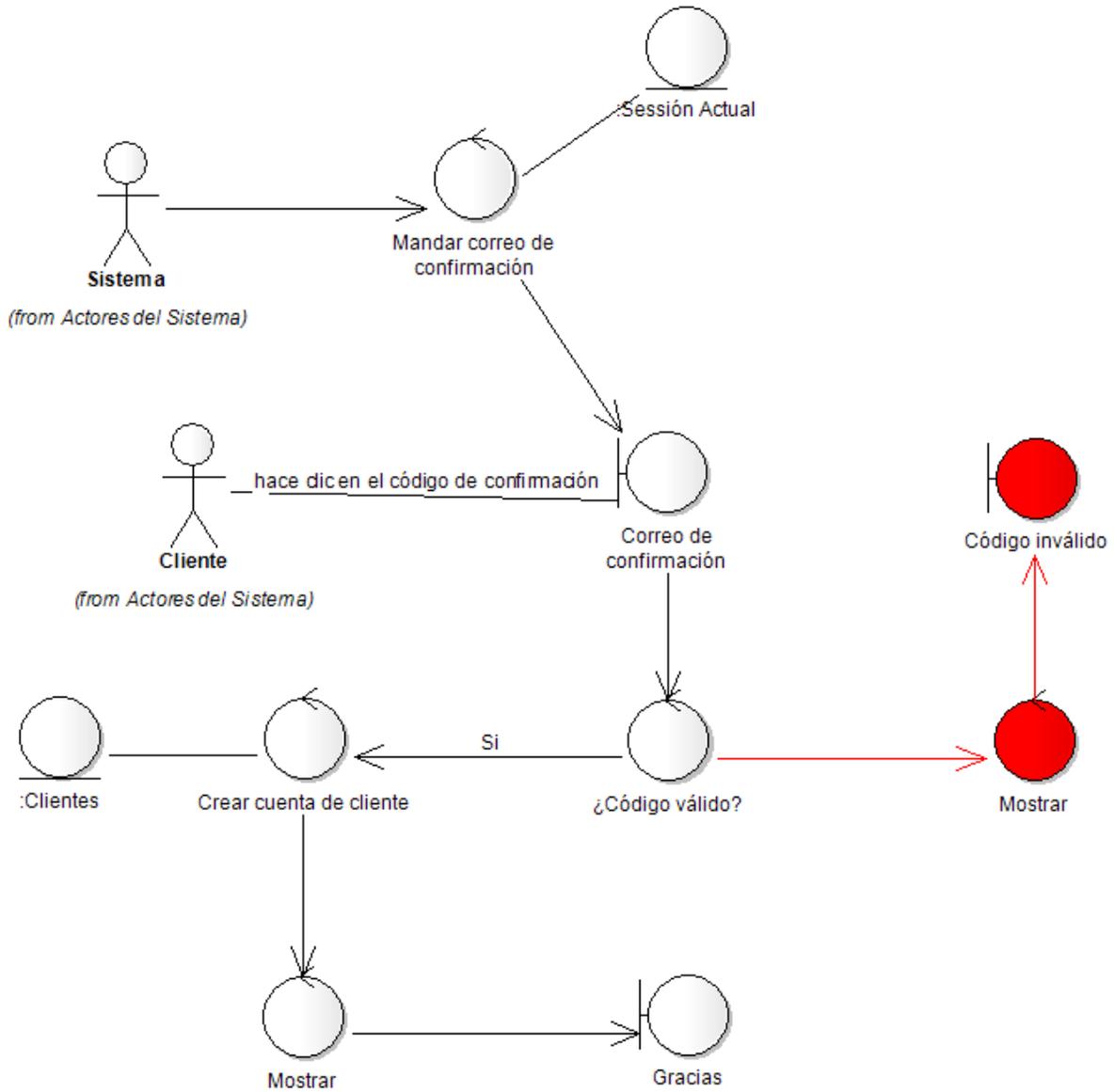


Figura A-13 Diagrama de Robustez del caso de uso "Verificar Correo"