



UNIVERSIDAD  
NACIONAL  
DE LA PLATA

## FACULTAD DE INFORMÁTICA

# TESINA DE LICENCIATURA

Programa de Apoyo al Egreso de Profesionales en Actividad

**TÍTULO:** Análisis y desarrollo de una aplicación web de liquidación de Siniestros de Riesgos Varios

**AUTOR:** Tonso, Fabrizio Jesús

**DIRECTOR ACADÉMICO:** Queiruga, Claudia **CODIRECTOR ACADÉMICO:** Rosso, Jorge Horacio

**DIRECTOR PROFESIONAL:** Laferrara, Víctor Gabriel

**CARRERA:** Licenciatura en Sistemas

### Resumen

Con el avance de la tecnología cada vez son más las empresas que buscan automatizar los servicios que brinda, no sólo para ofrecer información más confiable sino también para mejorar el control sobre la empresa y apoyar la toma de decisiones.

La compañía aseguradora donde trabajo, dispone de un sistema de administración de siniestros del ramo auto, moto vehículo y ART, sin embargo, no cuenta con una herramienta para administrar los siniestros de Riesgos varios, lo cual atrasa considerablemente la resolución de los casos de este tipo. El propósito de esta tesina es ampliar la aplicación web de Siniestros existente, con el previo análisis correspondiente. Esta ampliación permitirá a los liquidadores de Riesgos Varios gestionar todos los casos que le son derivados con información más amplia y ordenada. Además, le permitirá a la compañía tener un control de todos los casos gestionados a través de sus responsables internos y administradores.

### Palabras Clave

Siniestros. Estudios liquidadores. Aplicación web. Análisis y desarrollo. Compañía de seguros. Ramo. Póliza. Asegurados.

### Conclusiones

Los liquidadores de siniestros de Riesgos varios disponen de una herramienta para administrar, de una manera amigable y confiable, los casos que le son asignados.

Además, esta herramienta le permite al usuario llevar un control más ágil de todos sus casos, con el fin de presentar la indemnización correspondiente en el debido tiempo. A su vez la compañía puede tomar decisiones, en base a los tiempos empleados por cada liquidador en resolver un caso, para brindarles una mejor atención a sus asegurados.

### Trabajos Realizados

Se analizó e implementó una herramienta de software con interface web ampliando un sistema ya existente, denominado "Siniestros", para facilitar la gestión de los siniestros de Riesgos Varios. Para ello se aplicaron los conceptos aprendidos en las asignaturas Ingeniería de Software II, Base de Datos II, y Java y Aplicaciones Avanzadas sobre Internet (optativa del Área Arquitectura, Sistemas Operativos y Redes) de la Facultad de Informática de la UNLP.

Se realizaron cuestionarios, diagrama entidad relación, casos de usos, etc. También se llevaron a cabo reuniones con el negocio e internas para determinar las tablas existentes para su eventual reutilización, y para determinar la forma de la persistencia de datos. La aplicación web se desarrolló en el lenguaje JAVA, utilizando JSF y RichFaces.

### Trabajos Futuros

En una siguiente etapa, se espera poder ampliar el perfil del "Súper Administrador" para poder gestionar las restantes funcionalidades del sistema.

Además, se prevé agregar la posibilidad de adjuntar archivos en el perfil de los liquidadores del ramo auto y moto vehículo, atendiendo que actualmente los liquidadores envían la documentación pertinente del siniestro por mail a los empleados internos de la compañía.

Otro posible desarrollo, es la generación de órdenes de pago desde el perfil liquidador de los ramos anteriormente mencionados. Como próxima mejora, sería óptimo facilitarle el blanqueo de contraseña a los usuarios antes de ingresar al sistema.

Fecha de la presentación: Agosto 2020

**Universidad Nacional de La Plata**

**Facultad de Informática**



**Tesina de Grado PAEPA**

---

**Análisis y desarrollo de una aplicación web de  
liquidación de Siniestros de Riesgos Varios**

---

Directora Académica:  
Claudia Queiruga

Codirector Académico:  
Jorge Horacio Rosso

Director Profesional:  
Víctor Gabriel Laferrara

Autor:  
Fabrizio Jesús Tonso

agosto de 2020



# Índice General

Índice de Figuras	iii
Índice de tablas	iv
<i>Agradecimientos</i>	v
Introducción	1
1.1 Objetivos	1
1.2 Motivación	2
1.3 Organización General	3
Antecedentes y Fundamentos Teóricos	4
2.1 Introducción	4
2.1.1 Definiciones	5
2.1.2 Introducción a las metodologías de desarrollo de software	6
2.1.2.1 Metodologías tradicionales	6
2.1.2.2 Metodologías Ágiles	6
2.1.2.3 Comparación entre Metodologías Ágiles y Tradicionales	7
2.2 Marco Teórico	9
2.2.1 La propuesta del desarrollo	9
2.2.2 La metodología de desarrollo de la compañía	10
2.2.3 Metodología de desarrollo implementada	13
2.2.4 Tecnologías utilizadas	15
2.2.4.1 Java Server Faces	15
2.2.4.2 RichFaces	15
2.2.4.3 Base de datos Oracle	16
2.2.4.3.1 Triggers (o Disparadores)	16
2.3.4.3.2 Procedures (o Procedimientos)	17
2.3.4.3.3 Functions (o Funciones)	17
2.3.4.3.4 Packages (o Paquetes)	17
2.3 Estado del Arte	18
2.3.1 Comparación de la metodología de la compañía con las metodologías existentes	18
2.3.1.1 Modelo en Cascada	18
2.3.1.2 Modelo en V	20

2.3.2 Comparación de la metodología implementada con las metodologías existentes	21
2.3.2.1 Programación Extrema (XP)	21
2.3.2.2 Método de Desarrollo de Sistema Dinámico (MDSD)	24
3.3.3 Antecedentes	26
Implementación en la compañía	28
3.1 Contexto del problema	28
3.2 Análisis	29
3.3 Desarrollo	31
3.3.1 Creación de Perfiles y de Nuevas páginas	32
3.3.2 Asignación de un nuevo caso	32
3.3.3 Búsqueda de los siniestros de Rs. Vs.	34
3.3.4 Consulta de un siniestro	35
3.3.5 Contacto con el asegurado	35
3.3.6 Informe Preliminar	36
3.3.7 Informe Final	41
3.3.8 Generación de orden de pago	43
3.3.9 Carga de la factura de pago	44
3.3.10 Continuación del caso	48
3.3.11 Mensajería	49
3.3.12 Digitalización	49
3.3.13 Descarga de los Endosos de la Póliza y la Denuncia del Siniestro	49
3.3.14 Listado de Novedades	50
3.3.15 Estadísticas	50
3.3.16 Administración de usuarios	53
3.3.17 Seguridad en todas las páginas	55
Trabajos Futuros	56
4.1 Introducción	56
4.2 Ampliar el perfil Super Administrador	56
4.3 Agregar la suba de archivos digitales	57
4.4 Generación de órdenes de pago a los liquidadores	57
4.5 Blanqueo de clave por el usuario	58
Conclusiones	59
Bibliografía	61

## Índice de Figuras

<i>FIGURA 2.1 Gestión de un Nuevo Cambio</i>	12
<i>FIGURA 2.2 Modelo en Cascada</i>	19
<i>FIGURA 2.3 Modelo en V</i>	20
<i>FIGURA 2.4 Programación Extrema</i>	22
<i>FIGURA 2.5 Método de Desarrollo de Sistema Dinámico</i>	24
<i>FIGURA 3.1 Procesos Gestión de Caso</i>	30
<i>FIGURA 3.2 Diagrama Entidad-Relación</i>	30
<i>FIGURA 3.3 Trigger para actualizar los nuevos atributos de la tabla</i>	
<i>SINT_LIQUIDACION_ORAMOS</i>	33
<i>FIGURA 3.4 Trigger para notificar a los liquidadores el nuevo caso derivado</i>	33
<i>FIGURA 3.5 Pantalla de Búsqueda de Siniestros</i>	34
<i>FIGURA 3.6 Pantalla de Detalle de un Siniestro</i>	35
<i>FIGURA 3.7 Formulario Informe Preliminar</i>	37
<i>FIGURA 3.8 Selección de Cobertura Afectada al Siniestro</i>	38
<i>FIGURA 3.9 Coberturas Afectadas al Siniestro</i>	39
<i>FIGURA 3.10 Fragmento del Código de rich:fileUpload</i>	40
<i>FIGURA 3.11 Carga del Formulario</i>	40
<i>FIGURA 3.12 Formulario Informe Final</i>	42
<i>FIGURA 3.13 Fragmento del Procedimiento RESERVA_COBERTURA_UPD</i>	43
<i>FIGURA 3.14 Carga de la Factura</i>	46
<i>FIGURA 3.15 Json para Comunicarse con la AFIP</i>	47
<i>FIGURA 3.16 Obtención de la Respuesta de la AFIP</i>	48
<i>FIGURA 3.17 Estadísticas</i>	51
<i>FIGURA 3.18 Detalle Estadísticas</i>	52
<i>FIGURA 3.19 Panel de Control</i>	53
<i>FIGURA 3.20 Administración de Usuarios</i>	54

## Índice de tablas

*TABLA 2.1 Comparación entre las Metodologías Tradicionales y Ágiles*

8

## *Agradecimientos*

Agradezco haber llegado a este punto principalmente a mis padres que me apoyaron desde el comienzo de la carrera para que estudie y me auto supere día a día; también, les agradezco a ellos, mis “nonis” y tío que en los momentos más difíciles estuvieron ahí cuando, faltando pocas materias para culminar la carrera, quería abandonar sin ánimos para seguir. Sin embargo, gracias a su aliento y a la propuesta laboral que conseguí, en donde estoy trabajando actualmente, saque motivación y entusiasmo para seguir y llegar hoy a la confección de esta tesina.

Mis sinceros agradecimientos a mis directores académicos Claudia Queiruga y Jorge Rosso, por sus asesoramientos, sus importantes correcciones que me guiaron a escribir esta tesina, más aún en esta época de pandemia que, a pesar de la nueva modalidad de enseñanza web, pudieron hacerse un tiempo para responderme; y gracias a Jorge aprendí a programar en Java, lo que me abrió las puertas en el ámbito laboral y me dio el conocimiento para realizar esta tesina.

Gracias a mi asesor profesional, Víctor Laferrara, por su paciencia para guiarme, por sus comentarios y consejos para llevar a cabo el desarrollo a lo largo de la implementación de este proyecto. Así también, quiero hacer mención de alguien que fue como mi segunda asesora, Daniela Beloqui; gracias a ella por enseñarme conceptos claves del negocio y cómo aplicarlos, y por estar siempre a disposición, además de alegrarse por cada avance que realizaba en el proyecto.

Quiero agradecer a todos mis compañeros que estuvieron ahí por cada consulta y consejo que pedí, respondiendo siempre de la mejor manera. Tenemos un buen equipo laboral, ¡hay equipo Sistemas!

A la Facultad de Informática de la Universidad Nacional de La Plata, por darme la posibilidad de estudiar, brindarme la formación y herramientas para poder desempeñarme como profesional.

A la compañía de seguros, por permitirme llevar adelante este desarrollo dentro del ambiente laboral.

Agradezco de corazón a Marcelo Naiouf y Pablo Thomas por ponerse en contacto conmigo y comentarme la posibilidad de realizar la tesina en la modalidad PAEPA, fue el puntapié inicial para realizar esta tesina; a la Dirección Pedagógica de la facultad, Anahí Almán, por el taller de tesina que puso a disposición de los alumnos, junto con las actividades del taller y el material para guiarme en la confección de la tesina orientada a la puntuación; a María Julia Lozada y Florencia Bajo Isusi por acompañarme en la escritura de mi tesina, desde la portada a la disposición de los capítulos. Agradezco a todos los nombrados en este párrafo por estar a disposición por cada mail de consulta que envié.

Gracias a todos ellos, que de forma directa o indirectamente son partícipes de esta tesina.





# Capítulo 1

---

## Introducción

### 1.1 Objetivos

El objetivo general de esta tesina es **desarrollar una aplicación web** que le permita a los **estudios liquidadores**, asociados a la compañía aseguradora en la que desarrollo mi actividad profesional, **gestionar los siniestros de Riesgos Varios** (Rs. Vs.)<sup>1</sup> que le son asignados; y a su vez que le permita a la compañía tener control de todos los casos gestionados a través de sus responsables internos y administradores.

Los liquidadores cumplen un papel muy importante dado que son las personas que evalúan los siniestros declarados tanto por los propios asegurados como por terceros.

Una compañía de seguros es una sociedad anónima que tiene por objeto exclusivo realizar las actividades y operaciones de seguros, coaseguros y reaseguros en general, exclusivamente en las coberturas autorizadas por la SSN (Superintendencia de Seguros de la Nación). Son las únicas autorizadas por este organismo para celebrar contratos de seguro.

Los objetivos específicos son:

- Analizar los requerimientos del negocio (área de siniestros de la compañía).
- Evaluar el sistema de siniestros y demás sistemas actuales que utiliza la compañía.
- Analizar la base de datos para reutilizar, de ser posible, las tablas, los paquetes, procedimientos y funciones que actualmente existen; o crear nuevos.

---

<sup>1</sup> De ahora en más se dirá Rs. Vs. cuando se haga referencia a Riesgos Varios.

- Ampliar la funcionalidad del sistema de siniestros existente.
- Realizar las pruebas pertinentes para la implementación de la nueva herramienta.

## 1.2 Motivación

Como es sabido cada vez son más las empresas que utilizan aplicaciones web para automatizar los servicios que brinda, con el fin de ofrecer información confiable y en tiempo real para mejorar la eficiencia, para incrementar la rentabilidad, tener una mejor relación con los clientes, mejorar el control sobre la empresa y apoyar la toma de decisiones.

La compañía de seguros en la que me desempeño posee un sistema orientado a estudios liquidadores externos que trabajan con siniestros de autos (ramo 44), moto vehículo (ramo 44) y ART (ramo 80). A este sistema también tienen acceso abogados de legales, abogados penalistas, auditores médicos y médicos, como usuarios externos a la compañía; como así también los responsables internos y administradores de legales, responsables de médicos, y responsables internos y administradores de los siniestros de los ramos anteriormente nombrados, como usuarios internos a la compañía. Desde los perfiles de los liquidadores, auditores médicos, médicos, responsables de médicos y responsables internos de siniestros y administradores de siniestros se pueden consultar y gestionar los siniestros indicados al principio del párrafo; desde los perfiles de los abogados, abogados penalistas y responsables internos legales y administradores de legales se pueden consultar y gestionar los litigios<sup>2</sup> pertenecientes a esos siniestros. Toda la gestión previamente descripta está orientada a los terceros.

Dicho sistema es robusto, consistente, correcto y seguro, está implementado en JAVA utilizando las tecnologías JavaServer Faces (JSF)<sup>3</sup> y RichFaces<sup>4</sup>; se encuentra operativo desde el año 2008 y hasta la fecha sigue respondiendo correctamente. Sin embargo, es incompleto dado que en sus orígenes sólo fue pensado para los siniestros de los terceros y de los ramos nombrados anteriormente, sin tener en cuenta la gestión de los siniestros de Rs. Vs., o también llamados siniestros de “otros ramos”, los cuales son: incendio (ramo 1), transporte (ramo 2), cristales (ramo 5), robo (ramo 6), riesgos varios (ramo 7), responsabilidad civil (ramo 8), seguro técnico (ramo 9), accidentes personales (ramo 12), vida colectiva (ramo 18), vida individual (ramo 19), casco (ramo 22), combinado familiar (ramo 24), e integral de comercio (ramo 25). Ante la existencia y características del sistema de siniestros, denominado “Siniestros”, y su relación con los nuevos requisitos solicitados, se evaluó realizar una ampliación respetando las tecnologías utilizadas. Además, a un mismo liquidador se le pueden derivar

---

<sup>2</sup> Tipos de litigios: juicios, mediaciones, juicios ART, conciliaciones ART, beneficios para litigar sin gastos.

<sup>3</sup> JSF (JavaServer Faces): <http://www.java-serverfaces.org>

<sup>4</sup> RichFaces: <https://richfaces.jboss.org>

siniestros del ramo 4 y 44, y a su vez de Rs. Vs.; así el liquidador podrá gestionar todos sus casos en una sola aplicación, sólo cambiando de perfil.

La ampliación propuesta para desarrollar en mi tesina de grado cuenta con la incorporación de cuatro nuevos perfiles que permitirá gestionar los siniestros de Rs. Vs. que les son derivados a los liquidadores, dicha gestión está orientada a los asegurados; es decir, los estudios van a ponerse en contacto con los asegurados. Los perfiles nuevos que se incorporan son: Liquidador Rs. Vs., Responsable Interno Rs. Vs., Administrador Rs. Vs. y un perfil “Super Administrador” que podría gestionar todos los siniestros de Rs. Vs., todos los perfiles del sistema de siniestros, los siniestros de los ramos 4, 44 y 80, y los litigios. En un principio, este último perfil, va a poder gestionar sólo los siniestros de Rs. Vs. y los perfiles; dado la magnitud del cambio propuesto, se decidió dejar para otra etapa las demás gestiones descriptas anteriormente.

### **1.3 Organización General**

Esta tesina está organizada en 5 capítulos:

Capítulo 1: presentación del tema.

Capítulo 2: presentación de los antecedentes y fundamentos teóricos.

Capítulo 3: presentación del desarrollo realizado.

Capítulo 4: presentación de posibles trabajos futuros.

Capítulo 5: descripción de las conclusiones a las cuales se ha arribado.

# Capítulo 2

---

## **Antecedentes y Fundamentos Teóricos**

### **2.1 Introducción**

Esta tesina se inscribe en el área de ingeniería de software, específicamente en desarrollo web. La problemática principal es la gestión de los siniestros de Rs. Vs. sufridos por los asegurados de la compañía en la cual desempeño mi labor.

En mi rol de profesional de informática, detecté que la compañía se basa en las metodologías tradicionales de la ingeniería de software para llevar a cabo los diferentes desarrollos, por lo cual tomé la iniciativa de realizar el desarrollo de esta tesina orientado a las metodologías ágiles. De igual modo, en el análisis del caso de estudio identifiqué que los requerimientos solicitados por el usuario necesitaban de una mayor elaboración, dado que algunos eran ambiguos y otros estaban incompletos.

Las tecnologías utilizadas en el desarrollo de la presente tesina fueron JSF y RichFaces, siguiendo los lineamientos tecnológicos de la compañía, siendo éstas las tecnologías utilizadas en el actual sistema de siniestros de la compañía aseguradora, como se menciona en el primer capítulo. A su vez, la comunicación con la base de datos se realiza mediante la invocación de funciones y procedimientos pertenecientes a un paquete creado previamente en la base de datos.

### 2.1.1 Definiciones

Como se mencionó anteriormente, esta tesina se va a desarrollar en el lenguaje Java, la página de Java lo define como “un lenguaje de programación y una plataforma informática comercializada por primera vez en 1995 por Sun Microsystems [...] Java es rápido, seguro y fiable. Desde portátiles hasta centros de datos, desde consolas para juegos hasta súper computadoras, desde teléfonos móviles hasta Internet, Java está en todas partes” [1].

JavaServer Faces (JSF), la página oficial la define como “el ‘framework’ de interfaz de usuario (IU) orientado a componentes estándar para la plataforma Java EE. En términos que pueden sonar más familiares, es un ‘framework’ web basado en Java” [2].

La librería RichFaces amplía las capacidades de AJAX<sup>5</sup> de JSF con características avanzadas para el desarrollo de aplicaciones web empresariales [3].

El desarrollo se va a correr sobre una base de datos 12c<sup>6</sup> de Oracle<sup>7</sup>. Dicha base de datos, utiliza SQL (Structured Query Language, en castellano Lenguaje de Consultas Estructurado) como lenguaje de consultas de Base de Datos. Es considerado como el lenguaje estándar de las Bases de Datos relacionales, debido a que la mayoría de los fabricantes de DBMS<sup>8</sup> optaron por este como lenguaje de trabajo [4].

Se va a utilizar SOA (Service-Oriented Computing, o en castellano programación orientada a servicio) para validar la factura cargada por el liquidador contra la AFIP, dicha validación se va a llevar a cabo mediante REST. SOA es un término general que representa una plataforma informática distribuida distinta, es un tipo de arquitectura de software, la cual se basa en la integración de aplicaciones mediante servicios; un servicio es un programa de software que hace que su funcionalidad esté disponible a través de una interfaz técnica publicada, llamada contrato de servicio. REST es uno de los varios medios de implementación que se pueden utilizar para crear servicios, es una definición de arquitectura que nos indica cómo realizar el intercambio y manejo de datos a través de servicios web [5].

---

<sup>5</sup> El término AJAX es un acrónimo de Asynchronous JavaScript + XML, que se puede traducir como "JavaScript asíncrono + XML".

<sup>6</sup> La base de datos 12c de Oracle es una base de datos de alto rendimiento, de tipo empresarial. Oracle lanzó Oracle Database 12c para disponibilidad general el 1 de julio de 2013.

<sup>7</sup> <https://www.oracle.com/index.html>

<sup>8</sup> Data Base Management System (o en castellano, Sistema de Gestión de Bases de Datos).

## 2.1.2 Introducción a las metodologías de desarrollo de software

### 2.1.2.1 Metodologías tradicionales

Las metodologías tradicionales (también denominados modelos de procesos prescriptivos) fueron propuestas originalmente para brindar orden al caos del desarrollo de software. Al principio estas metodologías fueron una buena solución al trabajo de la ingeniería de software, sin embargo, el trabajo de ingeniería de software y el producto que produce permanecen en la “era del caos” [6].

Pressman, a estos modelos, los define como prescriptivos porque: “[...] ellos prescriben un conjunto de elementos de procesos-actividades de ‘framework’, acciones de ingeniería de software, tareas, productos de trabajos, seguro de calidad, y mecanismos de control de cambio por cada proyecto. Cada modelo de proceso también prescribe un flujo de proceso (también llamado flujo de trabajo), es decir, la manera en que los elementos del proceso están relacionados entre sí” [6].

### 2.1.2.2 Metodologías Ágiles

Pressman sostiene: “la ingeniería de software combina una filosofía y un conjunto de pautas de desarrollo. La filosofía alienta la satisfacción del cliente y la entrega incremental temprana del software; equipos de proyecto pequeños y altamente motivados; métodos informales; productos de trabajo de ingeniería de software mínimo; y simplicidad de desarrollo general. El conjunto de pautas enfatiza la entrega sobre el análisis y diseño (aunque estas actividades no se desanimen), y comunicación activa y continua entre desarrolladores y clientes” [6].

Los 12 principios ágiles [6]:

- Nuestra mayor prioridad es satisfacer al cliente a través de una temprana y continua entrega de software valioso. Para que una metodología sea ágil debe haber entregables útiles y funcionando en pocas semanas.
- Recibir los requisitos cambiantes, incluso los tardíos. Se aprovechan los cambios para la ventaja competitiva del cliente.
- Entregar software de trabajo con frecuencia, desde un par de días a un par de meses, con preferencia a la escala de tiempo más corta. El cliente siempre espera ver un programa funcionando, y eso es lo que hay que mostrarle cuanto antes
- Las personas a cargo del negocio (usuarios) y los desarrolladores deben trabajar juntos diariamente durante el proyecto. Si bien el usuario desconoce los conceptos informáticos, es él quien nos puede orientar desde el punto de vista de la necesidad del negocio.

- Construir proyectos en torno a individuos motivados. Hay que darles el entorno y el apoyo que necesitan, y confiarles la ejecución del trabajo. El buen estado de ánimo, el sentirse parte del todo y la disposición del equipo de trabajo son fundamentales en un proyecto de software.
- El método más eficiente y efectivo de comunicar información al equipo de desarrollo y entre sus miembros es la conversación cara a cara. El trabajo en equipo se basa en un buen ambiente de comunicación tanto entre los integrantes del equipo de desarrollo como entre éstos y el usuario.
- El software funcionando es la primera medida del progreso. La cantidad de requerimientos implementados y funcionando es lo que da la claridad y confiabilidad del avance del proyecto.
- Los procesos ágiles promueven el desarrollo sostenible. Los promotores, desarrolladores y usuarios deben ser capaces de mantener un ritmo constante de forma indefinida. Desde el comienzo del desarrollo se deben designar tareas de una forma que se puedan cumplir.
- La atención continua a la excelencia técnica y el buen diseño mejora la agilidad. No sólo se deben satisfacer los requerimientos del usuario, también los aspectos técnicos deben ser buenos.
- La simplicidad, es esencial. Se debe centrar el esfuerzo en lo solicitado por el usuario, de forma simple, sin realizar optimizaciones innecesarias.
- Las mejores arquitecturas, requisitos y diseño emergen de equipos autoorganizados. La responsabilidad es de todos.
- A intervalos regulares, el equipo reflexiona sobre cómo ser más efectivo para la continuación, ajustar y perfeccionar su comportamiento en consecuencia. El equipo de trabajo está siempre dispuesto a cambiar lo que sea necesario para mejorar.

### *2.1.2.3 Comparación entre Metodologías Ágiles y Tradicionales*

A continuación, se representa mediante una tabla un resumen de las diferencias encontradas entre las metodologías tradicionales y ágiles en base a toda la información recopilada de los diferentes libros y sitio web [6][7][8][9][10], y el conocimiento adquirido en la carrera.



<b>TRADICIONALES</b>	<b>ÁGILES</b>
Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo.	Basadas en heurísticas provenientes de prácticas de producción de código.
Cierta resistencia a los cambios.	Especialmente preparadas para cambios durante el proyecto.
Procesos con múltiples controles y numerosas políticas y normas.	Proceso menos controlado, con pocos principios.
Existe un contrato prefijado.	No existe contrato tradicional o al menos es bastante flexible.
El cliente interactúa con el equipo de desarrollo mediante reuniones.	El cliente es parte del equipo de desarrollo.
Grupos grandes y posiblemente distribuidos.	Grupos pequeños (se recomienda menor a 10 integrantes) y trabajando en el mismo sitio.
La arquitectura del software es esencial y se expresa mediante modelos desde el principio.	Menos énfasis en la arquitectura del software. Se va definiendo en el desarrollo.
Existe un compromiso respecto al tiempo de entrega del proyecto. (no siempre se cumple esta meta)	Existe incertidumbre respecto al tiempo de entrega de todo el producto. Lo cierto es que cada poco tiempo hay entrega de producto de valor para el cliente.
Desde un inicio se analiza el proyecto y se establece un costo para el mismo.	Existe incertidumbre respecto al costo del proyecto. Se invierte en las funcionalidades que más valor le dan al cliente y cíclicamente se avanza hasta que se logre, ya sea: el producto deseado o se acabe el presupuesto.
Atención exhaustiva a la documentación.	Solo se genera la documentación que genera valor al cliente y al proyecto.
Se valora especialmente el proceso.	Se valora especialmente al individuo y las interacciones de los mismos.
El éxito del proyecto está dado por el seguimiento del plan.	El éxito del proyecto está dado por la entrega continua de valor y funcionalidad al cliente.
Se generan entregables que requieren mucho tiempo de elaboración.	Se centran en hacer entregables en tiempos cortos con alta calidad incorporada.
La retroalimentación del cliente se produce al final, pudiendo generar insatisfacción.	La retroalimentación del cliente es constante a lo largo del proyecto.

*TABLA 2.1 Comparación entre las Metodologías Tradicionales y Ágiles*

## 2.2 Marco Teórico

### 2.2.1 La propuesta del desarrollo

El sector de Siniestros de la compañía<sup>9</sup> vio la necesidad de tener un sistema orientado a la gestión de los siniestros de Rs. Vs. que le son asignados a los estudios liquidadores externos. Con esta herramienta los estudios tendrían una aplicación web que centralice todos sus casos y le permitiría tener una gestión más ordenada de la información para liquidar, así podrían cerrar el caso para su eventual facturación. A su vez, El Sector dispondría de la información necesaria para realizar un seguimiento de los casos derivados, de este modo podría controlar a los estudios, aplicando penalidades por desfasaje de tiempo en la resolución.

Actualmente El Sector realiza la gestión de los siniestros de Rs. Vs. de la siguiente manera: el Responsable Interno envía un mail al estudio indicando el nuevo caso que se le ha derivado para su resolución, en este mail se le adjunta una copia de la denuncia y de la póliza asociada al siniestro. A partir de ese momento, el estudio procede a realizar la inspección correspondiente, luego envía un informe preliminar y posteriormente el informe final junto con la factura de sus honorarios y gastos. Una vez recibida esta documentación y verificada la correspondencia del liquidador con el siniestro liquidado, el Responsable Interno genera la orden de pago discriminando honorarios y gastos (de haberlos), así luego cargar la factura en el sistema. Todo este proceso descrito anteriormente puede tardar desde una semana hasta varios meses, según la complejidad y la correcta recepción de la documentación requerida al asegurado; como así también, el compromiso del liquidador para realizar la gestión a tiempo.

Para El Sector esta forma de proceder les resulta ineficiente y escasamente productiva, dado que una vez que se le envía el mail con el nuevo siniestro asignado, pierde el detalle de la información como la fecha de contacto con el asegurado, la fecha que se programó la visita con el mismo, si se concretó la visita o se tuvo que reprogramar (ya sea por cuestiones del asegurado o por el propio estudio), si el estudio le solicitó documentación complementaria al asegurado, cuando fue presentada esta documentación, etc. El Sector no puede determinar con precisión si la demora en la resolución de un caso es por la falta de colaboración del asegurado o por la falta de responsabilidad del estudio liquidador.

Asimismo, El Sector no cuenta con referencias estadísticas tales como: cuántos casos se derivaron a un determinado estudio, cuántos casos sin resolver tiene un determinado estudio, cuántos casos se tienen sin contactar con el asegurado, cuántos casos no poseen el informe preliminar, etc.

Ante estos faltantes El Sector propuso el desarrollo de un nuevo sistema de liquidadores de Rs. Vs. que le permita a cada estudio liquidador ingresar mediante un usuario y clave, y así poder visualizar

---

<sup>9</sup> De ahora en más se llamará “El Sector” al sector de Siniestros de la compañía.

rápidamente todos los casos asignados, ingresar al detalle de uno de ellos, contactar con el asegurado, informar la concreción de la visita o postergación de la misma, cargar el informe preliminar correspondiente indicando si le solicitó o no documentación complementaria al asegurado, cargar el informe final con la documentación solicitada (de ser necesario), generar una orden de pago, cargar los datos de su factura, descargar la documentación necesaria del siniestro. Además, el liquidador podrá subir archivos pertinentes al siniestro, ver las estadísticas de sus casos, enviar mensajes al Responsable Interno o Administrador y visualizar las novedades de cada siniestro.

Las tareas descritas en el párrafo anterior también se podrán realizar desde el perfil Responsable Interno Rs. Vs., con la diferencia que este perfil podrá gestionar los casos de varios liquidadores a la vez (esto es según se les fue asignado para su seguimiento); y también se podrá hacer desde el perfil Administrador Rs. Vs., pero abarcando todos los casos existentes. Además, el perfil Administrador Rs. Vs. podrá dar de alta los perfiles Liquidador Rs. Vs. a los estudios liquidadores y Responsable Interno Rs. Vs. a los empleados de la compañía, y gestionar los accesos.

### **2.2.2 La metodología de desarrollo de la compañía**

El sector de sistemas (o cómputos) hace un año adoptó una nueva metodología de trabajo de desarrollo propuesta por la compañía, para la interacción con los diversos sectores que la conforman, la cual se detalla a continuación.

La compañía dispone de dos nuevos sectores a partir de este cambio, el sector Gestión Interna Negocio Tecnología (GINT)<sup>10</sup> y el sector Centro Único de Acceso (CUA)<sup>11</sup>; ambos sectores con conocimientos del negocio en general y de tecnología, siendo el GINT más conocedor del negocio y el CUA más conocedor de tecnología. El GINT es el intermediario del negocio hacia sistemas, tienen más conocimiento de las necesidades reales del negocio; están en continuo contacto con la Alta Gerencia y evalúan las nuevas necesidades planteadas por cada sector. El CUA es el intermediario de sistemas hacia el negocio, tienen más conocimiento de los recursos de sistemas y la disponibilidad del capital humano (analistas y desarrolladores); deciden qué profesional atenderá la nueva necesidad requerida. Ambos sectores están en constante diálogo.

Al principio, cada sector de la compañía “compite” internamente por presentar las necesidades prioritarias, es decir, evalúan todas las necesidades existentes y eligen cuáles presentar para la eventual implementación, dándole un orden prioritario. Estas necesidades son presentadas al GINT por otro sector intermediario, llamado Optimización de Procesos, el cual refina los requerimientos de cada necesidad llevando a cabo reuniones con el sector que presentó la misma. Este sector hace la primera evaluación del impacto que tendría la no realización de dicha necesidad en la compañía,

---

<sup>10</sup> De ahora en más se llamará GINT al sector Gestión Interna Negocio Tecnología.

<sup>11</sup> De ahora en más se llamará CUA al sector Centro Único de Acceso.

determinan si la criticidad planteada es correcta y elaboran un documento llamado “Documento de Alcance de Negocio” (DAN)<sup>12</sup> explicando de forma detallada lo planteado por el sector solicitante.

Luego de creado el DAN, el sector Optimización de Procesos eleva dicho documento al GINT para su análisis. El GINT, sabiendo sobre las necesidades reales de la compañía, evalúa si es relevante la realización de lo descrito en el documento, y le asigna el verdadero orden prioritario de todas las necesidades planteadas por el sector solicitante; por ejemplo, si es el sector de Contaduría, se le da el orden prioritario correspondiente teniendo en cuenta todas las necesidades, planteadas como cambio, que existen en ese momento del sector Contaduría.

Con el DAN y la aprobación del GINT, se pone en conocimiento al CUA sobre la nueva necesidad para que éste le asigne a un analista. El analista lee el DAN y con el acompañamiento, de ser necesario, de un grupo de desarrolladores llevan a cabo una reunión de prefactibilidad. La misma está conformada por el sector de Optimización de Procesos, el GINT, el CUA y el analista para determinar, como dice su palabra, si es factible implementar lo solicitado por el sector solicitante en base a tiempos, costos, y disponibilidad de recursos. También se debate cómo se va a implementar la nueva necesidad planteada, y se despejan dudas surgidas a partir de la primera lectura del DAN.

Una vez concluida la reunión y confirmada la factibilidad del desarrollo, la necesidad es presentada formalmente a Sistemas mediante la generación de un cambio en el sistema Jira<sup>13</sup> de la compañía. Este cambio inicia como una necesidad y se lo representa: NIC-número, siendo la letra N de Necesidad, la letra I de Interna y la última letra pertenece al sector solicitante, siguiendo el ejemplo anterior sería C de Contaduría; siendo NIS<sup>14</sup> para Siniestros, NIA para ART y así para todos los sectores. Esa necesidad es asignada por el CUA al analista que estuvo presente en la reunión de prefactibilidad, con fecha de inicio y fin estimada de análisis; así dar una devolución concreta, y estimar los recursos que se necesitarán y los tiempos que van a emplear cada uno para llevar a cabo el desarrollo.

Concluido el análisis, el analista elabora un documento denominado “Documento de Alcance Técnico” (DAT)<sup>15</sup>, este documento se escribe en términos que el sector de Optimización de Recursos y el sector solicitante entiendan, es decir en términos poco técnicos. En él se escribe lo que se va a implementar de los requerimientos planteados en el DAN, pueden ser todos o algunos pueden quedar excluidos; si el sector solicitante está de acuerdo con lo resuelto, se cambia la necesidad a Requerimiento Funcional de Cambio (RFC)<sup>16</sup>.

Tanto el DAN como el DAT son contratos que se acuerdan al aceptar su contenido, que se deben respetar a rajatabla. El DAN es el contrato en representación del sector solicitante y el DAT es el contrato en representación del sector de Sistemas, siendo el DAT el más determinante dado que en éste se define qué se va a implementar y qué no.

---

<sup>12</sup> De ahora en más se llamará DAN al Documento de Alcance de Negocio.

<sup>13</sup> Jira Software es la herramienta que dispone la compañía para que los usuarios de los diferentes sectores puedan solicitar requerimientos, incidentes y cambios a sistemas. Más información disponible en <https://www.atlassian.com/software/jira>.

<sup>14</sup> De ahora en más se llamará NIS a la Necesidad Interna de Siniestros.

<sup>15</sup> De ahora en más se llamará DAT al Documento de Alcance Técnico.

<sup>16</sup> De ahora en más se llamará RFC al Requerimiento Funcional de Cambio.

Luego del común acuerdo de ambas partes, el analista crea las subtareas correspondientes para cumplir con lo descrito en el DAT, debe ser lo más específico posible para que el desarrollador sepa que se debe implementar; cada subtarea es implementada por un único desarrollador, pudiendo un mismo desarrollador implementar una o más subtareas de un cambio en particular. El proceso descrito se esquematiza en la figura 2.1.

La gestión de tareas y subtareas se hace mediante diagramas de Gantt<sup>17</sup>; así el CUA sabe cuándo arranca y cuándo termina una subtarea cada desarrollador, y qué analista está disponible o está analizando alguna necesidad. De esta forma sabe qué “recursos” están disponible y cuáles no.

La interacción entre el desarrollador y el negocio es nula o casi nula; si surgen dudas durante la etapa de desarrollo, el desarrollador debe notificarlo al analista para que éste a su vez se comunique con el CUA, para luego tratarlo con el GINT y, de ser necesario, convocar una nueva reunión con el sector Optimización de Procesos para tratar las dudas que fueron surgiendo.

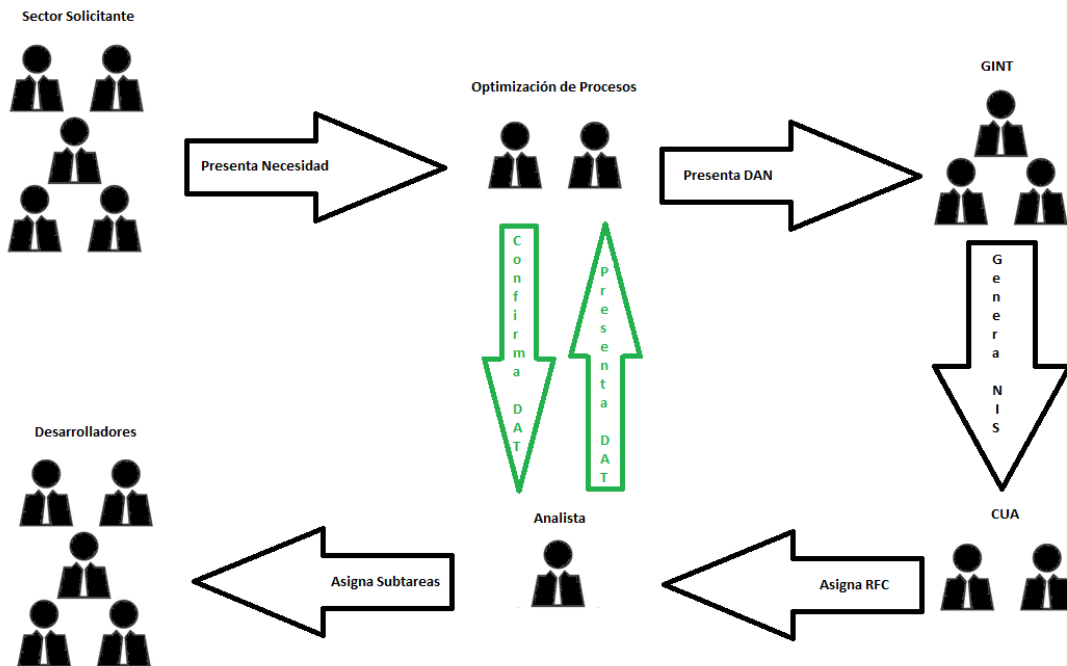


FIGURA 2.1 Gestión de un Nuevo Cambio, ilustración propia

La finalización del cambio es cuando se finalizan todas las subtareas pertenecientes a éste. Cuando esto ocurre, se pasa a TEST<sup>18</sup> la versión finalizada del cambio así el sector de Optimización de Procesos y el sector solicitante, pueden realizar las pruebas pertinentes. El analista, al pasar el cambio

<sup>17</sup> El diagrama (o gráfico) de Gantt es la herramienta utilizada para calendarizar el proyecto.

<sup>18</sup> Ambiente para realizar pruebas de nuevos desarrollos informáticos.

a TEST, realiza un documento denominado “Bitácora de Testeo”, donde se detalla paso a paso las pruebas a realizar. Los sectores involucrados en las pruebas siguen los pasos descritos en la bitácora para realizar las pruebas propuestas. Cabe destacar que el sector de Sistemas no posee personal para realizar las pruebas.

Finalizada la etapa de pruebas, el sector solicitante da su veredicto del sistema implementado; si se producen errores o hay alguna función que no cumple con lo descrito en el DAT, el sector de Optimización de Procesos los comunica al GINT para que éste cree subtareas de error denominadas “Tareas de Bug”, así el CUA asigna las tareas de bugs a los desarrolladores responsables de solucionarlas. Esto se hace respetando la disponibilidad de los profesionales, es decir, si el desarrollador empezó con una subtaska de otro cambio más prioritario, primero atenderá la de mayor prioridad; de igual modo, si la tarea de bug es más prioritaria que la subtaska de otro cambio, este atenderá la tarea de bug dejando en “Espera” la subtaska del otro cambio.

En principio, si el sistema es correcto se da por finalizado el cambio con el pasaje a producción PROD<sup>19</sup>, por definición un sistema es correcto si cumple con los requerimientos establecidos. Si el sector solicitante, ya sea a mitad del desarrollo o en la etapa de TEST, se percata que el sistema no está haciendo lo que se esperaba por una mala definición de los requerimientos, por ejemplo, no se puede cambiar el DAN y el desarrollo sigue su cauce hasta la conclusión del cambio. Si se quiere corregir los requerimientos mal definidos, se debe solicitar como otra necesidad del sector solicitante; lo que implica empezar desde el comienzo todo el circuito anteriormente mencionado. Quedando a criterio del sector solicitante darle más prioridad a la nueva necesidad para su pronta realización, así evitar publicar un sistema no correcto a PROD.

### 2.2.3 Metodología de desarrollo implementada

Si bien me adapté a la forma de trabajar de la compañía, traté de hacer un desarrollo orientado a una metodología ágil. Cuando el Sector de Optimización de Procesos propuso la reunión de prefactibilidad, con el DAN hecho, dejó en claro que El Sector no tenía las ideas claras de lo que estaban solicitando y que era necesario debatir internamente antes de convocar a dicha reunión, dado que existían definiciones que no estaban del todo claras y no eran concretas.

Si bien, a grandes rasgos, era factible lo que se estaba solicitando, los orienté a definir de una mejor manera el DAN. Les hice preguntas que surgieron a partir de la reunión, preguntas sobre cada perfil, y su interacción con el sistema y entre ellos, con el objetivo de lograr definir mejor el DAN. A su vez, algunas de las definiciones abarcaban conceptos muy amplios como por ejemplo “el sistema debía contar con un apartado de estadísticas”, en esos casos los motivé a que explicarían en detalle lo que

---

<sup>19</sup> Publicación en línea de un desarrollo informático.

requerían; siguiendo el ejemplo, qué tipos de estadísticas eran necesarias, cómo mostrarlas para que facilitar su interpretación y qué perfiles podrían acceder a las mismas.

Todas estas definiciones las fui analizando juntos con El Sector mientras confeccionaban el DAN, con el objetivo de orientarlos en sus necesidades. También les propuse la inclusión de un perfil “Super Administrador” para realizar la administración de usuarios; la idea predominante en el equipo de El Sector era que el perfil “Administrador Rs. Vs.” realice el alta de nuevos usuarios de Rs. Vs. y que cuente con permisos para blanquear sus claves. Les planteé la posibilidad de crear este nuevo perfil (Super Administrador) que gestione los accesos de todos los perfiles que existen actualmente en el sistema de siniestros. Esta sugerencia fue aceptada dado que resultó fiable teniendo en cuenta que actualmente, cada vez que se necesita dar de alta un nuevo usuario con perfil para el sistema de siniestros, es necesario realizar un requerimiento escrito completando un formulario de permisos y creando un requerimiento en el sistema Jira. Disponer de este nuevo perfil de usuario evitaría contactar al área de Sistemas cada vez que se necesita dar de alta un nuevo usuario en este sistema, optimizando la gestión y ganando tiempo administrativo.

Durante el desarrollo, al margen de la metodología de la compañía, realice demos<sup>20</sup> al finalizar cada apartado para mostrarle al sector de Optimización de Procesos los avances logrados; al hacer esto me fueron orientando y proponiendo mejoras antes de seguir con el siguiente apartado<sup>21</sup>, desde la disposición de los datos a cuáles datos mejor no mostrar y cuáles sí. Además, en las demos quedaron expuestas fallas por cada nueva funcionalidad implementada, facilitando su corrección rápidamente antes de pasar todo el desarrollo a TEST, considerando que es más trabajoso encontrar un error, en esta última instancia, con todo el desarrollo avanzado. En estas demos realizadas también se acordó con el sector de Optimización de Procesos dividir el desarrollo en tres etapas, en consecuencia, dos partes de toda la funcionalidad acordada al principio del análisis se dispondría en producción antes de finalizar todo el desarrollo. También le di la posibilidad al sector de Optimización de Procesos incorporar nuevos requerimientos omitidos en la etapa de confección del DAN, esto se acordó con el CUA y el GINT extendiendo los plazos de entrega final del desarrollo.

En conclusión, esta nueva metodología de desarrollo fue muy bien recibida por el sector de Optimización de Procesos resultando más eficiente que la utilizada habitualmente por la compañía, dado que les permitió participar de los avances del desarrollo sin tener que esperar hasta el final (normalmente meses), pudiendo incorporar dinámicamente nuevas funcionalidades sin la necesidad de pasar por todo el proceso de gestión de cambios. A su vez el proceso de testeó fue más rápido por la temprana detección de errores y se obtuvo un producto terminado (con las funcionalidades más importantes) antes de lo acordado.

---

<sup>20</sup> Una demo es un programa informático de demostración, es una versión reducida en prestaciones de un programa para poder utilizarlo.

<sup>21</sup> Cada apartado se explicará en detalle en el capítulo 3 de esta tesina.

## 2.2.4 Tecnologías utilizadas

En el presente apartado, se hará una breve descripción de las tecnologías empleadas en la ampliación del sistema. La mayoría de ellas, como se comentó al principio, fueron en base a las utilizadas en el sistema de siniestros. Así mismo, fue necesario adaptar el sistema a la nueva forma de persistir los datos en la base de datos; esto es mediante la invocación a funciones y procedimientos dentro de paquetes en común con otros sistemas de la compañía, estas herramientas se explican a continuación.

### 2.2.4.1 Java Server Faces

La compañía tomó como tecnología de desarrollo del back end JAVA, dadas sus características de portabilidad, contabilidad y seguridad, y eligió Java Server Faces (JSF) para el desarrollo del sistema de siniestros en sus comienzos porque era una herramienta segura y vigente en el año que se desarrolló la aplicación.

El sitio web oficial de JSF afirma: “JSF tiene dos funciones principales” [2]:

- La primera es generar una interfaz de usuario, generalmente una respuesta HTML que se sirve a un navegador y se ve como una página web. Esta interfaz de usuario está representada en el servidor por un árbol de componentes. Hay una asignación de 1 a 1 entre elementos en el árbol de componentes y elementos en la interfaz de usuario. La interfaz de usuario real se genera cuando el árbol de componentes se codifica (o representa). Esta separación entre el árbol de componentes y la interfaz de usuario permite que JSF admita diferentes lenguajes de marcado (HTML frente a XUL) o entornos de navegador alternativos.
- La segunda función de JSF es responder a los eventos generados por el usuario en la página invocando oyentes del lado del servidor, seguido de la generación de otra interfaz de usuario (por ejemplo, página web) o una actualización de la interfaz de usuario ya mostrada (posiblemente a través de Ajax). En este sentido, se dice que JSF es un framework web basado en eventos.

### 2.2.4.2 RichFaces

RichFaces es una librería de código abierto desarrollado por JBoss, la cual permite peticiones Ajax a la página. En el comienzo del desarrollo del sistema de siniestros se consideró eficiente utilizar esta librería, y en el módulo de siniestros de Rs. Vs., que corresponde a esta tesina, el acuerdo con el sector de Sistemas fue alinearse con esta tecnología.



JBoss señala las ventajas que provee al desarrollo de aplicaciones web el uso de RichFaces [3]:

- Crear vistas complejas de aplicaciones utilizando componentes listos para usar. La biblioteca de interfaz de usuario de RichFaces contiene componentes para agregar características interactivas ricas a las aplicaciones JSF. Extiende el marco de RichFaces para incluir un gran conjunto de componentes habilitados para Ajax que vienen con un amplio soporte de “skinning”. Además, el marco de RichFaces está diseñado para usarse sin problemas con otras bibliotecas de terceros en la misma página, por lo que tiene más opciones para desarrollar aplicaciones.
- Poder escribir sus propios componentes enriquecidos personalizados con soporte Ajax incorporado. El Kit de desarrollo de componentes, utilizado para la creación de la biblioteca de interfaz de usuario de RichFaces, incluye una instalación de generación de código y una instalación de plantillas que utiliza la sintaxis XHTML (lenguaje de marcado de hipertexto extendido).
- Generar recursos binarios sobre la marcha. Las extensiones a las instalaciones de manejo de recursos JSF 2 pueden generar imágenes, sonidos, hojas de cálculo de Microsoft Excel y más durante el tiempo de ejecución.
- Crear una interfaz de usuario moderna y rica con tecnología de “skinning”. RichFaces proporciona una función de “skinning” que le permite definir y administrar diferentes esquemas de color y otros parámetros de la apariencia. Es posible acceder a los parámetros del “skin” desde el código de la página durante el tiempo de ejecución. RichFaces viene empaquetado con una serie de máscaras para comenzar, pero también puede crear fácilmente sus propias máscaras personalizadas.

#### 2.2.4.3 Base de datos Oracle

Una de las principales características de la base de datos Oracle 12c es la optimización automática de datos, la cual le permite realizar una comprensión inteligente a partir de las consultas realizadas a ésta. Otra característica que posee es la seguridad de los datos, asignando privilegios y roles a determinados usuarios que interactúan con ella.

##### 2.2.4.3.1 Triggers

Una forma rápida de acceder a los datos para persistirlos es mediante Triggers o disparadores, éstos ejecutan una acción antes o después de insertar o actualizar un registro de una tabla. Se puede indicar que inserte o modifique un registro en una tabla a partir de la inserción o actualización de un registro de otra.

La ventaja de utilizar triggers es evitar modificar parte del código ya programado, esto sería cuando se tiene un sistema funcionando y se desea incorporar una nueva funcionalidad a éste.

#### 2.3.4.3.2 Procedures

Un procedure o procedimiento es un bloque PL/SQL almacenado en la base de datos que puede ejecutarse desde aplicaciones u otros procedimientos dentro de la base de datos.

Una política fundamental en la compañía para persistir los datos en la base de datos es utilizar procedimientos conocidos, es decir, cuando es necesario hacer una inserción o una actualización en la base se debe investigar si no existe un procedimiento que ya realiza la acción esperada; de esta forma se evita generar procedimientos repetidos.

#### 2.3.4.3.3 Functions

Igual al procedimiento, una function o función es un bloque de PL/SQL, pero devuelve un valor. Todos sus parámetros son de entrada.

Siguiendo la misma política de la compañía por los procedimientos, se utilizan funciones conocidas para obtener datos de la base de datos.

#### 2.3.4.3.4 Packages

Un package o paquete es un objeto de un esquema que agrupa elementos PL/SQL relacionados, entre los cuales se encuentran las funciones y los procedimientos.

Los paquetes se dividen en dos partes: una cabecera o especificación y un cuerpo (body). La cabecera permite describir el contenido del paquete, y conocer el nombre y los parámetros de llamada de las funciones y procedimientos; mientras que el código se incluye en el cuerpo del paquete. Esta separación de especificaciones y código permite utilizar un paquete sin que el usuario pueda visualizar el código.

Con el fin de organizar la interacción con la base de datos la compañía crea y utiliza paquetes específicos para cada sector, esto es mediante la creación de funciones y procedimientos comunes a un sector de la compañía dentro de un mismo paquete. Por ejemplo, toda la interacción del área de Contaduría de la compañía se realizará dentro del paquete “contaduría”.

## 2.3 Estado del Arte

### 2.3.1 Comparación de la metodología de la compañía con las metodologías existentes

La metodología de desarrollo de software adoptada por la compañía, descrita anteriormente, se aproxima a una metodología tradicional de software. A continuación, se describen sintéticamente las dos metodologías tradicionales que más se asemejan con la metodología empleada por la compañía.

#### 2.3.1.1 Modelo en Cascada

Fue presentado por Royce<sup>22</sup> en 1970, se basa en el desarrollo de etapas formando una cascada. En el mismo, cada fase no va a ser comenzada hasta que previamente haya finalizado la anterior. El problema más grande con este tipo de modelo, es el congelamiento temprano de los requerimientos, los cuales al presentar un cambio indican un cambio en el desarrollo de software. Para evitar esto, hay que eludir cambios en los mismos cuando nos encontramos en pleno desarrollo; lo cual es muy inusual. Ni los usuarios conocen sobre el desarrollo de software ni los desarrolladores conocen todo sobre el negocio, o como lo expresan Pfleeger y Atlee: “Ni los usuarios ni los desarrolladores saben todas las claves que afectan el resultado deseado [...]” [7].

Como lo explica Sommerville: “Debido a la cascada de una fase a otra, dicho modelo se conoce como modelo en cascada o como ciclo de vida del software [...]” [8].

De acuerdo a Sommerville [8], las etapas de este modelo se explican a continuación:

- **Análisis y definición de requerimientos.** Los servicios, restricciones y metas del sistema se definen a partir de las consultas con los usuarios. Entonces, se definen en detalle y sirven como una especificación del sistema. Su propósito es analizar las necesidades de la empresa y definir la creación o perfeccionamiento de un sistema.
- **Diseño del sistema y del software.** El proceso de diseño del sistema divide los requerimientos en sistemas hardware o software. Establece una arquitectura completa del sistema. El diseño del software identifica y describe las abstracciones fundamentales del sistema software y sus

---

<sup>22</sup> Winston W. Royce (15 de agosto de 1929 – 7 de junio de 1995).

relaciones. Consiste en el estudio del sistema actual de la empresa y establecer prioridades en las necesidades del negocio para el desarrollo del nuevo sistema.

- **Implementación y prueba de unidades.** Durante esta etapa, el diseño del software se lleva a cabo como un conjunto o unidades de programas. La prueba de unidades implica verificar que cada una cumpla su especificación. Si el diseño se lleva a cabo de una forma detallada, la implementación es más fácil de realizar. En esta fase se desarrolla lo especificado en los requerimientos.
- **Integración y prueba del sistema.** Los programas o las unidades individuales de programas se integran y prueban como un sistema completo para asegurar que se cumplan los requerimientos del software. Después de las pruebas, el sistema software se entrega al cliente. El resultado de esta fase es el sistema listo para realizar las pruebas pertinentes, buscando que los resultados de estas sean de acuerdo con los requerimientos preestablecidos.
- **Funcionamiento y mantenimiento.** Por lo general (aunque no necesariamente), ésta es la fase más larga del ciclo de vida. El sistema se instala y se pone en funcionamiento práctico. El mantenimiento implica corregir errores no descubiertos en las etapas anteriores del ciclo de vida, mejorar la implementación de las unidades del sistema y resaltar los servicios del sistema una vez que se descubren nuevos requerimientos.

La figura 2.2 ilustra esquemáticamente las etapas del diseño en cascada.

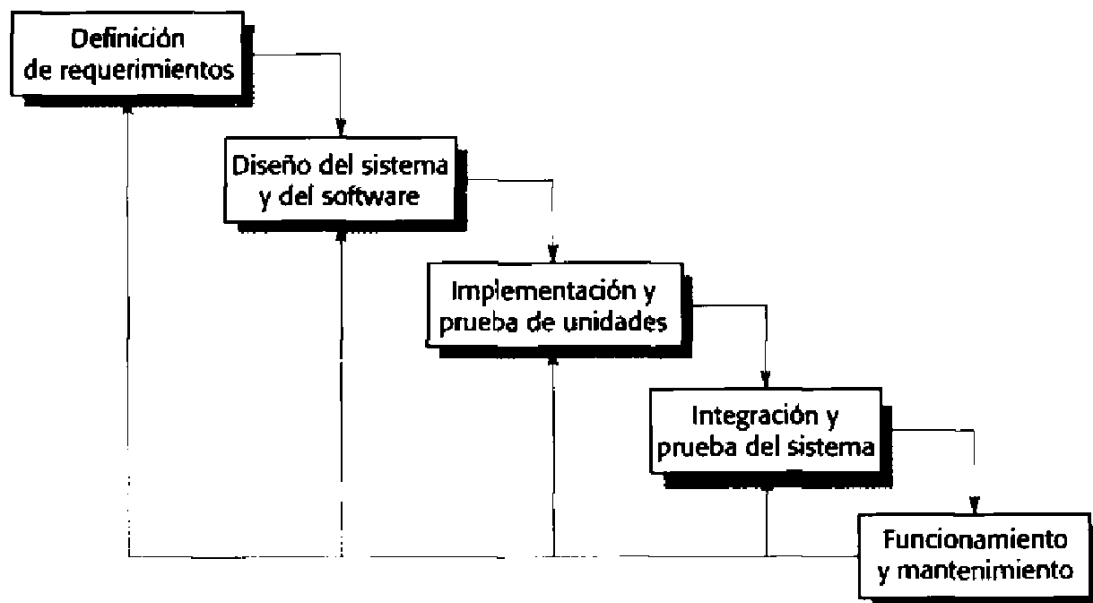


FIGURA 2.2 Modelo en Cascada, ilustración de Sommerville [8]

### 2.3.1.2 Modelo en V

El modelo en V es una variante de la representación del modelo en cascada. Según Pressman: “A medida que el equipo de software avanza hacia abajo desde el lado izquierdo de la V, los requerimientos básicos del problema mejoran hacia representaciones técnicas cada vez más detalladas del problema y de su solución. Una vez que se ha generado el código, el equipo sube por el lado derecho de la V, y en esencia ejecuta una serie de pruebas (acciones para asegurar la calidad) que validan cada uno de los modelos creados cuando el equipo fue hacia abajo por el lado izquierdo” [6].

A grandes rasgos no hay diferencias entre el modelo en cascada y el modelo en V. Este último proporciona una forma de visualizar el modo de aplicación de las acciones de verificación y validación al trabajo de ingeniería inicial [6].

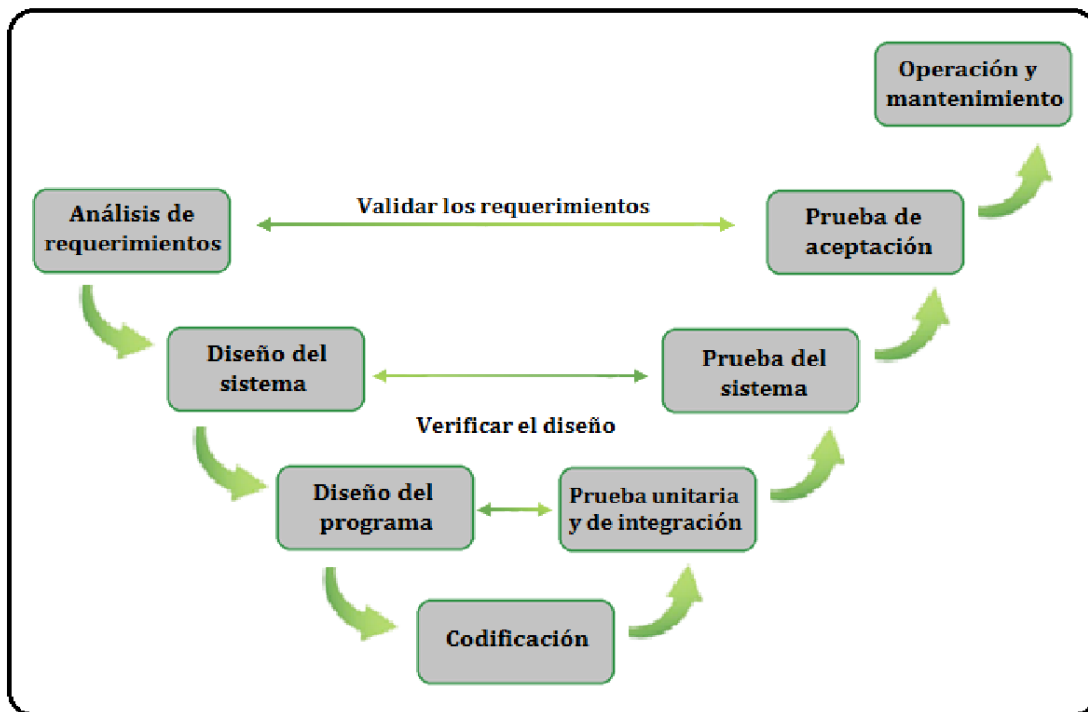


FIGURA 2.3 Modelo en V, ilustración de Pfleeger y Atlee [7]

Como se muestra en la figura 2.3, las pruebas unitarias y de integración deberían usarse para verificar el diseño del programa. Es decir, durante esta etapa los desarrolladores y el equipo de prueba se deben asegurar que se hayan implementado correctamente todos los aspectos del diseño del programa. De esta forma, también en la prueba del sistema se debería verificar el diseño del sistema, asegurando la correcta implementación de todo el sistema. Y por último en la prueba de aceptación, realizada por el cliente, validar los requerimientos solicitados al principio; en esta instancia se chequea que todos éstos se hayan implementado completamente antes que el sistema sea aceptado. Ante una prueba

errónea, el lado derecho se “linkea” con el lado izquierdo; así el lado izquierdo se puede volver a ejecutar para solucionar y mejorar los requerimientos, el diseño, y el código antes de seguir con el siguiente paso del lado derecho [7].

De acuerdo a lo descrito en el apartado del marco teórico, la metodología de desarrollo de software la compañía se aproxima al modelo en V.

### 2.3.2 Comparación de la metodología implementada con las metodologías existentes

Como describí en el apartado del marco teórico se decidió impulsar un cambio en la metodología de desarrollo de este proyecto aproximándose a las metodologías ágiles, respetando los 12 principios ágiles descritos al principio de este capítulo. Se podría considerar que no se cumplió con el principio de simplicidad, dado que propuse un nuevo perfil agregando funcionalidad adicional a la propuesta por El Negocio. Sin embargo, se consideró dado que aporta simplicidad a la gestión de usuarios en general permitiendo que un usuario administrador pueda gestionar la baja y alta de nuevos usuarios de todos los usuarios del sistema de siniestros.

A continuación, se describen las dos metodologías ágiles que más se asemejan con la metodología implementada en este proyecto.

#### 2.3.2.1 Programación Extrema (XP)

Aunque tiene sus orígenes en 1996 de la mano de Kent Beck<sup>23</sup>, las primeras actividades con las ideas y los métodos asociados a XP ocurrieron al final de la década de 1980 [6][8].

Pfleeger afirma: “Los defensores de XP enfatizan 4 características de agilidad: comunicación, simplicidad, coraje y ‘feedback’. La ‘comunicación’ implica el intercambio continuo entre clientes y desarrolladores. La ‘simplicidad’ alienta a los desarrolladores a seleccionar el diseño o implementación más simple para satisfacer las necesidades de sus clientes. El ‘coraje’ es descrito por los creadores de XP como el compromiso para brindar funcionalidad temprana y frecuente. El ciclo de ‘feedback’ se integra en las diversas actividades durante el proceso de desarrollo. [...]” [7].

---

<sup>23</sup> Kent Beck (n. 31 de marzo de 1961).

Pressman señala, lo dicho en el párrafo anterior, como “valores XP” y agrega al “respeto” como quinto valor; se inculca respeto entre los miembros del equipo, entre otros participantes, e indirectamente para el software en sí mismo [6].

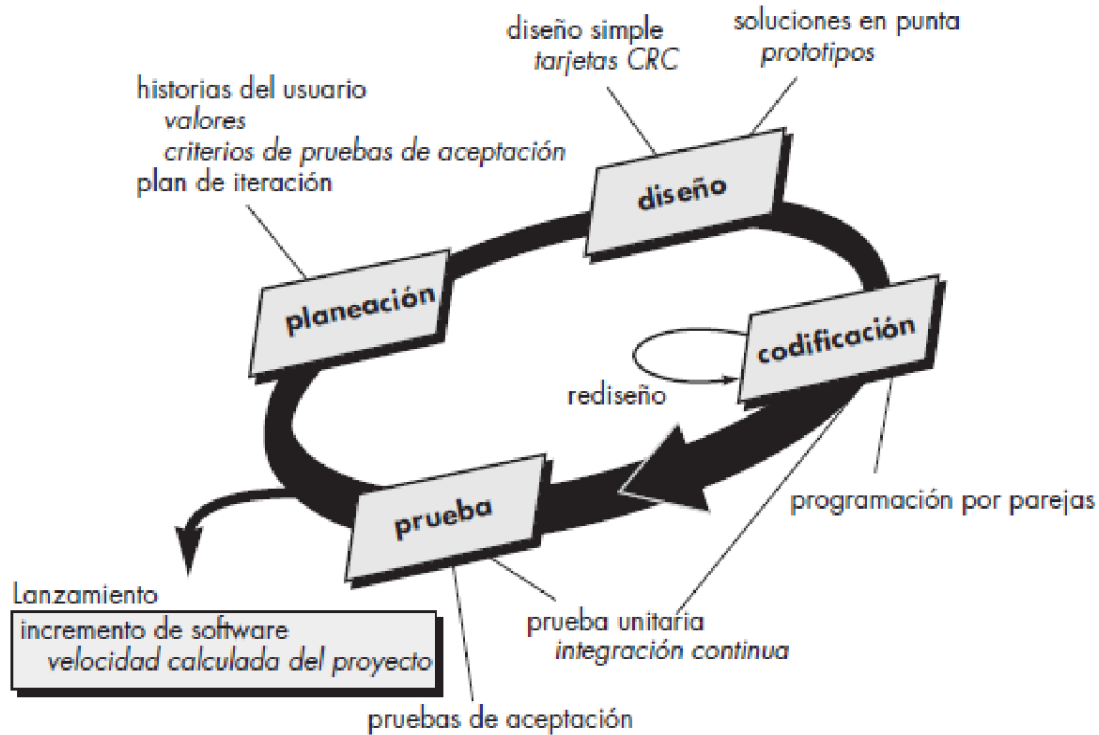


FIGURA 2.4 Programación Extrema, Ilustración de Pressman [6]

En la figura 2.4 se muestran las actividades involucradas en la metodología de XP, son las siguientes:

- Planeación (también llamada juego de planeación): comienza escuchando, esto significa conocer los requerimientos, permitiéndole a los analistas entender el contexto del negocio y los resultados y características principales que se requieren. En respuesta a esto se crean las denominadas historias del usuario<sup>24</sup> que describen los resultados esperados, características y funcionalidad del software que se va a desarrollar. Cada historia es escrita por el cliente y colocada en una tarjeta indexada; éste le asigna una prioridad a la historia en base a la importancia para el negocio. Luego los miembros del equipo de desarrollo evalúan cada historia y le asignan un costo. Si se estima que la historia tomaría más de tres semanas de desarrollo, se le pide al cliente que la divida en historias más chicas. Los clientes y desarrolladores trabajan juntos para decidir qué historias se trabajarán para la

<sup>24</sup> Las historias de usuario son una forma rápida de administrar los requisitos de los usuarios, permiten responder rápidamente a los requisitos cambiantes.

siguiente entrega. Cuando se llegue a un compromiso sobre la entrega se optará por: todas las historias se implementarán de inmediato, las historias con más valor se implementarán en primer lugar, o las más riesgosas se implementarán primero. Después de la primera entrega del proyecto, el equipo de desarrollo calcula la “velocidad” de éste, entendiéndose la velocidad del proyecto como el número de historias que se implementaron durante la primera entrega.

A medida que se avanza con el proyecto, el cliente puede ir agregando historias, cambiar el valor de una existente, dividirla o eliminarla.

- **Diseño:** sigue rigurosamente el principio “mantén lo sencillo”, dado que un diseño sencillo siempre se prefiere ante una representación más compleja. Se desalienta el diseño de funcionalidad adicional porque el desarrollador supone que se solicitará después. XP estimula el uso de tarjetas CRC (Clase-Responsabilidad-Colaborador), son el único producto del trabajo de diseño que se genera como parte del proceso XP. Si en el diseño de una historia se encuentra un problema de diseño difícil, XP recomienda la creación inmediata de un prototipo operativo de esa porción del diseño. El objetivo es disminuir el riesgo cuando se inicie la implementación verdadera y validar las estimaciones originales para la historia que contiene el problema de diseño. XP estimula el rediseño. El objetivo del rediseño es controlar las modificaciones que pueden y deben darse en el diseño. Sin embargo, el esfuerzo que requiere el rediseño aumenta de forma notable con la aplicación. Un concepto central en XP es que el diseño ocurre antes y después que comienza la codificación.
- **Codificación:** finalizadas las historias y el diseño preliminar, el equipo no inicia la codificación, sino que desarrolla una serie de pruebas unitarias a cada una de las historias que se van a incluir en la entrega en curso. Una vez creada la prueba unitaria, el desarrollador está más capacitado para centrarse en lo que se tiene que desarrollar para pasar la prueba. Un concepto clave para la codificación es la programación en pareja. XP recomienda que dos personas trabajen juntas en una estación de trabajo con el objeto de crear código para una historia. En la práctica cada persona adopta un papel diferente; por ejemplo, una de ellas tal vez piense en los detalles del código, mientras la otra se asegura que se sigan con los estándares de codificación. A medida que las parejas de programadores terminan su trabajo, el código que desarrollan se integra con el trabajo de los demás. Esta “integración continua” ayuda a evitar los problemas de compatibilidad e interfaces.
- **Pruebas:** la creación de pruebas unitarias antes que comience la codificación es clave en XP. Estas pruebas, se deben implementar de tal forma que permitan ser automatizadas, pudiendo así ejecutarse repetidas veces y con facilidad. Wells dice: “Corregir pequeños problemas cada cierto número de horas toma menos tiempo que resolver problemas enormes justo antes del plazo final.” Las pruebas de aceptación o pruebas del cliente son especificadas por el cliente y se centran en las características y funcionalidad general del sistema que son visibles y revisables por el cliente.



Retomando el trabajo de esta tesina, en relación a la metodología empleada, a diferencia de lo planteado por XP sobre mantener un diseño sencillo y desalentar funcionalidad adicional, se propuso una nueva funcionalidad incorporando el perfil de “Super Administrador” para que pueda gestionar los nuevos perfiles creados y los existentes en el Sistema de Siniestros, tal como se mencionó al comienzo de este apartado. Esta decisión se tomó con un sentido anticipatorio de los hechos y llevar a cabo un desarrollo íntegro a las necesidades de El Sector. La recomendación de XP de trabajar en pareja, no se pudo llevar a cabo dado que la empresa asigna subtarefas individuales a cada desarrollador. Sin embargo, en mi rol de desarrollador les pedí consejos y opiniones a mis compañeros a lo largo del desarrollo, esta técnica resultó sumamente útil para detectar errores de codificación no advertidos y poder encarar ciertas funcionalidades. Tampoco hubo pruebas unitarias, solamente pruebas del cliente, dado que El Sector y el sector de Optimización de Procesos realizaron las pruebas.

### 2.3.2.2 Método de Desarrollo de Sistema Dinámico (MDSB)

Pressman afirma: “La filosofía MDSB está tomada de una versión modificada de la regla de Pareto: 80 por ciento de una aplicación puede entregarse en 20 por ciento del tiempo que tomaría entregarla completa (100 por ciento)” [6].

MDSB es un proceso iterativo de software en el que cada iteración sigue la regla de hacer el trabajo suficiente para cada incremento con el objetivo de facilitar el paso siguiente. Los detalles se terminarán más tarde [6].

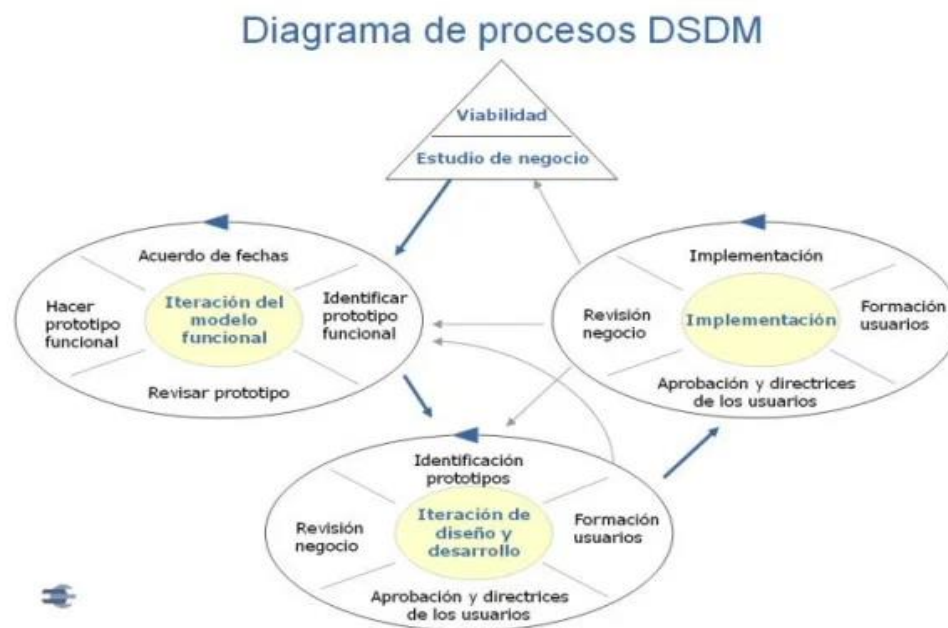


FIGURA 2.5 Método de Desarrollo de Sistema Dinámico, ilustración de Pérez y Ariza [9]

La figura 2.5 muestra las fases y etapas de MDSD, el cual se divide en 3 grandes etapas, según lo describe Pérez y Ariza [9]: “pre-proyecto”, “ciclo de vida del proyecto” y “post-proyecto”.

- Fase 1: pre-proyecto, se define que se pretende conseguir y quienes van a participar para conseguirlo.
- Fase 2: ciclo de vida del proyecto, se divide en 5 etapas:
  - Estudio de viabilidad: se evalúa si la aplicación es viable, teniendo en cuenta los requisitos básicos del negocio y sus restricciones asociadas.
  - Estudio de la empresa: establece los requerimientos e información funcionales que permitirán la aplicación para dar valor al negocio.
  - Iteración del modelo funcional: produce un conjunto de prototipos incrementales que demuestran al cliente la funcionalidad.
  - Diseño e iteración de la estructura: se revisa los prototipos construidos durante la iteración del modelo funcional; la iteración de construcción de diseño ocurre de manera concurrente, en algunas ocasiones.
  - Implementación: se entrega un incremento del software con el fin de capacitar al usuario y evaluar detalladamente los documentos del sistema.
- Fase 3: post-proyecto, se asegura que el sistema operativo acepte de manera eficaz y segura al proyecto.

Hay 9 principios subyacentes al MDSD. Estos principios forman los pilares del desarrollo de software basado en MDSD [9]:

- Involucrar al cliente es la clave para llevar un proyecto eficiente y efectivo, donde ambos, cliente y desarrolladores, comparten un entorno de trabajo para que las decisiones puedan ser tomadas con precisión.
- El equipo del proyecto debe tener el poder para tomar decisiones que son importantes para el progreso del proyecto, sin esperar aprobación de niveles superiores.
- MDSD se centra en la entrega frecuente de productos, asumiendo que entregar algo temprano es siempre mejor que entregar todo al final. Al entregar el producto frecuentemente desde una etapa temprana del proyecto, el producto puede ser verificado y revisado allí donde la documentación de registro y revisión puede ser tenida en cuenta en la siguiente fase o iteración.
- El principal criterio de aceptación de entregables en MDSD reside en entregar un sistema que satisfice las actuales necesidades de negocio. No está dirigida tanto a proporcionar un sistema perfecto que resuelva todas las necesidades posibles del negocio, sino que centra sus esfuerzos en aquellas funcionalidades críticas para alcanzar las metas establecidas en el proyecto/negocio.
- El desarrollo es iterativo e incremental, guiado por la realimentación de los usuarios para converger en una solución de negocio precisa.

- Todos los cambios durante el desarrollo son reversibles.
- Los requerimientos se establecen a un nivel general, antes de que comience el proyecto.
- Las pruebas son realizadas durante todo el ciclo vital del proyecto. Esto tiene que hacerse para evitar un caro coste extraordinario en arreglos y mantenimiento del sistema después de la entrega.
- La comunicación y cooperación entre todas las partes interesadas en el proyecto es un prerequisite fundamental.

La metodología de desarrollo ágil empleada para esta tesina se asemeja a la metodología DSDM, dado que se llevaron a cabo varios principios de la misma; entre los más destacables se encuentran, la integración del sector de Optimización de Procesos como parte del desarrollo, en consecuencia, las decisiones fueron tomadas de una mejor forma, y a la entrega frecuente de demos; de esta forma se pudo verificar y revisar parte del desarrollo por etapas y evitar solucionar varios errores juntos al finalizar las pruebas. A su vez se hizo una entrega a la mitad del tiempo estimado, poniendo en PROD las funcionalidades más críticas.

### 3.3.3 Antecedentes

En la actualidad, en el mercado del sector de seguros de nuestro país, no hay disponibles sistemas para la administración de siniestros de Rs. Vs. Esta investigación le realizó el sector Optimización de Procesos en noviembre del 2019 con el objetivo de contar con algún sistema similar como guía. Es por ello que se decidió desarrollar un sistema que atienda esta demanda interna de la compañía en la que me desempeño. Este sistema fue pensado por y para la compañía, así ganar la confianza de los estudios liquidadores del mercado ofreciéndoles un sistema que resguarde la documentación pertinente a la liquidación de un siniestro de Rs. Vs. y le permita disponer de ésta en el momento que se desee.

Actualmente la compañía posee un sistema de uso interno, programado en Oracle Forms<sup>25</sup> con PL/SQL<sup>26</sup>. Dicho sistema sólo es accedido por los empleados internos de la compañía, y posee toda la funcionalidad requerida por todos los sectores de la misma. En este sistema los empleados pueden gestionar los siniestros de Rs. Vs., registrando el estudio asociado a un determinado siniestro junto con la documentación pertinente, enviada por mail por el estudio. También se registran las coberturas de las pólizas afectadas al siniestro, que son informadas por el estudio vía mail. Para el desarrollo del

---

<sup>25</sup> Oracle Forms es un producto de software para crear pantallas que interactúan con una base de datos Oracle. Tiene un IDE que incluye un navegador de objetos, una hoja de propiedades y un editor de código que utiliza PL/SQL

<sup>26</sup> PL/SQL (Procedural Language/Structured Query Language) Es un lenguaje de programación en SQL.

trabajo de esta tesina se tomó como base ésta operatoria y las tablas asociadas de la base de datos, donde se persisten los datos del siniestro, para utilizarlas en el desarrollo de la ampliación del sistema de siniestro existente. Así mismo, se tomó de base la interacción entre la vista, modelo y controlador del sistema de siniestros, para implementar las páginas necesarias.

# Capítulo 3

---

## Implementación en la Compañía

### 3.1 Contexto del problema

El Sector creyó viable la implementación de un nuevo sistema para la liquidación de siniestros de Rs. Vs. y, luego de pasar por varias reuniones internas, presentó el proyecto como necesidad al sector Optimización de Procesos; el cual, mediante reuniones con El Sector, elaboró una maqueta en Excel para mostrar el dinamismo que debería poseer el nuevo sistema, en base a lo planteado en la necesidad. El sector Optimización de Procesos creó el DAN luego de varias reuniones con El Sector y presentó el mismo al GINT. Luego de leer el DAN y darle el orden de prioridad correspondiente, el GINT pone en conocimiento al CUA sobre el nuevo sistema que se está solicitando. El CUA evaluó mi disponibilidad y la estrecha relación del nuevo sistema con el sistema de Siniestros actual y teniendo en cuenta mi rol como responsable de dicho sistema, me convocó a la reunión de prefactibilidad como analista “de apoyo” del analista designado (quien sería mi asesor profesional, Víctor Laferrara) para evaluar la nueva necesidad de El Sector.

Participaron de la reunión un representante del GINT, un representante del CUA, dos representantes del sector Optimización de Procesos y dos analistas. La primera sugerencia del sector Optimización de Procesos fue desarrollar un nuevo sistema para la implementación de la necesidad planteada. Luego de debatir, se llegó a la conclusión que se podría ampliar el sistema actual de siniestros agregando nuevos perfiles que posean la funcionalidad necesaria para cubrir la liquidación de siniestros de Rs. Vs. Del acuerdo de ampliar el sistema de siniestros, resulté ser el analista “principal” del cambio.

Como se describió en el capítulo 2, colaboré en la mejora del DAN dado que se detectaron falencias en relación a la definición clara de los requerimientos solicitados. Este proceso se llevó adelante mediante la creación de cuestionarios, donde se consultaba por cada perfil en particular en base a lo interpretado en la reunión de prefactibilidad. Se realizaron reuniones con el sector de Optimización de Procesos, convocadas por mí, con el objetivo de precisar los requerimientos que estaban definidos de forma general; cómo, por ejemplo: “el sistema deberá contar con un apartado de estadísticas”, sin tener en claro para qué se querían hacer las estadísticas.

### 3.2 Análisis

Creada formalmente la NIS en el sistema Jira, comenzó mi tarea como analista. Leí con detenimiento el DAN y a su vez conversé con el sector de Optimización de Procesos cuando surgía alguna duda. Luego de esto, confeccioné el DAT, describiendo paso a paso qué se iba a implementar y cómo, detallando el tiempo empleado que se requeriría para cada apartado y el tiempo de resolución total del desarrollo.

En esta etapa, investigué en profundidad el actual sistema de siniestros, como es la interacción entre la vista, el controlador y el modelo, y como es utilizada cada componente de éste. También, llevé a cabo numerosas reuniones internas de sistemas, con compañeros con más de 15 años en la compañía, con el objetivo de: obtener el conocimiento necesario de las tablas, paquetes, funciones y procedimientos que actualmente existen en la base de datos y su utilización; saber cómo obtener, modificar y persistir los datos de “forma correcta” (para adaptarme a la forma de trabajar de la compañía); y conocer el funcionamiento del sistema programado en PL/SQL (Form), que hasta ese momento se utilizaba como único medio para dejar asentado las liquidaciones de siniestros de RS. Vs., como se explicó anteriormente.

De estas interacciones obtuve la información necesaria, por ejemplo, mis compañeros me indicaron que hay una tabla de liquidación de Rs. Vs. y un paquete específico de Rs. Vs.; por lo cual, en base a estos, la investigación de los sistemas indicados anteriormente y el análisis de los requerimientos, empecé a documentar cómo se debería llevar a cabo el desarrollo. Todo esto se realizó en simultáneo con la confección del DAT.

En la documentación destinada a los desarrolladores, se encuentran: casos de uso, un proceso de gestión de casos (como se muestra en la figura 3.1), y un diagrama entidad relación con la interacción de las tablas existentes y las nuevas que se debían crear (figura 3.2).

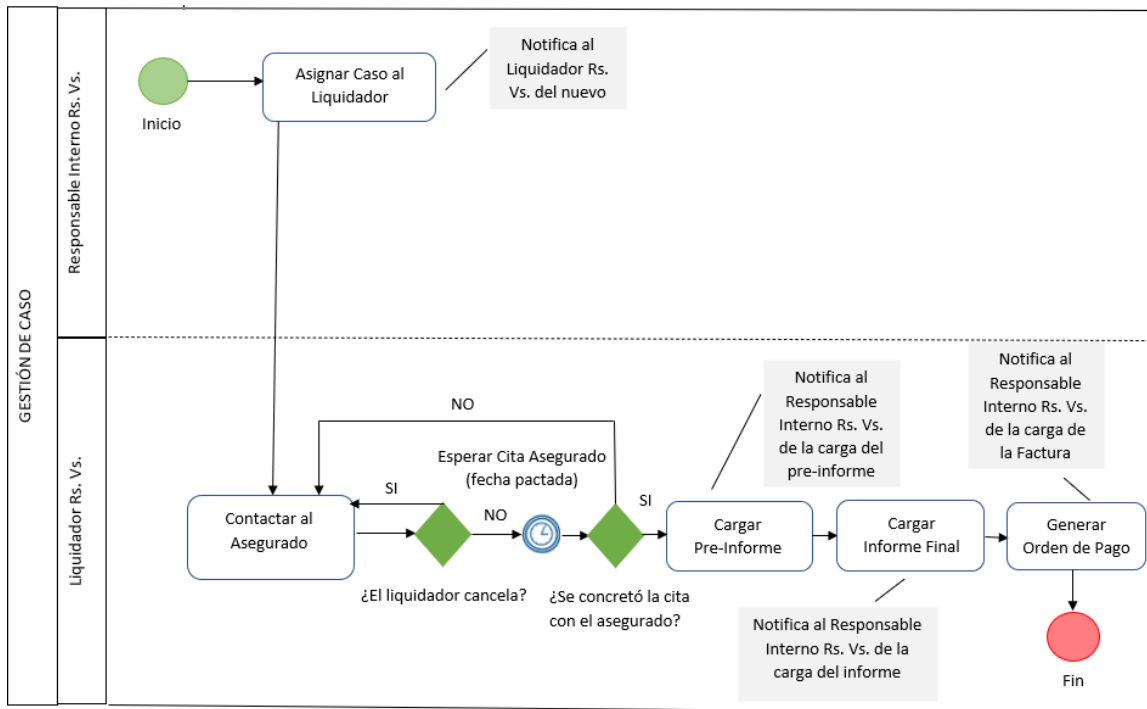


FIGURA 3.1 Procesos Gestión de Caso, ilustración propia

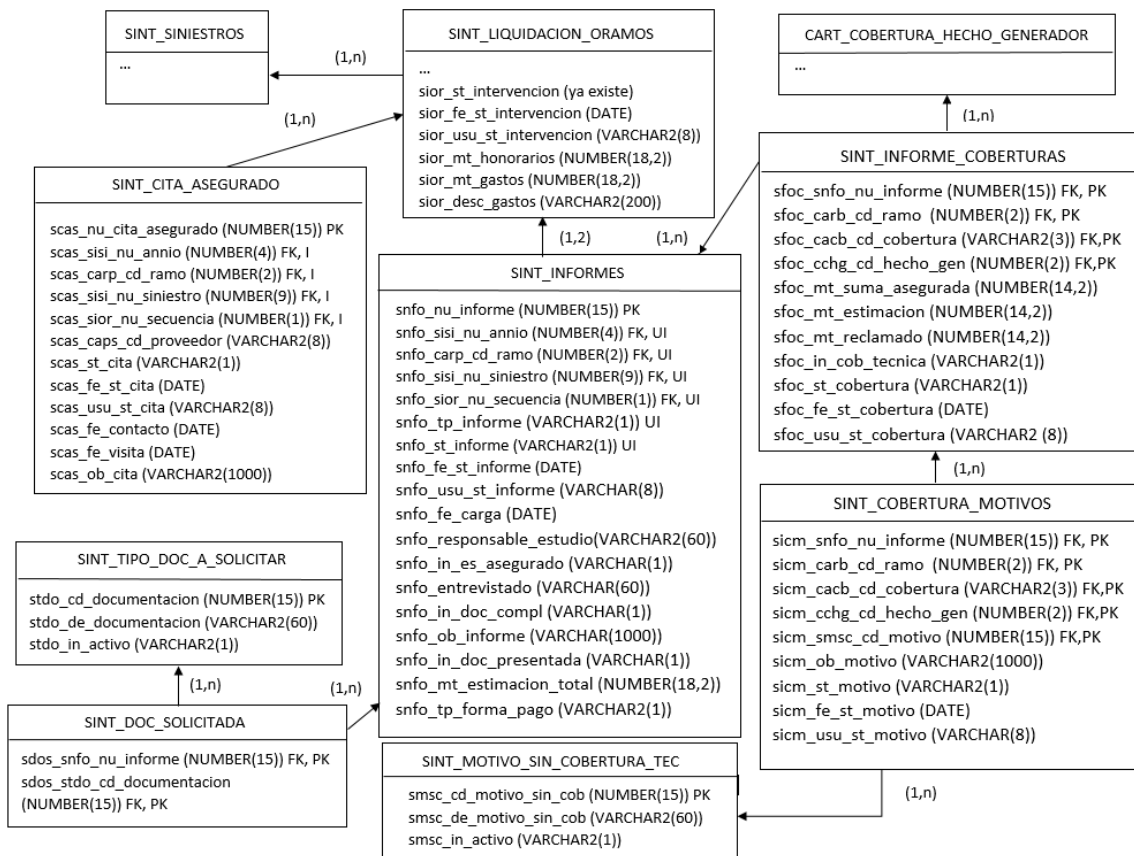


FIGURA 3.2 Diagrama Entidad-Relación, ilustración propia

Los nombres a las tablas y a cada atributo de la misma, se le da de la siguiente manera: los nombres de las tablas comienzan con el prefijo correspondiente al sector que pertenece cada tabla, en este caso: SINT es SINiestro Table, y CART es CARtera<sup>27</sup> Table; y los prefijos de cada atributo de una tabla se basan en el nombre de esta, por ejemplo: los campos de la tabla SINT\_CITAS\_ASEGURADO es SCAS siendo por Sint Citas ASegurado. Estos prefijos deben ser únicos para una misma tabla, es decir, no puede haber dos tablas cuyos atributos comienzan igual. Lo más común es asignar como prefijo la primera letra de cada palabra, pero, muchas combinaciones de estas ya se utilizan; por lo que tuve que utilizar letras intermedias cómo, por ejemplo: SNFO de SINT\_INFORMES siendo por Sint iNFormes. Otro aspecto a destacar es, si una tabla tiene una clave foránea de otra, esto se indica agregando el prefijo de la tabla padre; por ejemplo, en la tabla SINT\_DOC\_SOLICITADA tiene como atributo sdos\_snfo\_nu\_informe, tiene el prefijo SNFO luego de SDOS debido a que es clave foránea de la tabla SINT\_INFORMES. Un último comentario sobre esta modalidad para identificar tablas en la compañía es: si una tabla tiene un atributo “nieto”, es decir, es foráneo de una tabla hija que a su vez ese campo es foráneo; se respeta el prefijo de la tabla “abuelo”, por ejemplo: el atributo sfoc\_cacb\_cd\_cobertura es clave foránea de la tabla CART\_COBERTURA\_HECHO\_GENERADOR, esta tabla a su vez es hija de la tabla CART\_COBERTURA, la cual comienza con CACB cada atributo de esta.

### 3.3 Desarrollo

Aprobado el DAT por el sector Optimización de Procesos, con asesoramiento de El Sector, se cambia la NIS a RCF, para que pueda crear las subtarefas correspondientes a lo planteado en el DAT. Estas subtarefas las iba a asignar el CUA a los desarrolladores que estaban disponible, fijando fechas de inicio y fin de cada subtarea, según lo especifiqué en el DAT. Sin embargo, pedí permiso para poder desarrollar gran parte del proyecto. Esto lo hice debido a la nueva modalidad de tesina propuesta por la Facultad de Informática, PAEPA, la cual estoy llevando a cabo.

---

<sup>27</sup> Cartera hace referencia a la cartera del productor, por lo que serían tablas de producción. La cartera de un productor es el conjunto de pólizas pertenecientes al productor, se le da el nombre de producción porque son las pólizas que se emiten, se producen.



### 3.3.1 Creación de Perfiles y de Nuevas páginas

Primero me dediqué a crear los nuevos perfiles tanto de los empleados como del liquidador: los perfiles de los empleados se manejan mediante permisos, estos se guardan en una tabla llamada CART\_PERMISOS, en esta se indica el número de permiso, la descripción del permiso y la fecha de alta del permiso; siguiendo esta lógica di de alta 3 nuevos permisos, uno para cada perfil. Los perfiles de los prestadores<sup>28</sup>, en el sistema de siniestros, se manejan mediante un campo de la tabla CART\_PROVEEDORES\_SERVICIOS, en este campo se indica cada perfil con una letra diferente, siendo: “A” para Auditores Médicos, “B” para Abogados, “L” para Liquidadores, “M” para Médicos. Yo incorporé una nueva letra identificatoria “R” para Liquidadores Rs. Vs., siguiendo la lógica actual del sistema.

Luego creé las primeras vistas, modelo y controlador para todos los perfiles, así luego sería más dinámico la creación de las nuevas funciones. Una vez creado esto, comencé en orden de ejecución como lo plasmé en el proceso “Gestión de Caso” (figura 3.1).

### 3.3.2 Asignación de un nuevo caso

Este apartado se hace desde el sistema programado en PL/SQL. El Sector creyó conveniente dejar la forma de asignar un nuevo caso como se venía haciendo, por lo que creé un trigger; así cada vez que se asigna un liquidador a un caso pasará dos cosas: la primera, se completan los atributos nuevos de la tabla que se utiliza para persistir las liquidaciones de Rs. Vs. SINT\_LIQUIDACION\_ORAMOS<sup>29</sup> (como se muestra en la figura 3.3); la segunda, se enviará una notificación al liquidador del nuevo caso derivado (como se muestra en la figura 3.4), la cual se visualizará en la opción de menú de novedades, este se explicará más en detalle en el apartado Listado de Novedades.

La asignación de los casos sólo lo pueden hacer los responsables internos de Rs. Vs. ingresando por el sistema programado en PL/SQL mediante un permiso. Al asignar un nuevo caso a un liquidador, automáticamente se asigna como el responsable de ese caso al usuario que realiza dicha acción.

---

<sup>28</sup> Se denomina prestadores a los que trabajan para la compañía, pero son externos, estos pueden ser: liquidadores, abogados, médicos, vendedores de repuestos, etc.

<sup>29</sup> La palabra ORAMOS del nombre de la tabla es por Otros RAMOS, es otra forma de nombrar a Rs. Vs. por lo que en esta tabla se persisten los datos de las liquidaciones de Rs. Vs.

```

CREATE OR REPLACE TRIGGER SISTEMA.INS_ASIGNA_LIQUIDADOR
BEFORE
INSERT
ON GIOSEG.SINT_LIQUIDACION_ORAMOS
REFERENCING OLD AS OLD NEW AS NEW
FOR EACH ROW
DECLARE
numero NUMBER;
BEGIN
IF :NEW.SIOR_CD_PRESTADOR1 IS NOT NULL
THEN
BEGIN
:NEW.SIOR_ST_INTERVENCION := 1;
:NEW.SIOR_FE_ST_INTERVENCION := SYSDATE;
:NEW.SIOR_USU_ST_INTERVENCION := USER;
END;
END IF;
END;
/

```

FIGURA 3.3 Trigger para actualizar los nuevos atributos de la tabla SINT\_LIQUIDACION\_ORAMOS, ilustración propia

```

CREATE OR REPLACE TRIGGER SISTEMA.INSERTA_NOVEDAD_BOGA
AFTER
INSERT OR UPDATE OF SIOR_CD_PRESTADOR1
ON GIOSEG.SINT_LIQUIDACION_ORAMOS
REFERENCING OLD AS OLD NEW AS NEW
FOR EACH ROW
DECLARE
numero NUMBER;
BEGIN
IF :OLD.SIOR_CD_PRESTADOR1 IS NULL AND :NEW.SIOR_CD_PRESTADOR1 IS NOT NULL
THEN
BEGIN
SELECT NVL (MAX (nro_nota), 0)
INTO numero
FROM boga.novedades
WHERE annio = :NEW.SIOR_NU_ANNIO
AND ramo = :NEW.SIOR_CARP_CD_RAMO
AND siniestro = :NEW.SIOR_NU_SINIESTRO;
EXCEPTION
WHEN NO_DATA_FOUND
THEN
numero := 0;
END;
INSERT INTO boga.novedades a
VALUES (numero + 1, :NEW.SIOR_NU_ANNIO,
:NEW.SIOR_CARP_CD_RAMO, :NEW.SIOR_NU_SINIESTRO,
USER, sysdate, 'Siniestro nuevo derivado', 24, 0, 1, 1, 1,
1, 1, 'L', 1, 1, 0, 1, 1, 0);
END IF;
IF :OLD.SIOR_CD_PRESTADOR1 IS NOT NULL AND :NEW.SIOR_CD_PRESTADOR1 IS NOT NULL AND :OLD.SIOR_CD_PRESTADOR1 != :NEW.SIOR_CD_PRESTADOR1
THEN
BEGIN
SELECT NVL (MAX (nro_nota), 0)
INTO numero
FROM boga.novedades
WHERE annio = :NEW.SIOR_NU_ANNIO
AND ramo = :NEW.SIOR_CARP_CD_RAMO
AND siniestro = :NEW.SIOR_NU_SINIESTRO;
EXCEPTION
WHEN NO_DATA_FOUND
THEN
numero := 0;
END;
INSERT INTO boga.novedades a
VALUES (numero + 1, :NEW.SIOR_NU_ANNIO,
:NEW.SIOR_CARP_CD_RAMO, :NEW.SIOR_NU_SINIESTRO,
'selfProd', sysdate, 'Siniestro nuevo derivado', 24, 0, 1, 1, 1,
1, 1, 'L', 1, 1, 0, 1, 1, 0);
END IF;
END;
/

```

FIGURA 3.4 Trigger para notificar a los liquidadores el nuevo caso derivado, ilustración propia

### 3.3.3 Búsqueda de los siniestros de Rs. Vs.

A este apartado van a poder acceder todos los perfiles de Rs. Vs., lo que difiere es en la búsqueda: los liquidadores de Rs. Vs. sólo van a poder buscar los siniestros que tengan asignados, cada Responsable Interno de Rs. Vs. sólo va a poder buscar los siniestros que está como responsable, y el perfil administrador va a poder buscar todos los siniestros sin restricción.

En esta pantalla se podrá filtrar los siniestros por año<sup>30</sup>, ramo, número de siniestro, fecha desde y hasta, y por estado del siniestro como muestra la figura 3.5. El estado de siniestro puede ser: Derivado, cuando se deriva por primera vez; Pendiente de Visita, cuando el liquidador contacta al asegurado para visitarlo; En Análisis, cuando el liquidador confirma la visita con el asegurado; Facturado, cuando el liquidador guarda su factura; y Cerrado, cuando el Responsable Interno concilia la factura y le paga al liquidador.

El administrador Rs. Vs. además cuenta con dos filtros extras que le permitirá buscar los siniestros por Responsable Interno Rs. Vs. y por agencia, esto fue a pedido de El Sector.

LOGO DE LA COMPAÑÍA

FTONSO - TONSO FABRIZIO JESUS  
Perfil: Responsable Interno Riesgos Varios

← Operaciones sobre cuenta Listados Estadísticas Circulares Salir

Panel de Control

Filtros de búsqueda

Año/Ramo/Siniestro: Ingrese el año / Todos... / Ingrese el número de siniestros

Fecha desde - Fecha hasta: 26/04/2019 - 26/04/2020

Pendientes  Realizados

Estados del expediente:  Derivado  Pendiente de Visita  En Análisis  
 Facturado  Cerrado

Aplicar filtros

Lista de siniestros - Cantidad encontrados: 0

FIGURA 3.5 Pantalla de Búsqueda de Siniestros, ilustración propia

<sup>30</sup> El año de un siniestro está dado por el ejercicio de la empresa. Cada ejercicio comienza el primero de julio y termina el treinta de junio del otro año. La compañía donde trabajo va por el ejercicio 97 ya que hace 97 años que está en el mercado asegurador.

### 3.3.4 Consulta de un siniestro

En este apartado se detallan todos los datos relevantes del siniestro para informarle al Liquidador y demás perfiles, los cuales son: información del expediente, información del asegurado, información del productor, información de la póliza y el detalle del productor. Además, en este apartado todos los perfiles dispondrán de todos los endosos de la póliza al momento del siniestro y la denuncia del siniestro para descargar en formato PDF; podrán enviarse mensajes internos los liquidadores y los empleados internos; y podrán subir documentación relacionada el siniestro (figura 3.6), estos últimos se explicarán con más detalle en sus respectivos apartados más adelante.

LOGO DE LA COMPAÑÍA Perfil: Liquidador Riesgos Varios

← Listados Estadísticas Circulares Salir

Mensajería

**Detalle del siniestro** NUMERO DEL SINIESTRO

Detalle del expediente	
Estado:	Derivado
Indicador de fraude:	No
Fecha de derivación:	13/04/2020
Fecha de modificación:	13/04/2020 11:41
Agencia donde ingreso la denuncia:	BUENOS AIRES (3)
Cobertura financiera:	CUBIERTO
Plan elegido:	100
Exclusión de cobertura:	
Fecha de contacto:	
Fecha de visita:	
Tipo de endoso:	E - Emision

Menús desplegables

- Información General
- Información del asegurado
- Información del productor
- Información de la póliza
- Detalle del siniestro

Contactar al asegurado Historial de Mensajes Documentación digitalizada

Contactar al asegurado

Descarga de la póliza

Ver Póliza

Descargar Denuncia

Descarga de la denuncia

FIGURA 3.6 Pantalla de Detalle de un Siniestro, ilustración propia

### 3.3.5 Contacto con el asegurado

En este apartado se podrá guardar la cita para visitar al asegurado que sufrió el siniestro; así el liquidador podrá inspeccionar los objetos siniestrados y dar su veredicto si está cubierto por la póliza o no. Esto sólo estará disponible para el perfil Liquidador Rs. Vs., en los demás perfiles se visualizará la leyenda “espere confirmación de visita del liquidador”.

Al principio, como se muestra en la figura 3.6, se visualiza el botón “Contactar al asegurado”; al ingresar, le muestra la información de contacto del asegurado y del productor de la póliza, y un calendario para indicar la fecha de visita pactada; cuando confirma la cita por primera vez, el caso cambia a estado “Pendiente de Visita”.

Una vez confirmada la cita en la solapa “Contactar al asegurado” desaparece el botón antes mencionado y le aparece el detalle del contacto guardado, el liquidador tiene dos opciones: cancelar la cita antes que transcurra el día de la misma, mediante un botón cancelar que se muestra en el detalle, o esperar a la fecha; si cancela: se le permitirá postergar la cita, volviendo el estado del caso a “Derivado”, y le aparecerá nuevamente el botón “Contactar al asegurado” para repetir la acción indicada al comienzo del párrafo; si llega el día de la cita: se le mostrará dos opciones, “Visita fallida” o “Visita concretada”. Si elige la primera opción: se procede igual que al cancelarla, si elige la otra opción: automáticamente se le informará que la acción fue exitosa, el caso cambia a estado “En Análisis” y se visualiza un botón para poder cargar el informe preliminar en el detalle del siniestro. Así mismo, se muestra un historial de contactos en la solapa “Contactar al asegurado” y se cambia el nombre de esta por “Contactos con el asegurado”, este historial es visible a todos los perfiles.

### 3.3.6 Informe Preliminar

En este apartado el Liquidador Rs. Vs. y los demás perfiles de empleado, pueden guardar la información relevante al informe preliminar como muestra la figura 3.7. Dispone de los datos básico del caso a su disposición en la parte superior para tener una visión de cómo está el siniestro al cargar el informe preliminar, luego se visualiza los campos a completar para continuar.

Los campos a completar son: nombre del liquidador que realiza la carga del informe, algunos usuarios del sistema con perfil Liquidador Rs. Vs. son personas individuales pero otros están asociado a un estudio de liquidadores, por lo que debe indicar el nombre el liquidador que realiza la carga; el nombre de la persona entrevistada, puede ser el asegurado o un familiar directo, si es el asegurado la persona entrevistada se le da la oportunidad de tildar “¿Es el asegurado?” y se completa automáticamente el nombre de este; las coberturas<sup>31</sup> afectadas al siniestro, se indica el hecho generador<sup>32</sup>, monto estimado por el liquidador, el monto reclamado por el asegurado, y si está cubierto o no lo denunciado, como se muestra en la figura 3.8; una observación del caso; y debe indicar si le solicitó documentación complementaria al asegurado, de ser así se visualiza los tipos de documentación para que indique el(o los) tipo(s) de documentación que les solicitó; adjuntando el formulario correspondiente firmado por el asegurado, donde certifica la documentación presentada.

---

<sup>31</sup> Una cobertura es los riesgos que se van a cubrir dado un siniestro; por ejemplo: la cobertura 070 de combinado familiar (ramo 24) es “todo riesgo en domicilio sin nómina”, este caso está cubierto los daños a los electrodomésticos ya sea por rayo, baja tensión, etc. Las coberturas se establecen cuando se contrata una póliza.

<sup>32</sup> Un hecho generador es el motivo por el cual ocurrió el siniestro; por ejemplo: el hecho generador 4 de la cobertura 070 es “rayo”.

Todos los campos a ingresar son obligatorios, y se valida que haya al menos una cobertura seleccionada antes de guardar los cambios. En caso de haber un campo vacío se visualiza un mensaje de error indicando cuál es el campo que se debe completar para continuar.

Al completar todos los campos, y confirmar la guarda del informe preliminar, por cada cobertura que posea cobertura técnica se actualiza la reserva de cada cobertura por el monto indicado para su eventual pago. La reserva para una cobertura es el capital que se guarda para indemnizar un siniestro, la reserva total guardada es la suma de los montos reservados de todas las coberturas.

**Informe Preliminar**

**Información del Caso**

Siniestro:	Número de siniestro				
	Fecha Ocurrencia:	18/07/2019	Fecha de derivación:	13/02/2020 00:00	
	Fecha de contacto:	27/04/2020 19:22	Fecha de visita:	28/04/2020	
	Póliza/Certificado:	Número de póliza	Sección:	24	
	Vigencia desde:	08/05/2019	Vigencia hasta:	08/05/2020	
	Asegurado:	Nombre asegurado			
Hora Ocurrencia:	11:30	Lugar de ocurrencia:	Lugar de ocurrencia		

Liquidador:

Persona Entrevistada:

¿Es el asegurado?

Coberturas Afectadas

Seleccione las coberturas afectadas

Cobertura - Suma Asegurada

Seleccione una Cobertura... ▼

Monto estimado en \$	Monto Reclamado en \$ (Opcional)	¿Sin Cobertura Técnica? <input type="checkbox"/>
<input style="width: 90%;" type="text"/>	<input style="width: 90%;" type="text"/>	
formato 99999999,99	formato 99999999,99	

No se cargaron Coberturas

Monto Estimado Total (con cobertura técnica):	\$0,00	Monto Reclamado Total:	\$0,00
---	--------	------------------------	--------

Observaciones:

¿Solicitó documentación complementaria?

FIGURA 3.7 Formulario Informe Preliminar, ilustración propia

**Coberturas Afectadas**

**Seleccione las coberturas afectadas**

Cobertura - Suma Asegurada

070) TODO RIESGO EN DOMICILIO - SIN NOMINA - \$120.000,00

Hecho Generador

4 - VARIACIÓN DE TENSIÓN

Monto estimado en \$ 100000 <small>formato 99999999,99</small>	Monto Reclamado en \$ (Opcional) 120000 <small>formato 99999999,99</small>	<input checked="" type="checkbox"/> ¿Sin Cobertura Técnica?
--	--	---

**Seleccione un motivo o más de uno de ser necesario**

- 1) No cuenta con el adicional correspondiente
- 2) Exclusión específica en póliza
- 3) Falta de documentación habilitante
- 4) Bien no nominado en póliza
- 5) El lugar de ocurrencia no coincide con el Dom. de Riesgo
- 6) Hecho sucedido fuera del lugar de Riesgo Cubierto
- 7) Otros

**No se cargaron Coberturas**

Monto Estimado Total (con cobertura técnica):	\$0,00	Monto Reclamado Total:	\$0,00
---	--------	------------------------	--------

FIGURA 3.8 Selección de Cobertura Afectada al Siniestro, ilustración propia

Cuando se tilda “¿Sin cobertura técnica?” se muestran los distintos motivos por lo cual no aplica la cobertura al siniestro, como se muestra en la figura 3.8. A medida que el usuario va agregando nuevas coberturas afectadas al siniestro, se va calculando el total de los montos indicados en los campos “Monto estimado” y “Monto Reclamado”, que, tal como su nombre lo indica, se guarda el monto que el usuario estima que cuesta la indemnización para esa cobertura y el monto que el asegurado reclama en su lugar (este puede ser nulo, ya que a veces el asegurado no reclama un monto específico para una cobertura). El total estimado se calcula sólo de los montos de las coberturas que posean cobertura técnica y el monto reclamado se suma sin considerar esto. Al agregar una cobertura, se visualizan los datos indicados a modo de tabla, así siguiendo con todas las coberturas que seleccione como se muestra en la figura 3.9

**Coberturas Afectadas**

Seleccione las coberturas afectadas

Cobertura - Suma Asegurada

Seleccione una Cobertura...

Monto estimado en \$  Monto Reclamado en \$ (Opcional)  ¿Sin Cobertura Técnica?

formato 99999999,99 formato 99999999,99

Agregar Cobertura

Cobertura	Suma Asegurada	Hecho Generador	Monto Estimación	Monto Reclamado	Cob. Técnica	Motivo/s	-
070) TODO RIESGO EN DOMICILIO - SIN NOMINA	\$120.000,00	4 - VARIACIÓN DE TENSIÓN	\$100.000,00	\$120.000,00	N	No cuenta con el adicional correspondiente, Bien no nominado en póliza	✘
010) INCENDIO EDIFICIO	\$3.487.581,50	3 - DAÑOS POR CAIDA DE RAYO	\$1.234.555,00	\$3.000.000,00	S	No aplica	✘
025) GASTOS DE LIMPIEZA Y RETIRO DE ESCOMBROS	\$174.379,08	1 - GASTOS DE LIMPIEZA Y RETIRO ESCOMBROS	\$100.000,00	\$170.005,00	S	No aplica	✘
<b>Monto Estimado Total (con cobertura técnica):</b>		<b>\$1.334.555,00</b>	<b>Monto Reclamado Total:</b>		<b>\$3.290.005,00</b>		

*FIGURA 3.9 Coberturas Afectadas al Siniestro, ilustración propia*

La suba del archivo lo gestiono con un componente de RichFaces llamado “rich:fileUpload” (figura 3.10), este componente consiste en dos partes: por un lado está la lista de archivos que el usuario quiere cargar; por el otro, los archivos cargados, para el “preview” de la imagen cargada utilice un servlet, llamado showImageServlet; en él, obtengo la imagen redimensionándola a un tamaño más chico y la muestro en miniatura; si el usuario quiere visualizarla en tamaño completo, lo podrá hacer al hacer click sobre la misma. Este componente lo adapté para que sea más amigable con el usuario cambiándole el nombre de los botones al castellano. En cuanto al funcionamiento, el componente llama a un “listener” que se ejecuta al cargar cada archivo, en este “listener” se puede gestionar el Blob<sup>33</sup> del archivo para guardarlo en una variable hasta que el usuario guarde el informe preliminar. Me tuve que adaptar a la forma de persistir los documentos de la compañía, esta es guardando el Blob directamente en la base. La figura 3.11 muestra la visual de la carga del formulario con el “preview”. Los archivos que se pueden adjuntar son en formato JPG o PDF.

<sup>33</sup> Los BLOB (Binary Large Objects, objetos binarios grandes) son elementos utilizados en las bases de datos para almacenar datos de gran tamaño. Generalmente, estos datos son imágenes, archivos de sonido y otros objetos multimedia.



```

<rich:fileUpload fileUploadListener="#{siniestros@sbean.archivoListener}" render="archivoUpload"
    id="archivoUpload"
    acceptedTypes=".pdf, .PDF, .jpg, .JPG"
    doneLabel="Listo"
    deleteLabel="Eliminar"
    clearLabel="Limpiar"
    clearAllLabel="Limpiar Todo"
    addLabel="Agregar"
    uploadLabel="Cargar Archivo"
    maxFilesQuantity="#{siniestros@sbean.archivoPreInf}"
    onTypeRejected="alert('Solo archivos PDF y JPG son aceptados');";
    <aj:ajax id="archivoUploadEvent" event="uploadComplete" execute="@form" render="infoFormulario" />
</rich:fileUpload>
</h:column>
<h:column span="6" id="infoFormulario" layout="Block" >
    <rich:panel style="height: 245px; overflow: auto;";
    <facet name="header">
        <h:outputText value="Información del formulario adjunto" style="padding: 20px;"/>
    </facet>
    <h:outputText value="No se ha adjuntado el formulario" rendered="#{siniestros@sbean.imagenDigitalizada==null}" />
    <rich:table id="columns" value="#{siniestros@sbean.imagenDigitalizada}" var="archivo" rowKeyVar="row"
    <rich:panel bodyClass="rich-Laguna-panel-no-header">
        <h:panelGrid id="panelFotos" rendered="#{archivo.extension ne 'PDF'}">
            <h:commandLink id="link">
                
                <rich:componentControl target="panelFotosArchivo" operation="show" />
            <aj:commandLink action="#{siniestros@sbean.eliminarArchivo}" render="infoFormulario" style="vertical-align: top; immediate="true">
                
            </aj:commandLink>
            <h:commandLink>
                <h:panelGrid id="columns" 2">
                    <h:outputText value="Nombre:" />
                    <h:inputText value="#{archivo.nombre}" />
                </h:panelGrid>
                <rich:popupPanel id="panelFotosArchivo" width="1000" height="600" resizable="true">
                    <facet name="header">
                        <h:panelGroup>
                            <h:outputText value="#{archivo.nombre}" />
                        </h:panelGroup>
                    </facet>
                    <facet name="controls">
                        <h:outputLink onclick="#{rich:component('panelFotosArchivo').hide()}; return false;" />
                        
                    </facet>
                </rich:popupPanel>
            </h:column>
            
        </rich:panel>
    </h:column>

```

FIGURA 3.10 Fragmento del Código de rich:fileUpload, ilustración propia

¿Solicitó documentación complementaria?


Indique el/los documento/s solicitado/s

Presupuesto  Informe Técnico  Denuncia Policial  Constancia de CBU  Otros

Adjunte el formulario de requerimientos firmado por el asegurado

+ Agregar

Información del formulario adjunto



Nombre:

FIGURA 3.11 Carga del Formulario, ilustración propia

Una vez que el usuario guarde el informe preliminar se visualiza el botón para crear el informe final. El liquidador puede modificar cada parte del informe preliminar hasta que guarde el informe final por primera vez; se registra un historial por cada modificación del informe preliminar.

### 3.3.7 Informe Final

Al igual que el apartado de Informe Preliminar, esta opción está disponible para todos los perfiles. Si se indica que se solicitó documentación al asegurado, antes de poder continuar con la creación del informe final, debe ingresar la documentación solicitada; para esto utilizo el mismo componente que comenté en el apartado anterior para guardar el formulario, sólo que esta vez se le permite guardar hasta 10 archivos, pueden ser en formato JPG o PDF. Esto es bloqueante, hasta que no se adjunte la documentación requerida no se va a poder avanzar.

Una vez que se adjunta la documentación, o si no se solicitó dicha documentación en el informe preliminar, se visualiza el formulario para cargar el informe final. En este apartado también se muestra la información relevante del siniestro al principio; y, en base a lo completado en el informe preliminar, se mostrarán las coberturas seleccionadas, en las cuales se deberá indicar el monto definitivo con el cual se va a indemnizar al asegurado por cada una de ellas. Si se indicó que no aplica cobertura técnica una determinada cobertura entonces, el monto a indemnizar es 0 (cero) ya que no está cubierto lo siniestrado, siendo este campo no modificable. También, al igual que el Informe Preliminar, se tiene un campo para agregar una observación; se debe adjuntar el informe final en formato PDF; y se debe indicar la forma de pago elegida por el asegurado, la cual puede ser transferencia bancaria o cheque (figura 3.12).

El archivo que se adjunta como informe final es un documento confeccionado por cada liquidador, el mismo no sigue un estándar; es por eso, que se les da la posibilidad de adjuntar el informe y a confeccionados por ellos. Si el liquidador indica que el asegurado optó por la opción de transferencia bancaria, debe adjuntar el CBU del asegurado, el mismo puede ser en formato JPG o PDF. En ambas subidas de archivos se utilizó el mismo componente nombrado anteriormente.

Resumen de la liquidación						
Cobertura	Suma Asegurada	Hecho Generador	Cob. Técnica	Motivo/s	Monto a indemnizar en \$	Monto Reclamado
070) TODO RIESGO EN DOMICILIO - SIN NOMINA	\$120.000,00	4 - VARIACIÓN DE TENSIÓN	N	No cuenta con el adicional correspondiente, Bien no nominado en póliza	0,00	\$120.000,00
010) INCENDIO EDIFICIO	\$3.487.581,50	3 - DAÑOS POR CAIDA DE RAYO	S	No aplica	1.234.555	\$3.000.000,00
025) GASTOS DE LIMPIEZA Y RETIRO DE ESCOMBROS	\$174.379,08	1 - GASTOS DE LIMPIEZA Y RETIRO ESCOMBROS	S	No aplica	100.000	\$170.005,00
<b>Monto a indemnizar total</b>					<b>\$1.334.555,00</b>	

Observaciones:

**Adjunte el Informe Final**


Se puede cargar sólo un archivo de informe final, si desea reemplazarlo: elimine el existente primero

export(1).pdf  
Listo

Limpiar

X Limpiar Todo

**Información del Informe Final adjunto**



Nombre:

¿Cual es el método de pago elegido por el asegurado?

Transferencia Bancaria
  Cheque

Cancelar

Guardar Informe Final

FIGURA 3.12 Formulario Informe Final, ilustración propia

Cuando se guarda el informe final, al igual como expliqué en el apartado Informe Preliminar, se actualizan las reservas de las coberturas. Esta actualización se hace mediante la invocación a dos procedimientos que creé en el paquete de liquidación de Rs. Vs. que actualizan los datos de dos formas del sistema programado en PL/SQL (figura 3.13).

Cabe destacar que estos datos, tanto cuándo el estudio liquidador presentaba el informe preliminar e informe final, anteriormente eran actualizados por un Responsable Interno mediante dichas formas.

```

PROCEDURE RESERVA_COBERTURA_UPD (
    p_Annio          gioseg.sint_movimientos_coberturas.smcc_sisi_nu_anno%TYPE,
    p_Ramo          gioseg.sint_movimientos_coberturas.smcc_carp_cd_amo%TYPE,
    p_Siniestro     gioseg.sint_movimientos_coberturas.smcc_sisi_nu_siniestro%TYPE,
    p_Cd_Suc       gioseg.sint_siniestros_coberturas.sisc_casu_cd_sucursal%TYPE,
    p_Nu_Poliza    gioseg.sint_movimientos_coberturas.smcc_capo_nu_poliza%TYPE,
    p_Nu_Certificado gioseg.sint_movimientos_coberturas.smcc_cace_nu_certificado%TYPE,
    p_Nu_Endoso     gioseg.sint_movimientos_coberturas.smcc_carc_nu_endoso%TYPE,
    p_Nu_Asegurado gioseg.sint_movimientos_coberturas.smcc_carc_nu_asegurado%TYPE,
    p_Cd_Ramo      gioseg.sint_movimientos_coberturas.smcc_carb_cd_amo%TYPE,
    p_Cd_Cobertura gioseg.sint_movimientos_coberturas.smcc_cach_cd_cobertura%TYPE,
    p_Cd_Hecho_Gen gioseg.sint_movimientos_coberturas.smcc_cchg_cd_hecho_gen%TYPE,
    p_Usuario      VARCHAR2,
    p_Monto        gioseg.cart_cobertura_hecho_generador.cchg_estimacion_promedio%TYPE,
    p_datos_ok_out OUT VARCHAR2,
    p_mensaje_error_out OUT VARCHAR2) IS
    l_Ajuste          gioseg.sint_movimientos_coberturas.smcc_mt_movimiento%TYPE:=0;
    l_Monto_Actual   gioseg.sint_movimientos_coberturas.smcc_mt_movimiento%TYPE:=0;
    l_Cant           NUMBER(10);
    l_Estimacion     gioseg.cart_cobertura_hecho_generador.cchg_estimacion_promedio%TYPE;
    l_Cd_Considera_Mov gioseg.sint_movimientos_coberturas.smcc_cd_considera_mov%TYPE;
BEGIN
    BEGIN
        SELECT sisc_mt_reserva_actual
        INTO l_Monto_Actual
        FROM gioseg.sint_siniestros_coberturas a
        WHERE a.sisc_sisi_nu_anno      = p_Annio
        AND a.sisc_carp_cd_amo        = p_Ramo
        AND a.sisc_sisi_nu_siniestro   = p_Siniestro
        AND a.sisc_casu_cd_sucursal    = p_Cd_Suc
        AND a.sisc_capo_nu_poliza      = p_Nu_Poliza
        AND a.sisc_cace_nu_certificado = p_Nu_Certificado
        AND a.sisc_carc_nu_endoso      = p_Nu_Endoso
        AND a.sisc_carc_nu_asegurado   = p_Nu_Asegurado
        AND a.sisc_carb_cd_amo        = p_Cd_Ramo
        AND a.sisc_cach_cd_cobertura   = p_Cd_Cobertura
        AND a.sisc_cchg_cd_hecho_gen   = p_Cd_Hecho_Gen;
    EXCEPTION
        WHEN no_data_found THEN
            l_Monto_Actual:=0;
            INSERT INTO gioseg.sint_siniestros_coberturas

```

FIGURA 3.13 Fragmento del Procedimiento RESERVA\_COBERTURA\_UPD, ilustración propia

Una vez que el usuario guarde el informe final se visualiza el botón para generar la orden de pago. El usuario puede modificar cada parte del Informe Final hasta que se genera la orden de pago; y, al igual que en el Informe Preliminar, se registra un historial de cada modificación del informe final.

### 3.3.8 Generación de orden de pago

En este apartado todos los perfiles van a poder generar una orden de pago relacionada al siniestro; y los perfiles de empleado, no así el Liquidador Rs. Vs., van a poder generar una segunda orden de ser necesario, una vez conciliada y pagada la primera.

En la orden se indica el monto de honorarios del liquidador por los servicios brindados y los gastos que ocasionó realizar la liquidación, si los hubiera, con el ingreso de una descripción de estos. Se creó un nuevo tipo de orden para guardar estos conceptos.

La gestión de ordenes ya se realizaba en la compañía a través del sistema de PL/SQL, por lo que se tomaron las ordenes existentes como ejemplo para la creación de la orden propia para la liquidación

de siniestros de Rs. Vs.; la misma está compuesta por dos conceptos “Honorarios Prestadores” y “Gasto Liquidación Siniestros”. En su creación este tipo de orden comienza en “Generada” (estado 1); luego que el liquidador carga la factura relacionada a la orden (esto se explicará en profundidad en el apartado siguiente), la orden pasa a estar “Facturada” (estado 4) como así también el estado del caso; posteriormente a la carga de la factura, el Responsable Interno valida la orden y decide si anularla (cambiar a estado 5) o conciliarla (cambiar a estado 7), una vez que se concilia se genera el pago. Estos últimos cambios de estados el Responsable Interno Rs. Vs. lo realiza desde el sistema programado en PL/SQL.

Cuando el Responsable Interno Rs. Vs. cambie una orden a estado anulada, se le dará otra oportunidad al Liquidador Rs. Vs. para generar otra orden de pago nueva, pasando por los mismos estados que se indicaron anteriormente, y el estado del caso vuelve a “En Análisis”. Si el Responsable Interno Rs. Vs. concilia la orden: el caso pasa a estado “Cerrado”, y finaliza la gestión de un caso. Existe la posibilidad que el Responsable Interno Rs. Vs. considere necesario la generación de una segunda orden de pago; para esto, al momento que se concilia lo que sería la primera orden, aparece la opción para generar la segunda orden solamente en los perfiles Responsable Interno Rs. Vs., Administrador Rs. Vs y Super Administrador. Una vez que el empleado interno de la compañía genera la segunda orden, el Liquidador Rs. Vs. puede volver a cargar la factura relacionada a esta, volviendo el estado del caso a “Facturado”; para luego repetir los cambios de estados como se indicó anteriormente. Por consiguiente, el estado del caso puede volver a “En Análisis” una vez “cerrado” (esto representa el 5% de los casos, por lo que es muy poco probable que suceda), finalizando nuevamente a “Cerrado” cuando el Responsable Interno concilie la segunda orden.

Una vez creada la orden se le mostrará la información de ésta a todos los perfiles ingresando al detalle del siniestro, botón “Ver órdenes de pago”, ingresando por ese link el liquidador va a poder cargar la factura relacionada a esta.

### **3.3.9 Carga de la factura de pago**

Este apartado fue el más importante que implementé ya que posee múltiples validaciones, las cuales se hacen sobre los datos de la factura cargada por los estudios liquidadores. A esta funcionalidad sólo tiene acceso el perfil Liquidador Rs. Vs., y en la página se carga todos los datos referentes a la factura del siniestro liquidado.

La página cuenta con un campo para ingresar el código de barras, el cual, una vez que se presiona la tecla “enter” valida cada parte de los dígitos ingresados. Este campo está pensado para ingresarlo con una lectora de código de barras para facilitar la carga de la factura. Una vez que se ingresan todos los dígitos del código se valida la longitud de este para completar los diferentes campos requeridos para guardar la facturación; estos son: CUIT (once dígitos), código de la factura (tres dígitos), punto de

venta (cinco dígitos), CAE<sup>34</sup> (catorce dígitos), fecha de vencimiento de CAE en formato AAAAMMDD (ocho dígitos) y el código verificador (un dígito).

Además de los campos indicados en el párrafo anterior, el liquidador debe indicar el número de la factura y la fecha de emisión de esta; debe ingresar el monto neto, y los montos IIBB<sup>35</sup> y Tasa de Seguridad e Higiene; también se le requiere que adjunte la factura en formato digital, puede ser en formato PDF o JPEG/JPG. Se muestra la pantalla para la carga de todos estos datos en la figura 3.14.

Todos los campos son numéricos, por esto se valida que cuando se ingresa un caracter sea un número, en el caso de los montos se permite ingresar la “,” para indicar la parte decimal, si el usuario ingresa una letra o caracter especial directamente no se visualiza en el campo.

Anteriormente se explicó cómo se autocompleta los campos cuando se ingresa un número de barras; así mismo, se cuenta con la validación de la CUIT del liquidador y se autocompletan algunos campos a partir de la misma: al completar la CUIT se verifica que la ingresada corresponda con el usuario que inició sesión, y del mismo modo, se verifica que la CUIT del liquidador actualmente en sesión corresponde a la orden que se quiere facturar, si las validaciones son exitosas se muestra en pantalla el nombre del liquidador (o nombre del estudio liquidador). Continuado a esto, se obtienen automáticamente los posibles códigos de comprobantes pertenecientes al liquidador a partir de la CUIT, para mostrarlos en el campo de selección “Código de comprobante”, por si hubiera más de una opción para seleccionar (figura 3.14); una vez seleccionado el código se auto completan los campos “Tipo de comprobante” y “Sigla” asociados al código seleccionado. De esta forma, el liquidador debe seguir completando los demás campos de forma manual en base a su factura.

Paso seguido a completar los datos de la factura, se le solicita que indique los montos de la factura, debiendo ingresar el monto neto como obligatorio, y los montos IIBB y Tasa de Seguridad e Higiene si los hubiera. Una vez ingresado el monto neto, se verifica si se le debe aplicar el IVA del 21% o no a la factura para obtener el monto total; esto depende del tipo de factura que se indicó en el primer cuadrante de la figura 3.14. si la sigla es “A” es altamente probable que se le aplique el IVA, esto se determina si el liquidador es de Tierra del Fuego o no: si es de esta Provincia la factura queda exenta de IVA. En consecuencia, no se les aplica el IVA a los liquidadores de Tierra del Fuego y a los que facturen con sigla distinta a “A”.

Posteriormente indicado el monto neto y calculado el IVA, el liquidador debe completar los montos IIBB y Tasa de Seguridad e Higiene de corresponder; pudiendo dejar en nulo o en cero ambos montos. A partir a la suma de todos los montos indicados se obtiene el total, el cual se muestra al lado de la leyenda “Total Facturado” de la figura 3.14.

---

<sup>34</sup> CAE es la sigla de Código de Autorización Electrónico.

<sup>35</sup> IIBB es Ingresos Brutos.

## Facturación siniestro Nro del Siniestro

**Datos de la factura (Orden de servicio nro. orden I)**

**Sr. prestador, tenga en cuenta que los campos de esta pantalla deben llenarse con los datos que estan en la factura que va a presentar.**

Codigo de barras factura:	<input type="text"/>		
CUIT:	<input type="text"/>		
Codigo de comprobante:	<input type="text" value="Seleccione"/>		
Tipo de comprobante:		Sigla:	
Punto de venta:	<input type="text"/>	Número de factura:	<input type="text"/>
Fecha de emisión:	<input type="text"/>		
CAE:	<input type="text"/>	Fecha de vencimiento CAE:	<input type="text"/>

Monto neto:	\$	<input type="text"/>
IVA:		
Neto con IVA:		
Percepción IIBB	\$	<input type="text"/>
Tasa de Seg. e Hig.	\$	<input type="text"/>

**Total facturado:**

**Adjunte la Factura**

**Se puede cargar sólo una Factura, si desea reemplazarla: elimine la existente primero**

<div style="border: 1px solid #ccc; background-color: #f0f0f0; padding: 5px; text-align: center; margin-bottom: 10px;"><span style="font-size: 24px; color: #007bff;">+</span> Agregar</div>	<div style="border: 1px solid #ccc; background-color: #f0f0f0; padding: 5px; text-align: center;"><p><b>Información de la Factura adjunta</b></p><p>No se ha adjuntado la Factura</p></div>
--	---

FIGURA 3.14 Carga de la Factura, ilustración propia

Finalmente, como último paso, debe adjuntar la factura en formato digital. La suba de este archivo se realiza como se aplica para las demás páginas que requiera adjuntar un archivo.

Una vez completado todos los campos, se llevan a cabo una serie de validaciones sobre los datos cargados. Se vuelve a validar que la CUIT ingresada corresponde al liquidador que inició sesión y al

liquidador que generó la orden; se verifica que el CAE ingresado es válido; se verifica que la factura ingresada no haya sido dada de alta antes; se verifica que la fecha de emisión sea menor o igual al día de la fecha; se verifica que los montos IIBB y Tasa de Seguridad e Higiene ingresados no excedan al máximo permitido; se verifica el tipo de la factura y se valida contra los servicios de AFIP que existe una factura cargada en sus bases de datos.

La Validación contra la AFIP se hace a través de web services utilizando SOA con REST, los servicios de AFIP son en SOAP. Se genera un token de validación para conectarse con la AFIP, y la comunicación con esta es mediante un Json con los datos de la factura (figura 3.15); la AFIP responde en resultado si la acepta mediante una “A” o si encontró un error o no la encontró mediante una “R” con un código y un mensaje relacionado a ese código en observaciones, manejo esa respuesta como se muestra en la figura 3.16

Una vez que pasaron todas las validaciones, se guarda la factura y se cambia el estado del caso a “Facturado”; el estado de la orden también pasa a “Facturada” y se espera por la conciliación, si la factura es correcta a la orden, o anulación, si no se está de acuerdo con algún dato o valor de la factura.

```
{
  "cmpResp": {
    "cbteFch": "string",
    "cbteModo": "string",
    "cbteNro": 0,
    "cbteTipo": 0,
    "codAutorizacion": "string",
    "cuitEmisor": 0,
    "docNroReceptor": "string",
    "docTipoReceptor": "string",
    "impTotal": 0,
    "opcionales": [
      {
        "id": "string",
        "valor": "string"
      }
    ],
    "ptoVta": 0
  },
  "errors": [
    {
      "code": 0,
      "msg": "string"
    }
  ],
  "events": [
    {
      "code": 0,
      "msg": "string"
    }
  ],
  "fchProceso": "string",
  "observaciones": [
    {
      "code": 0,
      "msg": "string"
    }
  ],
  "resultado": "string"
}
```

FIGURA 3.15 Json para Comunicarse con la AFIP, ilustración propia



```

public Boolean useToken(String token) throws JSONException, Exception{
    //try{
        Client client = ClientBuilder.newClient();
        WebTarget webTarget = (WebTarget) client.target(SERVICE_URL + URI_DOCUMENT_RECOGNITION);
        String respuesta = webTarget.request(MediaType.APPLICATION_JSON_TYPE)
            .header("Authorization", "Bearer " + token)
            .post(Entity.json(requestDto), String.class);

        CmpResponse obj = new Gson().fromJson(respuesta, CmpResponse.class);
        String response=obj.getResultado();
        if (response!=null && (obj.getErrors()==null)){
            if (response.equals("A")){
                this.resultado=true;
                return true;}
            else
            {if (obj.getObservaciones() !=null){
                Obs[] observaciones=obj.getObservaciones();
                throw new JSONException(observaciones[0].getMsg());}
            }
        }
        else {
            if (obj.getErrors() !=null){
                Err[] errores=obj.getErrors();
                throw new JSONException(errores[0].getMsg());}
        }
    }

    return false;
}

```

FIGURA 3.16 Obtención de la Respuesta de la AFIP, ilustración propia

Como se explica en el apartado anterior, se definió que un siniestro tenga asociada una orden de pago conciliada, o a lo sumo dos órdenes, sin posibilidad de tener al mismo tiempo las dos órdenes para conciliar: primero se concilia 1 y luego se habilita la posible gestión de una segunda. A su vez, cada orden sólo tiene asociada una sola factura, pudiéndola eliminar si se decide cancelar una orden y agregar la misma factura nuevamente a una nueva orden.

### 3.3.10 Continuación del caso

Como se explicó en los apartados anteriores, el cambio de orden a “Conciliada” o “Cancelada” la realiza un empleado de la compañía, muy probablemente el asignado como Responsable Interno Rs. Vs. del caso, desde el sistema de Forms que se tenía; esto fue decidido por El Sector, para poder seguir operando como se venía haciendo. Se agregó, a las formas involucradas, la funcionalidad para cambiar a estado “Cerrado” el caso, si se concilia, y cambiar a estado “En análisis” si se cancela; habilitando la opción para editar el informe final al Liquidador Rs. Vs.. De esta forma, el liquidador

puede modificar el informe final nuevamente de ser necesario, para luego volver a generar la orden de pago y finalmente volver a cargar la factura.

Hay casos excepcionales que requieren el pago de una segunda factura, por eso, una vez “cerrado” el caso, el Responsable Interno Rs. Vs., el Administrador Rs. Vs. o el Super Administrador van a poder cargar una segunda orden de pago, habilitando la posibilidad al Liquidador Rs. Vs. de cargar la factura asociada a la orden. Una vez que el liquidador carga la nueva factura, el caso vuelve a estado “Facturado”, para luego volver a esperar la finalización del ciclo previamente explicado.

### **3.3.11 Mensajería**

En este apartado, tanto el Liquidador Rs. Vs., el Responsable Interno Rs. Vs. como el Administrador Rs. Vs. pueden enviarse mensajes entre ellos. Es una pantalla simple que cuenta con la opción para elegir los destinatarios del mensaje, un panel para ingresar el texto del mensaje y un botón para enviar el mensaje. Los mensajes se manejan como novedades de tipo “mensaje” y se muestran tanto en la opción de menú “Lista de novedades” como en la solapa “Historial de Mensajes” de la consulta de siniestros; en esta solapa se muestran todos los mensajes según el usuario que inicia sesión.

### **3.3.12 Digitalización**

En este apartado, todos los usuarios van a poder subir documentos pertinentes al siniestro en formato PDF o JPG; pueden subir hasta 10 archivos por vez, pudiendo repetir el procedimiento tantas veces como sea necesario. Esto fue implementado de la misma forma que se explica en apartados anteriores

### **3.3.13 Descarga de los Endosos de la Póliza y la Denuncia del Siniestro**

Todos los usuarios pueden descargar todos los endosos de la póliza al momento del siniestro, y la denuncia de este. La generación del reporte se hace mediante Oracle Report; es una herramienta de

diseño de informes de Oracle que permite generar archivos PDF. Los reportes que se generan ya son utilizados en el sistema programado en PL/SQL y la implementación a la invocación de los diversos reportes ya estaba implementada en el sistema de siniestros, lo que hice fue adaptar la invocación existente, cambiando los parámetros y los nombres de los reportes, así generar el archivo que corresponda.

### 3.3.14 Listado de Novedades

Todos los usuarios, dependiendo de sus perfiles, pueden visualizar las novedades de todos los siniestros que están asociados, con la condición que no tengan más de 15 días de fecha de creación, accediendo a este desde la opción de menú listados – lista de novedades. Se muestran todos los tipos de novedades: “nuevo siniestro derivado” al perfil Liquidador Rs. Vs. cuando el Responsable Interno Rs. Vs. le asigna cada caso; “Nuevo Informe Preliminar cargado”, “Nuevo Informe Final cargado”, “Carga de factura” a los perfiles Responsable Interno Rs. Vs., siendo el responsable del siniestro en cuestión, y al Administrador Rs. Vs., cuando el Liquidador Rs. Vs. cargue un nuevo informe preliminar, cargue un nuevo informe final y cargue la factura al siniestro, respectivamente; y “Mensaje” cuando se le envía un mensaje al usuario, como se explicó en el apartado de Mensajería. Además, esta pantalla cuenta con la posibilidad de eliminar las novedades, pudiendo eliminarlas todas juntas o las que seleccione.

### 3.3.15 Estadísticas

En este apartado todos los perfiles pueden crear gráficos de barra o de torta en base a los casos que tienen asignados, siguiendo la lógica según cada perfil como se indicó anteriormente. Se muestra para elegir la fecha desde y hasta para tomar las estadísticas; los tipos de gráficos que se quiere generar; si las estadísticas son por ramo o por estado de los casos; y se dan a elegir en base a que ramos y estados se quiere generar el o los gráficos, se da la posibilidad de poder elegir todos los ramos y todos los estados al seleccionar la opción “Seleccionar todos”, ya que lo más común es que se saquen estadísticas en base a todos; pero, si quiere comparar la cantidad de siniestros de Incendio con respecto a Combinado Familiar para saber cuál es el porcentaje de casos en análisis de ambos, se puede hacer, como se muestra en la figura 3.17.

Los perfiles Responsable Interno Rs. Vs. y Administrador Rs. Vs. además de contar con lo anterior, también les aparece la opción para seleccionar un liquidador en particular; para el caso del primer

perfil, estará limitado a los liquidadores que tenga alguno de sus casos asignado, para el segundo, podrá elegir todos los liquidadores que haya.

Para la generación de los gráficos utilicé la librería jfreechart que cuenta con la funcionalidad para crear un gráfico de barra y de tortas, mediante las funciones `ChartFactory.createBarChart` y `ChartFactory.createPieChart3D` respectivamente. A estas funciones se le pasan como parámetro el título del gráfico, los valores a mostrar y algunas condiciones para mostrar los mismo; cómo, por ejemplo, el gráfico de barras grafique las barras en vertical. Al igual que se explica en el apartado Informe Preliminar, utilicé el servlet `showImageServlet` para visualizar los gráficos en la página que se obtienen a partir de las opciones seleccionadas, si es de barra, torta o ambos.

### Estadísticas

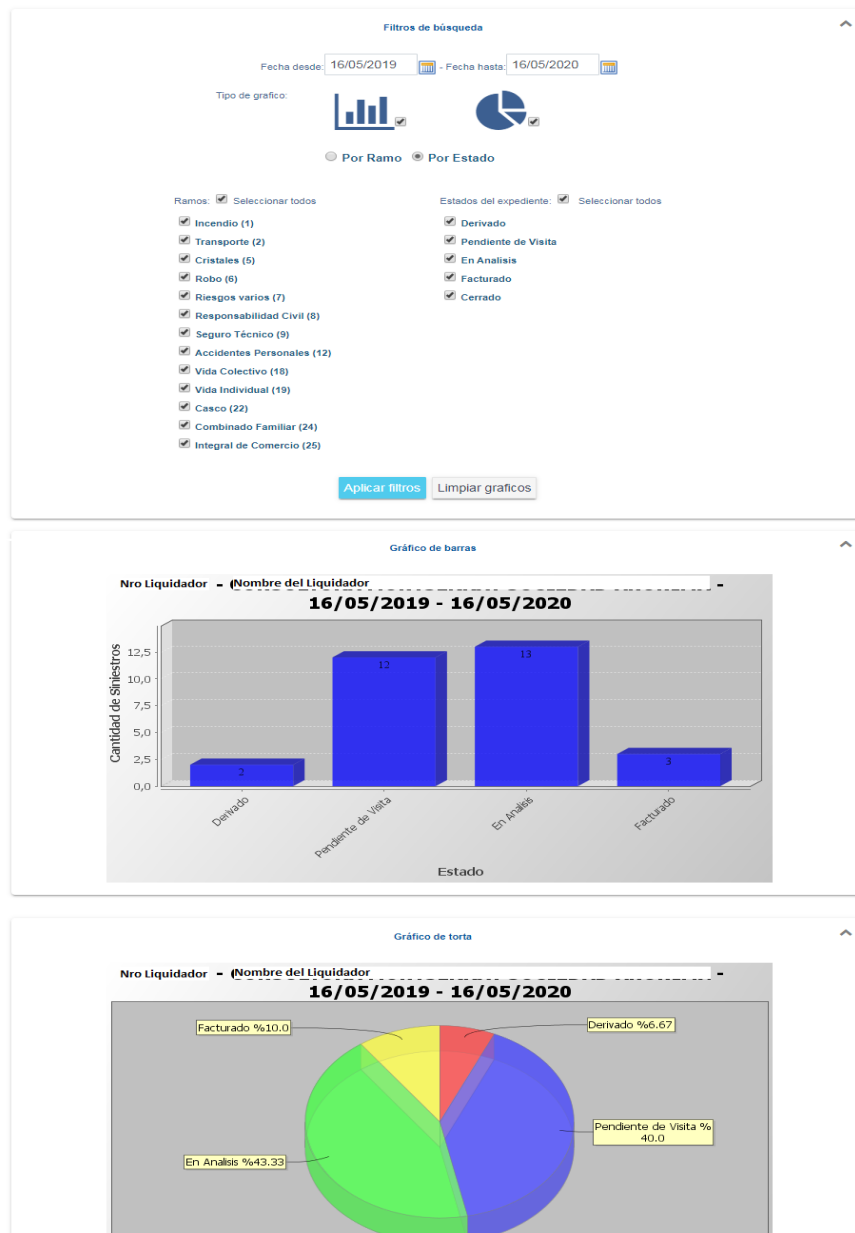


FIGURA 3.17 Estadísticas, ilustración propia

Al hacer “click” sobre los gráficos se muestra un listado con los siniestros discriminados según se indicó en el filtro de las estadísticas, de esta forma el usuario puede ingresar a un determinado siniestro a partir de la estadística generada, como se muestra en la figura 3.18.

Detalle de las estadísticas generadas por Estado: **Nro Liquidador - Nombre Liquidador**

Derivado (1), Cantidad encontrados: 2												
Año	Ramo	Siniestro	Póliza	Demora	Fe. Ocurrencia	Fe. Derivacion	Agencia	Sección	Causa Siniestro	Informe Preliminar	Informe Final	
año	ramo	siniestro	póliza	1	08/06/2018	15/05/2020	Agencia	INTEGRAL DE COMERCIO	OTRAS CAUSAS	❌	❌	
año	ramo	siniestro	póliza	1	29/03/2020	15/05/2020	Agencia	COMBINADO FAMILIAR	OTRAS CAUSAS	❌	❌	
Pendiente de Visita (2), Cantidad encontrados: 12												
En Analisis (3), Cantidad encontrados: 13												
Facturado (4), Cantidad encontrados: 3												
Año	Ramo	Siniestro	Póliza	Demora	Fe. Ocurrencia	Fe. Derivacion	Agencia	Sección	Causa Siniestro	Informe Preliminar	Informe Final	
año	ramo	siniestro	póliza	39	24/06/2017	07/04/2020	Agencia	RESPONSABILIDAD CIVIL	ROBO	✅	✅	
año	ramo	siniestro	póliza	33	03/12/2017	13/04/2020	Agencia	COMBINADO FAMILIAR	INCENDIO COBERTURA BASICA SIN RAYO	✅	✅	
año	ramo	siniestro	póliza	4	03/07/2019	12/05/2020	Agencia	COMBINADO FAMILIAR	OTRAS CAUSAS	✅	✅	

FIGURA 3.18 Detalle Estadísticas, ilustración propia

Dentro de este apartado se decidió hacer hincapié sobre los casos “demorados” de los Liquidadores de Rs. Vs., así poder tener un mejor control sobre los retrasos, indicado en los objetivos de esta tesina. Para esto, se muestra en la búsqueda de los siniestros, indicado en el apartado búsqueda de los siniestros Rs. Vs., un panel de control donde se visualizan 3 paneles mostrando la cantidad de casos demorados en el contacto con el asegurado, demorados en la carga del informe preliminar y demorados en la carga del informe final. Se visualiza el primer panel cuando haya siniestros en estado “derivado” o “pendiente de visita”, y hayan pasado 5 días de la fecha de derivación del caso; se visualiza el segundo panel cuando haya siniestros en estado “en análisis” y sin informe preliminar guardado, y hayan pasado 10 días de la fecha de derivación del caso; y se visualiza el tercer panel cuando haya siniestros en estado “en análisis” y sin informe final guardado (pero, si posea informe preliminar), y hayan pasado 30 días de la derivación del caso. Se informa, al posicionar el mouse sobre cada panel, la descripción de cada panel para que el usuario sepa a qué se debe cada uno, cómo se visualiza en la figura 3.19. Si el usuario no posee casos demorados, se muestra un mensaje de felicitaciones.

Siguiendo la lógica hasta ahora dicha, a cada perfil le aparecerá, lo explicado en el párrafo anterior, según los casos que tengan asignados.



FIGURA 3.19 Panel de Control, ilustración propia

### 3.3.16 Administración de usuarios

Sólo los perfiles administradores pueden administrar los usuarios, estos son: Administrador Liquidadores, Administrador Rs. Vs., Administrador Legales, Administrador Médicos y Super Administrador. Cada perfil administrador puede dar de alta los perfiles pertenecientes a su área, no así a otros administradores de su área, y el Super Administrador puede dar de alta todos los perfiles del sistema de siniestros. Los usuarios a administrar son empleados de la compañía y prestadores de servicios, estos pueden ser: Liquidador, Liquidador Rs. Vs., Médico, Auditor Médico, Abogado Legales, Abogado Penalista.

Los administradores disponen de una búsqueda por usuario, debe indicar el usuario y el tipo de usuario. Una vez que buscan un determinado usuario, se visualiza el nombre de este, el mail, la fecha del primer ingreso, la fecha de recuperación de clave, la fecha de expiración de clave, y los intentos fallidos de ingresos (figura 3.20). En esta pantalla se puede modificar el mail, setear la clave como clave inicial y administrar los perfiles del usuario; al indicar que se cambie a estado inicial, se genera una nueva clave actualizando la fecha de recupero de esta, y se envía un mail automático al usuario.

## Administración de usuarios

### Ingrese el usuario

Usuario:  Tipo:

### Resultado de la búsqueda

#### Detalle del Usuario

Nombre:

Mail:

Fecha 1er Ingreso:

Fecha Recupero de Clave:

Fecha Expiración de Clave:

Intentos Incorrectos:

Setear Clave Inicial:

#### Detalle de los Perfiles

- Visita
- Responsable Penalista
- Responsable Interno de Liquidadores
- Responsable Interno de Legales
- Responsable Interno de Rs. Vs.
- Responsable Médicos
- Administrador de Liquidadores
- Administrador de Rs. Vs.
- Administrador de Legales

FIGURA 3.20 Administración de Usuarios, ilustración propia

El envío de mail es algo que se disponía de antes, lo que hice fue adaptar la invocación a la función que envía el mail modificando el contenido de este, y enviarlo a la dirección de mail del usuario.

Por cada gestión de usuario se registra un log, si se realiza el alta o baja de perfiles: se guarda el usuario que inició sesión, los perfiles que fueron dados de alta o removidos; como así también, si se cambia el mail o se setea la clave en inicial: se guarda el usuario que inició sesión, el mail modificado

y a que mail se le envió la clave inicial. Todo lo indicado en este párrafo es por cuestiones de seguridad.

### **3.3.17 Seguridad en todas las páginas**

En el acceso a todos los apartados se valida que el usuario, que quiere realizar una determinada operación, está habilitado para realizarla. Primero que nada, se verifica que haya una sesión iniciada, y de ser así, se verifica que el perfil con el cual ingresó, cumpla los requisitos para visualizar la página solicitada; del mismo modo, una vez que pasó las validaciones, se mostrarán los datos pertenecientes al perfil, como se indicó en los diferentes apartados. Si el perfil no cumple para acceder a la página solicitada, se muestra un mensaje de error 403 (Forbidden), indicando que no tiene permiso para acceder a esa página. Por ejemplo, si un usuario con perfil Responsable Interno Rs. Vs. quiere ingresar a la página para administrar usuarios, le mostrará este mensaje de error.



# Capítulo 4

---

## Trabajos Futuros

### 4.1 Introducción

A continuación, se detallan los posibles trabajos futuros que se desprenden de lo presentado en esta tesina. Estos temas, por el momento, exceden los límites de alcance de este desarrollo, pero, sería útil su implementación para tener un sistema de siniestros aún más completo y auto gestionable; con el fin de poder independizar de ciertas tareas administrativas a los empleados de la compañía.

### 4.2 Ampliar el perfil Super Administrador

Actualmente se dispone de un perfil Super Administrador muy similar al Administrador de Rs. Vs., lo único en que se diferencian es en la administración de usuarios, como se explica en el capítulo anterior. Se deja la posibilidad de ampliar el perfil de este para administrar todo el sistema de siniestros, esto quiere decir, acceder a todas las funcionalidades de cada administrador del sistema de siniestros; siendo de gran utilidad para la alta gerencia.

Con este perfil, la alta gerencia podría ver todos los siniestros y litigios de todos los ramos y la gestión de todos los siniestros de los diferentes perfiles, pudiendo acceder a las estadísticas de los liquidadores, los médicos y los liquidadores de Rs. Vs. que cuenta cada perfil administrador. De esta forma, tendría centralizado todo lo relacionado a los siniestros en un solo perfil.

### **4.3 Agregar la suba de archivos digitales**

Hoy en día los perfiles Liquidador, Abogado Legales, Abogado Penalista, Médico y Auditor Médico no tiene la funcionalidad para subir documentación relevante a un siniestro del ramo 4, 44 y 80, así tampoco a los litigios relacionados a estos; para poder hacerlo, le deben enviar un mail a un empleado interno de la compañía y, éste por medio del sistema programado en PL/SQL, lo agrega al siniestro en cuestión. Si se agrega la posibilidad de adjuntar documentación, como se hizo en el apartado de digitalización desarrollado en esta tesina, los prestadores podrían adjuntar la documentación requerida sin pasar primero por un empleado interno, el cual puede estar ocupado atendiendo otras cuestiones referentes a su rol en la empresa.

### **4.4 Generación de órdenes de pago a los liquidadores**

En esta tesina se implementó un apartado para que los Liquidadores Rs. Vs. pudieran generar una orden de pago a los siniestros que han gestionado, para luego cargar su factura al mismo. Actualmente, los liquidadores de los ramos 4, 44 y 80 no pueden gestionar esta acción. Para poder cobrar sus honorarios, le deben enviar por mail la factura a un empleado interno de la compañía; éste, una vez que comprueba que el caso fue gestionado por el liquidador como se indica en el mail, genera una orden de pago por el sistema programado en PL/SQL y le asocia la factura enviada por mail. Lo anterior descrito, genera un retraso al momento de pagar los honorarios a los liquidadores, lo que genera descontento. Además, los empleados internos pierden tiempo investigando si efectivamente el liquidador que quiere facturar es el que llevó a cabo la gestión del siniestro.

#### **4.5 Blanqueo de clave por el usuario**

Se decidió que el blanqueo de clave la haga un administrador en vez de darle la posibilidad al usuario que quiere ingresar al sistema, es decir, de autogestionar su recupero de clave. A mi entender, como próxima mejora, sería funcional que el usuario, al olvidarse la clave, pueda recuperarla al ingresar a un link en el “login” que diga “¿Ha olvidado su contraseña?” para que valide el usuario y mail registrado, si son válidos se enviaría la clave inicial al usuario; sino, se le avisaría de la discrepancia en pantalla y, en ese caso, se avisaría a la compañía siendo el administrador quien gestione el cambio de mail de ser necesario.

# Capítulo 5

---

## Conclusiones

En esta tesina se llevó a cabo el análisis de la nueva funcionalidad solicitada por El Sector y el desarrollo de la ampliación del actual sistema de siniestros que dispone la compañía; permitiéndole al Liquidador Rs. Vs. las siguientes funcionalidades:

- Organizar todos los siniestros derivados para su gestión en un solo lugar, sin tener que revisar el historial de mail al querer recordar cuales siniestros debería atender.
- Dejar asentado cuando contactó al asegurado para encarar el siniestro.
- Cargar el Informe Preliminar, indicando si se solicita o no documentación adicional al asegurado.
- Cargar el Informe Final.
- Generar una orden de pago al siniestro.
- Cargar una factura a la orden de pago.
- Subir documentación relevante al siniestro.
- Consultar la información pertinente del siniestro.
- Enviar mensajes al Responsable Interno Rs. Vs. o al Administrador Rs. Vs. mediante la aplicación.
- Descargar todos los endosos de la póliza al momento del siniestro.
- Descargar la denuncia de la póliza.
- Realizar estadísticas de todos sus siniestros.

Y al empleado interno de la compañía, con perfil Administrador Rs. Vs. o Super Administrador, las siguientes funcionalidades:

- Hacer un seguimiento de todos los casos a los cuales debe atender.

- Enviar mensajes al Liquidador Rs. Vs. mediante la aplicación.
- Realizar estadísticas de todos los siniestros.
- Gestionar los perfiles de los usuarios, ya sean prestadores o empleados de la compañía.
- Cambiar el mail y blanquear la clave de los usuarios.

Se llegaron a los resultados esperados en los objetivos, el liquidador tiene a disposición un sistema para centralizar todos sus casos y la compañía dispone de un sistema que le permite tener un mayor control sobre los liquidadores y su gestión de los casos.

En el transcurso del análisis y desarrollo de esta tesina se han ido adaptando las demandas a lo propuesto por la compañía, solicitando los permisos para cambiar ciertos aspectos y hacerlos más ágiles. Esta forma de trabajo, le pareció más óptima al negocio, teniéndola como referente para futuros cambios. Esto se debe a que tuvieron un sistema correcto, confiable, eficiente y amigable en el plazo de cinco meses. Así mismo, fueron parte del desarrollo al poder opinar y mejorar ciertas cuestiones en simultáneo, sin tener que esperar a hacer el test al final de todo el proyecto y volver a generar otro cambio si había algo que no respondía a las necesidades.

Adicionalmente cabe destacar el contexto en el cual se realizó esta tesina: los dos primeros meses de trabajo tuvieron lugar yendo a la compañía con regularidad de lunes a viernes en el horario laboral correspondiente; a la mitad del desarrollo de este trabajo, la Organización Mundial de la Salud (OMS) declara la pandemia por el brote del Covid-19 (o también llamado Coronavirus) y, ante la presencia de varios casos importados del virus en el país, el gobierno nacional decreta el Aislamiento Social, Preventivo y Obligatorio para frenar el avance del virus. Ante la repentina cuarentena, la compañía decide implementar la modalidad de trabajo “home office” para poder seguir con las actividades laborales de sus empleados. Gracias a esto y a la pronta respuesta del personal del sector Soporte Técnico, pude continuar con el desarrollo propuesto para esta tesina.

La segunda mitad del desarrollo fue más compleja y difícil ya que no se pudo sostener las reuniones y demos del sistema como se hizo en la primera mitad del estudio. En su lugar, realicé videoconferencias y envié capturas de pantalla o video tutoriales para mostrar el avance del sistema y las nuevas funcionalidades implementadas.

Al día de la entrega de esta tesina se encuentra todo lo desarrollado en PROD, funcionando perfectamente. A tres meses de la puesta en PROD, sólo se detectó una falla al momento de cargar una factura de tipo “Nota de Crédito”, debido a la falta de la prueba necesaria cuando se encontraba en TEST, por lo poco frecuente que se factura por este tipo.

## Bibliografía

- [1] *What is Java?* (s.f.). Recuperado el 25 de febrero de 2020, de [https://www.java.com/es/download/faq/whatis\\_java.xml](https://www.java.com/es/download/faq/whatis_java.xml)
- [2] *JavaServer™ Faces*. (s.f.). Recuperado el 25 de febrero de 2020, de [http://www.java.com/es/download/faq/whatis\\_java.xml](http://www.java.com/es/download/faq/whatis_java.xml)
- [3] Leathem B., Fryc L. y Rogers S. (22 de febrero de 2011). *jboss: Developer Guide*. Recuperado el 25 de febrero de 2020, de [https://docs.jboss.org/richfaces/latest\\_4\\_5\\_X/Developer\\_Guide/en-US/html/index.html](https://docs.jboss.org/richfaces/latest_4_5_X/Developer_Guide/en-US/html/index.html)
- [4] Bertone R. y Thomas P. (2011). *Introducción a las Bases de Datos. Fundamentos y Diseño*, primera edición. Buenos Aires, Argentina: Pearson Prentice Hall de Argentina.
- [5] Erl T., Carlyle B., Pautasso C., y Balasubramanian R. (2013). *SOA with REST. Principles, Patterns & Constraints for Building Enterprise Solutions with REST*, primera edición. New Jersey, Estados Unidos: Pearson Prentice Hall.
- [6] Pressman, R. S. (2010). *Software Engineering: A practitioner's approach*, séptima edición. New York, Estados Unidos: McGraw-Hill.
- [7] Pfleeger S. L. y Atlee J. M. (2010). *Software Engineering. Theory and Practice*, cuarta edición. New Jersey, Estados Unidos: Pearson Prentice Hall.
- [8] Sommerville I. (2005). *Ingeniería del software*, Séptima edición. Madrid, España: Pearson Educación de España.
- [9] Pérez E. L. y Ariza J. (11 de marzo de 2014). *Wordpress: DSDM Dynamic Systems Development Method*. Recuperado el 27 de marzo de 2020, de <https://metodologiadsdm.wordpress.com/2014/03/11/>
- [10] Calderón A., Dámaris S., Rebaza V. y Carlos J. (2007). *Metodologías Ágiles*. Trujillo, Perú: Universidad Nacional de Trujillo Facultad de Ciencias Físicas y Matemáticas.