



Manual de Usuario

Trabajo de tesina

Autor
Erasmo Jordán

Profesor
Mario Cárdenas



Aseguradora
Sociedad Anónima

Manual de Software para el Usuario



LiceoInformático II
Colegio Universitario

Contexto

Para el usuario

La aplicación Aseguradora es un sistema concebido para registrar, analizar y presentar datos relacionados con la situación y las necesidades en materia de las empresas de rubro asegurador.

Es fácil aprender a usar este software, pero se asume que el usuario estará ya familiarizado con los términos, conceptos y métodos presentados en los Indicadores de carga de trabajo del personal necesario: Capítulo segundo – Modelo de negocio. Se debe estudiar detenidamente dicho capítulo antes de empezar a usar el software.

En esta guía del usuario se ofrece una visión general de las características de la aplicación y se indican las instrucciones que deben seguirse paso a paso para realizar diversas tareas.

¿Cómo se organiza la documentación?

El índice de la documentación es conformado por una estructura jerárquica en la que, a medida que se avance, se irán sumando tantos niveles de contenido como sea necesario. El máximo de niveles con el que nos encontraremos en el manual es de siete. A continuación, se detallan la siguiente representación:

Representación gráfica	Ejemplo
Título del capítulo	Capítulo 3º: Guía de procedimientos
➤ 1.1 Nivel 2	3.3 Utilizar el sistema
★ 1.1.1 Nivel 3	3.3.6 Pagos
➡ 1.1.1.1 Nivel 4	3.3.6.1 Facturar
* 1.1.1.1.1 Nivel 5	3.3.6.1.1 Consultar facturas de pago
> 1.1.1.1.1.1 Nivel 6	3.3.6.1.1.1 Emitir recibo
- 1.1.1.1.1.1.1 nivel 7	-

Índice

1 ANÁLISIS	1
1.1 ¿EN QUÉ CONSISTE UNA ASEGURADORA?	4
1.2 RELEVAMIENTO	4
1.2.1 Identificación del problema	5
1.2.2 Cuestionario	6
1.2.3 Encuesta	8
1.3 OBJETIVOS DEL SISTEMA	10
1.3.1 Objetivos generales	10
1.3.2 Objetivos específicos	10
1.4 DIAGNOSTICO	10
1.4.1 Propuesta narrativa	11
1.4.1.1 Prestaciones	11
1.5 ESTUDIO DE FACTIBILIDAD	12
1.5.1 Factibilidad técnica	12
1.5.1.1 Infraestructura de red	12
1.5.1.2 Software	12
* 1.5.1.2.1 Lenguaje de programación	12
* 1.5.1.2.2 Servicio web	13
* 1.5.1.2.3 Sistema gestor de bases de datos	13
* 1.5.1.2.4 Software utilizado para el desarrollo	14
1.5.1.3 Hardware	14
* 1.5.1.3.1 Apache	14
* 1.5.1.3.2 MySQL	14
* 1.5.1.3.3 Requerimientos mínimos para computadoras de desarrollo	14
* 1.5.1.3.4 Hardware para la implementación del proyecto	16
* 1.5.1.3.5 Requerimientos mínimos de computadoras de usuarios	17
1.5.1.4 Recurso humano	17
* 1.5.1.4.1 Estructura del grupo de trabajo	18
1.5.1.5 Conclusión de la factibilidad técnica	18
1.5.2 Factibilidad económica	18
1.5.2.1 Beneficios	18
* 1.5.2.1.1 Beneficios no cuantificables	19
* 1.5.2.1.2 Beneficios cuantificables	19
1.5.2.2 Costos del sistema a desarrollar	19
* 1.5.2.2.1 Costos no cuantificables	19
* 1.5.2.2.2 Costos cuantificables	19
2 MODELO DE NEGOCIO	25
2.1 GLOSARIO DEL SISTEMA	27
2.1.1 Pólizas	28
2.1.1.1 ¿Qué es una póliza?	28
2.1.1.2 Número o identificador de póliza	28
2.1.1.3 Estados de una póliza	28
2.1.1.4 ¿Qué es la vigencia de la póliza?	28

2.1.1.5 ¿Que es el tomador?	29
2.1.1.6 ¿Que es el número o identificador del tomador?	29
2.1.1.7 ¿Que es el bien asegurado?	29
2.1.1.8 ¿Que es la suma asegurada?	29
2.1.1.9 ¿Que es la cobertura?	29
2.1.1.10 ¿Que es la cobertura adicional?	29
2.1.1.11 ¿Que significa renovación de una póliza?	30
2.1.2 Endosos	30
2.1.2.1 ¿Que es un endoso?	30
2.1.2.2 Número o identificador del endoso	30
2.1.2.2 Tipos de endosos	30
2.1.3 Siniestros	30
2.1.3.1 ¿Que es un siniestro?	30
2.1.3.2 Número o identificador del siniestro	31
2.1.3.3 Estados del siniestro	31
2.1.3.4 Fecha de denuncia del siniestro	31
2.1.3.5 Fecha y hora de ocurrencia del siniestro	31
2.1.3.6 Que significa lesión o muerte	31
2.1.3.7 Tipo de siniestros	31
2.1.3.8 ¿Que es el denunciante?	32
2.1.3.9 ¿Que es el reclamante?	32
2.1.3.10 ¿Que es el conductor?	32
2.1.4 Pagos	32
2.1.4.1 ¿Que es un pago?	32
2.1.4.2 Número de pago	32
2.1.4.3 Estados del pago	32
2.1.4.4 ¿Que es la prima total?	33
2.1.4.5 Fecha de registro del pago	33
2.1.4.6 Método de pago	33
2.1.4.7 ¿Que es una cuota?	33
2.1.4.8 ¿Que es el número de cuota?	33
2.1.4.9 ¿Que es el identificador de cuota?	33
2.1.4.10 ¿Que es la prima mensual?	33
2.1.4.11 ¿Que es el lapso de una cuota?	34
2.1.4.12 Fecha de facturación de una cuota	34
2.1.4.13 Fecha de vencimiento de facturación de una cuota	34
2.1.4.14 Estados de una cuota	34
2.1.5 Usuarios	34
2.1.5.1 ¿Que es un usuario?	34
2.1.5.2 ¿Que es el número o identificador de usuario?	34
2.1.5.3 Estados de un usuario	35
2.1.5.4 ¿Que es el nombre de usuario?	35
2.1.5.5 ¿Que es la contraseña?	35
2.1.5.6 ¿Que es el rol del usuario?	35
2.1.6 Sucursales	35
2.1.6.1 ¿Que es una sucursal?	35
2.1.6.2 ¿Que es el numero o identificador de una sucursal?	35
2.2 DESCRIPCIÓN DE PATRONES DE FUNCIONAMIENTO	36
2.2.1 Patrón de funcionamiento de pólizas	36
2.2.1.1 Patrones de estados de pólizas	36
2.2.1.2 Patrones de renovación de póliza	37
2.2.2 Patrón de funcionamiento de pagos	37
2.2.2.1 Patrones de modificación de pagos	38

2.2.2.2 Patrones de estados de pagos	38
2.2.2.3 Patrones de estados de cuotas	39
2.2.3 Patrón de funcionamiento de siniestros	39
2.2.3.1 Patrones de estados de siniestros	40
3 GUÍA DE PROCEDIMIENTOS	41
3.1 INTERFACES DE NAVEGACIÓN	45
3.1.1 Formulario de acceso al sistema	45
3.1.2 Barra superior	45
3.1.2.1 Menú de acciones de usuario	46
3.1.3 Menú principal	46
3.1.4 Contenedor general	47
3.1.4.1 Titulo de navegación	47
3.1.4.2 Subtitulo de navegación	47
3.1.4.3 Cuadro de interacción	47
3.1.5 Mensajes y alertas de interacción	47
3.1.5.1 Mensaje de éxito	48
3.1.5.2 Mensaje informativo	48
3.1.5.3 Mensaje de error	48
3.2 INTERACCIÓN CON LAS INTERFACES DE NAVEGACIÓN	50
3. 2.1 Ingresar al sistema	50
3.2.2 Menú de acciones	52
3.2.2.1 Administrar foto de perfil	52
3.2.2.2 Administrar mis datos	53
* 3.2.2.2.1 Modificar información general	54
* 3.2.2.2.2 Modificar contraseña	55
3.2.2.3 Salir del sistema	56
3.2.3 Menú principal	56
3.3 UTILIZAR EL SISTEMA	56
3.3.1 Inicio	56
3.3.2 Pólizas	57
3.3.2.1 Emitir nueva póliza	57
* 3.3.2.1.1 Sección póliza	59
* 3.3.2.1.2 Sección tomador	60
* 3.3.2.1.3 Sección vehículo	62
* 3.3.2.1.4 Sección cobertura	64
* 3.3.2.1.5 Sección pago.	65
3.3.2.2 Consultar pólizas	66
* 3.3.2.2.1 Ver la información de la póliza	67
3.3.2.3 Renovar póliza	70
* 3.3.2.3.1 Proceso de renovación denegado	70
* 3.3.2.3.2 Proceso de renovación	71
3.3.3 Endosos	73
3.3.3.1 Emitir un endoso	73
* 3.3.3.1.1 Modificación de riesgo	75
* 3.3.3.1.2 Modificación de datos nominales	78
* 3.3.3.1.3 Modificación de datos del vehículo	79
* 3.3.3.1.4 Modificación de asegurado	81
* 3.3.3.1.5 Cambio de cobertura	82
3.3.3.2 Consultar endosos	83
* 3.3.3.2.1 Ver endosos de una póliza	84

* 3.3.3.2.2 Modificar la descripción de un endoso	85
3.3.4 Siniestros	86
3.3.4.1 Emitir nuevo siniestro	86
* 3.3.4.1.1 Información general	88
* 3.3.4.1.2 Denunciante	90
* 3.3.4.1.3 Accidente	91
* 3.3.4.1.4 Reclamante tercero	93
* 3.3.4.1.5 Conductor tercero	93
3.3.4.2 Administrar siniestros	94
* 3.3.4.2.1 Modificar estado del siniestro	95
* 3.3.4.2.2 Modificar información del siniestro	96
* 3.3.4.2.3 Administrar imágenes	97
3.3.4.3 Consultar siniestros	101
3.3.5 Tomadores	102
3.3.5.1 Agregar nuevo tomador	103
3.3.5.2 Administrar tomadores	104
* 3.3.5.2.1 Modificar información	105
* 3.3.5.2.2 Ver información	106
* 3.3.5.2.3 Dar de baja del sistema	106
3.3.6 Pagos	107
3.3.6.1 Facturar	108
* 3.3.6.1.1 Consultar facturas de pago	109
* 3.3.6.1.2 Consultar recibos de pagos caducados	115
3.3.6.2 Modificar cuotas	117
3.3.7 Administrador	121
3.3.7.1 Gestionar usuarios	121
* 3.3.7.1.1 Nuevo usuario	122
* 3.3.7.1.2 Listado de bajas	125
* 3.3.7.1.3 Dar de baja	126
* 3.3.7.1.4 Modificar rol	127
* 3.3.7.1.5 Reenviar datos de acceso a usuarios	128
* 3.3.7.1.6 Consultar información	129
3.3.7.2 Gestionar sucursales	130
* 3.3.7.2.1 Agregar nueva sucursal	131
* 3.3.7.2.2 Consultar información de una sucursal	133
* 3.3.7.2.3 Modificar información de una sucursal	134
* 3.3.7.2.4 Eliminar una sucursal	134
4 DOCUMENTACIÓN TÉCNICA	136
4.1 DIAGRAMAS DEL SISTEMA	138
4.1.1 Diagrama de relación de entidades	138
4.1.2 Diagrama de flujo de datos	139
4.1.2.1 D.F.D. Nivel 1 Pólizas (1)	141
4.1.2.2 D.F.D. Nivel 1 Endosos (2)	145
4.1.2.3 D.F.D. Nivel 1 Siniestros (3)	148
4.1.2.4 D.F.D. Nivel 1 Tomadores (4)	152
4.1.2.5 D.F.D. Nivel 1 Pagos (5)	156
4.1.2.6 D.F.D. Nivel 1 Usuarios (6)	159
4.1.2.7 D.F.D. Nivel 1 Sucursales (7)	163
4.1.3 Diagrama de clases	167
4.1.4 Diagrama de casos de uso (C.U)	168
4.1.5 Diagrama de secuencias	169

4.1.5.1 Diagrama de secuencia para C.U. cargar/eliminar foto de perfil	170
4.1.5.2 Diagrama de secuencia para C.U emitir póliza:	172
4.1.5.3 Diagrama de secuencia para C.U renovar póliza:	173
4.1.5.4 Diagrama de secuencia para C.U emitir endoso:	174
4.1.5.5 Diagrama de secuencia para C.U cargar siniestro:	175
4.1.5.6 Diagrama de secuencia para C.U modificar siniestro:	176
4.1.5.7 Diagrama de secuencia para C.U agregar tomador:	177
4.1.5.8 Diagrama de secuencia para C.U modificar tomador:	178
4.1.5.9 Diagrama de secuencia para C.U efectuar pago:	179
4.1.5.10 Diagrama de secuencia para C.U enviar recibo:	180
4.1.5.11 Diagrama de secuencia para C.U modificar pago:	181
4.1.5.12 Diagrama de secuencia para C.U agregar usuario:	182
4.1.5.13 Diagrama de secuencia para C.U alta/baja usuario:	183
4.1.5.14 Diagrama de secuencia para C.U modificar rol usuario:	185
4.1.5.15 Diagrama de secuencia para C.U agregar sucursal:	186
4.1.5.16 Diagrama de secuencia para C.U modificar sucursal:	187
4.2 LISTADO DE TABLAS DE LA BASE DE DATOS	188
4.3 CÓDIGO COMENTADO	201
4.3.1 Carpeta “config”	202
4.3.2 Carpeta “controlador”	203
4.3.2.1 fichero “endososController.php”	204
4.3.2.2 fichero “funcionesGlobalesController.php”	209
4.3.2.3 fichero “pagosController.php”	213
4.3.2.4 fichero “polizasController.php”	234
4.3.2.5 fichero “siniestrosController.php”	263
4.3.2.6 fichero “sistemaController.php”	284
4.3.2.7 fichero “sucursalesController.php”	287
4.3.2.8 fichero “tomadoresController.php”	293
4.3.2.9 fichero “usersController.php”	302
4.3.3 Carpeta “js”	318
4.3.3.2 fichero “endosos.js”	331
4.3.3.3 fichero “login.js”	348
4.3.3.4 fichero “menu.js”	352
4.3.3.5 fichero “pago.js”	357
4.3.3.6 fichero “poliza.js”	379
4.3.3.7 fichero “siniestros.js”	403
4.3.3.8 fichero “sucursales.js”	433
4.3.3.9 fichero “users.js”	441
4.3.4 Carpeta “scripts”	462
4.3.4.1 Carpeta “emails”	462
4.3.4.2 Carpeta “imagenes”	469
4.3.4.3 Carpeta “PDFs”	473
4.3.4.4 Carpeta “tablas”	474
4.3.4.5 Fichero “aseguradora.php”	498
4.3.5 Carpeta “vista”	506
4.3.5.1 Carpeta “endosos”	506
4.3.5.2 Carpeta “maqueta”	518
4.3.5.3 Carpeta “pagos”	530
4.3.5.4 Carpeta “poliza”	546
4.3.5.5 Carpeta “siniestro”	564
4.3.5.6 Carpeta “sucursales”	581
4.3.5.7 Carpeta “tomador”	587

4.3.5.8 Carpeta “users”	592
4.3.6 Fichero “index.php”	601
4.3.7 Fichero “indexMaqueta.php”	602

Capítulo primero

1º

Análisis

Contenido

1.1 ¿En qué consiste una aseguradora?

1.2 Relevamiento

 1.2.1 Problema

 1.2.2 Cuestionario

 1.2.3 Encuesta

1.3 Objetivos de sistema

 1.3.1 Objetivos generales

 1.3.2 Objetivos específicos

1.4 Diagnóstico

 1.4.1 Propuesta narrativa

 1.4.1.1 Prestaciones

1.5 Estudio de factibilidad

 1.5.1 Factibilidad técnica

 1.5.1.1 Infraestructura de red

 1.5.1.2 Software

 1.5.1.2.1 Lenguaje de programación

 1.5.1.2.2 Servidor web

 1.5.1.2.3 Sistema gestor de bases de datos

 1.5.1.2.4 Software utilizado

 1.5.1.3 Hardware

 1.5.1.3.1 Apache

 1.5.1.3.2 MySql

 1.5.1.3.3 Requerimientos mínimos de pcs. de desarrollo

 1.5.1.3.4 Hardware para la implementación del proyecto

 1.5.1.3.5 Requisitos mínimos de computadoras de usuarios

 1.5.1.4 Recurso humano

 1.5.1.4.1 Estructura de grupo de trabajo

 1.5.1.5 Conclusión de la factibilidad técnica

 1.5.2 Factibilidad económica

 1.5.2.1 Beneficios

 1.5.2.1.1 Beneficios no cuantificables

 1.5.2.1.2 Beneficios cuantificables

- 1.5.2.2 **Costos del sistema a desarrollar**
 - 1.5.2.2.1 **Costos no cuantificables**
 - 1.5.2.2.2 **Costos cuantificables**
 - 1.5.2.2.2.1 **Costos del desarrollo del proyecto**
 - 1.5.2.2.2.1.1 **Costos de recursos humanos**
 - 1.5.2.2.2.1.1.1 **Recurso humano**
 - 1.5.2.2.2.1.1.2 **Supuestos**
 - 1.5.2.2.2.1.2 **Costos de recursos tecnológicos**
 - 1.5.2.2.2.1.2.1 **Costos hardware**
 - 1.5.2.2.2.1.2.2 **Costos software**
 - 1.5.2.2.2.1.2.3 **Costos fijos**
 - 1.5.2.2.2.1.2.4 **Imprevistos**
 - 1.5.2.2.2.2 **Costos de operación del sistema**
 - 1.5.2.2.2.2.1 **Costos de hardware**
 - 1.5.2.2.2.2.2 **Costos de software**
 - 1.5.2.2.2.2.2.1 **Computadoras de usuarios**
 - 1.5.2.2.2.2.3 **Costos de mantenimiento**
 - 1.5.2.2.2.2.3.1 **Recursos humanos**
 - 1.5.2.2.2.2.3.2 **Costos fijos**

➤ 1.1 ¿En qué consiste una aseguradora?

Una aseguradora o compañía de seguros es una empresa que se encarga de asegurar riesgos a terceros, de tal manera que protege o resguarda los bienes materiales de los riesgos a los que estos están expuestos.

La misión principal de una aseguradora es proteger desde el aspecto económico aquellos bienes que son potencialmente dañables y se encuentran bajo algún tipo de riesgo, como puede ser una vivienda, un automóvil o incluso a las propias personas físicas con los seguros de vida. De este modo, si ocurre algún accidente o eventualidad a lo protegido será la compañía quien se encargue de subsanar o reparar los daños.

A cambio de este tipo de protección la compañía de seguros exige unos pagos periódicos, también conocidos como cuotas o primas, para que esta cobertura tenga lugar. El importe de estos abonos suele ser determinado por el nivel de riesgo que calcule la compañía al ofrecer cobertura ante daños, lesiones o pérdidas.

A través de un contrato con sus correspondientes cláusulas y condiciones, la aseguradora y el asegurado pactan las condiciones en las que se lleva a cabo esta relación de protección frente a riesgos y la manera en que la parte aseguradora arregla posibles daños o paga una cantidad de dinero a la otra parte en caso de accidentes u otras situaciones. A este documento se le llama también póliza de seguros.

Para que una aseguradora tenga la posibilidad de acometer gran número de contratos y proteger un alto número de cuentas y clientes, es necesario que cuente con una considerable reserva para así tener un alto margen de solvencia ante futuros pagos propios de su actividad económica. Es por ello que el mercado de las compañías de seguros cuenta con un gran tamaño e importancia, siendo un pilar del mercado financiero junto a las entidades bancarias.

➤ 1.2 Relevamiento

A continuación, se detallará el contenido con el que se lleva a cabo el proceso de relevamiento del sistema de aseguradora.

1.2.1 Identificación del problema

- 1-Gran cantidad de información en papel
 - 2-Mayor margen de error al manipular la información
 - 3-No quedan registrados el 100% de las modificaciones de los datos
 - 4-Mayor dificultad para llevar a cabo los cambios de datos
 - 5-Mayor dificultad para acceder a determinada información
 - 6-Mayores gastos económicos para la empresa
- 1- Toda la información que tenga que ser registrada en la empresa, será escrita en papel, lo que es una técnica poco ágil.
- 2- Cuando se desea acceder a cierta información, es más difícil y demanda mayor tiempo ubicar ese registro, además de que estas técnicas son muy propensas a provocar que las personas olviden la inclusión de ciertos datos.
- 3- Usando la técnica de registro en papel, puede suceder con más frecuencia que algunos datos no sean incluidos dentro de la información correspondiente.
- 4- Cuando se quiere realizar una modificación, por más mínima que sea, se debe de reescribir todo el registro y descartar el anterior.
- 5-Uno de las principales dificultades es la búsqueda de información, y de no tener un orden de los registros muy estudiado y organizado, el trabajo del personal puede tornarse muy tardado y engorroso.
- 6-El hecho de tener registrado en papel cada una de las acciones que se realiza en la empresa, conlleva un gasto extra en elementos para la impresión, el papel y el personal encargado.

*¿Qué es un Sistema de Gestión Informático?

Un sistema de gestión informático es una técnica que permite el almacenamiento y el proceso de información, para lo cual se vale de un grupo de elementos que se relacionan entre sí: hardware, software y personal informático. Los sistemas de gestión hacen referencia a todos aquellos sistemas informáticos con los que se apoya una empresa para poder desarrollar correctamente las actividades de la misma.

1.2.2 Cuestionario

1. ¿Conoce Usted que es un Sistema de Gestión Informático?

Si No (Leer *)

2. ¿Alguna vez ha manejado algún sistema de Gestión?

Si No

3. ¿Cree usted que necesita un analista o persona capacitada para ayudarlo?

Si No

4. ¿Usted cree que necesitaría algún curso rápido de manejo como administrador del Sistema?

Si No

5. ¿Considera que el personal se adaptara fácilmente al nuevo sistema?

Si No

6. ¿Cree que el personal se beneficiaría con algún curso previo operativo del sistema?

Si No

7. ¿El Sistema actual que utiliza es netamente manual?

Si No

Otros: _____

8. ¿Qué tipo de conocimientos tiene sobre navegación web?

Ninguno

Básico

Intermedio

Avanzado

9. ¿Tiene acceso a Internet?

Si No

10. ¿Qué tipo de acceso posee a Internet?

ADSL

Cable Modem

3G/4G

Otros: _____

11. ¿Cuál es la velocidad de su conexión a Internet?

1 Mb

2Mb

3Mb

5Mb

Otros: _____

12. ¿Usted cree que la labor de su empresa será más fácil de llevar a cabo con un sistema de gestión informático?

Si No

13. ¿Cree usted que la implementación del nuevo sistema informático optimizara el funcionamiento de su empresa?

Si No

14. ¿Cree Usted que la implementación del sistema le será rentable económicoamente?

Si No

15. ¿Cree que la calidad del servicio para sus clientes mejorará con un Sistema de Gestión Informático?

Si No

1.2.3 Encuesta

1. ¿Cuál es el fin de su empresa?

2. ¿Cómo funciona su empresa?

3. ¿Qué tipo de dificultades se encuentran en cada área de trabajo?

4. ¿Con que finalidad necesitaría el Sistema?

5. ¿Cuál es la división de áreas o sectores en su empresa?

6. ¿Qué sector considera más crítico?

7. ¿Qué método utiliza actualmente para el registro de la información?

8. ¿Actualmente tiene un registro de todas las acciones realizadas en su empresa?

9. ¿Actualmente como hace para verificar los detalles de la información manipulada en el pasado?

10. ¿Desea que el sistema guarde otro tipo de información? ¿De qué tipo?

➤ 1.3 Objetivos del sistema

Los objetivos pueden subdividirse en dos tipos, a continuación, serán especificados:

✳ 1.3.1 Objetivos generales

Implementar un sistema informático para empresas aseguradoras orientadas principalmente a vehículos, que reúna las principales funcionalidades de la misma y que permita al usuario poder gestionarlas acorde a su modelo de negocio de manera más rápida, flexible y cómoda.

✳ 1.3.2 Objetivos específicos

- * Determinar los atributos necesarios a registrar dentro del sistema
- * Elaborar una descripción detallada del sistema propuesto
- * Desarrollar la interfaz del sistema de información
- * Elaborar un manual de usuario para el manejo del sistema

➤ 1.4 Diagnóstico

Un diagnóstico de gestión es un proceso de evaluación de la gestión de una determinada organización contrastándolo con los requisitos que figuren en una norma o en los requisitos de la propia organización.

La finalidad de un diagnóstico de gestión es obtener una imagen sobre el grado de cumplimiento de los requisitos y una serie de orientaciones para la mejora de esa gestión.

1.4.1 Propuesta narrativa

Una empresa de rubro asegurador se somete a muchos procesos y a la manipulación de información de gran escala, por más básico que sea el servicio que ofrezca. Aquellas pequeñas aseguradoras que quieran integrarse en el mercado deben tener un sistema informático para lidiar de la mejor manera posible con este problema, y de no ser así, la compañía se vería afectada por una saturación de datos e información en papel, lo que genera una mayor pérdida de tiempo en la realización de muchas tareas.

La idea principal es dejar asentados todos los archivos de datos en la web para llevar a cabo la manipulación de la información de manera rápida y fácil, así evitando y dejando atrás los métodos anticuados de utilización de carpetas y libros físicos para hacerlo.

1.4.1.1 Prestaciones

El sistema permitirá administrar todos los datos de la empresa y sabrá gestionarlos de la manera que el usuario necesite para realizar una acción.

Principales funcionalidades:

Inicio de sesión por roles (administrador y usuarios), limitando las funcionalidades según corresponda.

Gestionar la emisión de una póliza, ingresando datos del tomador, del vehículo, el tipo de cobertura y las condiciones de pago.

Brindar información de las pólizas, que permitirá visualizar su estado, el número, a que tomador corresponde, el tipo de cobertura, el estado del pago y las cuotas, los endosos y siniestros realizados e información del riesgo.

Gestionar renovaciones de pólizas vencidas.

Gestionar datos del contrato de póliza (Endosos), como modificación de riesgo, datos nominales del riesgo, asegurados y coberturas.

Gestionar el registro y almacenamiento de un siniestro y todos sus detalles.

Brindar información de todos los siniestros registrados y sus estados.

Modificar toda la información permitida de un siniestro, como poder cargar una imagen, modificar el estado en que se encuentra, así como el resto de los datos.

Consultar los siniestros registrados, brindando toda la información detallada de cada uno y el estado en que se encuentra.

Listado de tomadores registrados, pudiendo crear nuevos registros, consultar su información, modificarla o eliminar del sistema.

Sección de pago para para las pólizas, en la que podrán establecerse un numero de cuotas con sus correspondientes lapsos de facturación y fechas vencimiento y anulación.

Gestión de usuarios del sistema, permitiendo agregar nuevos, dar de baja y alta, modificar su rol, y consultar su información.

➤ 1.5 Estudio de factibilidad

En esta instancia se logra determinar si el sistema que se solicito es factible. Para llevar a cabo este proceso se toma en cuenta:

- Factibilidad Técnico.
- Factibilidad Económico.

✳ 1.5.1 Factibilidad técnica

El objetivo de la factibilidad técnica es verificar si la capacidad tecnológica que se tiene a disposición puede ser considerada y aprovechada para llevar a cabo el desarrollo del proyecto.

➤ 1.5.1.1 Infraestructura de red

La infraestructura requerida es básicamente contar con tecnología ethernet y servicio web.

➤ 1.5.1.2 Software

Esto es una evaluación para el desarrollo e implementación que se lleva a cabo estableciendo el más conveniente.

* 1.5.1.2.1 Lenguaje de programación

Los lenguajes de programación que se utilizan son los mencionados a continuación, teniendo en cuenta que no son propietarios y con licencia gratuita, y pueden trabajar bajo cualquier sistema operativo: PHP, HTML, JAVASCRIPT, JQUERY.

* 1.5.1.2.2 Servicio web

PHP es un lenguaje que requiere de un servidor web para ser interpretado, por ello es que se realizara una evaluación para determinar cuál es el idóneo, y para esto los que se tomaran en cuenta son Microsoft IIS y Apache, ya que son los dos más usados por excelencia en la web.

	IIS	Apache
Soporte para Sistema Operativo	Windows	Linux, Unix, Windows, Mac OS
Soporte	Corporativo	Comunidad
Costo	Gratis solo en Windows	Gratis
Desarrollo	Propietario	Open source
Seguridad	Excelente	Buena
Performance	Buena	Buena
Participación en el Mercado	30 %	35%

* 1.5.1.2.3 Sistema gestor de bases de datos

La evaluación del Sistema Gestor de Bases de Datos se realizará tomando en cuenta los más populares como ORACLE, SQL SERVER y MySQL.

	ORACLE	MySQL	SQL Server
Interfaz	GUI, SQL	SQL	GUI, SQL
Lenguajes Soportados	C, C#, C++, Java, Ruby	Ruby, Objetive C y PHP	Visual Basic, .NET y PHP
Sistemas Operativos	Windows, Linux, Solaris, HP-UX, OS X, z/OS, AIX	Windows, Linux, OS X, FreeBSD, Solaris	Windows
Licencia	Propietario	Código Libre	Propietario

* 1.5.1.2.4 Software utilizado para el desarrollo

Se muestra de forma detallada el software a utilizar en el desarrollo del proyecto, así como las herramientas de apoyo en la realización de cada una de las actividades que se ejecutarán para el desarrollo del sistema.

	Software	Descripción
Entorno de Desarrollo	Visual Studio Code	Editor de código fuente utilizado para desarrollar la aplicación web.
Software generador de diagramas	Lucidchart	Aplicación usada para diseñar los diagramas
Sistema Operativo	Windows 10	Sistema Operativo que se utilizara en las máquinas para el desarrollo
Navegador Web	Brave Browser	Se usará como navegador web en el proyecto.

➔ 1.5.1.3 Hardware

A continuación, se alistarán los recursos de Hardware que serán utilizados en el proyecto.

* 1.5.1.3.1 Apache

Servidor Web que interpretara el lenguaje PHP.

* 1.5.1.3.2 MySQL

Sistema gestor de base de datos utilizado en el proyecto.

* 1.5.1.3.3 Requerimientos mínimos para computadoras de desarrollo

Las características del hardware para las computadoras de desarrollo se han determinado examinando cada uno de los requerimientos mínimos del software a instalar en el equipo. Para el caso del servidor de desarrollo se instalarán: Sistema Operativo, PHP, MySQL, Apache y Brave Browser. En este caso se ha tomado como base el software de mayor requerimiento de hardware, el cual es MySQL.

Requerimientos mínimos para el servidor para desarrollo:

Componente	Requerimiento Mínimo
Velocidad del Procesador	2,5 GHz
Memoria RAM	2 Gb
Almacenamiento de Disco Duro	500 Gb

En las computadoras para el desarrollo se instalará:

- * Sistema Operativo
- * PHP (+Gestión de paquetes Composer)
- * MySQL
- * Apache
- * Brave Browser

Requerimientos mínimos para las computadoras de desarrollo:

Componente	Requerimiento Mínimo
Velocidad del Procesador	2,2 GHz
Memoria RAM	4 Gb
Almacenamiento de Disco Duro	1 Tb

* 1.5.1.3.4 Hardware para la implementación del proyecto

Para la implementación del proyecto se utilizó el siguiente hardware:

Recurso	Características	
Equipo para la Implementación	Modelo	Lenovo Ideapad 320
	Procesador	Intel Core i5 7200u 2.71 GHz
	Sistema Operativo	Windows 10
	Memoria RAM	8 Gb
	Disco Duro	HDD 1Tb / SSD WD 128 Gb
	Monitor	Lenovo Generic 1080P
	Mouse	Logitech M280
	Auriculares	Genius Hs Bt 940

* 1.5.1.3.5 Requerimientos mínimos de computadoras de usuarios

Las características del hardware para las computadoras de usuarios para operación se han determinado examinando cada uno de los requerimientos mínimos del software a instalar en el equipo. En el caso de las computadoras de usuarios se instalarán: Sistema Operativo y Brave Browser

Características de equipos para usuarios:

Recurso	Características	
	Modelo	Desktop
Equipo de Usuario	Procesador	Intel Pentium/ AMD Athlon (4 núcleos)
	Sistema Operativo	Windows 10
	Memoria RAM	4 Gb
	Disco Duro	HDD 500Gb
	Monitor	LED VGA
	Puertos	Mouse USB, Teclado USB, RJ 43 Ethernet
	Dispositivo de Red	Placa Wifi

➔ 1.5.1.4 Recurso humano

El recurso humano utilizado en el desarrollo del proyecto se detalla en la siguiente tabla:

Cantidad	Perfil	Función
1	Estudiante Analista de Sistemas	Analista Programador

* 1.5.1.4.1 Estructura del grupo de trabajo

Personal		Cantidad	Descripción
Grupo de desarrollo del proyecto	Administrador del proyecto	1	Administrar las tareas que se realizaran en el desarrollo del proyecto.
	Analista Programador	1	Analizar, diseñar y programar el sistema.
Asesoría del grupo de desarrollo	Docente director Programador	1	Brindar asesoría al grupo de desarrollo.
	Docente Trabajo de Diploma	1	Brindar observaciones al grupo de desarrollo.
	Docente Diagramador	1	Brindar asesoría para diagramar al grupo de desarrollo.

➔ 1.5.1.5 Conclusión de la factibilidad técnica

Se concluye que el proyecto es técnicamente factible, ya que se cuenta con el software, hardware y recurso humano para desarrollar e implementar el proyecto.

✳ 1.5.2 Factibilidad económica

En este tipo de factibilidad se determina si el sistema de información proporcionara beneficios mayores que los costos de su desarrollo, además se detalla si es posible obtener los recursos económicos para el desarrollo del proyecto o si ya se cuenta con el hardware para la implementación del sistema.

➔ 1.5.2.1 Beneficios

En el momento que se aplica el sistema en una empresa, se pueden detectar dos tipos de beneficios:

- Beneficios Cuantificables.
- Beneficios no Cuantificables.

* 1.5.2.1.1 Beneficios no cuantificables

Son beneficios intangibles que la empresa obtiene con el hecho de solo implantar el sistema. Abajo se detallan algunos:

- * Mejora en los procesos de toma de decisiones de la empresa.
- * Información de acciones realizadas.
- * Registro de datos históricos.
- * Reducción de tiempo en actividades de la empresa.

* 1.5.2.1.2 Beneficios cuantificables

Son beneficios tangibles de la empresa que se miden en términos monetarios, y los cuales pueden mejorar la productividad de los procesos y el personal.

➔ 1.5.2.2 Costos del sistema a desarrollar

Costos que serán invertidos para llevar a cabo el desarrollo e implementación del sistema.

* 1.5.2.2.1 Costos no cuantificables

Se listan los costos no cuantificables asociados al sistema, que se relacionan a factores como:

- * Toma de decisiones ineficaz, por información inaccesible o inoportuna.
- * Mala calidad de atención a los clientes, por información inaccesible o inoportuna.

* 1.5.2.2.2 Costos cuantificables

Aquellos costos que pueden ser cuantificados, relacionados directamente con:

➤ 1.5.2.2.2.1 Costos del desarrollo del proyecto

Los costos de desarrollo se han determinado en concepto de costos de recursos humanos, costos de recursos tecnológicos (Hardware y Software), costos fijos e imprevistos.

- 1.5.2.2.2.1.1 Costos de recursos humanos

Aquí se estudian los costos del recurso humano relacionados al desarrollo del proyecto.

1.5.2.2.2.1.1 Recurso humano

*Recurso humano: equipo de desarrollo del proyecto.

Los costos relacionados al equipo de desarrollo del proyecto son de \$369.600, se muestran los supuestos tomados en cuenta para el cálculo de este costo.

Tipo	Cantidad	Salario (\$/hora)	Salario (\$/día)	Salario (\$/mes)	Horas laborales por día	Horas laboradas al mes (\$/mes)	Duración del proyecto (meses)	Monto (\$)
Estudiante Analista de Sistemas	1	\$300	\$2,400	\$52,800	8	176	7	\$369.600

1.5.2.2.2.1.1.2 Supuestos

- * Se trabajarán 8 horas diarias.
 - * Se trabajarán 22 días al mes, de lunes a viernes.
- *1.5.2.2.1.2 Costos de recursos tecnológicos*

1.5.2.2.2.1.2.1 Costos de hardware

El costo del equipo computacional se detalla a continuación, este consta de una computadora personal.

Equipo: \$109.990 (https://articulo.mercadolibre.com.ar/MLA-880953110-notebook-lenovo-ideapad-s340-intel-156-8gb-ram-128gb-ssd-JM?searchVariation=64900522151#searchVariation=64900522151&position=6&type=item&tracking_id=15acc129-72d6-4d4c-b64f-14721ea1465c)

1.5.2.2.1.2.2 Costos de software

Se detalla el software que se utilizara para el desarrollo del proyecto:

Software	Costo (\$)	Licencias	Subtotal
Office 365	\$ 4099	1 año	\$ 4099
Windows 10 Home	\$ 8899	Definitiva	\$ 8899
MySQL	\$ 0	Definitiva	\$ 0
Apache	\$ 0	Definitiva	\$ 0
Brave Browser	\$ 0	Definitiva	\$ 0
Visual Studio Code	\$ 0	Definitiva	\$ 0
PHP/JavaScript	\$ 0	Definitiva	\$ 0
Día	\$ 0	Definitiva	\$ 0
TOTAL			\$ 12.998

Monto Total: \$12.998

Referencias:

Office 365: <https://www.microsoft.com/es-ar/microsoft-365/buy/compare-all-microsoft-365-products?&activetab=tab:primaryr1>

Windows 10 Home: <https://www.microsoft.com/es-ar/p/windows-10-home/d76qx4bznwk4?rtc=1&activetab=pivot%3aoverviewtab>

1.5.2.2.2.1.2.3 Costos fijos

Los costos fijos que serán necesarios para el desarrollo del proyecto se calculan según lo siguiente:

Detalle del servicio	Costo Mensual (estimado)(\$)	Detalle	Meses	Subtotal (\$)
Energía Eléctrica	\$528,13	Consumo aproximado mensual de 35 kWh.	7	\$ 3696,91
Servicio de Internet	\$1345,10	Velocidad de 25 Mb	7	\$ 9415,7
TOTAL				\$ 13.112,91

Monto Total: \$13.112,91

1.5.2.2.2.1.2.4 Costos imprevistos

Para el manejo de imprevistos, se tomará un 10% del total de costos del proyecto, con el fin de solventar cualquier gasto que no se haya considerado y que pueda afectar al desarrollo del proyecto.

TIPO	MONTO	SUBTOTAL
RECURSOS HUMANOS		
Equipo de desarrollo	\$369.600	
		\$369.600
RECURSOS TECNOLOGICOS		
Hardware	\$109.990	
Software	\$12.998	
Costos Fijos	\$13.112,91	
		\$136.100,91
SUBTOTAL		\$505.700,91
IMPREVISTOS (10%)		\$50.570,00
TOTAL		\$556.271,00

> 1.5.2.2.2.2 Costos de operación del sistema

Dentro de este apartado se consideran los costos de hardware, software, de mantenimiento y costos fijos necesarios para que se pueda usar el sistema de información.

- 1.5.2.2.2.2.1 Costos de hardware

Para poder usar el sistema se requiere contar con el equipo informático adecuado que cumpla con los requerimientos necesarios para su correcto funcionamiento. El equipo mínimo consta de:

- * Combo computadora + monitor para correr el sistema (Por sucursal)
- * Red de internet por cable o wifi

Referencias:

Pc escritorio + teclado + mouse: https://articulo.mercadolibre.com.ar/MLA-818224506-pc-computadora-intel-pentium-gold-g5400-4gb-ssd-240gb-p1-JM#position=3&type=item&tracking_id=f3918706-53ab-4924-9f51-826105258fd3

Monitor VGA 18,5: https://www.mercadolibre.com.ar/monitor-hp-19ka-led-185-negro-110v220v/p/MLA15133503?searchVariation=MLA15133503&pdp_filters=ITEM_CONDITION:2230284&source=search#searchVariation=MLA15133503&position=1&type=product&tracking_id=068fd37b-37a0-4f17-a124-8aa075d94570

- 1.5.2.2.2.2.2 Costos de software

1.5.2.2.2.2.2.1 Computadoras de usuarios

Software para computadoras de usuarios:

- * Sistema operativo: \$0.
- * Navegador Web: \$0.

- 1.5.2.2.2.2.3 Costos de mantenimiento

1.5.2.2.2.2.3.1 Recursos humanos

Se tomará en cuenta el costo de mantenimiento del sistema una vez implementado y puesto en funcionamiento, el costo será el siguiente:

PERSONAL	SALARIO POR MES (\$)
Administrador de base de datos	\$800
Analista	\$800
TOTAL MENSUAL	\$1.600
TOTAL ANUAL	\$19.200

1.5.2.2.2.3.2 Costos fijos

TIPO	MONTO	SUBTOTAL
COSTOS DE HARDWARE (Por Sucl.)		
Combo Computadora	\$35.000	
Monitor	\$12.000	
		\$47.000 x 1
COSTOS DE SOFTWARE		
Sistema Operativo	\$0	
Navegador Web	\$0	
		\$0
COSTOS DE MANTENIMIENTO		
Recursos Humanos	\$19.200	
		\$19.200
TOTAL		\$66.200

RESUMEN

TIPO	MONTO
Costo de Desarrollo	\$556.271,00
Costo de Operación	\$66.200,00
TOTAL ESTIMADO	\$622.471,00

Capítulo segundo

2º

Modelo de negocio

Contenido:

2.1 Glosario del sistema

2.1.1 Pólizas

- 2.1.1.1 ¿Que es una póliza?
- 2.1.1.2 Número o identificador de póliza
- 2.1.1.3 Estados de una póliza
- 2.1.1.4 ¿Que es la vigencia de la póliza?
- 2.1.1.5 ¿Que es el tomador?
- 2.1.1.6 ¿Que es el número o identificador del tomador?
- 2.1.1.7 ¿Que es el bien asegurado?
- 2.1.1.8 ¿Que es la suma asegurada?
- 2.1.1.9 ¿Que es la cobertura?
- 2.1.1.10 ¿Que es la cobertura adicional?
- 2.1.1.11 ¿Que significa renovación de una póliza?

2.1.2 Endosos

- 2.1.2.1 ¿Que es un endoso?
- 2.1.2.2 Número o identificador del endoso
- 2.1.2.2 Tipos de endosos

2.1.3 Siniestros

- 2.1.3.1 ¿Que es un siniestro?
- 2.1.3.2 Número o identificador del siniestro
- 2.1.3.3 Estados del siniestro
- 2.1.3.4 Fecha de denuncia del siniestro
- 2.1.3.5 Fecha y hora de ocurrencia del siniestro
- 2.1.3.6 Que significa lesión o muerte
- 2.1.3.7 Tipo de siniestros
- 2.1.3.8 ¿Que es el denunciante?
- 2.1.3.9 ¿Que es el reclamante?
- 2.1.3.10 ¿Que es el conductor?

2.1.4 Pagos

- 2.1.4.1 ¿Que es un pago?
- 2.1.4.2 Número de pago
- 2.1.4.3 Estados del pago
- 2.1.4.4 ¿Que es la prima total?

- 2.1.4.5 Fecha de registro del pago
- 2.1.4.6 Método de pago
- 2.1.4.7 ¿Que es una cuota?
- 2.1.4.8 ¿Que es el número de cuota?
- 2.1.4.9 ¿Que es el identificador de cuota?
- 2.1.4.10 ¿Que es la prima mensual?
- 2.1.4.11 ¿Que es el lapso de una cuota?
- 2.1.4.12 Fecha de facturación de una cuota
- 2.1.4.13 Fecha de vencimiento de facturación de una cuota
- 2.1.4.14 Estados de una cuota

2.1.5 Usuarios

- 2.1.5.1 ¿Que es un usuario?
- 2.1.5.2 ¿Que es el número o identificador de usuario?
- 2.1.5.3 Estados de un usuario
- 2.1.5.4 ¿Que es el nombre de usuario?
- 2.1.5.5 ¿Que es la contraseña?
- 2.1.5.6 ¿Que es el rol del usuario?

2.1.6 Sucursales

- 2.1.6.1 ¿Que es una sucursal?
- 2.1.6.2 ¿Que es el numero o identificador de una sucursal?

2.2 Descripción de patrones de funcionamiento

- 2.2.1 Patrón de funcionamiento de pólizas
 - 2.2.1.1 Patrones de estados de pólizas
 - 2.2.1.2 Patrones de renovación de póliza
- 2.2.2 Patrón de funcionamiento de pagos
 - 2.2.2.1 Patrones de modificación de pagos
 - 2.2.2.2 Patrones de estados de pagos
 - 2.2.2.3 Patrones de estados de cuotas
- 2.2.3 Patrón de funcionamiento de siniestros
 - 2.2.3.1 Patrones de estados de siniestros

➤ 2.1 Glosario del sistema

El glosario del sistema tiene como finalidad que, al momento de estar utilizando el sistema, la persona pueda asimilar la terminología con la que el equipo hace referencia a sus características propias, más que nada cuando se utilizan denominaciones las cuales pueden llegar a tomar diferentes significados en otros rubros, pudiendo provocar confusiones o malos entendidos.

★ 2.1.1 Pólizas

➤ 2.1.1.1 ¿Qué es una póliza?

El término póliza hace referencia al documento donde se detallan los beneficios a los que tiene derecho el asegurado cuando abona una prima de seguro. En otras palabras, es el contrato en sí.

➤ 2.1.1.2 Número o identificador de póliza

Es un número único que asigna el sistema aleatoriamente con la finalidad de distinguir cada una de las pólizas registradas a la hora de hacer referencia a las mismas. Su estructura puede variar dependiendo la compañía, por defecto consta de dos letras mayúsculas seguido de quince números entre 1 y 9.

Ej.: FZ884138568612735

➤ 2.1.1.3 Estados de una póliza

Un estado hace referencia a la forma en la que se presentan las características de la póliza en ese momento, dependiendo de este se podrán realizar determinadas acciones o no. La cantidad de estados estará definida por la compañía.

Se detalla cada uno de ellos en el apartado 2.3.1.1 (Patrones de estados de pólizas). Por ejemplo:

Desde el 24/10/2020 (15:16) hasta 26/10/2020 (11:59): En espera

Desde el 26/10/2020 (11:59) hasta 26/10/2021 (11:59): Vigente

Desde el 26/10/2021 (11:59) hasta 30/10/2021 (11:59): Vencida

Desde el 30/10/2021 (11:59) hasta 30/10/2022 (11:59): Renovada

➤ 2.1.1.4 ¿Qué es la vigencia de la póliza?

Es el período durante el cual la compañía de seguros toma bajo su cargo los riesgos cubiertos por la póliza. Se designa con una fecha de inicio de vigencia y con una fecha de fin de vigencia.

➔ 2.1.1.5 ¿Que es el tomador?

El tomador es la persona que contrata el Seguro y suscribe las condiciones de la póliza.

El asegurado es la persona que está bajo la cobertura de la póliza, y puede o no ser la misma persona que el tomador.

➔ 2.1.1.6 ¿Que es el número o identificador del tomador?

Es un número único generado por el sistema con la finalidad de facilitar el poder hacer referencia a un tomador particular. Su estructura por defecto consta de ocho números entre 1 y 9.

➔ 2.1.1.7 ¿Que es el bien asegurado?

El bien asegurado es el elemento correspondiente al tomador al cual se le aplican los procesos de seguro, en este caso sería el vehículo registrado.

➔ 2.1.1.8 ¿Que es la suma asegurada?

Es el valor atribuido por el tomador de un contrato de seguro a los bienes cubiertos por la póliza. En caso de siniestro, es el importe máximo que está obligado a pagar el asegurador.

➔ 2.1.1.9 ¿Que es la cobertura?

Obligación principal del asegurador que consiste en hacerse cargo, hasta el límite de la suma asegurada, de las consecuencias económicas que se deriven de un siniestro. Las coberturas están predefinidas, y por defecto existen seis tipos:

TIPO A Responsabilidad Civil Hacia Terceros

TIPO B Pérdidas Totales

TIPO C Responsabilidad Civil con Robo/Incendio Total

TIPO C1 Terceros Completo

TIPO D Todo Riesgo con Franquicia fija

TIPO E Todo Riesgo sin Franquicia

➔ 2.1.1.10 ¿Que es la cobertura adicional?

Estas coberturas son cláusulas que se pueden adicionar al bien asegurado, es decir, seguros adicionales, opcionales y especiales dentro de tu contrato de seguro. Por defecto existen cinco tipos denominadas “Accidentes Personales”, “Granizo”, “Parabrisas y Lunetas”, “Rotura de Cerraduras” y “Rotura de Cristales”

➔ 2.1.1.11 ¿Que significa renovación de una póliza?

Proceso al que se somete una póliza que ya no tiene validez para que pueda retomar su vigencia.

✳ 2.1.2 Endosos

➔ 2.1.2.1 ¿Que es un endoso?

El endoso es una modificación que se realiza sobre una póliza de seguro vigente (contrato de seguro) y altera las condiciones de la misma. Es decir, se refieren a la acción particular por la que un documento incorpora un cambio determinado a una póliza.

➔ 2.1.2.2 Número o identificador del endoso

Son un conjunto de caracteres numéricos únicos generados y asignados por el sistema a cada endoso realizado en las pólizas, tienen como finalidad distinguir y aumentar la facilidad al momento de referirse a cada uno. Su estructura por defecto consta de diez números entre 1 y 9.

➔ 2.1.2.2 Tipos de endosos

Hacen referencia a las distintas variedades particulares de endosos que existen y pueden ser aplicadas a las pólizas vigentes. Por defecto existen cinco tipos:

Modificación de riesgo: Modificar el bien asegurado y/o su información.

Modificación de datos nominales: Comprende la modificación de la suma asegurada, la matrícula, el número de chasis y/o el número de motor.

Modificación datos del vehículo: Comprende la modificación de toda la información que no sean datos nominales, como por ejemplo la clase, cantidad de kilómetros o coberturas adicionales.

Modificación de asegurado: Se modifica tomador o asegurado de la póliza.

Cambio de cobertura: Comprende la modificación del tipo de cobertura del bien asegurado o vehículo.

✳ 2.1.3 Siniestros

➔ 2.1.3.1 ¿Que es un siniestro?

Un siniestro es cuando ocurre cualquier acontecimiento o hecho asegurado que se contemplan en la póliza de seguros, es decir, todo accidente que haya protagonizado el asegurado.

➔ 2.1.3.2 Número o identificador del siniestro

Es un número único que asigna el sistema aleatoriamente con la finalidad de distinguir cada uno de los siniestros de una póliza registrados a la hora de hacer referencia a las mismas. Su estructura puede variar dependiendo la compañía, por defecto consta de quince números entre 1 y 9 seguido de dos letras mayúsculas. Por ejemplo: 564966459478417NK

➔ 2.1.3.3 Estados del siniestro

Un estado hace referencia a la forma en la que se presentan las características del siniestro en ese momento, dependiendo de este, el siniestro formara parte de una etapa determinada de su ciclo de vida. La cantidad de estados estará definida por la compañía. Por defecto existen cuatro tipos: en trámite, con reclamo en juicio, rechazado y finalizado. Se detalla cada uno de ellos en el apartado 2.3.3.1 (Patrones de estados de siniestros).

➔ 2.1.3.4 Fecha de denuncia del siniestro

Fecha correspondiente al día, mes y año presente en el que el tomador o asegurado realiza la denuncia del siniestro (El sistema guardara la fecha en que se grabe el siniestro).

➔ 2.1.3.5 Fecha y hora de ocurrencia del siniestro

Hace referencia al día, mes, año, hora y minutos en que el tomador o asegurado se vio afectado en el siniestro.

➔ 2.1.3.6 Que significa lesión o muerte

Opción que determina si terceras personas se vieron afectadas con lesiones o muertes en el siniestro ocurrido a registrar.

➔ 2.1.3.7 Tipo de siniestros

Hace referencia a las distintas variedades particulares de siniestros que existen. Por defecto existen nueve tipos:

Destrucción Total

Destrucción Parcial

Rotura de Luneta o Parabrisas

Deterioro por Granizo

Accidente/Choque

Incendio Parcial

Incendio Total

Robo Parcial

Robo Total

Con respecto al funcionamiento se detalla más en el apartado 2.3.3 (Patrón de funcionamiento de siniestros)

► 2.1.3.8 ¿Que es el denunciante?

Persona que realiza la denuncia del siniestro la cual tiene relación directa, o en su caso es el conductor del vehículo (bien asegurado).

► 2.1.3.9 ¿Que es el reclamante?

Tercera persona involucrada en el siniestro.

► 2.1.3.10 ¿Que es el conductor?

Tercera persona involucrada en el siniestro, puede o no ser la misma persona que el reclamante.

✳ 2.1.4 Pagos

► 2.1.4.1 ¿Que es un pago?

Es la denominación que el sistema le adjudica al conjunto de cuotas generadas correspondientes a la prima total de una póliza emitida.

► 2.1.4.2 Número de pago

El número de pago un conjunto de caracteres numéricos únicos generadas por el sistema al momento de registrarse el pago de una póliza, tienen la finalidad de representar a su conjunto de cuotas correspondientes. La estructura por defecto consta de once números entre 1 y 9.

► 2.1.4.3 Estados del pago

Un estado hace referencia a la forma en la que se presentan las características del pago en ese momento, dependiendo de este, se determinara si el pago se encuentra disponible para operar o no. Su finalidad es para informar al usuario. Por defecto existen tres tipos: En vigencia, Caducado y Finalizado. Se detalla cada uno de ellos en el apartado 2.3.2.2 (Patrones de estados de pagos).

► 2.1.4.4 ¿Que es la prima total?

La prima total es la cantidad que el asegurado tiene que pagar a la entidad aseguradora para tener la cobertura necesaria de los bienes o servicios contratados.

► 2.1.4.5 Fecha de registro del pago

Se refiere al día, mes y año en que se registró el pago en el sistema, generalmente es el mismo valor que la fecha de registro de póliza, a no ser que haya sido modificado, de esta manera el valor registrado será la fecha de modificación.

► 2.1.4.6 Método de pago

Se refiere a la metodología que el asegurado utilizara para abonar cada cuota de la prima total a la entidad aseguradora.

► 2.1.4.7 ¿Que es una cuota?

Hace referencia a la cantidad de veces en la que se abonara la prima total.

► 2.1.4.8 ¿Que es el número de cuota?

El sistema realiza un listado de todas las cuotas correspondientes a un pago asignándole un numero único para determinar cuántos recibos fueron generados para ese en particular. Por ejemplo: Se decide abonar la prima total de una póliza en 6 pagos, entonces cada cuota tendrá uno de los 6 números de manera ascendente. Los números de cuota se pueden repetir en caso de pertenecer a pagos diferentes.

► 2.1.4.9 ¿Que es el identificador de cuota?

Es un conjunto de caracteres numéricos únicos generados y asignados por el sistema a cada cuota independientemente del pago al que correspondan, su finalidad es poder referenciarlas de manera más fácil en todo el sistema. Su estructura consta de diez números entre 1 y 9.

► 2.1.4.10 ¿Que es la prima mensual?

Es el monto individual asignado a cada una de las cuotas de un determinado pago.

➔ 2.1.4.11 ¿Que es el lapso de una cuota?

Hace referencia al margen de tiempo de cobertura que tendrá el tomador al realizar el pago de esa cuota.

➔ 2.1.4.12 Fecha de facturación de una cuota

Se refiere al día, mes y año en que la cuota estará disponible para realizar su correspondiente pago.

➔ 2.1.4.13 Fecha de vencimiento de facturación de una cuota

Es el día, mes y año en el que una cuota dejara de estar disponible para realizar su correspondiente pago.

➔ 2.1.4.14 Estados de una cuota

Un estado hace referencia a las condiciones en la que se presenta una cuota en ese momento, dependiendo de este, el sistema podrá permitir realizar determinadas acciones en una póliza o no. La cantidad de estados estará definida por la compañía.

Se detalla cada uno de ellos en el apartado 2.3.2.3 (Patrones de estados de cuotas).

✳ 2.1.5 Usuarios

➔ 2.1.5.1 ¿Que es un usuario?

Un usuario es una persona física encargada de llevar a cabo todas las acciones del sistema, puede haber tantos como la compañía requiera.

➔ 2.1.5.2 ¿Que es el número o identificador de usuario?

Es un conjunto de caracteres numéricos únicos generados y asignados por el sistema a cada usuario, con la finalidad de aumentar la facilidad de hacer referencia a cada uno.

➔ 2.1.5.3 Estados de un usuario

Es la condición con la que se presenta un usuario, dependiendo del estado este último podrá o no acceder al sistema.

➔ 2.1.5.4 ¿Que es el nombre de usuario?

El nombre de usuario es la denominación única con la que el mismo desee figurar públicamente en el sistema.

➔ 2.1.5.5 ¿Que es la contraseña?

Es la clave oculta necesaria correspondiente a cada usuario para poder ingresar al sistema.

➔ 2.1.5.6 ¿Que es el rol del usuario?

Refiere a la posición de las funciones o tareas a desarrollar por orden de rango, grado o importancia dentro de la organización. Por defecto el sistema posee dos tipos: Usuario y Administrador.

✳ 2.1.6 Sucursales

➔ 2.1.6.1 ¿Que es una sucursal?

Refiere a la empresa física sobre la que está operando el sistema al momento de utilizarlo.

➔ 2.1.6.2 ¿Que es el numero o identificador de una sucursal?

Es un conjunto de caracteres alfanuméricos único que es generado y asignado por el sistema a cada una de las sucursales para poder distinguirlas y hacer referencia a las mismas más fácilmente. Su estructura consta de dos letras mayúscula seguido de un guion y seis números aleatorios entre 1 y 9. Por ejemplo: BB-435476.

➤ 2.2 Descripción de patrones de funcionamiento

❖ 2.2.1 Patrón de funcionamiento de pólizas

Las pólizas registradas pueden estar sometidas a seis estados diferentes dependiendo de los factores por los que se vea afectada, los estados son "En espera", "Vigente", "Vigente-deuda", "Vencida", "Anulada" y "Renovada".

Cuando una póliza es emitida, se registran cuatro fechas diferentes, primero la fecha de emisión, segundo la fecha de inicio de vigencia, tercero la fecha de fin de vigencia y cuarto la fecha de anulación (Un mes posterior exacto a la fecha de fin de vigencia), y quedara asignado el estado "En espera" por defecto.

Una vez llegados a la fecha de emisión, la póliza automáticamente cambiara el estado a "Vigente" manteniéndose así hasta el momento en que nos ubiquemos en la fecha de fin de vigencia (suponiendo que todas las cuotas correspondientes se encuentren al día), que será cuando se actualice a "Vencida", durante este estado es cuando debe de renovarse la póliza en caso de así requerirlo, pero una vez llegada la fecha de anulación, el estado de la póliza se actualizara a "Anulada" y ya no será posible.

➤ 2.2.1.1 Patrones de estados de pólizas

A continuación, se detallan los estados, y a que factores se encuentra sometida la póliza al poseer cada uno.

-En espera: El estado comprende el lapso entre la fecha de emisión (inclusive) y la fecha de inicio de vigencia (no inclusive), en esta etapa la póliza no cuenta con cobertura, solo se encuentra registrada y programada para que la vigencia comience cuando finalice este plazo.

-Vigente: El estado comprende desde la fecha de inicio de vigencia (inclusive) hasta la fecha de fin de vigencia (no inclusive), esta etapa es cuando la póliza tiene cobertura (con todas las cuotas en estado "A facturar" y "Pago aprobado") pudiendo realizarse cualquier interacción con la misma.

-Vigente con deuda: Es el estado que se asigna temporalmente a la póliza cuando en pleno lapso de vigencia, una cuota o más con cualquiera de los estados "Facturada (Sin pagar)", "Pago rechazado", o "Deuda", supera la fecha de vencimiento de facturación. En este periodo la póliza se encuentra sin cobertura hasta que todas las cuotas estén con el estado "Pago aprobado".

-Vencida: Este estado comprende el lapso entre la fecha de fin de vigencia (inclusive) y la fecha de anulación (no inclusive), en esta etapa la póliza deja de tener cobertura y se encontrara con la opción de renovación solo por un mes posterior a la fecha de fin de vigencia, siempre y cuando todas las cuotas se encuentren con el estado "Pago aprobado" sin excepción.

-Renovada: Este estado se asigna a una póliza vencida en el momento en que el usuario decide renovarla, por lo que quedara sin vigencia ni cobertura, habiendo finalizado el ciclo de vida.

-Anulada: Es el estado que se asigna a una póliza vencida en el momento en que esta alcanzó la fecha de anulación sin anteriormente haber sido renovada por el usuario, por lo que estará sin vigencia ni cobertura habiendo finalizado su ciclo de vida.

➔ 2.2.1.2 Patrones de renovación de póliza

La opción de renovación solo estará disponible para todas aquellas pólizas las cuales se encuentren con el estado "Vencida" y con todas las cuotas con el estado "Pago aprobado", ambas condiciones sin excepción, de lo contrario no se permitirá realizar el proceso.

Una vez que se haya realizado el proceso de renovación, lo que sucede es que la póliza en la que se está operando cambiara su estado de "Vencida" a "Renovada", y se generara una nueva póliza con las condiciones de renovación establecidas y con el estado "En espera".

✳ 2.2.2 Patrón de funcionamiento de pagos

En el proceso de registro de una póliza, en la última sección se deben establecer los parámetros de pago para designar en qué momento y de qué forma se cobrara cada prima mensual durante la vigencia.

En esa etapa el sistema autocompletara un campo con un ejemplar de la prima total del riesgo en relación a su suma asegurada junto con otros estándares (como cantidad de meses de duración de vigencia, etc.), pudiendo ser modificada por el usuario.

También se deberá de seleccionar la cantidad de cuotas con las que se desee pagar la prima total. Las opciones serán generadas por el sistema dependiendo la cantidad de meses de vigencia, partiendo de un mínimo de 2 cuotas hasta un máximo de 12 cuotas según sea posible.

Una vez se seleccione la cantidad de cuotas, se calculará la prima mensual correspondiente, no pudiendo modificar esta opción.

La siguiente opción a seleccionar es el día de cada mes en que el tomador decida que se facture y se genere el recibo de pago de la cuota. En el caso de que el día seleccionado sea menor al día de inicio de vigencia de ese mismo mes, la

primera facturación será un mes después, para evitar que el sistema la interprete como una cuota en deuda una vez finalizado el registro de la póliza.

El formulario finaliza seleccionando el método de pago que prefiera el cliente, pudiendo ser asociando una tarjeta de crédito o en efectivo en la sucursal o medios seleccionados por los administradores de la empresa.

Una vez registrada la póliza, se almacenará en la base de datos un pago relacionado a esta, y las cuotas relacionadas al pago. Si la cantidad de cuotas elegidas es menor a la cantidad de meses de vigencia de la póliza, estas se almacenarán en lapsos, es decir, cada una comprenderá dos o más meses.

Ejemplo:

Si la vigencia de la póliza comienza el 20/06/2020, y finaliza el 20/06/2021, la vigencia tiene 12 meses, en la cual se puede establecer 2, 3, 4, 6 y 12 cuotas, y al momento de elegir una opción, el sistema dará diferentes opciones:

2 cuotas: Se generaran dos lapsos de 6 meses en los cuales se facturara el día en que decida el usuario, en caso de que el día seleccionado sea menor al día de inicio de vigencia en ese mismo mes (es decir un día ya pasado, anterior al 20/06/2020), para evitar que el sistema la interprete como una cuota en deuda, se facturara ese mismo número del día pero del mes siguiente (por ejemplo, el tomador decide pagar el 10 de cada mes, el sistema registrara la primera facturación el 10/07/2020 en lugar de 10/06/2020, y de ahí calcula 6 meses exactos para la siguiente cuota).

3 cuotas: Se generan 3 lapsos de 4 meses, el criterio de pago funciona de igual manera que el anterior.

4 cuotas: Se generan 4 lapsos de 3 meses con el mismo criterio de pagos.

6 cuotas: Se generan 6 lapsos de 2 meses.

12 cuotas: Se debe pagar la cuota mensualmente durante 12 lapsos.

➔ 2.2.2.1 Patrones de modificación de pagos

Todos los parámetros de cuotas de los pagos de las pólizas establecidos pueden ser modificados luego de haber emitido una nueva póliza, pero solo si esa misma se encuentra en estado "Espera" o "vigente". De todas maneras, no se recomienda realizar esta acción ya que pueden generarse filtraciones o vulnerabilidades como fechas de cobro obsoletas, o cuotas en deuda a la hora de actualizar automáticamente esta información, o al facturar los nuevos lapsos ya habiendo realizado esta operación con los anteriores.

➔ 2.2.2.2 Patrones de estados de pagos

Los pagos generales de las primas totales (Es decir, el conjunto de cuotas de una póliza) pueden encontrarse en tres tipos de estados los cuales son:

-En vigencia: Cuando el pago se encuentra activo y funcional con respecto a una póliza con estado diferente a "Anulada" o "Renovada". Todos los pagos con este estado pueden ser modificados si así es permitido por el sistema.

-Caducado: El pago tendrá este estado, cuando la póliza correspondiente terminó su ciclo de vida habiendo sido renovada o anulada. Significa que el pago solo estará disponible para visualizar el conjunto de cuotas de esa póliza.

-Modificado: Se aplica este estado a todos aquellos pagos cuya póliza no se encuentra en estado "Anulada" o "Renovada" y fueron modificados, pudiendo modificarse nuevamente. Sirve para alertar que ese pago ya se modificó.

➔ 2.2.2.3 Patrones de estados de cuotas

Las cuotas de un pago de una prima total de una póliza pueden encontrarse en cuatro tipos de estados:

-A facturar: Se aplica este estado a todas aquellas cuotas las cuales no fueron facturadas para emitir el cobro, este estado permanecerá hasta una vez llegada la fecha de facturación.

-Facturada (Sin pagar): Se aplicará este estado a la cuota una vez llegada la fecha de facturación, y permanecerá hasta que el usuario emita el cobro correspondiente, o en caso de no realizarse, por un mes hasta la fecha de vencimiento.

-Pago aprobado: Se aplica este estado al momento en que el usuario emita el cobro de la cuota. No puede ser revertido.

-Pago rechazado: Cuando ocurre un problema con la tarjeta de crédito asociada (Sea falta de fondos, etc.) al intentar realizar el cobro de la prima. Este estado puede ser revertido al emitir el cobro manualmente.

-Deuda: Una cuota se encuentra en deuda cuando se alcanzó o superó la fecha de vencimiento de la factura. El hecho de que una cuota se encuentre en este estado, ocasiona que la póliza deje de tener cobertura temporalmente hasta que se revierta la opción (Emitiendo el cobro).

★ 2.2.3 Patrón de funcionamiento de siniestros

Al momento de registrar un nuevo siniestro, en la primera etapa del formulario debemos de seleccionar el tipo de siniestro ocurrido, dependiendo de esta opción, será fundamental para el sistema para saber qué información debe de brindarse posteriormente.

Otra sección para completar es la de los datos del denunciante de ese siniestro, que solo se permite que sea el mismo asegurado o personas cercanas a su vínculo social (depende del modelo de negocio que lleve a cabo la compañía, pudiendo adaptar el sistema al mismo). En el caso de que el tomador sea el denunciante, el sistema brinda la funcionalidad de omitir el ingreso de los datos

del mismo siempre y cuando el usuario haya seleccionado la opción "Si" en un campo específico para estas ocasiones.

En el caso de seleccionar "Accidente/choque" como tipo de siniestro, significa que existen tercera personas involucradas en el mismo, las cuales toman el rol de reclamante y/o conductor de la acción, pudiendo ser o no la misma persona. En caso de serlo, una vez completada la información del reclamante, el sistema brinda una opción la cual, si el usuario así lo requiere, podrá omitir los campos del conductor solo con un click.

Si la opción seleccionada como tipo de siniestro es diferente a "Accidente/choque", el sistema quita del formulario las secciones de "Reclamante" y de "Conductor".

► 2.2.3.1 Patrones de estados de siniestros

Pueden aplicarse cuatro tipos de estados a los siniestros para determinar su situación actual.

-En trámite: Se aplica este estado a aquellos siniestros los cuales están en el proceso de negociación entre el tercero damnificado y la compañía.

-Con reclamo en juicio: Se aplica a aquellos siniestros los cuales se encuentran intervenidos por demandas aplicadas y abogados en el proceso de negociación.

-Rechazado: Se aplica cuando el siniestro fue rechazado por determinadas razones (Falta de carnet, falta de pago, circular en contramano o bajo efectos de estupefacientes, etc.).

-Finalizado: Se aplica este estado a todos los siniestros cuya negociación entre la compañía y el reclamante finalizó legalmente ya sea retribuyendo económico al tercero o bajo cualquier otra medida.

Capítulo tercero

3º

Guía de procedimientos

Contenido:

3.1 Interfaces de navegación

3.1.1 Formulario de acceso al sistema

3.1.2 Barra superior

 3.1.2.1 Menú de acciones de usuario

3.1.3 Menú principal

3.1.4 Contenedor general

 3.1.4.1 Título de navegación

 3.1.4.2 Subtítulo de navegación

 3.1.4.3 Cuadro de interacción

3.1.5 Mensajes y alertas de interacción

 3.1.5.1 Mensaje de éxito

 3.1.5.2 Mensaje informativo

 3.1.5.3 Mensaje de error

 3.1.5.4 Mensajes de notificación informativos

3.2 Interacción con las interfaces de navegación

3.2.1 Ingresar al sistema

3.2.2 Menú de acciones

 3.2.2.1 Foto de perfil

 3.2.2.2 Mis datos

 3.2.2.2.1 Modificar información general

 3.2.2.2.2 Modificar contraseña

 3.2.2.3 Salir del sistema

3.2.3 Menú principal

3.3 Utilizar el sistema

3.3.1 Inicio

3.3.2 Pólizas

 3.3.2.1 Emitir nueva póliza

 3.3.2.1.1 Sección póliza

 3.3.2.1.2 Sección tomador

 3.3.2.1.3 Sección vehículo

 3.3.2.1.4 Sección cobertura

 3.3.2.1.5 Sección pago

3.3.2.2 Consultar pólizas

3.3.2.2.1 Ver la información de la póliza

3.3.2.3 Renovar póliza

3.3.2.3.1 Proceso de renovación denegado

3.3.2.3.2 Proceso de renovación

3.3.2.3.2.1 Datos de póliza

3.3.2.3.2.2 Datos del pago

3.3.3 Endosos

3.3.3.1 Emitir un endoso

3.3.3.1.1 Modificación de riesgo

3.3.3.1.2 Modificación de datos nominales

3.3.3.1.3 Modificación de datos del vehículo

3.3.3.1.4 Modificación de asegurado

3.3.3.1.5 Cambio de cobertura

3.3.3.2 Consultar endosos

3.3.3.2.1 Ver endosos de una póliza

3.3.3.2.2 Modificar la descripción de un endoso

3.3.4 Siniestros

3.3.4.1 Emitir nuevo siniestro

3.3.4.1.1 Información general

3.3.4.1.2 Denunciante

3.3.4.1.3 Accidente

3.3.4.1.4 Reclamante tercero

3.3.4.1.5 Conductor tercero

3.3.4.2 Administrar siniestros

3.3.4.2.1 Modificar estado del siniestro

3.3.4.2.2 Modificar información del siniestro

3.3.4.2.3 Administrar imágenes

3.3.4.3 Consultar siniestros

3.3.5 Tomadores

3.3.5.1 Agregar nuevo tomador

3.3.5.2 Administrar tomadores

3.3.5.2.1 Modificar información

3.3.5.2.2 Ver información

3.3.5.2.3 Dar de baja del sistema

3.3.6 Pagos

3.3.6.1 Facturar

3.3.6.1.1 Consultar facturas de pago

3.3.6.1.1.1 Emitir recibo

3.3.6.1.1.2 Ver recibo

3.3.6.1.1.3 Enviar recibo

3.3.6.1.1.4 Imprimir recibo

3.3.6.1.2 Consultar recibos de pagos caducados

3.3.6.1.2.1 Enviar listado

3.3.6.1.2.2 Descargar listado

3.3.6.2 Modificar cuotas

3.3.7 Administrador

3.3.7.1 Usuarios

3.3.7.1.1 Nuevo usuario

3.3.7.1.2 Listado de bajas

3.3.7.1.2.1 Reactivar usuario

3.3.7.1.3 Dar de baja

3.3.7.1.4 Modificar rol

3.3.7.1.5 Reenviar datos de acceso a usuarios

3.3.7.1.6 Consultar información

3.3.7.2 Sucursales

3.3.7.2.1 Agregar nueva sucursal

3.3.7.2.2 Consultar información de una sucursal

➤ 3.1 Interfaces de navegación

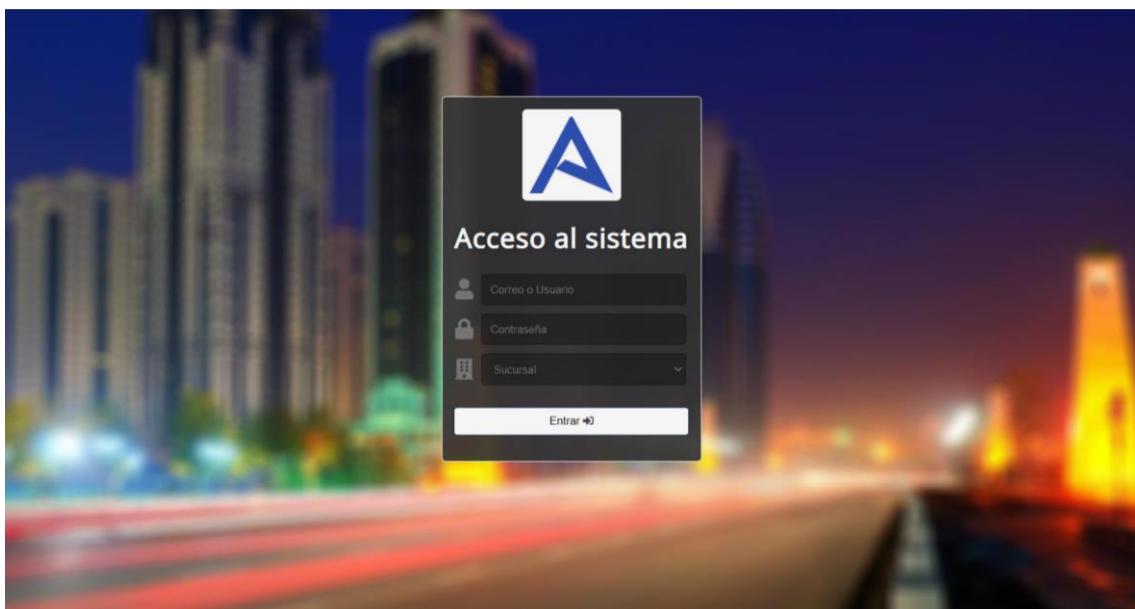
En esta sección se realizará una presentación de todas las interfaces con las que el usuario se verá relacionado al interactuar con el sistema.

La finalidad es conocer la estructura y las interfaces, lo que será de utilidad para que el usuario aumente la familiarización con el sistema y lleve a cabo las tareas con mayor facilidad, como así poder enterarse de aquellos términos o denominaciones con las que el sistema hará referencia ocasionalmente a sus secciones, ya sea en notificaciones internas o en el contenido del manual.

Para conocer cómo utilizar el sistema directamente, vea la sección 3.2 (Interacción con las interfaces de navegación).

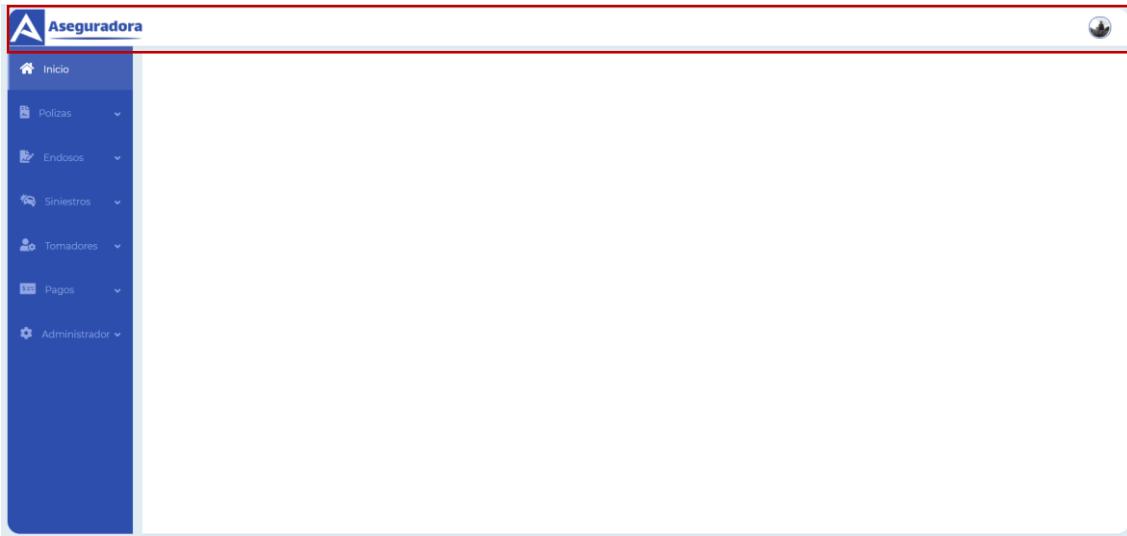
✳ 3.1.1 Formulario de acceso al sistema

Este formulario es lo primero que veremos si somos nuevos usuarios, es donde debemos ingresar nuestra información privada de acceso la cual hayamos recibido por parte del personal autorizado de la compañía.



✳ 3.1.2 Barra superior

Una vez ingresados los datos de acceso y ya dentro del sistema, comenzando de arriba hacia abajo, lo que primero veremos es la barra superior, dentro de ella a su izquierda veremos el logo de la compañía y a su derecha la foto del usuario. La barra superior no es un elemento dinámico dentro del sistema, pero si es posible agregar determinados elementos adicionales (Botones, opciones, etc.) si así lo requiere la compañía.



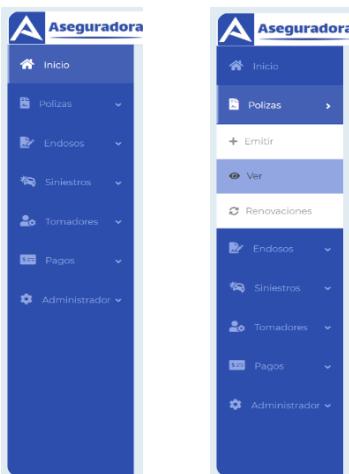
➔ 3.1.2.1 Menú de acciones de usuario

Al hacer click sobre la foto de perfil, se desplegará el menú de acciones de usuario en la parte superior derecha de la pantalla. Es el elemento donde el sistema reserva el listado de acciones relacionadas con el usuario.



✳ 3.1.3 Menú principal

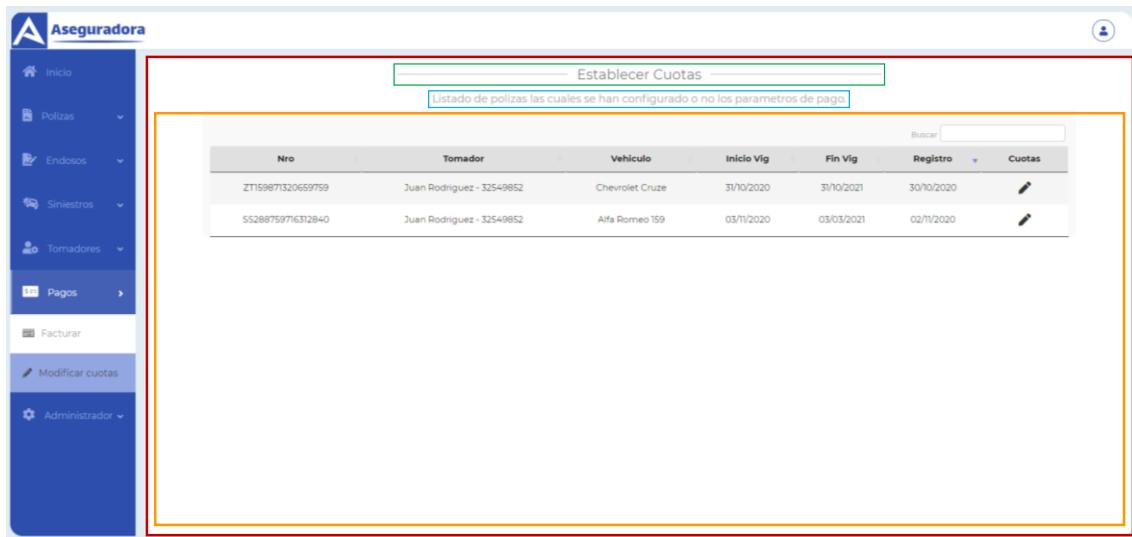
Este es el elemento con el que más va a interactuar el usuario, ya que es donde se concentra el listado de todas las opciones para llevar a cabo las acciones principales del sistema.



Al presionar en una opción, se desplegará el listado de cada una de las sub acciones disponibles para llevarse a cabo.

✳ 3.1.4 Contenedor general

Sección del sistema donde se alojará el contenido de la acción que deseé llevar a cabo el usuario. Se remarca en rojo.



➔ 3.1.4.1 Titulo de navegación

Es un elemento dinámico, básicamente informa al usuario que tipo de acción va a realizar en el sistema. (Se remarca en verde).

➔ 3.1.4.2 Subtitulo de navegación

Este elemento se presenta ocasionalmente para dar una breve descripción de las acciones o para informar de algún dato necesario al usuario. (Se remarca en celeste).

➔ 3.1.4.3 Cuadro de interacción

Es donde se mostrarán los formularios, tablas y demás elementos con los que se podrán realizar las acciones principales. (Se remarca en naranja).

✳ 3.1.5 Mensajes y alertas de interacción

Son todos los tipos de notificaciones que puedan aparecer a lo largo del uso del sistema. Se ejecutarán dependiendo del estado en que se encuentre la acción realizada, ya sea en su proceso o al finalizar la misma.

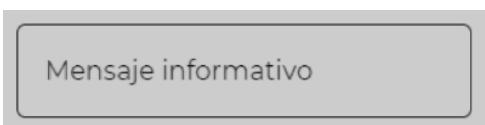
Este tipo de mensajes al ejecutarse se verán específicamente en la parte inferior central, y en el centro del cuadro de interacción. Aquellos cuya información no es tan relevante para el usuario en sí, permanecerán por 5 segundos aproximadamente, y aquellos que brindan información relevante como los mensajes de notificación, permanecerán hasta que el usuario decida cerrarlo o realice la acción pretendida.

➔ 3.1.5.1 Mensaje de éxito



Este mensaje aparece cuando una acción es completada y su correspondiente proceso fue realizado correctamente, generalmente cuando se trata de grabar, modificar o eliminar registros.

➔ 3.1.5.2 Mensaje informativo



Este mensaje aparece cuando una acción es completada y su correspondiente proceso fue realizado correctamente, generalmente cuando se realiza una modificación de información y no se registraron cambios.

➔ 3.1.5.3 Mensaje de error



Este mensaje aparece cuando una acción es completada y su correspondiente proceso se vio interrumpido o denegado, generalmente cuando ocurre un fallo al realizar una consulta a la base de datos y la operación no pudo ser finalizada.

En caso de presentarse, se recomienda:

-Verificar de todas maneras si la acción realizada se llevó a cabo, ya que dependiendo del error a veces las tareas pueden ser completadas. Por ejemplo, si se emite una póliza, puede que al final se haya grabado con éxito y el error es indistinto para esa acción específica.

-Reiniciar la página y reintentar el procedimiento que se estaba realizando, ya que puede tratarse de una falla temporal.

Se den o no los anteriores casos, siempre que se presente este mensaje, debe de contactarse con el personal técnico para solucionarlo.

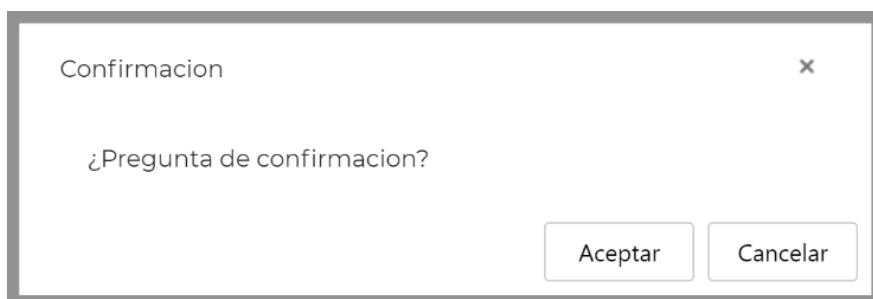
➔ 3.1.5.4 Mensajes de notificaciones

Las notificaciones aparecen cuando en el proceso de realizar una acción determinada, el sistema prohíbe la ejecución temporalmente para informar al usuario sobre una tarea específica a realizar, generalmente cuando se intenta registrar, modificar, eliminar o enviar registros con características las cuales no se les permitido llevar a cabo.

Existen tres tipos dependiendo del tipo de información brindada:

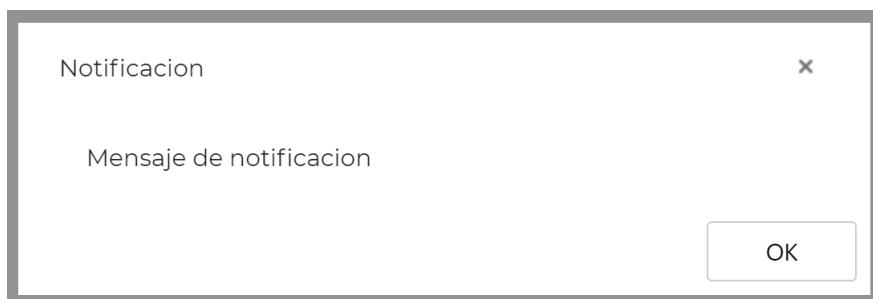
-De confirmación: Notifica si el usuario desea realizar una acción o no. Suele utilizarse para registrar o cancelar procesos.

Por ejemplo, para cancelar la información del proceso de emisión de póliza, o para dar de baja a un usuario.



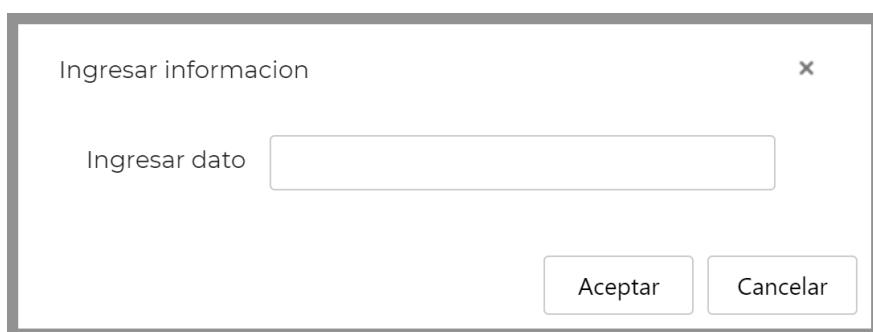
-De notificación informativa: Notifica al usuario un mensaje informativo.

Por ejemplo, cuando no se permite eliminar un cliente ya que es tomador de una póliza.



-De ingresar información: Notifica al usuario que debe ingresar cierta información o datos para realizar esa acción.

Por ejemplo, para enviar el recibo de pago de cuota de una póliza por email.



➤ 3.2 Interacción con las interfaces de navegación

En esta sección se realizará una guía detallada de procedimientos para que el usuario pueda realizar cada una de las acciones disponibles en el sistema.

✳ 3.2.1 Ingresar al sistema

Para acceder al sistema por primera vez, primero debemos tener disponible el usuario y la contraseña que haya sido brindada por el administrador.

Esta información debe de llegar a la casilla de correo electrónico que el usuario haya brindado a la compañía al momento de ser dado de alta en el sistema.

La interfaz con la que nos encontraremos será la siguiente:



The image shows a dark-themed login interface for a system. At the top is a large blue letter 'A'. Below it, the text 'Acceso al sistema' is displayed. There are three input fields: 'Correo o Usuario' with a user icon, 'Contraseña' with a lock icon, and 'Sucursal' with a building icon. A dropdown arrow is visible next to the 'Sucursal' field. At the bottom is a white button labeled 'Entrar ➔'.

Diseño del formulario para
ingresar los datos de acceso
al sistema



This image shows the same login interface as above, but with red outlines around all three input fields ('Correo o Usuario', 'Contraseña', and 'Sucursal'). Below the fields, a list of validation errors is shown: '• Debe ingresar un Usuario', '• Debe ingresar una Contraseña', and '• Debe seleccionar una Sucursal'. The 'Entrar ➔' button is at the bottom.

Diseño del formulario al
momento de intentar ingresar
sin completar los campos
requeridos

Se debe de ingresar:

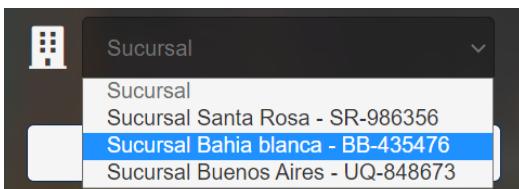
El nombre de usuario



La contraseña

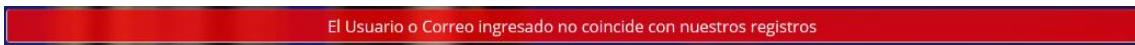


Seleccionar la sucursal en la que se va a operar



Después de ingresar la información, al presionar en comenzará la etapa de validación interna , por lo que, al finalizar, si todo es correcto, ya se habrá iniciado sesión sin problemas, de lo contrario, el usuario puede recibir tres tipos de respuestas a través de un mensaje a lo largo de la parte superior de la pantalla del formulario de acceso:

-Información no valida:

 El Usuario o Correo ingresado no coincide con nuestros registros

Este mensaje informa que tanto el nombre de usuario ingresado no se relaciona con ningún registro de usuario dado de alta en el sistema.

-Contraseña no valida:

 La contraseña ingresada es incorrecta

Este mensaje aparece cuando el nombre de usuario ingresado tiene relación con un registro de usuario dado de alta, pero que su correspondiente contraseña no coincide con la ingresada.

-Usuario no autorizado:

 Usted no tiene permiso para acceder al sistema permanentemente

Este tipo de mensaje significa que el usuario ingresado y la contraseña se relacionan con un registro de usuario que en algún momento fue dado de alta en el sistema, pero temporalmente se encuentra dado de baja por decisión directa del administrador, y permanecerá en este estado hasta que el mismo decida revertirlo.

En caso de persistir cualquier tipo de mensaje anterior sin existir un motivo previo, se debe contactar con el administrador del sistema.

3.2.2 Menú de acciones



El menú de acciones se encuentra en la parte superior derecha de la pantalla y se mostrara al presionar sobre la foto de perfil del usuario. En él, se ubica el nombre completo del usuario que inicio sesión, y debajo el listado de opciones con las que se podrá interactuar.

A través de este menú, tendremos la posibilidad de realizar las siguientes acciones:

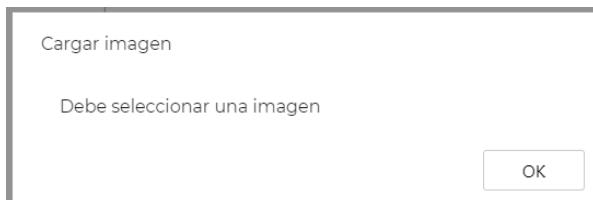
3.2.2.1 Administrar foto de perfil



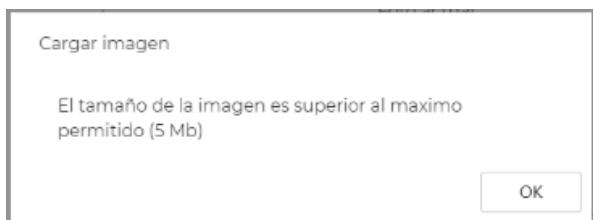
Para añadir una nueva foto de perfil, primero debemos presionar en el botón “seleccionar nuevo archivo”, seguido de esto se abrirá una nueva ventana donde podremos acceder a los datos de la computadora local, debemos seleccionar la imagen (Formatos disponibles: JPG, PNG, JPEG, tamaño máximo: 5 Megabytes), y al presionar en “Aceptar” deberíamos poder visualizarla al mismo tiempo de recibir un mensaje de éxito en caso de que el proceso se haya realizado sin problemas, de lo contrario se lanzara un mensaje de error. Al presionar en “Cancelar” se cerrará la ventana de administración de la foto de perfil.

En el caso en que la imagen no cumpla con las características permitidas, se visualizaran las notificaciones correspondientes:

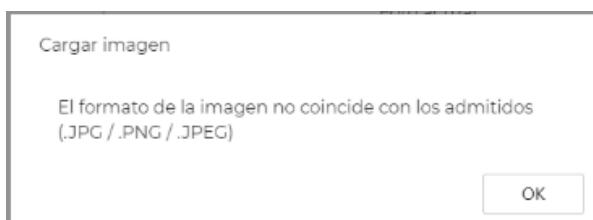
Imagen no seleccionada:



Tamaño de imagen superior a 5 Megabytes:



Formato de imagen no permitido:



En caso de recibir los siguientes avisos en una alerta de error (Sección 3.1.5.3) en reiteradas ocasiones, debe de ponerse en contacto con el administrador del sistema o con el personal técnico.

- “No se ha podido registrar la imagen solicitada”
- “Ha ocurrido un problema al realizar la petición al servidor”
- “Ha ocurrido un problema interno. Reintente más tarde”

➡ 3.2.2.2 Administrar mis datos



Para gestionar la información del usuario primero se debe presionar en el botón “Mis datos” del menú de acciones, al hacerlo se presentará la siguiente interfaz:

 **Modificar Mi Información**

Nombre completo <input type="text" value="Erasmo Jesus Jordan"/>	Documento <input type="text" value="41037199"/>	Nacionalidad <input type="text" value="Argentina"/>
Provincia/Distrito <input type="text" value="La Pampa"/>	Localidad <input type="text" value="Winifreda"/>	Dirección <input type="text" value="Zona Rural Lote 20"/>
Fecha nacimiento <input type="text" value="19/11/1998"/>	Teléfono <input type="text" value="2302694413"/>	E-Mail <input type="text" value="erasmo7jordan@hotmail.com"/>
Género <input type="text" value="Masculino"/>	<input type="checkbox"/> Modificar Contraseña	Usuario <input type="text" value="jordanerasmo7"/>
		Contraseña nueva <input type="text" value="jordanerasmo7"/>
		Repetir <input type="text" value="jordanerasmo7"/>
◀ Volver		 Modificar

En el formulario se encuentran los datos personales del usuario registrado. A continuación, se listará en detalle la información que debería de ingresar en cada campo:

Nombre completo: Campo de texto, debe ingresar su nombre y apellido.

Documento: Campo numérico, debe ingresar su número de documento.

Nacionalidad: Campo de texto, debe ingresar su país de nacimiento.

Provincia: Campo seleccionable, con opción de búsqueda, debe seleccionar su provincia de residencia.

Localidad: Campo seleccionable, con opción de búsqueda, debe seleccionar su actual localidad de residencia.

Dirección: Campo de texto, debe ingresar su actual dirección de residencia, preferentemente con la estructura “Barrio-Calle-Altura o número”.

Fecha de nacimiento: Campo de fecha seleccionable, debe ingresar su fecha de nacimiento.

Teléfono: Campo numérico, debe ingresar su número de teléfono.

E-Mail: Campo de texto específico, debe ingresar su correo electrónico, no se aceptan datos que no cumplan con la estructura de un e-mail.

Género: Campo seleccionable, debe seleccionar el género que corresponda.

Modificar contraseña: Campo de tipo seleccionable, si se activa, se permitirá modificar la contraseña actual.

Usuario: Campo alfanumérico, debe ingresar su nombre de usuario, este debe ser único en todo el sistema.

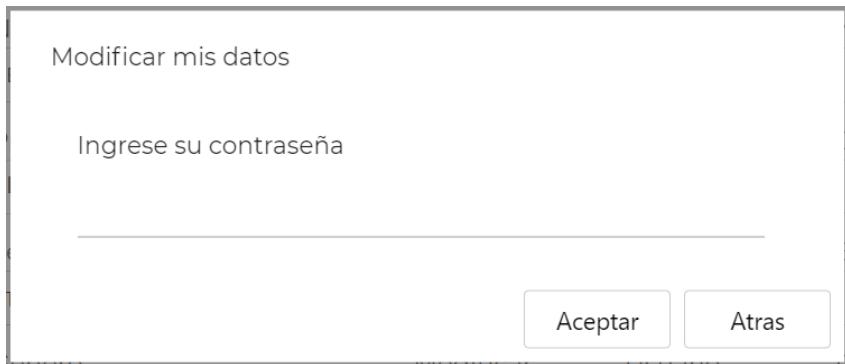
Contraseña nueva – Repetir: Campo alfanumérico, debe ingresar la nueva contraseña que va a utilizar en el sistema. Debe de contener mínimamente cinco caracteres.

Al presionar en “Volver”, se cerrará el formulario de administración y el usuario será redireccionado a la pantalla principal del sistema.

* 3.2.2.1 Modificar información general

Primero: Reemplazar el contenido de los campos deseados con la nueva información.

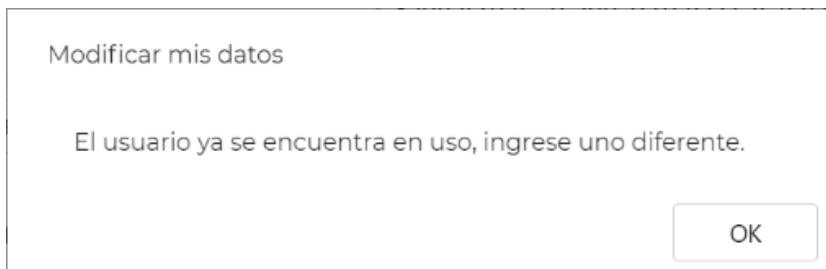
Segundo: Presionar en la opción en “Modificar”:



Tercero: Ingresar la contraseña actual y presionar en “Aceptar”. Si la contraseña es correcta, los cambios serán modificados y se verá un mensaje de éxito, en caso contrario, el proceso no se llevará a cabo y el sistema pedirá que ingrese nuevamente su contraseña.

Para cerrar la ventana y volver al formulario, presione en “Atrás”.

*Si modificamos el usuario, en el caso de que el ingresado ya se encuentre en uso, se desplegará el siguiente aviso:



En este caso, presione en “OK”, ingrese un nuevo usuario y presione en “Reintentar”. Si los cambios se registraron sin problemas, se desplegará un mensaje de éxito.

* 3.2.2.2.2 Modificar contraseña

Primero: Activar la opción  si se desactiva, se revierten los cambios.

Segundo: Ingresar la nueva contraseña en los campos  

Tercero: Seguir el mismo procedimiento desde el segundo paso del apartado 3.2.2.2.1 (Modificar información general).

➡ 3.2.2.3 Salir del sistema

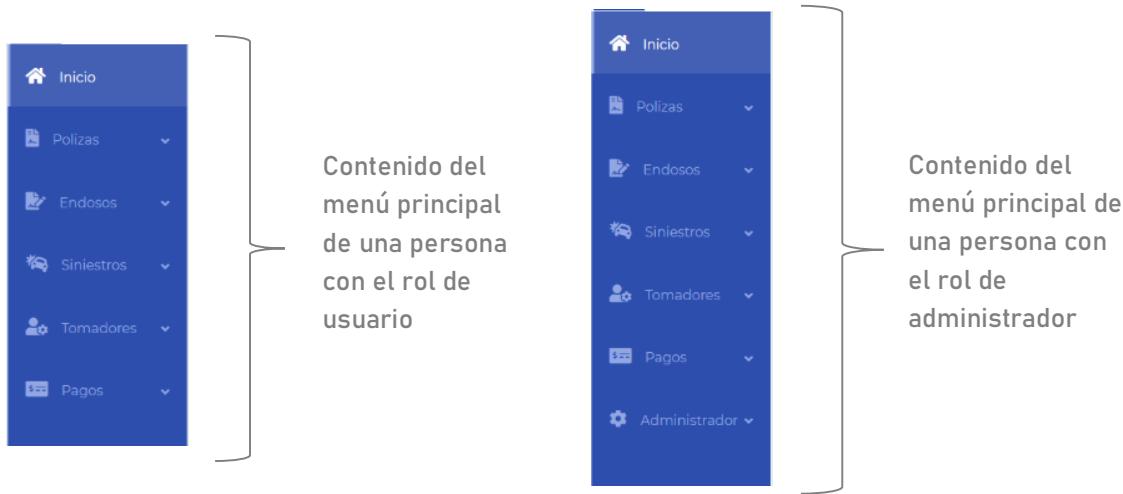
Primero: Presionar en la opción [Salir del sistema](#)

Segundo: En la siguiente alerta de confirmación, presionar en la opción “Si”.



Para volver al sistema, presione en la opción “Atrás”.

✳ 3.2.3 Menú principal

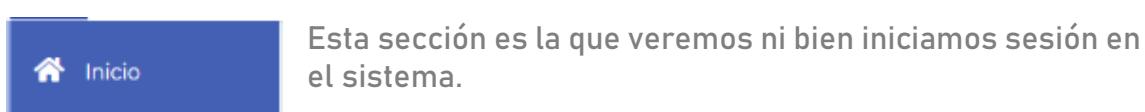


El menú principal se encuentra en el sector izquierdo de la pantalla, es donde se encuentra el listado de opciones de todas las actividades disponibles para realizar en el sistema. El contenido de este varía según el rol asignado de la persona que inicia sesión. Cada una de las acciones que se pueden realizar serán detalladas en la sección 3.3, (Utilizar el sistema).

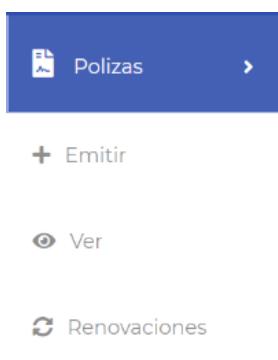
➡ 3.3 Utilizar el sistema

En esta sección se detallará todo lo que el usuario debe tener en cuenta para llevar a cabo cada tipo de acción, así como cada uno de los procedimientos a seguir para poder realizar las actividades correctamente.

✳ 3.3.1 Inicio



3.3.2 Pólizas



En esta sección podremos llevar a cabo todas las actividades relacionadas a las pólizas.

Dentro de las opciones disponibles podremos emitir una nueva paliza, consultar toda la información relacionada a una póliza, y llevar a cabo las renovaciones de aquellas que se encuentren vencidas.

3.3.2.1 Emitir nueva póliza



Primero: Nos dirigimos al menú principal y presionamos en “Pólizas”, una vez desplegado el listado de las acciones, presionar en la opción “Emitir”, seguido de esto, se presentará el siguiente formulario:

Cronología de navegación de las secciones del registro de una póliza.

Emision De Poliza

Poliza ▶ Tomador ▶ Vehiculo ▶ Cobertura ▶ Pago

Datos De Poliza

Se asignara el numero EH846298624806575 a la poliza a registrar

Inicio de vigencia
Seleccione

Fin de vigencia
Seleccione vigencia ▾

Fecha de finalizacion
Seleccione una opcion anterior

Siguiente ▶

Es requerido llenar la información solicitada de cada una de las secciones en orden cronológico, es decir, para que el sistema permita finalizar el proceso, no debemos saltarnos ningún paso de la sucesión, como tampoco ninguno de sus campos requeridos.

En caso de que la información de cada campo no sea ingresada o no sea permitida, los formularios de cada sección tomaran la siguiente apariencia:

En la imagen se ilustra como ejemplo la sección de vehículo, pero el funcionamiento es el mismo para todas las secciones.

Los campos resaltados en rojo son aquellos que no se ingresó correctamente la información

Emission De Poliza

Poliza Tomador Vehiculo Cobertura Pago

Datos Del Vehiculo

Suma Asegurada 560.000	Marca Seleccione	Modelo Seleccionar Marca	Patente ABC123 (Sin espacios)	Motor 2.0 TSI 20V 230CV		
Nro Chasis B2N5R6S9E9R6VJ5TB	Nro Motor GH323JK	Clase Seleccione	Uso Seleccione			
GPS <input type="checkbox"/>	0 Km <input type="checkbox"/>	Kilometros 124.000	Codigo Postal Seleccione	Localidad Seleccionar Codigo Postal		
Coberturas Adicionales Seleccione	Combustible Seleccione	Gas <input type="checkbox"/>	Año 2017	Pasajeros 5	Asientos 5	Color Blanco

[Atras](#) [Siguiente ▶](#)

Debe ingresar el Numero de Chasis
Debe ingresar el Numero de Motor
Debe seleccionar la Clase
Debe seleccionar el Uso

En esta parte, se mostrarán los errores correspondientes a los campos los cuales no haya sido ingresada la información correcta, detallando que tipo de dato debe ingresar.

Segundo: Completar cada una de las secciones con la información requerida:

Emision De Poliza

Poliza
Tomador
Vehiculo
Cobertura
Pago

Datos De Poliza

Se asignara el numero **EH846298624806575** a la poliza a registrar

Inicio de vigencia	Fin de vigencia	Fecha de finalizacion
<input type="text" value="14/11/2020"/>	<input type="button" value="Seleccione vigencia"/>	<input type="button" value="Seleccione una opcion anterior"/>

* 3.3.2.1.1 Sección póliza

En la parte superior, el sistema nos muestra el número aleatorio con el que se va a registrar la póliza.

Inicio de vigencia:

Inicio de vigencia

Seleccione

Noviembre 2020

Do	Lu	Ma	Mi	Ju	Vi	Sa
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

Campo de tipo fecha, debe seleccionar la fecha en la que iniciara la vigencia de la póliza, solo estarán disponibles las fechas futuras, incluyendo el mismo día de emisión en caso de no haber superado las 12:00 hs del mediodía, de lo contrario, no estará disponible. Este campo es obligatorio.

Fin de vigencia:

Fin de vigencia

Seleccione vigencia

Seleccione vigencia

- 4 Meses
- 5 Meses
- 6 Meses
- 7 Meses
- 8 Meses
- 9 Meses
- 10 Meses
- 11 Meses
- 12 Meses

Campo seleccionable, debe seleccionar cuantos meses tendrá de vigencia la póliza, este campo es obligatorio y estará disponible una vez seleccione la fecha de inicio de vigencia.

Emision De Poliza

Poliza Tomador Vehiculo Cobertura Pago

Datos De Poliza

Se asignara el numero EO6669961I6536531 a la poliza a registrar

Inicio de vigencia 19/11/2020	Fin de vigencia 4 Meses	Fecha de finalizacion 19/03/2021
----------------------------------	----------------------------	-------------------------------------

La vigencia establecida sera desde el Jueves 19 Noviembre 2020 hasta el Viernes 19 Marzo 2021 a las 11:59 AM

Siguiente ►

Al seleccionar la opción, se mostrará la vigencia escrita en un recuadro ubicado en la zona inferior a los campos.



Fecha de finalización: Campo que no requiere interacción del usuario, muestra la fecha exacta en la que finalizara la vigencia en el momento en que seleccione el campo anterior.

Para finalizar, presione en “Siguiente”, y si todos los campos fueron completados con la información correcta, pasaremos a la siguiente sección.

* 3.3.2.1.2 Sección tomador

Emision De Poliza

Poliza **Tomador** Vehiculo Cobertura Pago

Datos Del Tomador

Nombre	Nacionalidad	Codigo Postal	Telefono
Documento	Provincia	Calle	Correo
Tipo de Persona	Localidad	Genero	Nacimiento

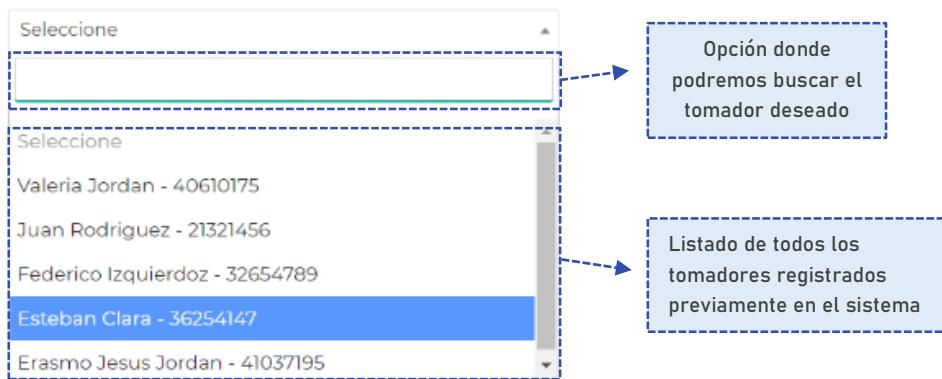
Atras + Nuevo Siguiente ►

Presione en “Atrás”, para volver a la etapa anterior. Al hacerlo, no se perderán los cambios.

En la sección de tomador tenemos dos opciones:

-Seleccionar un tomador ya registrado en el sistema:

Presionamos en “Seleccione”



Una vez hayamos encontrado el tomador deseado, presionamos sobre la opción para que quede seleccionada.

-Registrar un nuevo tomador desde cero:

Para registrar un nuevo tomador, debe presionar en el botón “Nuevo” ubicado en la parte superior derecha de la sección, seguido de esto se abrirá la siguiente ventana:

Datos Del Nuevo Tomador		
Nombre/Razon Social Nombre Apellido	Documento/CIUT 000000000000 (Sin espacios)	Persona Seleccione
Nacionalidad/Pais Nombre Pais	Nacimiento/Orig 00/00/0000 (Seleccione)	Provincia/Distrito Seleccione
Código Postal Seleccione Provincia	Localidad Seleccione Código Postal	Teléfono 000000000000
Calle Nombre/Numero Calle	Género Seleccione	Correo correo@ejemplo.com (Opcional)

X Cerrar Grabar

El proceso a seguir para llenar los campos del formulario es exactamente el mismo que se detalla en la sección 3.2.3.5.1 (Agregar nuevo tomador), y una vez realizado, para finalizar debe presionar en “Grabar”. Presione “Cerrar” para salir de la ventana, al hacerlo, se perderán los cambios.

Una vez se haya realizado cualquiera de los dos procesos para ingresar un tomador, la interfaz será la siguiente:

Emision De Poliza

Poliza ➔ Tomador ➔ Vehiculo ➔ Cobertura ➔ Pago

Datos Del Tomador

Erasmo Jesus Jordan - 41037195				+ Nuevo
Nombre	Nacionalidad	Codigo Postal	Telefono	
Erasmo Jesus Jordan	Argentina	6313	2302694415	
Documento	Provincia	Calle	Correo	
41037195	La Pampa	Zona Rural		
Tipo de Persona	Localidad	Genero	Nacimiento	
Fisica	Winifreda	Masculino	19/11/1998	

[Atras](#)

[Siguiente ➔](#)

Recuadro donde se detallan todos los datos del tomador seleccionado.

Presione “Atrás” para volver a la etapa anterior, no se perderán los cambios.

Presionar en “Siguiente” para pasar a la siguiente sección.

* 3.3.2.1.3 Sección vehículo

Emision De Poliza

Poliza ➔ Tomador ➔ **Vehiculo ➔** Cobertura ➔ Pago

Datos Del Vehiculo

Suma Asegurada	Marca	Modelo	Patente
560.000	Seleccione	Seleccionar Marca	ABC123 (Sin espacios)
Nro Chasis	Nro Motor	Clase	Uso
B2N5R6S9E9R6VJ5TB	GH323JK	Seleccione	Seleccione
GPS	0 Km	Kilometros	Codigo Postal
<input type="checkbox"/>	<input type="checkbox"/>	124.000	Seleccione
Coberturas Adicionales	Combustible	Gas	Año
Seleccione	Seleccione	<input type="checkbox"/>	2017
Pasajeros	Asientos	Color	
5	5	Blanco	

[Atras](#)

[Siguiente ➔](#)

Presione “Atrás” para volver a la etapa anterior, no se perderán los cambios.

Suma asegurada: Campo de texto, debe ingresar el valor asegurado del vehículo. Este campo es obligatorio.

Marca: Campo seleccionable con buscador integrado, debe seleccionar la marca del vehículo. Este campo es obligatorio.

Modelo: Campo de tipo seleccionable con buscador integrado, debe seleccionar el modelo del vehículo. Este campo es obligatorio.

Patente: Campo alfanumérico, debe ingresar la matrícula del vehículo. Este campo es obligatorio.

Motor: Campo alfanumérico, debe ingresar algunas características del tipo de motor del vehículo (Cilindrada, cantidad de válvulas, potencia, etc.). Este campo es obligatorio.

Número de chasis: Campo alfanumérico, debe ingresar el número de registro del chasis del vehículo. Este campo es obligatorio.

Número de motor: Campo alfanumérico, debe ingresar el número de registro del motor del vehículo. Este campo es obligatorio.

Clase: Campo seleccionable, debe seleccionar la clase de vehículo de la que se trata. Este campo es obligatorio.

Uso: Campo seleccionable, debe seleccionar el tipo de uso que se le brinda al vehículo. Este campo es obligatorio.

GPS: Campo seleccionable, debe marcarlo en caso de que el vehículo requiera la funcionalidad de GPS adicional. En caso de que la respuesta sea "NO" debe dejarlo en blanco.

0 Km: Campo seleccionable, debe marcarlo en caso de que el vehículo sea cero kilómetro. En caso de que la respuesta sea "NO" debe dejarlo en blanco.

Kilómetros: Campo numérico, debe ingresar la cantidad de kilómetros que tiene realizado el vehículo. En el caso de que la opción anterior sea "No", este campo es obligatorio, de lo contrario, no se encontrara disponible para completar.

Código postal: Campo seleccionable con buscador integrado, debe seleccionar el código postal correspondiente a la ubicación donde se encuentra el vehículo. Este campo es obligatorio.

Localidad: Campo seleccionable con buscador integrado, debe seleccionar la localidad correspondiente a la ubicación donde se encuentra el vehículo. Este campo es obligatorio y se habilita al completar el anterior.

Coberturas adicionales: Campo seleccionable, debe seleccionar el tipo de cobertura adicional que elija tener el tomador para el vehículo. Este campo es obligatorio, en caso de que el tomador decida no tener ninguna, se encuentra esta opción dentro de las opciones.

Combustible: Campo seleccionable, debe seleccionar el tipo de combustible que utiliza el vehículo. Este campo es obligatorio.

Gas: Campo seleccionable, debe marcarlo en el caso de que el vehículo utilice equipo de gas integrado. En caso de no poseer, debe dejarlo en blanco.

Año: Campo numérico, debe ingresar el año en que el vehículo fue fabricado. Este campo es obligatorio.

Pasajeros: Campo numérico, debe ingresar la capacidad máxima de pasajeros que puedan habitar el vehículo. Este campo es opcional.

Asientos: Campo numérico, debe ingresar la cantidad máxima de asientos con los que cuente el vehículo. Este campo es opcional.

Color: Campo alfanumérico, debe ingresar el color del vehículo, puede hacerse escribiéndolo o colocar el valor numérico del color. Este campo es opcional.

Una vez se completen todos los campos con la información adecuada, presione en "Siguiente" para pasar a la sección que sigue.

* 3.3.2.1.4 Sección cobertura

Emission De Poliza

Poliza Tomador Vehiculo Cobertura Pago

Tipo De Cobertura

Seleccione

[Atras](#) [Siguiente](#)

Presione "Atrás" para volver a la etapa anterior, no se perderán los cambios.

Presione en la opción "Seleccione" para que se despliegue el listado de todas las coberturas disponibles:

Tipo De Cobertura

Seleccione

TIPO A: Responsabilidad Civil Hacia Terceros
TIPO B: Perdidas Totales
TIPO C: Responsabilidad Civil con Robo/Incendio Total
TIPO C1: Terceros Completo
TIPO D: Todo Riesgo con Franquicia fija
TIPO E: Todo Riesgo sin Franquicia

Una vez haga click en la cobertura deseada, esta pasara a seleccionarse.

Tipo De Cobertura

TIPO C1: Terceros Completo

En este caso, la póliza irá a cubrir Responsabilidad Civil hacia Terceros de la cobertura A, más la cobertura de Robo y/o Hurto e Incendio, en aquellos casos que el perjuicio para el vehículo sea parcial o total. En el caso de Daños por accidente, este seguro automotor cubre solamente la Destrucción Total del vehículo.

[Atras](#) [Siguiente](#)

Recuadro donde se mostrará la descripción de la cobertura seleccionada

Para finalizar con esta sección, presione en “Siguiente”, y pasaremos a la siguiente sección.

* 3.3.2.1.5 Sección pago.

Emision De Poliza

Poliza Tomador Vehiculo Cobertura Pago

Datos De Pago

Prima total \$ 6566	Cant cuotas Seleccione	Prima mensual \$ 	Dia a facturar Seleccione	Metodo de pago Seleccione
------------------------	---------------------------	----------------------	------------------------------	------------------------------

Atras Grabar Poliza Cancelar

Presione “Atrás” para volver a la etapa anterior, no se perderán los cambios.

Prima total: Campo numérico, debe ingresar la prima total correspondiente a las características de la póliza en proceso. El valor se genera automáticamente por el sistema, pero se puede modificar a criterio del usuario. Este campo es obligatorio.

Cantidad de cuotas: Campo seleccionable, debe seleccionar la cantidad de cuotas a generar para el pago de la póliza según las opciones disponibles. Este campo es obligatorio.

Prima mensual: Campo que no requiere interacción del usuario, se asignara el valor de la prima mensual al momento de que la anterior opción se seleccionada.

Día a facturar: Campo seleccionable, debe seleccionar el día en que el tomador decida que se facture cada prima mensual. Este campo es obligatorio.

Método de pago: Campo seleccionable, debe seleccionar el tipo de método que el tomador utilizara para pagar cada prima mensual. Este campo es obligatorio. La opción “Asociar tarjeta de crédito” se encuentra deshabilitada temporalmente hasta que la compañía decida qué pasarela de pago utilizara para asociar las tarjetas de crédito.

Tercero:

En cualquier momento del registro, presione en “Cancelar” seguido de la opción “Si” para anulará el proceso de registro realizado hasta el momento.

Una vez que todas las anteriores secciones cuyos campos fueron completados con su respectiva información correspondiente, para terminar el proceso presione en “**Grabar Póliza**”, seguido aparecerá una notificación la cual contiene un informe resumiendo todos los parámetros a tener en cuenta para el pago de la póliza:

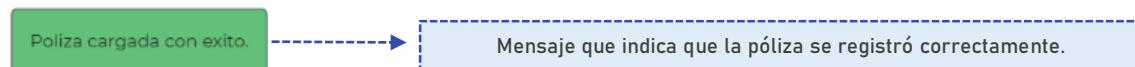
Registrar Pago

Valor de la prima total de \$6580 que se realizará en 4 pagos de \$1645.
El pago de la prima mensual se facturara cada 1 mes/es, en el dia 10 del mes que así corresponda, durante un plazo de 4 mes/es de vigencia.

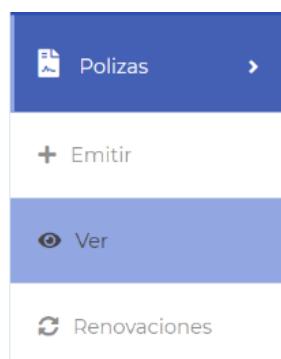
Registrar Atras

Presione en “Atrás” para regresar al formulario en el caso que desee realizar alguna modificación en cualquiera de las secciones.

Presione en “Registrar” para terminar con el proceso de registro de la póliza. En el caso de que el registro haya finalizado correctamente, se mostrará un mensaje de éxito, en caso contrario, será un mensaje de error.



➔ 3.3.2.2 Consultar pólizas



Nos dirigimos al menú principal y presionamos en “Pólizas”, una vez desplegado el listado de las acciones, presionar en la opción “Ver”, seguido de esto, se mostrará el listado de todas las pólizas registradas en el sistema en la siguiente interfaz:

Consultar Polizas

Opción para filtrar el contenido de la tabla. Al presionar en una opción, podrá ver solo aquellas pólizas cuyo estado sea el mismo que haya seleccionado en el filtro.

Campo para poder buscar una póliza específica a través de cualquiera de sus características (Sea por número, tomador, fechas, etc.).

Listado de todas las pólizas registradas en el sistema, así se encuentren en cualquiera de sus estados.

Icono para acceder a la información de la póliza (Instrucción 3.2.3.2.2.1).

Botón para actualizar en tiempo real el estado de las pólizas.

Descripción de los parámetros que aparecen en la tabla:

- Nro.: Número de la póliza.
- Tomador: Tomador correspondiente de la póliza.
- Vehículo: Marca y modelo del vehículo registrado en esa póliza.
- Vigencia: Fecha de inicio y Fecha de finalización de la vigencia de la póliza.
- Anulación: Fecha en la que se anulará la póliza.
- Emisión: Fecha y hora en la que se registró esa póliza.
- Estado: Estado actual con el que se encuentra la póliza.
- Info: Resumen del estado actual con el que se encuentra la póliza.
- Ver: Botón de interacción.

* 3.3.2.2.1 Ver la información de la póliza

Primero: Buscar la póliza deseada a través del buscador

Segundo: Presionar en el botón para acceder a la información de la póliza, detallados en el siguiente formulario:

Consultar Polizas

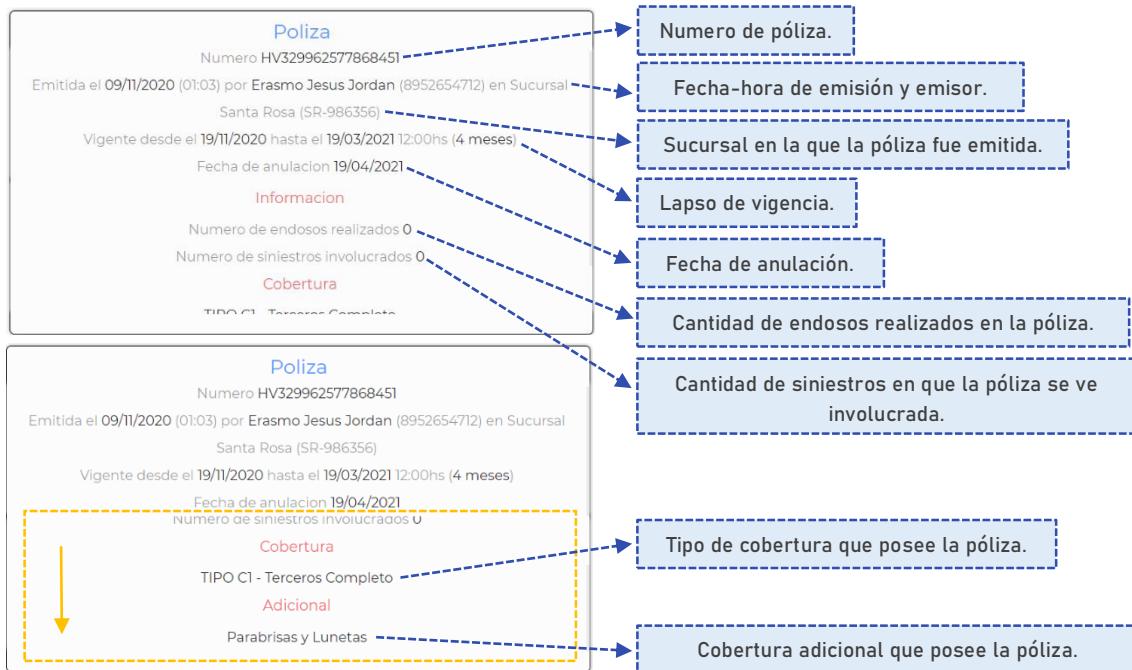
Poliza Numero: HV329962577868451 Emitida el 09/11/2020 (01:03) por Erasmo Jesus Jordan (8952654712) en Sucursal Santa Rosa (SR-986356) Vigente desde el 19/11/2020 hasta el 19/03/2021 12:00hs (4 meses) Fecha de anulacion 19/04/2021 numero de endosos realizados 0 Numero de siniestros involucrados 0 Cobertura TIPO C1 - Terceros Completo Adicional	Tomador Numero 99496322 Nombre completo Erasmo Jesus Jordan Nro documento 41037195 Tipo de persona Física Nacionalidad Argentina Domicilio Zona Rural, Winifreda La Pampa Código postal 6313 Fecha nacimiento 19/11/1998 Genero Masculino Telefono 2302694415 Correo	Pagos Método de pago Numero de pago 03861353458 Fecha de registro 09/11/2020 Prima total \$6580 Método de pago Pago en efectivo Estado actual En vigencia Cuotas <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Número</th> <th>Identificador</th> <th>Prima</th> <th>Estado</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>1646687074</td> <td>\$1645</td> <td>A facturar</td> </tr> <tr> <td>2</td> <td>5683979204</td> <td>\$1645</td> <td>A facturar</td> </tr> <tr> <td>3</td> <td>2142596775</td> <td>\$1645</td> <td>A facturar</td> </tr> <tr> <td>4</td> <td>7308952974</td> <td>\$1645</td> <td>A facturar</td> </tr> </tbody> </table>	Número	Identificador	Prima	Estado	1	1646687074	\$1645	A facturar	2	5683979204	\$1645	A facturar	3	2142596775	\$1645	A facturar	4	7308952974	\$1645	A facturar
Número	Identificador	Prima	Estado																			
1	1646687074	\$1645	A facturar																			
2	5683979204	\$1645	A facturar																			
3	2142596775	\$1645	A facturar																			
4	7308952974	\$1645	A facturar																			
Estados Actual Espera: Desde 09/11/2020 hasta Actualidad Anteriores	Vehículo / Riesgo Suma asegurada \$987000 Marca Volkswagen Modelo Bora Motor 1.8T 20V 180CV Patente MH-Z782 Nro chasis QWV2E3R4T5Y6U71809 Nro motor E562AE Clase Automóvil Uso Particular o Privado GPS No Cero Km No Cantidad de Kilometros: 67056 Kms Localidad Santa Rosa Código postal: 6300 Combustible Nafta Equipo de gas No Año 2014 Color Negro Asientos 5 Pasajeros 5	Actualizar																				

[Volver](#)

[Actualizar](#)

Presione en “Volver” para regresar al listado de pólizas.

En caso de presentarse algún fallo al momento de mostrar la información (como, por ejemplo, solo mostrarse la pantalla en blanco, o alguna alerta de error por alguna falla momentánea) presione en “Actualizar”, y en caso de persistir repetidas veces, comunicarse con el servicio técnico.



Tomador Número 98496322 Nombre completo: Erasmo Jesus Jordan . Nro documento 41037195 Tipo de persona Física Nacionalidad Argentina Domicilio Zona Rural, Winifreda La Pampa Código postal 6313 Fecha nacimiento 19/11/1998 Género Masculino Teléfono 2302694415 Correo erasmo7jordan@gmail.com	Datos personales del tomador.
--	--------------------------------------

Pagos Metodo de pago Número de pago 03861353458 Fecha de registro 09/11/2020 Prima total \$6580 Método de pago Pago en efectivo Estado actual En vigencia Cuotas <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Número</th> <th>Identificador</th> <th>Prima</th> <th>Estado</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>1645651074</td> <td>\$1645</td> <td>A facturar</td> </tr> <tr> <td>2</td> <td>5683979294</td> <td>\$1645</td> <td>A facturar</td> </tr> <tr> <td>3</td> <td>2142596775</td> <td>\$1645</td> <td>A facturar</td> </tr> <tr> <td>4</td> <td>7308952974</td> <td>\$1645</td> <td>A facturar</td> </tr> </tbody> </table>	Número	Identificador	Prima	Estado	1	1645651074	\$1645	A facturar	2	5683979294	\$1645	A facturar	3	2142596775	\$1645	A facturar	4	7308952974	\$1645	A facturar	Parámetros de pago. Información del pago y sus detalles. Detalles de todas las cuotas de ese pago.
Número	Identificador	Prima	Estado																		
1	1645651074	\$1645	A facturar																		
2	5683979294	\$1645	A facturar																		
3	2142596775	\$1645	A facturar																		
4	7308952974	\$1645	A facturar																		

Estados Actual Renovada: Desde 09/11/2020 - 22:16 hasta Actualidad Anteriores <div style="border: 1px dashed orange; padding: 5px; margin-bottom: 5px;"> Vencida: Desde 30/10/2020 - 11:59 hasta 09/11/2020 - 22:16 </div> <div style="border: 1px dashed orange; padding: 5px; margin-bottom: 5px;"> Vigente: Desde 30/10/2019 - 11:59 hasta 30/10/2020 - 11:59 </div> <div style="border: 1px dashed orange; padding: 5px; margin-bottom: 5px;"> Espera: Desde 28/10/2019 - 23:33 hasta 30/10/2019 - 11:59 </div>	Actual estado de la póliza. Línea del tiempo de los estados por los que haya pasado esa póliza, detallados con fecha y hora desde el primero hasta el actual. Historial donde se muestra el listado de estados anteriores al actual.
--	---

Vehículo / Riesgo Suma asegurada \$987.000 Marca Volkswagen Modelo Bora Motor 1.8T 20V 180CV Patente MH2782 Nro chasis QIW2E3R4T5Y6U71809 Nro motor E1652AE Clase Automóvil Uso Particular o Privado GPS No Cero Km No Cantidad de Kilometros 67.056 Kms Localidad Santa Rosa Código postal 6300 Combustible Nafta Equipo de gas No Año 2014 Color Negro Asientos 5 Pasajeros 5	Información del vehículo
--	---------------------------------

➔ 3.3.2.3 Renovar póliza



Primero: Nos dirigimos al menú principal y presionamos en “Pólizas”, una vez desplegado el listado de las acciones, presionar en la opción “Renovaciones”, seguido de esto, se mostrará el listado de todas las pólizas en estado “Vencida” que se encuentran registradas en el sistema:

En el ejemplo, solo hay una póliza vencida, por eso es la única que se encuentra en el listado.

The screenshot shows a table titled "Renovacion De Polizas" with the subtitle "Listado de polizas vencidas con su respectiva vigencia finalizada". The table has columns: Nro, Tomador, Vigencia, Emision, Anulacion, and Renovar. A search bar labeled "Buscar" is at the top right. A dashed box encloses the "Renovar" button in the last row of the table, which corresponds to the policy with Nro ZT159871320659759.

Segundo: Una vez ubiquemos la póliza deseada, presionamos en la opción “Renovar” que se representa en cada fila con el icono: ➔

* 3.3.2.3.1 Proceso de renovación denegado

Si al momento de intentar iniciar el proceso de renovación de una póliza, el sistema lo interrumpe mostrando el siguiente mensaje:



Este aviso nos indica que la póliza no tiene el total de prima abonada, lo cual es requisito indispensable para realizar la renovación.

La solución consiste en abonar las cuotas que, por un motivo u otro, no han sido pagadas por el tomador. Puede hacerlo desde el método de pago con el que trabaje la compañía, o a través de los procedimientos detallados en la instrucción 3.2.3.6.1.1 (Emitir recibo).

En el ejemplo se puede ver que el pago de la póliza tiene una cuota en estado “deuda”, teniendo en cuenta que todas deben estar con el estado “Pago aprobado”. Para poder ubicar las cuotas en deuda y posteriormente emitir su respectivo recibo, se recomienda seguir el siguiente instructivo:

Primero: Copiamos el número de pago que se detalla en el mensaje:

Verifique el pago nro: **06133413978**

Segundo: Nos dirigimos a la sección de facturar pagos (Instrucción 3.2.3.6.1) y lo copiamos en el buscador:

Facturas De Cuotas				
Listado de pagos registrados				
Pago num	Poliza num	Tomador	Nº Cuotas	Estado
06133413978	ZT159871320659759	Juan Rodriguez - 32549852	4	Modificado

Tercero: Presionamos en la opción y ya podremos acceder a las cuotas correspondientes.

Consultar Facturas De Pago							
Poliza ZT159871320659759 - Juan Rodriguez (32549852)				Prima Total \$20000		Buscar	
Ident	Nº	Prima	Lapso	Factura	Vto	Estado	Factura
8009411157	1	\$5000	30/10/2019 - 30/01/2020	09/11/2019	09/12/2019	Pago aprobado	
2884849634	2	\$5000	30/01/2020 - 30/04/2020	09/02/2020	09/03/2020	Pago aprobado	
5777360281	3	\$5000	30/04/2020 - 30/07/2020	09/05/2020	09/06/2020	Pago aprobado	
2105319905	4	\$5000	30/07/2020 - 30/10/2020	09/08/2020	09/09/2020	Deuda	Saldar

Se puede visualizar la cuota con deuda.

Cuarto: Realizamos el procedimiento descrito en la instrucción 3.2.3.6.1.1.1 (Emitir recibo) para cada una de las cuotas cuyo pago no se encuentra aprobado, y una vez hecho esto, el sistema permitirá realizar el proceso de renovación.

* 3.3.2.3.2 Proceso de renovación

Tercero: Luego de presionar en la opción de la póliza a renovar, se mostrará un formulario en el que tendremos que ingresar la siguiente información:

> 3.3.2.3.2.1 Datos de póliza

Renovacion De Polizas

Datos De Poliza ZT159871320659759

Inicio de vigencia	Fin de vigencia	Fecha de finalizacion
<input type="button" value="Seleccione"/>	<input type="button" value="Seleccione vigencia"/>	<input type="button" value="Seleccione una opcion anterior"/>

[Atras](#) [Siguiente ▶](#)

En este formulario debemos llenar los campos relacionados a la nueva vigencia que se aplicara en la póliza renovada. El procedimiento es igual al realizado en la instrucción 3.2.3.2.1.1 (Sección póliza) al emitir una nueva póliza. Una vez completados los campos, presione en “Siguiente”.

> 3.3.2.3.2.2 Datos del pago

Renovacion De Polizas

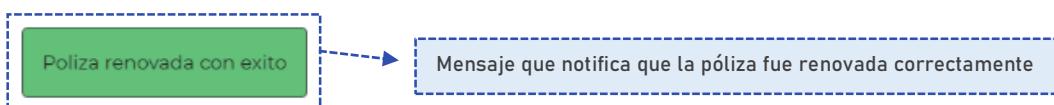
Datos Del Pago

Prima total \$	Cant cuotas	Prima mensual \$	Dia a facturar	Metodo de pago
<input type="text" value="6666"/>	<input type="button" value="Seleccione"/>	<input type="text"/>	<input type="button" value="Seleccione"/>	<input type="button" value="Seleccione"/>

[Atras](#) [Renovar](#)

En este formulario debemos llenar los campos correspondientes a los nuevos parámetros de pago para generar la nueva póliza renovada. El procedimiento es igual al realizado en la instrucción 3.2.3.2.1.5 (Sección pago) al emitir una nueva póliza. Una vez completados los campos, presione en “Renovar”.

Si el proceso fue realizado correctamente, se notificará con un mensaje de éxito y la póliza vencida desaparecerá de la lista de aquellas disponibles para ser renovada, de lo contrario se mostrará un mensaje de error.



Posteriormente si nos dirigimos al listado de todas las pólizas registradas en el sistema, y buscamos aquella que fue renovada recientemente, podremos ver que su estado fue modificado a “Renovada”:

Nueva póliza generada con el mismo número que la vencida.

Consultar Polizas

Todas	Nro	Tomador	Vehículo	Vigencia	Anulación	Emisión	Estado	Info	Ver
	ZT159871320659759	Juan Rodriguez - 32549852	Chevrolet Cruze	10/11/2020 - 10/03/2021	10/04/2021	09/11/2020 - 22:16	Espera	Sin cobertura - A espera del inicio de vigencia	
	HV329962577868451	Erasmo Jesus Jordan - 98496322	Volkswagen Bora	19/11/2020 - 19/03/2021	19/04/2021	09/11/2020 - 01:03	Espera	Sin cobertura - A espera del inicio de vigencia	
	SM674838607172036	Esteban Clara - 13947637	Dacia Logan	18/11/2020 - 18/11/2021	18/12/2021	06/11/2020 - 03:12	Espera	Sin cobertura - A espera del inicio de vigencia	
	ID298018993893817	Erasmo Jesus Jordan - 98496322	Abarth Punto Evo	06/11/2020 - 06/03/2021	06/04/2021	06/11/2020 - 02:30	Vigente	Con cobertura - Sin deudas	
	SS288759716312840	Juan Rodriguez - 32549852	Volkswagen Vento	03/11/2020 - 03/03/2021	03/04/2021	02/11/2020 - 15:24	Vigente	Con cobertura - Sin deudas	
	ZT159871320659759	Juan Rodriguez - 32549852	Chevrolet Cruze	30/10/2019 - 30/10/2020	30/11/2020	28/10/2019 - 23:33	Renovada	Sin cobertura - Se renovó una vez estando vencida	

Póliza vencida que fue renovada, finalizando su ciclo de vida.

3.3.3 Endosos

Endosos

- + Emitir
- Ver

En esta sección podremos llevar a cabo todas las actividades relacionadas a la modificación de los datos de las pólizas, o los denominados endosos.

Dentro de las opciones disponibles podremos emitir un nuevo endoso, como también poder consultar la información de cada uno de ellos.

3.3.3.1 Emitir un endoso

Endosos

- + Emitir
- Ver

Primero: Nos dirigimos al menú principal y presionamos en “Endosos”, una vez desplegado el listado de las acciones, presionar en la opción “Emitir”, seguido de esto, se mostrará el listado de todas las pólizas registradas en el sistema cuyo estado sea “Vigente” (o Vigente-deuda), las cuales son las únicas permitidas para realizar modificaciones o endosos:

Generar Endoso

Campo para buscar una póliza específica.

Buscar poliza

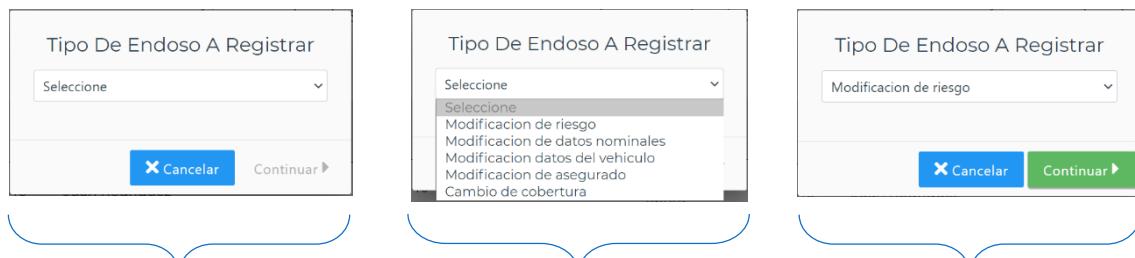
Poliza Nro	Tomador	Vehículo	Inicio Vig	Fin Vig	Estado	
SS288759716312840	Juan Rodriguez	Volkswagen Vento	03/11/2020	03/03/2021	Vigente	
ID298018993893817	Erasmo Jesus Jordan	Abarth Punto Evo	06/11/2020	06/03/2021	Vigente	
ZT159871320659759	Juan Rodriguez	Chevrolet Cruze	10/11/2020	10/03/2021	Vigente	

Listado de pólizas disponibles para realizar endosos.

Segundo: Una vez hayamos ubicado la póliza a la que se le realizara el endoso, debe presionar en la opción  que se encuentra a la derecha de la fila:

Tercero: Se mostrará una ventana en la que debe seleccionar el tipo de endoso que desee realizarle a la póliza, y a continuación presionar en “Continuar”.

Para cerrar la ventana y regresar a la tabla anterior, presione en “Cancelar”.

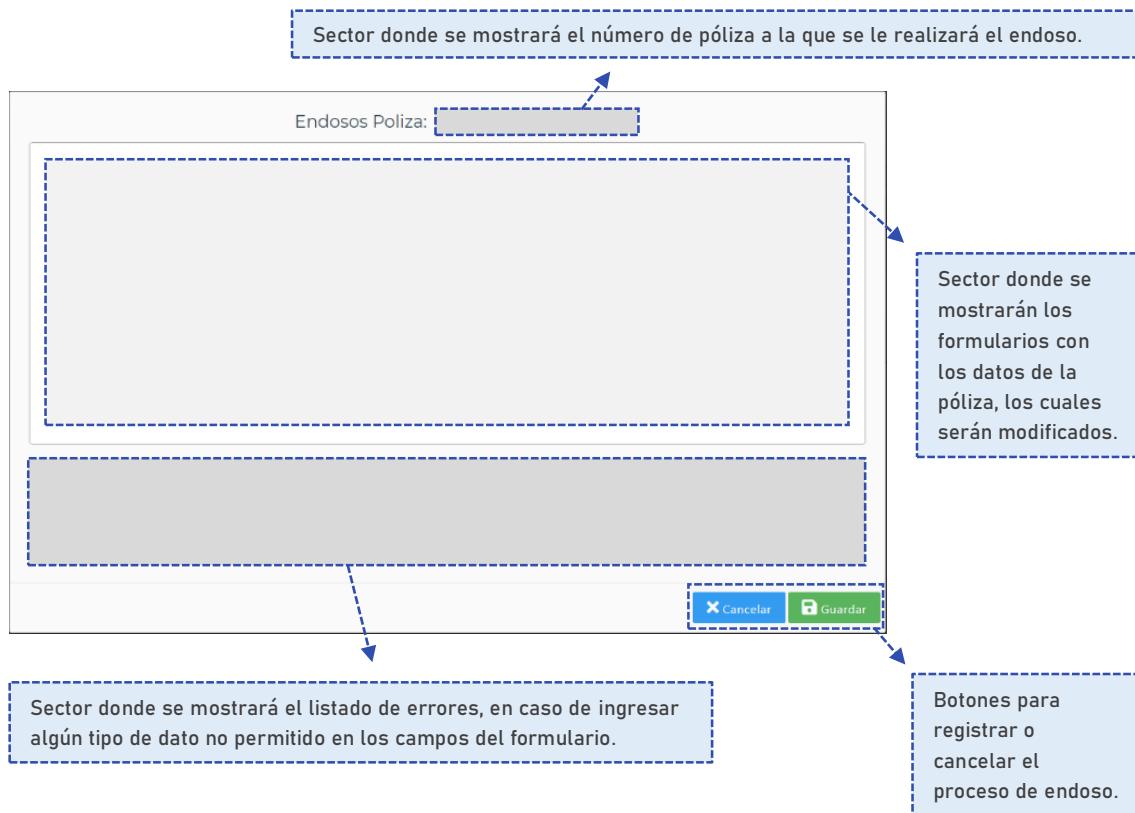


Ventana modal que se muestra al presionar en la opción para emitir un nuevo endoso en la póliza.

Listado de los tipos de endosos disponibles para realizar en una póliza.

Un tipo de endoso específico seleccionado. Al realizar esto, se habilita la opción de “Continuar”.

Estructura de como se muestra la ventana modal donde se realizaran los endosos:



Para cerrar la ventana y regresar al listado de pólizas, presione en “Cancelar”, y se mostrara una alerta de confirmación:



Al presionar en “Si”, además de re direccionar al listado de pólizas, se perderán los cambios que haya realizado hasta el momento.

Presione en “Atrás” si desea cerrar esta ventana y regresar al modal y continuar con la modificación de los datos de la póliza.

* 3.3.3.1.1 Modificación de riesgo

Seleccione
Modificación de riesgo
Modificación de datos nominales
Modificación datos del vehículo
Modificación de asegurado
Cambio de cobertura

Este tipo de endoso consiste en modificar el vehículo registrado en la póliza. Puede llevarse a cabo modificando solo el modelo, marca y modelo, o marca, modelo y el resto de la información.

Al presionar en “Continuar” se mostrará una ventana con el siguiente formulario:

Endosos Poliza: SS288759716312840

Modificación De Riesgo

Suma Asegurada 1.430.000	Marca Volkswagen	Modelo Vento	Patente MJI791	Motor 2.0 TSI DSG 211CV
Nro Chasis K7L6N4N516L3H0F8U6	Nro Motor T9987LP	Clase Automóvil	Uso Particular o Privado	
GPS <input checked="" type="checkbox"/>	0 Km <input checked="" type="checkbox"/>	Kilómetros 57	Código Postal 6300	Localidad Santa Rosa
Coberturas Adicionales Accidentes Personales	Combustible Nafta	Gas <input checked="" type="checkbox"/>	Año 2014	Pasajeros Asientos 5 5
				Color Negro

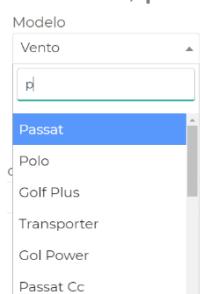
Cancelar Guardar

Los campos que se muestran son exactamente los mismos que los descritos en la instrucción 3.2.3.2.1.3 (Sección vehículo) al registrar una nueva póliza.

-Todos los campos de que no sean “Marca” y “Modelo” estarán deshabilitados, ya que corresponden al resto de información del vehículo (Que, al ser modificados, corresponden a otros tipos de endosos).

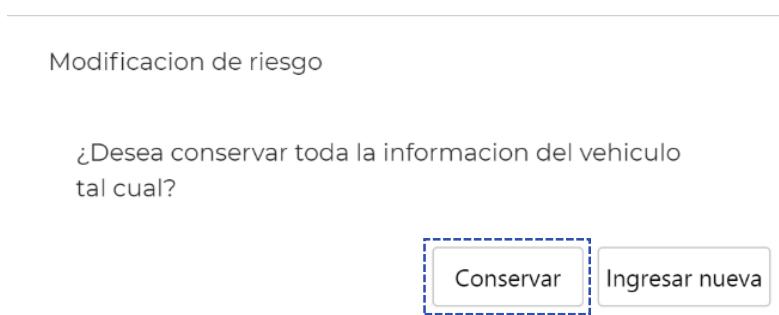
Se pueden realizar 2 tipos de modificaciones:

*Modificar solo la marca y modelo, o solo el modelo: En este caso, se modifica la marca y el modelo o solo el modelo del vehículo, y se **conserva el resto de la información tal cual**. Estos casos pueden darse, por ejemplo, cuando se registra el vehículo al emitir una nueva póliza, y toda la información ingresada es correcta, pero hubo una equivocación al seleccionar el modelo.



Para realizar el procedimiento, primero presione en el campo “Marca” o “Modelo” (dependiendo si desea modificar uno, otro o ambos) y seleccione en la opción deseada.

El sistema mostrara la siguiente alerta de confirmación:



Presione en “Conservar”, y de esta manera quedara modificado el campo que haya seleccionado, y la información restante se mantendrá tal cual estaba. En el ejemplo se conservó la marca “Volkswagen”, y el modelo “Vento” fue modificado a “Passat”, sin modificar el resto de los datos.

A screenshot of a web form titled 'Modificación De Riesgo'. The form is for policy number SS288759716312840. It includes fields for Suma Asegurada (1430.000), Marca (Volkswagen), Modelo (Passat), Nro Chasis (K7L6N4N5i6L3H0FB6), Nro Motor (TS987LP), Clase (Automóvil), Uso (Particular o Privado), GPS (checkbox), Kilometros (0 Km), Código Postal (6300), Localidad (Santa Rosa), Coberturas Adicionales (Accidentes Personales), Combustible (Nafta), Gas (checkbox), Año (2014), Pasajeros Asientos (5), and Color (Negro). At the bottom are 'Cancelar' and 'Guardar' buttons.

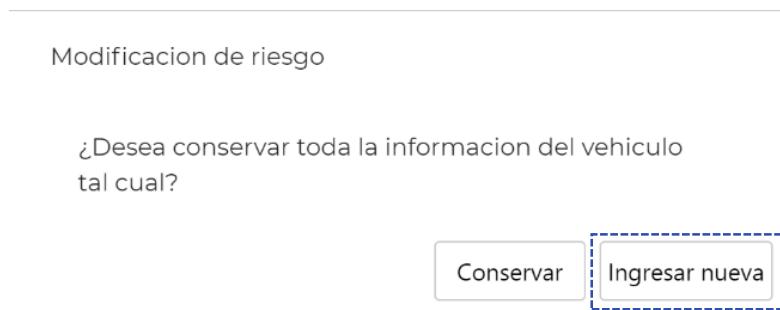
Formulario, donde fue modificado el modelo del vehículo, y conservando el resto de la información.

***Modificar la marca y modelo, o solo el modelo junto al resto de los datos:** En este caso, se modifica la marca y/o modelo o solo el modelo del auto, pero también debe completar todos los demás campos con información nueva. Se recomienda llevar a cabo esta acción en aquellos casos donde, **en la póliza, se deba modificar desde cero el vehículo registrado**. Esta situación generalmente se da cuando el tomador cambia de vehículo y desea aplicarle las coberturas de su actual póliza. No es buena práctica llevar a cabo este punto para modificar uno o dos campos, y por ende repetir la información en el resto, ya que para esas situaciones específicas existen otros tipos de endosos como pueden ser “Modificación de datos del vehículo” o “Modificación de datos nominales”.

Modelo
Passat
T-Cross
Voyage
Bora
Virtus
New Beetle
Suran

Para realizar el procedimiento, primero presione en el campo “Marca” o “Modelo” (dependiendo si desea modificar uno, otro o ambos) y seleccione en la opción deseada.

El sistema mostrara la siguiente alerta de confirmación:



Presione en “Ingresar nueva”, y de esta manera quedara modificado el campo que haya seleccionado, y el resto se habilitara para que ingrese la nueva información: (Realizaremos el endoso al Volkswagen Passat modificado previamente).

Endoso Poliza: SS288759716312840

Modificación De Riesgo

Suma Asegurada 560.000	Marca Volkswagen	Modelo Bora	Patente ABC 123	Motor 2.0 TSI 20V 230CV
Nro Chasis BZNSR659E9RGVJSTB	Nro Motor GH323JK	Clase Selección	Uso Selección	
GPS <input type="checkbox"/>	0 Km <input type="checkbox"/>	Kilometros 124.000	Código Postal Selección	Localidad Selección Localidad
Coberturas Adicionales Selección	Combustible Selección	Gas <input type="checkbox"/>	Año 2017	Pasajeros Asientos 5 5
Color Blanco				

Cancelar **Guardar**

Formulario donde todos los campos quedan disponibles para ser llenados con los datos del nuevo vehículo asignado a la póliza.

En el ejemplo se conservó la marca “Volkswagen”, y el modelo “Passat” fue modificado a “Bora”, y el resto de los campos fueron rellenados con los datos de este último vehículo.

Formulario, donde fueron modificados por completo los campos con la información del nuevo vehículo perteneciente a la póliza.

Una vez realizada la modificación de los datos del vehículo, de cualquiera de las formas permitidas (Apartados *), presione en “Guardar” para finalizar con el proceso de endoso de tipo “Modificación de riesgo”. Se mostrará un mensaje de éxito en caso de que se haya finalizado el proceso correctamente, de lo contrario, se notificara con un mensaje de error.



* 3.3.3.1.2 Modificación de datos nominales

Seleccione
Modificación de riesgo
Modificación de datos nominales
Modificación datos del vehículo
Modificación de asegurado
Cambio de cobertura

Este tipo de endoso consiste en modificar información específica del vehículo. Comprende: la suma asegurada, el número de matrícula, el número de chasis y el número de motor.

Más allá de la mencionada, la información que formará parte de los datos nominales depende del modelo de negocio con el que se rija la compañía.

Al presionar en “Continuar” se mostrará una ventana con el siguiente formulario:

Endosos Poliza: SS288759716312840

Modificacion De Datos Nominales

Suma Asegurada	Marca	Modelo	Patente	Motor
<input type="text" value="792.000"/>	<input type="text" value="Volkswagen"/>	<input type="text" value="Bora"/>	<input type="text" value="LKO349"/>	<input type="text" value="1.8T MT 180CV"/>
Nro Chasis	Nro Motor	Clase	Uso	
<input type="text" value="G8I7U6Y5T4R3E2LO8J"/>	<input type="text" value="MT898LI"/>	<input type="text" value="Automovil"/>	<input type="text" value="Particular o Privado"/>	
GPS	0 Km	Kilometros	Codigo Postal	Localidad
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="text" value="89.800"/>	<input type="text" value="6300"/>	<input type="text" value="Santa Rosa"/>
Coberturas Adicionales	Combustible	Gas	Año	Pasajeros Asientos
<input type="text" value="Parabrisas y Lunetas"/>	<input type="text" value="Nafta"/>	<input checked="" type="checkbox"/>	<input type="text" value="2013"/>	<input type="text" value="5"/>
				<input type="text" value="5"/>
				<input type="text" value="Gris"/>

X Cancelar
Guardar

Los campos que se muestran son exactamente los mismos que los descritos en la instrucción 3.2.3.2.1.3 (Sección vehículo) al registrar una nueva póliza.

En este tipo de endoso particular, solo estarán habilitados los campos “Suma asegurada”, “Patente”, “Nro de chasis” y “Nro de motor”, mientras que no se permite manipular el resto desde esta sección.

En el ejemplo, modificamos la suma asegurada de “792.000” a “807.000”.

Una vez hayamos modificado los campos deseados, presione en “Guardar”, y se mostrará un mensaje de éxito en caso de haber finalizado el proceso de registro del endoso correctamente, en caso contrario, el mensaje será de error.



* 3.3.3.1.3 Modificación de datos del vehículo

Seleccione

- Modificacion de riesgo
- Modificacion de datos nominales
- Modificacion datos del vehiculo**
- Modificacion de asegurado
- Cambio de cobertura

Este tipo de endoso consiste en modificar aquella información que no forme parte de los datos nominales del vehículo (la suma asegurada, el número de matrícula, el número de chasis y el número de motor).

Los denominados “Datos del vehículo” hacen referencia a los siguientes campos específicos: “Motor”, “Clase”, “Uso”, “GPS”, “0Km”, “Kilómetros”, “Código postal”, “Localidad”, “Coberturas adicionales”, “Combustible”, “Gas”, “Año”, “Pasajeros”, “Asientos” y “Color”. No permitiendo manipular el resto.

Los campos con los que podrá interactuar el usuario, son exactamente los mismos que los descritos en la instrucción 3.2.3.2.1.3 (Sección vehículo) al registrar una nueva póliza.

Al presionar en “Continuar” se mostrará una ventana con el siguiente formulario:

Endoso Poliza: SS288759716312840

Modificacion Datos Del Vehiculo

Suma Asegurada 807.000	Marca Volkswagen	Modelo Bora	Patente LKO349	Motor 1.8T MT 180CV
Nro Chasis G8I7U6Y5T4R3E2LO8J	Nro Motor MT998LI	Clase Automovil	Uso Particular o Privado	
GPS <input type="checkbox"/>	0 Km <input type="checkbox"/>	Kilometros 89.800	Codigo Postal 6300	Localidad Santa Rosa
Coberturas Adicionales Parabrisas y Lunetas	Combustible Nafta	Gas <input type="checkbox"/>	Año 2013	Pasajeros Asientos 5
				Color Gris

X Cancelar Guardar

En el ejemplo, se harán las siguientes modificaciones:

Campo Kilómetros: De “89.800” a “98.300”

Campo Código Postal: De “6300” a “6313”

Campo Localidad: De “Santa Rosa” a “Winifreda”

Campo Coberturas Adicionales: De “Parabrisas y Lunetas” a “Granizo”

Una vez modificada la información de los campos deseados, presione en “Guardar” y se mostrará un mensaje de éxito en caso de haber finalizado el proceso correctamente, de lo contrario, será un mensaje de error.



* 3.3.3.1.4 Modificación de asegurado

Seleccione
Modificación de riesgo
Modificación de datos nominales
Modificación datos del vehículo
Modificación de asegurado
Cambio de cobertura

Este tipo de endoso consiste en modificar la persona que es tomador de la póliza.

Una vez que presione en “Continuar”, se mostrara el siguiente formulario:

Endosos Poliza: SS288759716312840

Modificación De Asegurado

Juan Rodriguez - 21321456			
Nombre	Nacionalidad	Código Postal	Teléfono
Juan Rodriguez	Argentina	5340	3265888665
Documento	Provincia	Calle	Correo
21321456	Catamarca	Spinetto 325	rodriguez.juan@gmail.com
Tipo de Persona	Localidad	Género	Nacimiento
Física	Los Guaytmas	Masculino	16/07/1969

Cancelar **Guardar**

El campo con el que interactuara el usuario, es exactamente el mismo que el descrito en la instrucción 3.2.3.2.1.2 (Sección tomador) al registrar una nueva póliza.

Al abrirse la ventana, en el campo, por defecto ya se encontrará seleccionado el registro de la persona la cual es actualmente tomador.

En el ejemplo, se modificó el actual tomador “Juan Rodríguez” por “Federico Izquierdoz” quien será el nuevo tomador.

Una vez seleccione en la persona la cual será el nuevo tomador, presione en “Guardar”, y si el proceso se realizó correctamente, se mostrará un mensaje de éxito, en caso contrario, el mensaje será de error o informativo.

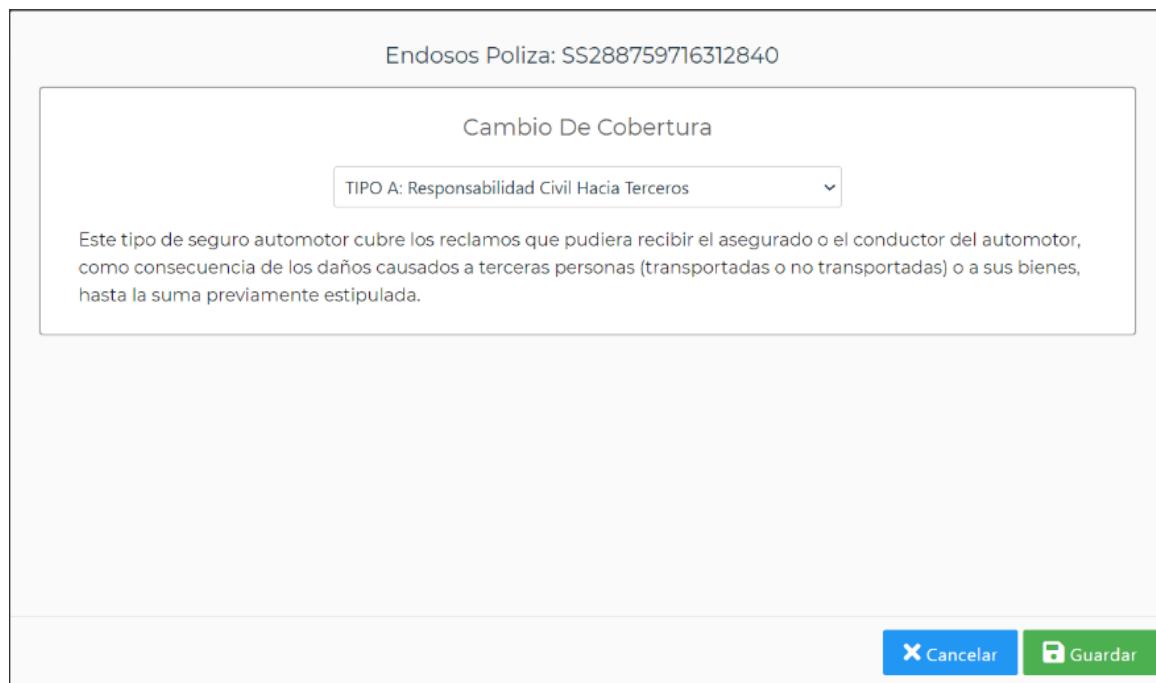


* 3.3.3.1.5 Cambio de cobertura

Seleccione
Modificación de riesgo
Modificación de datos nominales
Modificación datos del vehículo
Modificación de asegurado
Cambio de cobertura

Este tipo de endoso se lleva a cabo cuando se desea modificar el tipo de cobertura con el que la póliza asegura los riesgos.

Al presionar en “Continuar”, se mostrará el siguiente formulario:



Endoso Poliza: SS288759716312840

Cambio De Cobertura

TIPO A: Responsabilidad Civil Hacia Terceros

Este tipo de seguro automotor cubre los reclamos que pudiera recibir el asegurado o el conductor del automotor, como consecuencia de los daños causados a terceras personas (transportadas o no transportadas) o a sus bienes, hasta la suma previamente estipulada.

Cancelar **Guardar**

El campo con el que interactuara el usuario, es exactamente el mismo que el descrito en la instrucción 3.2.3.2.1.4 (Sección cobertura) al registrar una nueva póliza.

Al abrirse la ventana, en el campo se encontrará por defecto seleccionado el tipo de cobertura con el que actualmente la póliza brinda el seguro.

En el ejemplo, se modificó la cobertura “TIPO A: Responsabilidad civil Hacia Terceros” por “TIPO C1: Terceros Completo” la cual implementara desde ese momento la póliza.

Una vez seleccione el tipo de cobertura deseado, presione en “Guardar”, y si el proceso se realizó correctamente, se mostrará un mensaje de éxito, en caso contrario, el mensaje será de error o informativo.



➔ 3.3.3.2 Consultar endosos



Nos dirigimos al menú principal y presionamos en "Endosos", una vez desplegado el listado de las acciones, presionar en la opción "Ver", seguido de esto, se mostrará en la siguiente interfaz, el listado de todas aquellas pólizas las cuales en algún momento se les hayan realizado uno o más endosos:

The screenshot shows a table titled 'Consultar Endosos' with four columns: 'Poliza Nro', 'Tomador', 'Estado', and a fourth column with four small search icons. A dashed box highlights the first row of the table. Another dashed box highlights the search bar at the top right. A third dashed box highlights the bottom right corner of the table area. Arrows point from each of these three boxes to their respective explanatory text boxes below them.

Poliza Nro	Tomador	Estado	
ZT159871320659759	Juan Rodriguez	Renovada	🔍
SS288759716312840	Federico Izquierdoz	Vigente	🔍
ID298018993893817	Erasmo Jesus Jordan	Vigente	🔍
ZT159871320659759	Juan Rodriguez	Vigente	🔍

Listado de todas las pólizas a las cuales se les realizo algún endoso.

Campo para buscar una póliza específica.

Opción que nos permite visualizar el historial de todos los endosos realizados en esa póliza

En el listado, se mostrarán todas las pólizas a las que se les realizo uno o más endosos en algún momento durante su vigencia.

En el contenido principal podremos ver cada póliza representada en una fila de la tabla, donde la principal información, comenzando de izquierda a derecha es la siguiente:

Póliza nro.: Hace referencia al número de póliza.

Tomador: Actual tomador de la póliza a la que se le realizo el endoso.

Estado: Actual estado en el que se encuentra la póliza. Debemos tener en cuenta que, si en algún momento se le realizo un endoso a una póliza la cual actualmente se encuentra, ya sea vigente, vigente con deudas, vencida, renovada o anulada, igualmente se mostrara en la tabla, siendo indistinto si se le es permitido realizar nuevos endosos o no. Por este motivo es que pueden darse ocasiones en las que veamos la misma póliza repetida dos veces como se puede apreciar en el ejemplo (Póliza número ZT159871320659759 perteneciente a Juan Rodríguez, la cual fue renovada posteriormente a realizarle un endoso).

* 3.3.3.2.1 Ver endosos de una póliza

Una vez ubicada la póliza deseada, presionar en la opción  y a continuación se abrirá la siguiente ventana:



The screenshot shows a list of endorsements for policy number SS288759716312840. The table has the following columns: Nro end. (Endorsement Number), Registro (Registration Date), Hora (Hour), Tipo (Type), and Descripción (Description). The table contains six rows of data, each with a small edit icon (pencil) next to the description.

Nro end.	Registro	Hora	Tipo	Descripción
9776615777	11/11/2020	20:40	Cambio de cobertura	Cambio de cobertura poliza SS28875971...
4601718938	11/11/2020	20:26	Modificación de asegurado	Modificación de asegurado poliza SS28...
7031298648	11/11/2020	19:21	Modificación datos del vehículo	Modificación de datos del vehículo po...
8881273801	11/11/2020	18:58	Modificación de datos nominales	Modificación de datos nominales poliz...
6498038713	10/11/2020	19:54	Modificación de riesgo	Modificación de riesgo poliza SS28875...
5268665715	10/11/2020	19:09	Modificación de riesgo	Modificación de riesgo poliza SS28875...

Annotations in the screenshot:

- A callout box points to the top-left area of the table with the text: "Listado de todos los endosos realizados en la póliza (Aplicados en los ejemplos anteriores.)."
- A callout box points to the search bar with the text: "Campo para buscar un endoso específico, pudiendo hacerse desde cualquiera de sus características, como, por ejemplo, el número, fecha de registro, el tipo, etc."
- A callout box points to the "Modificar" (Edit) button in the bottom right corner of the table area with the text: "Botón que permite acceder a la ventana para modificar la descripción de ese endoso (Instrucción 3.2.3.3.2.2.)."
- A callout box points to the "Cerrar" (Close) button in the bottom right corner of the window with the text: "Botón para cerrar la ventana actual."

Cada endoso se ve representado en una fila de la tabla, en las columnas se detalla la información principal de cada uno, y se ven ordenados por fecha y hora, apareciendo en primer lugar aquellos que ultimo fueron registrados.

Nro. end.: Número único que se asigna al momento de registrar un nuevo endoso.

Registro: Fecha en la que el endoso fue registrado.

Hora: Hora y minutos en que fue registrado el endoso.

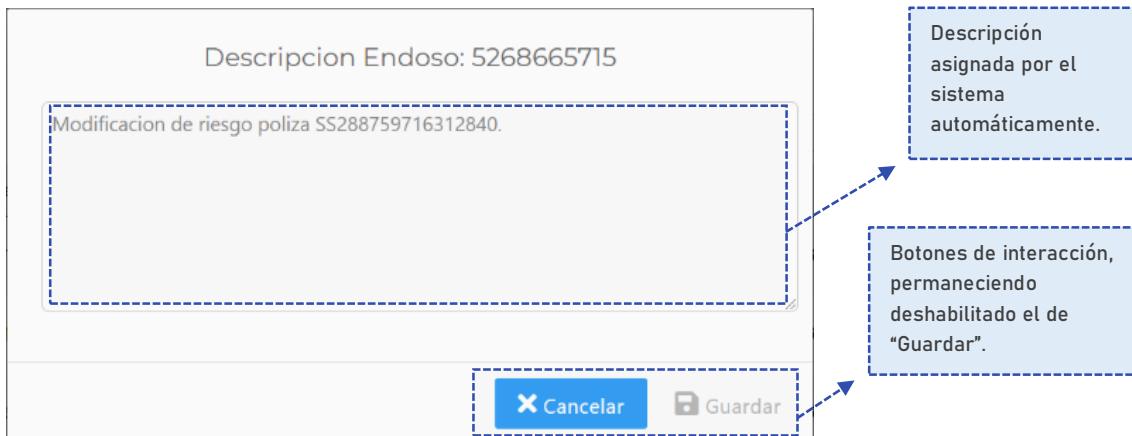
Tipo: Es el tipo de endoso al que hace referencia.

Descripción: Fragmento de la descripción del endoso que fue registrada por el usuario o por el sistema automáticamente.

Para salir de la ventana actual y volver al listado de pólizas, presione en "Cerrar"

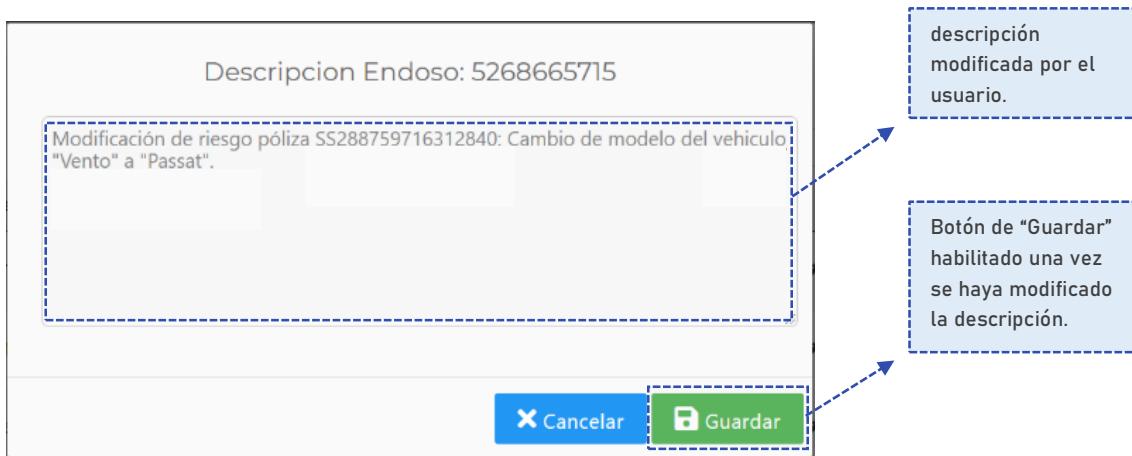
* 3.3.3.2.2 Modificar la descripción de un endoso

Una vez ubiquemos el endoso de una póliza deseado, presione en la opción a la derecha de la fila () seguido se abrirá la siguiente ventana:



En la ventana podremos visualizar un campo de texto donde se encontrará la descripción que actualmente tiene el endoso. Cabe destacar que, al momento de registrar un endoso específico, el sistema guardara una descripción por defecto automáticamente, y lo que permite esta opción es poder modificar ese dato. La descripción de un endoso está pensada para facilitar la lectura de la información y para que el usuario le resulte más fácil saber exactamente cuál fue la modificación que se realizó en ese momento.

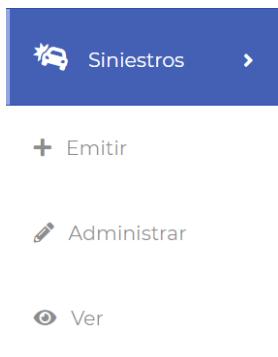
Primeramente, la opción “Guardar” se encontrará deshabilitada, y permanecerá así hasta que se realice una modificación en el campo de texto.



Una vez realizada la modificación del campo, presione en “Guardar” para terminar con el proceso de registro, en caso de llevarse a cabo correctamente, se lanzará un mensaje de éxito, en caso que ingrese la misma información, el mensaje será informando esto último, y en caso de que se haya producido un problema en el servidor, el mensaje será de error.



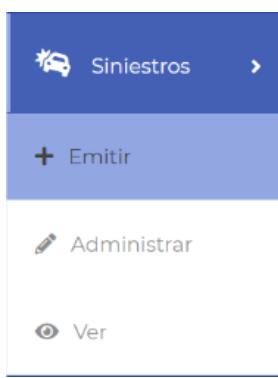
3.3.4 Siniestros



En esta sección podremos llevar a cabo todas las actividades relacionadas a los siniestros en los que se vea relacionado el asegurado de la póliza.

Dentro de las opciones disponibles podremos emitir un nuevo siniestro, modificar la información permitida de aquellos siniestros ya registrados, y consultar toda la información relacionada a un siniestro.

3.3.4.1 Emitir nuevo siniestro



Primero: Nos dirigimos al menú principal y presionamos en “Siniestros”, una vez desplegado el listado de las acciones, presionar en la opción “Emitir”, seguido de esto, se presentará el siguiente formulario:

Navegación donde se encuentra el listado de pasos a seguir para registrar un nuevo siniestro.

Emitir Nuevo Siniestro

Siniestro

Se asignara el numero 586691299298960ZT al siniestro a registrar

Fecha de Ocurrencia: Seleccionar

Hora de Ocurrencia: Seleccionar

Hubo Lesión/Muerte: Seleccionar

Poliza: Seleccionar una Poliza

Tipo de Siniestro: Seleccionar

Siguiente ➔

Para completar el registro de un nuevo siniestro, el formulario constara de tres o cinco pasos, partiendo de izquierda a derecha dependiendo del tipo de siniestro del que se trate.

Cada paso puede verse reflejado en la navegación ubicada en la parte superior, en la cual podremos presionar sobre las opciones para visualizar el formulario y el tipo de información que debemos ingresar.

Estructura general de las secciones al intentar continuar con el registro sin haber completado la información de los campos requeridos obligatoriamente:

Emitir Nuevo Siniestro

Siniestro

Se asignara el numero 586691299298960ZT al siniestro a registrar

Fecha de Ocurrencia: Seleccionar

Hora de Ocurrencia: Seleccionar

Hubo Lesion/Muerte: Seleccionar

Poliza: Seleccionar una Poliza

Tipo de Siniestro: Seleccionar

Siguiente ➔

**Debe seleccionar la Fecha de Ocurrencia
Debe seleccionar la Hora de Ocurrencia
Debe seleccionar si Hubo Lesión/Muerte
Debe seleccionar una Poliza**

Sector donde se mostrará el listado errores en los datos de cada campo requerido.

Al principio el menú de navegación se encontrará con todos los pasos disponibles, más adelante explicaremos en qué momento y en qué circunstancias se deshabilitan ciertas secciones de la navegación superior.

En el último paso tendremos disponible la opción de “Grabar siniestro”, y en todos los anteriores la opción “Siguiente”, **siempre y cuando en los pasos anteriores se completen todos los campos requeridos**, ya que debe seguirse obligatoriamente la estructura lineal del formulario, de lo contrario, cada una de las secciones posteriores estará disponible solo para su visualización sin tener dichas opciones habilitadas.

Emitir Nuevo Siniestro

Accidente

Descripción

Provincia: Seleccionar

Calle

Localidad: Seleccionar una Provincia

Altura

Opción de "Siguiente" no disponible.

Accediendo a la sección "Accidente" desde la navegación superior, sin antes haber completado las secciones de "Información general" y "Denunciante".

Emitir Nuevo Siniestro

Accidente

Descripción

Provincia: Seleccionar

Calle

Localidad: Seleccionar una Provincia

Altura

Opción de "Siguiente" disponible.

Accediendo a la sección "Accidente" habiendo previamente completado las secciones "Información general" y "Denunciante".

Segundo: Completar la información de cada sección:

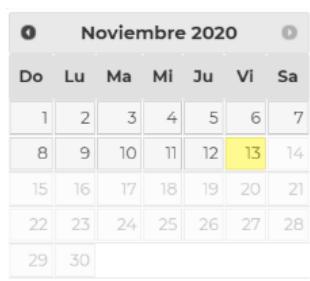
* 3.3.4.1.1 Información general

Emitir Nuevo Siniestro

Informacion General	Denunciante	Accidente	Reclamante	Conductor
Siniestro				
Se asignara el numero 586691299298960ZT al siniestro a registrar				
Fecha de Ocurrencia Seleccione	Hora de Ocurrencia --::--	Hubo Lesion/Muerte Seleccione		
Poliza Seleccione una Poliza	Tipo de Siniestro Seleccione			
Siguiente ➔				

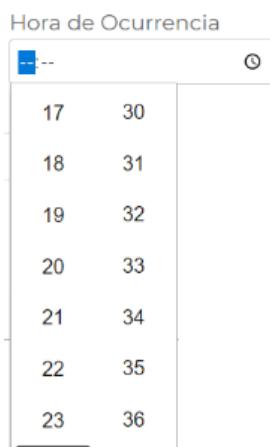
En el recuadro superior nos indica el número que se le asignará al siguiente siniestro que será registrado.

En esta sección, se ingresarán los datos en los siguientes campos:



Fecha de ocurrencia: Campo de tipo fecha, debe seleccionar la fecha en que ocurrió el siniestro, solo estarán disponibles aquellos días anteriores al día en que el usuario registre el siniestro, siendo este inclusive. Este campo es obligatorio.

El día presente en el ejemplo corresponde al 13/11/2020, por lo que estarán disponibles solo aquellos anteriores a este.



Hora de ocurrencia: Campo de tipo texto, solo es permitido ingresar un horario con el formato de 24 horas (15:00) sin más, en caso contrario, el dato será considerado invalido. En el caso de haber seleccionado el día presente en el campo “Fecha de ocurrencia” (En el ejemplo 13/11/2020), solo estarán disponibles aquellos horarios **diez minutos anteriores al actual**, y en caso de haber seleccionado cualquier día anterior, **todos los horarios estarán permitidos**.

Por ejemplo, si el día seleccionado es 13/11/2020 (actual), siendo en ese momento las 16:00, solo estarán disponibles aquellos horarios anteriores a las 15:50. En caso de que el valor ingresado no sea permitido, se mostrara el siguiente error:

La hora de Ocurrencia ingresada no está permitida Este campo es obligatorio.

Hubo lesión/muerte: Campo de selección, en caso de que en el siniestro hayan resultado personas con heridas o se presenten defunciones, seleccione la opción “Si”, en caso de no haberse llevado a cabo ninguna de las anteriores situaciones, seleccione la opción “No”. Este campo no puede dejarse en blanco.

Póliza:

Poliza

Seleccione una Poliza

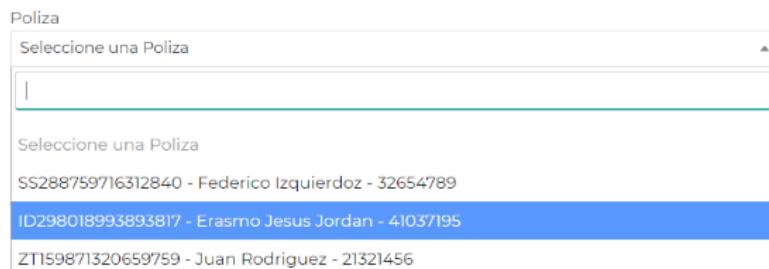
|

Seleccione una Poliza

SS288759716312840 - Federico Izquierdoz - 32654789

ID298018993893817 - Erasmo Jesus Jordan - 41037195

ZT159871320659759 - Juan Rodriguez - 21321456



Campo de selección, debe seleccionar la póliza que cubrirá el siniestro en proceso de registro, se desplegarán solo aquellas que se encuentren en estado “Vigente”, no permitiendo la selección de las que se encuentren con los estados “Vigente-deuda”, “Vencida”, “Renovada” o “Deuda”. Se integra una entrada que funciona como buscador. Este campo es obligatorio.

Seleccione una Poliza

SS288759716312840 - Federico Izquierdoz - 32654789

ID298018993893817 - Erasmo Jesus Jordan - 41037195 - (Poliza Vencida)

ZT159871320659759 - Juan Rodriguez - 21321456 - (Poliza con deudas)



Opción deshabilitada ya que la póliza se encuentra sin vigencia.

Opción deshabilitada ya que la póliza se encuentra con deuda.

Tipo de siniestro:

Tipo de Siniestro

Seleccione

Seleccione

Destrucción Total

Distracción Parcial

Rotura de Luneta o Parabrisas

Deterioro por Granizo

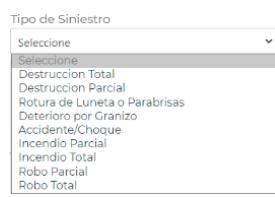
Accidente/Choque

Incendio Parcial

Incendio Total

Robo Parcial

Robo Total



Campo de selección, debe seleccionar el tipo de siniestro del que se trata, dependiendo de este campo, estarán deshabilitadas algunas opciones secciones del formulario que no serán necesarias llenar. Este campo es obligatorio.

Si selecciona el tipo de siniestro “Accidente/Choque”, automáticamente se habilitarán las secciones “Reclamante” y “Conductor” en la navegación superior para tener acceso a sus campos y poder rellenarlos, dada la situación de que si existe una tercera persona la cual estuvo involucrada en el acontecimiento.

Emitir Nuevo Siniestro

Informacion General Denunciante Accidente Reclamante Conductor

Siniestro

Se asignara el numero 533644920054132DK al siniestro a registrar

Fecha de Ocurrencia: Seleccionar Hora de Ocurrencia: Seleccionar Hubo Lesion/Muerte: Seleccionar

Poliza: Seleccionar una Poliza

Tipo de Siniestro: Accidente/Choque

Secciones de "Reclamante" y de "Conductor" habilitadas.

Opción "Accidente/Choque" seleccionada.

Siguiente ►

En caso de seleccionar cualquiera de las otras opciones, el formulario finaliza en la sección “Accidente”, ya que no se encontrarán habilitados los de “Reclamante” y “Conductor”.

Emitir Nuevo Siniestro

Informacion General Denunciante Accidente Reclamante Conductor

Siniestro

Se asignara el numero 533644920054132DK al siniestro a registrar

Fecha de Ocurrencia: Seleccionar Hora de Ocurrencia: Seleccionar Hubo Lesion/Muerte: Seleccionar

Poliza: Seleccionar una Poliza

Tipo de Siniestro: Deterioro por Granizo

Secciones de "Reclamante" y de "Conductor" deshabilitadas.

Opción diferente a "Accidente/Choque" seleccionada.

Siguiente ►

Una vez complete todos los campos requeridos, presione en “Siguiente” para ser dirigido a la siguiente sección “Denunciante”.

* 3.3.4.1.2 Denunciante

Emitir Nuevo Siniestro

Informacion General Denunciante Accidente Reclamante Conductor

Denunciante

El Denunciante es la misma persona que el Asegurado

Seleccionar

Nombre Completo: _____ DNI: _____ Domicilio: _____

Telefono: _____ E-mail: _____

Atras ▲ Siguiente ►

En esta sección debe ingresar los datos de la persona que realiza la denuncia del siniestro. Esa persona, por medidas del modelo de negocio, debe tener relación directa con el conductor del vehículo asegurado, o directamente ser el mismo tomador.

El denunciante es la misma persona que el Asegurado: Campo seleccionable, si la persona que lleva a cabo la denuncia es el tomador, seleccione en la opción “Si”, en caso contrario, si quien hace la denuncia es otra persona, seleccione la opción “No”, y seguido de esto debe completar los campos ubicados en la parte inferior, los cuales “Nombre Completo”, “DNI”, “Domicilio” y “Teléfono” son requeridos obligatoriamente, mientras que “E-mail” es opcional.

Una vez seleccionada la opción, presione en “Siguiente” para pasar a la sección siguiente.

* 3.3.4.1.3 Accidente

Emitir Nuevo Siniestro

Informacion General ➔ Denunciante ➔ **Accidente ➔** Reclamante ➔ Conductor

Accidente

Descripción

Provincia
Seleccione

Calle

Localidad
Seleccione una Provincia

Altura

Atras ▶ Siguiente ▶

En esta sección deberá describir los detalles del accidente en sí, ingresando la ubicación exacta y una descripción del suceso.

Descripción: Campo de texto, debe describir paso por paso lo más detalladamente posible como se llevó a cabo el accidente. Este campo es obligatorio.

Provincia: Campo de selección, debe seleccionar la provincia en la que ocurrió el accidente. Este campo es obligatorio.

Localidad: Campo de selección, debe seleccionar la localidad en la que ocurrió el accidente. Este campo es obligatorio.

Calle: Debe ingresar la denominación de la calle en la que ocurrió el accidente, este campo no es obligatorio ya que no siempre pueden llevarse a cabo los acontecimientos sobre una calle, pero es altamente recomendable rellenarlo en situaciones que si se requiera.

Altura: Debe ingresar la altura en la que ocurrió el accidente, al igual que el campo “Calle”, se recomienda rellenarlo siempre y cuando sea posible, aunque no es obligatorio.

En esta sección pueden darse dos casos, dependiendo del tipo de siniestro seleccionado en la instrucción 3.2.3.4.1.1 (Información general):

Si se seleccionó el tipo de siniestro diferente a “Accidente/Choque”, el formulario finalizaría en esta sección de “Accidente”, ya que no sería necesario ingresar los datos de una tercera persona:

Emitir Nuevo Siniestro

Información General Denunciante Accidente Reclamante Conductor

Accidente

Descripción
Vehículo marca Peugeot 206 se encontraba libre de ocupantes sobre calle Güemes altura 989 en sentido cardinal de este a oeste sobre su vera izquierda, cuando llegó al mismo me encuentro con el vehículo chocado y al consultar una cámara de seguridad ubicada en frente del lugar del acontecimiento, se aprecia claramente que la parte delantera rodado es colisionado por la parte trasera de una camioneta marca Ford Ranger dominio UZ665 color rojo, la cual intentaba estacionar, dicho vehículo se marcha del lugar sin dar informe de lo ocurrido ni dejar un contacto. Luego logro dar con el dueño de la Ford Ranger y obtuve sus datos personales y los de su compañía de seguros.

Provincia: La Pampa Localidad: Santa Rosa
Calle: Güemes Altura: 988

Atras Cerrar Grabar Siniestro

En este punto, para terminar con el proceso de registro del siniestro, presione en “Grabar Siniestro” en caso de haberse finalizado correctamente, se mostrará un mensaje de éxito, de lo contrario será un mensaje de error.

Presione en “Cancelar” si desea no continuar con el proceso, y reiniciar el formulario, presione en “Cancelar”.

En caso de haber seleccionado el tipo de siniestro “Accidente/Choque”, se mostrará la opción “Siguiente”:

Emitir Nuevo Siniestro

Información General Denunciante Accidente Reclamante Conductor

Accidente

Descripción
Vehículo marca Peugeot 206 se encontraba libre de ocupantes sobre calle Güemes altura 989 en sentido cardinal de este a oeste sobre su vera izquierda, cuando llegó al mismo me encuentro con el vehículo chocado y al consultar una cámara de seguridad ubicada en frente del lugar del acontecimiento, se aprecia claramente que la parte delantera rodado es colisionado por la parte trasera de una camioneta marca Ford Ranger dominio UZ665 color rojo, la cual intentaba estacionar, dicho vehículo se marcha del lugar sin dar informe de lo ocurrido ni dejar un contacto. Luego logro dar con el dueño de la Ford Ranger y obtuve sus datos personales y los de su compañía de seguros.

Provincia: La Pampa Localidad: Santa Rosa
Calle: Güemes Altura: 998

Atras Siguiente

En este caso, una vez ingrese la información correcta en los campos, presione en “Siguiente” para pasar a la siguiente sección.

* 3.3.4.1.4 Reclamante tercero

Emitir Nuevo Siniestro

Informacion General Denunciante Accidente **Reclamante** Conductor

Reclamante (Tercero)

Nombre Completo Domicilio
Telefono DNI

[Atras](#) [Siguiente ▶](#)

En esta sección, se debe ingresar la información de la tercera persona involucrada en el siniestro.

Todos los campos son obligatorios.

Una vez se hayan completado todos los campos, presione en “Siguiente”.

* 3.3.4.1.5 Conductor tercero

Emitir Nuevo Siniestro

Informacion General Denunciante Accidente **Reclamante** **Conductor**

Conductor (Tercero)

El Conductor es la misma persona que el Reclamante

Nombre Completo Domicilio
Telefono DNI

[Atras](#) [Cancelar X](#) [Grabar Siniestro F](#)

Esta corresponde a la última sección del formulario para registrar un siniestro, es donde debe ingresarse la información de aquella persona la cual conducía el vehículo (Dado el caso) del reclamante (Tomador de la póliza tercera).

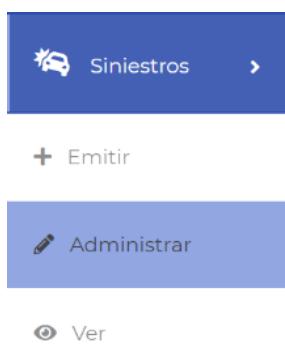
El conductor es la misma persona que el reclamante: Campo de selección, si la persona reclamante es quien conducía el vehículo al momento del siniestro, seleccione la opción “Si”, en caso contrario, si el vehículo se encontraba libre de ocupantes o quien conducía es otra persona diferente al reclamante, seleccione

la opción “No”, y a continuación deberá completar la información de los campos ubicados en la parte inferior, siendo todos requeridos obligatoriamente.

Una vez haya llenado el formulario con la información correcta, para terminar el proceso de registro presione en “Grabar Siniestro”, y se mostrará un mensaje de éxito en caso de haber finalizado correctamente, en caso contrario se mostrará un mensaje de error.

Si desea no continuar con el registro, y reiniciar el formulario, presione en “Cancelar”.

→ 3.3.4.2 Administrar siniestros



Primero: Nos dirigimos al menú principal y presionamos en “Siniestros”, una vez desplegado el listado de las acciones, presionar en la opción “Administrar”, seguido de esto, se presentará la siguiente tabla:

The screenshot shows a table titled "Administrador Siniestros" displaying three rows of accident information. The columns are: Nro Siniestro, Nro Poliza, Tomador, Tipo, Denuncia, Estado, and Administrar. Each row has edit and delete icons. A search bar labeled "Buscar" is at the top right. Callouts explain: "Listado de todos los siniestros registrados en el sistema." points to the table; "Campo para realizar una búsqueda de un siniestro específico." points to the search bar; "Opción para cambiar de estado el siniestro (Instrucción 3.2.3.4.2.1)." points to the "Estado" column; "Opción para modificar la información del siniestro." points to the "Administrador" column; and "Opción para administrar las imágenes del siniestro (Instrucción 3.2.3.4.2.3)." points to the "Administrador" column.

Nro Siniestro	Nro Poliza	Tomador	Tipo	Denuncia	Estado	Administrador
203634861927185MK	ZT159871320659759	Juan Rodriguez	Accidente/Choque	31/10/2020 - 23:40	En trámite	
31728587489878AO	SS288759716312840	Federico Izquierdoz	Accidente/Choque	13/11/2020 - 20:45	En trámite	
553511054963650XW	ZT159871320659759	Juan Rodriguez	Destrucción Total	01/11/2020 - 17:29	En trámite	

Cada siniestro es representado por una fila de la tabla, y su correspondiente información es representada en cada columna.

Nro. siniestro: Es el número único generado y asignado por el sistema a ese siniestro al momento de ser registrado.

Nro. póliza: Es el número único identificativo que representa a la póliza la cual cubre ese siniestro.

Tomador: Es la persona que es tomador de la póliza.

Tipo: Es el tipo de siniestro del que se trata.

Denuncia: Fecha representada en día, mes, año, hora y minuto en que se registró el siniestro.

Estado: Es el actual estado en el que se encuentra el siniestro, junto a la opción para poder modificarlo ().

Administrar: Esta columna consta de dos opciones:

 : Es la opción para acceder al formulario donde podremos modificar aquella información permitida del siniestro.

 : Opción para acceder a la ventana donde podremos administrar las imágenes que representan al siniestro.

* 3.3.4.2.1 Modificar estado del siniestro

Una vez encontrado en la tabla el siniestro deseado, debe presionar en la opción que se ubica en la columna “Estado” (), y se abrirá la siguiente ventana:



Cambiar Estado De Siniestro
En trámite
Seleccione
En trámite
Con reclamo en juicio
Rechazado
Finalizado

Debe seleccionar el estado que quiera asignarle al siniestro, dentro de las opciones se encuentran los siguientes:

En trámite: Aplique este estado al siniestro si este se encuentra en negociación entre la tercera persona damnificada y la compañía.

Con reclamo en juicio: Seleccione este estado si el siniestro se encuentra intervenido por demandas o abogados en el proceso de negociación.

Rechazado: Aplique este estado si la negociación fue rechazada por motivos que no abalan seguir con el proceso.

Finalizado: Aplique este estado si la negociación entre la compañía y el reclamante fue resuelta legalmente.

Una vez haya seleccionado el estado, presione en la opción  y en caso de finalizar el proceso correctamente, se mostrará un mensaje de éxito:

* 3.3.4.2.2 Modificar información del siniestro

Una vez encontrado en la tabla el siniestro deseado, de las dos opciones que se encuentran en la columna “administrar”, debe presionar en la que se ubica a la izquierda (), y se mostrara la siguiente interfaz:

La interfaz es exactamente la misma que la que se presenta en la instrucción 3.2.3.4.1 (Emitir un nuevo siniestro), pero sin las opciones de “Siguiente” en cada sección (aunque si se podrá utilizar la navegación superior), en este caso solo estará el botón “Confirmar” y “Volver”.

Para realizar una modificación, solo debemos buscar la sección deseada e ingresar la información en los campos requeridos.

-El campo “Póliza” no puede ser modificado una vez registrado el siniestro.

-La fecha límite en el campo “Fecha de ocurrencia” será la misma que la fecha límite establecida al momento del registro. En el ejemplo, el día máximo al momento del registro fue el día presente (13/11/2020), por consiguiente, esta será la fecha límite permitida en días posteriores al momento de realizar modificaciones en el siniestro (En este caso estamos en la fecha 14/11/2020).

-El valor máximo que permite el campo “Hora de ocurrencia”, es exactamente el mismo horario máximo permitido al momento de registrar el siniestro. En el ejemplo el valor máximo que se permitía al momento de realizar el registro del siniestro fue 15:50, por ende, este será el límite máximo que pueda modificarse **en caso de seleccionar el día máximo permitido en el momento del registro (13/11/2020)**, de lo contrario el valor no estará permitido.

En el ejemplo, modificaremos el siniestro que fue registrado en la instrucción 3.2.3.4.1 (Emitir nuevo siniestro). Se realizarán las siguientes modificaciones:

Información general: Fecha de ocurrencia de “13/11/2020” a “12/11/2020”.

Denunciante: El denunciante es la misma persona que el asegurado de “Si” a “No” e ingresaremos la información de la persona.

Accidente: Calle de “Güemes” a “La Rioja”, y la descripción.

Reclamante: Teléfono de “2302654319” a “2302654329”.

Conductor: El Conductor es la misma persona que el Reclamante de “Si” a “No” e ingresaremos los datos de la persona.

Para cancelar el proceso de modificación y regresar al listado de siniestros presione en “Volver”.

Una vez haya modificado los campos de interés, presione en “Confirmar”, y si el proceso se realizó correctamente, se mostrará un mensaje de éxito, de lo contrario, el mensaje será de error.

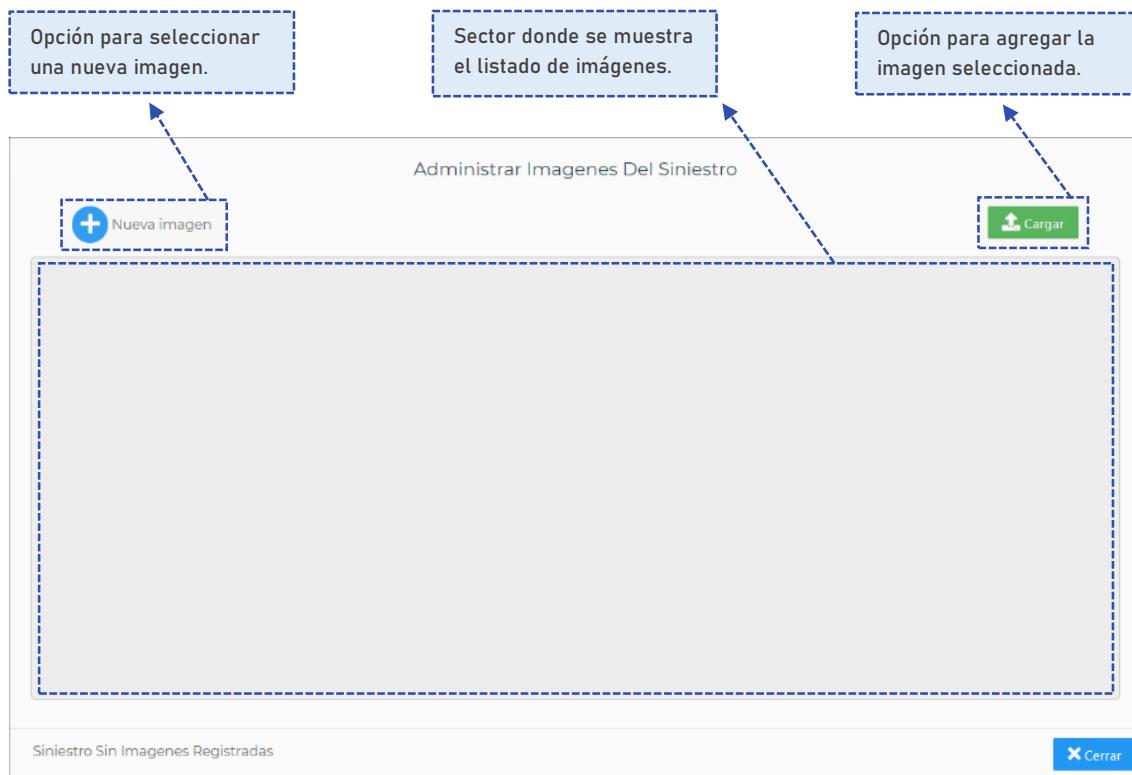


* 3.3.4.2.3 Administrar imágenes

Una vez registrado un siniestro, es posible guardar imágenes de las consecuencias del suceso del mismo, ya que muchas veces son necesarias para llevar a cabo el proceso de gestión legal.

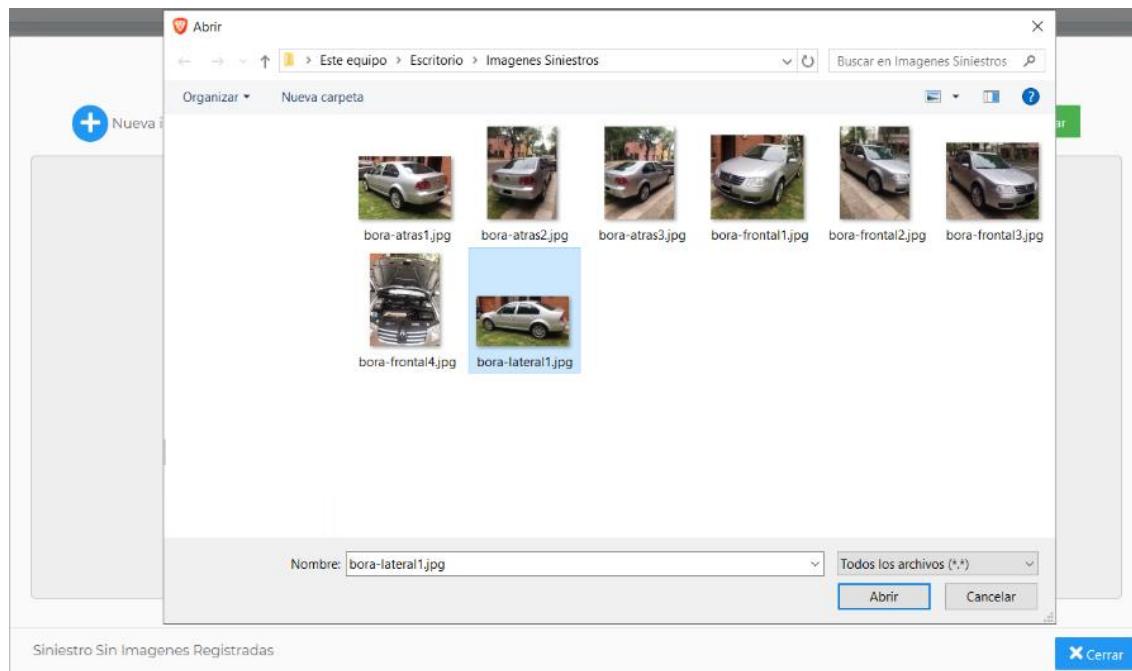
En esta sección podremos agregar hasta doce imágenes, con un tamaño máximo de cinco megabytes permitido para cada una, con formatos “JPG”, “PNG” y “JPEG”, y se encontrara disponible una opción para que cada una de ellas puedan ser eliminadas.

Una vez encontrado en la tabla el siniestro deseado, de las dos opciones que se encuentran en la columna “administrar”, debe presionar en la que se ubica a la derecha (), y se abrirá la siguiente ventana:



Agregar una nueva imagen:

Primero: Presione en la opción “Nueva imagen” y seleccione la imagen que desea subir y presione en “Abrir”:



Segundo: Una vez que se cierra la ventana, presione en la opción “Cargar” y ya podremos visualizar la imagen en el contenedor de la ventana.

Se deben seguir estos pasos para cada imagen que se desee subir.

Al agregar una nueva imagen, a la izquierda de la opción “Cargar” se mostrará una opción que representa el estado del registro de la imagen, que nos indicará con el siguiente ícono (✓) si la imagen se registró correctamente, y en caso contrario se mostrara el ícono de error (!) ya sea por los siguientes motivos:

1- No se seleccionó ninguna imagen: En caso de que intentemos agregar una imagen que nunca fue seleccionada, se mostrara el siguiente mensaje de notificación informativo:



2- Tamaño de imagen no permitido: En caso de intentar agregar una imagen cuyo tamaño es superior a cinco megabytes (5 Mb), se mostrará el siguiente mensaje de notificación informativo:



3- Formato de imagen no compatible: Si se intenta agregar una nueva imagen cuyo formato es diferente de “JPG”, “PNG” o “JPEG”, el sistema no permitirá seguir con el proceso y mostrará el siguiente mensaje de notificación informativo:



Una vez hayamos cargado todas las imágenes, al presionar sobre cualquiera de ellas, podremos visualizarla con un tamaño mayor, y también podremos navegar entre todas ellas a través de las flechas que tendremos disponibles:



Eliminar una imagen:

Para poder eliminar una imagen ya registrada en el sistema, solo tendremos que seleccionarla con el mouse, y en la parte superior derecha debemos presionar en el siguiente ícono: ()

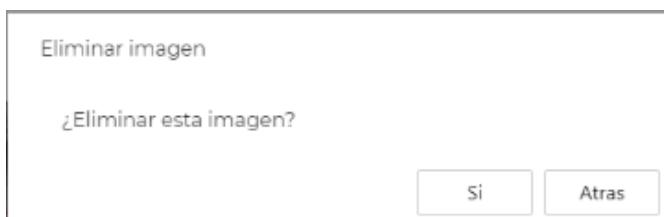
Administrador Imágenes Del Siniestro

Nueva imagen Cargar

Opción para eliminar la imagen.

Siniestro 31728587489878AO Volkswagen Bora Cerrar

Una vez presionada la opción de eliminar, se mostrará un mensaje de confirmación:



Presione en “Si” si desea eliminar la imagen.

Presione “Atrás” si desea volver al listado de imágenes.

➔ 3.3.4.3 Consultar siniestros

Primero: Nos dirigimos al menú principal y presionamos en “Siniestros”, una vez desplegado el listado de las acciones, presionar en la opción “Ver”, seguido de esto, se presentará la siguiente tabla:

Nro Siniestro	Nro Poliza	Tomador	Fecha Denuncia	Estado
203634861927185MK	ZT159871320659759	Juan Rodriguez	31/10/2020 - 23:40	En trámite
553511054963650XW	ZT159871320659759	Juan Rodriguez	01/11/2020 - 17:29	En trámite
31728587489878AO	SS288759716312840	Federico Izquierdoz	13/11/2020 - 16:00	Con reclamo en juicio

Campo para buscar un siniestro específico.

Buscar

Opción para visualizar la información del siniestro.

Cada siniestro es representado por una fila, mientras que en cada columna se mostrara información general del mismo, la cual ya hemos visto en la instrucción 3.2.3.4.2 (Administrar siniestros), pero con algunas opciones menos. Una vez ubicado el siniestro deseado, presione en la opción de visualización ubicada a la derecha de la fila () para acceder a sus datos, y se mostrara la siguiente interfaz:

Siniestro	Poliza	Accidente
Numero #31728587489878AO	Numero de poliza #SS288759716312840	Ubicacion Santa Rosa, La Pampa, La Rioja 988
Denuncia 12/11/2020 - 15:50	Tomador Federico Izquierdoz - #85935423	Descripcion Vehículo marca Peugeot 206 se encontraba libre de ocupantes sobre calle La Rioja altura 988 en sentido cardinal de este a oeste sobre su vera izquierda, cuando llegó al mismo me
Ocurrencia	Vigencia	

Denunciante	Reclamante (Tercero)	Conductor (Tercero)
Misma persona que el Tomador NO	Misma persona que el tomador NO	Misma persona que el reclamante NO
Nombre Ernesto Garcia	Nombre Juan Pedro Garcia	Nombre Enrique Palacios
Documento	Domicilio	Domicilio

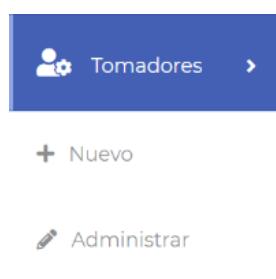
Volver

En cada recuadro podremos apreciar la información específica de los elementos que componen el siniestro, en los siguientes ejemplos podemos ver la información detallada del siniestro registrado anteriormente (Número 31728587489878AO):

Siniestro	Poliza	Accidente
<p>Numero #31728587489878AO</p> <p>Denuncia 12/11/2020 - 15:00</p> <p>Ocurrencia 13/11/2020 - 16:00</p> <p>Tipo Accidente/Choque</p> <p>Lesión/muerte NO</p> <p>Estado Con reclamo en juicio</p>	<p>Numero de poliza #55288759716312840</p> <p>Tomador Federico Izquierdoz - #85935423</p> <p>Vigencia 03/11/2020 al 03/03/2021 a las 11:59</p> <p>Estado actual Vigente</p>	<p>Ubicación Santa Rosa, La Pampa, La Rioja 988</p> <p>Descripción Vehículo marca Peugeot 206 se encontraba libre de ocupantes sobre calle La Rioja altura 988 en sentido cardinal de este a oeste sobre su vera izquierda, cuando llegó al mismo me encuentro con el vehículo chocado y al consultar una cámara de seguridad ubicada en frente del lugar del acontecimiento, se aprecia claramente que la parte delantera rodada es colisionado por la parte trasera de una camioneta marca Ford Ranger dominio LIZ665 color rojo, la cual intentaba estacionar, dicho vehículo se marcha del lugar sin dar informe de lo ocurrido ni dejar un contacto. Luego logró dar con el dueño de la Ford Ranger y obtuve sus datos personales y los de su compañía de seguros.</p>
Denunciante	Reclamante (Tercero)	Conductor (Tercero)
<p>Misma persona que el Tomador NO</p> <p>Nombre Ernesto García</p> <p>Documento 35963741</p> <p>Domicilio San Justo 563</p> <p>Telefono 2302592417</p> <p>Email</p>	<p>Misma persona que el tomador NO</p> <p>Nombre Juan Pedro García</p> <p>Domicilio Avellaneda 89</p> <p>Telefono 2302654319</p> <p>Documento 20698741</p>	<p>Misma persona que el reclamante NO</p> <p>Nombre Enrique Palacios</p> <p>Domicilio Avellaneda 89</p> <p>Telefono 2302968741</p> <p>Documento 36852147</p>

Para regresar al listado de siniestros, presione en “Volver”.

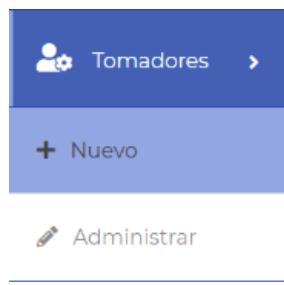
3.3.5 Tomadores



En esta sección podremos llevar a cabo todas las actividades relacionadas a los tomadores de las pólizas del sistema.

Dentro de las opciones disponibles podremos agregar un nuevo tomador, consultar y modificar su información y dar de baja aquellos que sean permitidos.

➔ 3.3.5.1 Agregar nuevo tomador



Primero: Nos dirigimos al menú principal y presionamos en “Tomadores”, una vez desplegado el listado de las acciones, presionar en la opción “Nuevo”, seguido de esto, se mostrará en el contenedor principal el siguiente formulario:

Registrar Nuevo Tomador

Datos Del Nuevo Tomador

Nombre/Razon Social <input type="text" value="Nombre Apellido"/>	Documento/CUIT <input type="text" value="00000000000 (Sin espacios)"/>	Persona <input type="text" value="Seleccione"/>
Nacionalidad/Pais <input type="text" value="Nombre Pais"/>	Nacimiento/Orig <input type="text" value="00/00/0000 (Seleccione)"/>	Provincia/Distrito <input type="text" value="Seleccione"/>
Codigo Postal <input type="text" value="Seleccione Provincia"/>	Localidad <input type="text" value="Seleccione Codigo Postal"/>	Teléfono <input type="text" value="00000000000 (Sin espacios)"/>
Calle <input type="text" value="Nombre/Numero Calle"/>	Género <input type="text" value="Seleccione"/>	Correo <input type="text" value="correo@ejemplo.com (Opcional)"/>

✗ Cancelar Grabar

Segundo: Ingresamos la información en todos los campos y presionamos en “Grabar”, en caso de no haber llenado uno o más campos requeridos, se mostrara un listado de todos aquellos que deben de ser modificados:

Registrar Nuevo Tomador

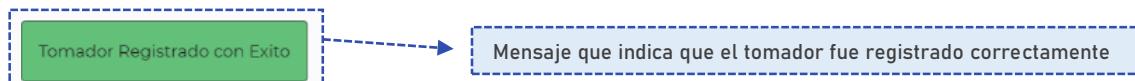
Datos Del Nuevo Tomador

Nombre/Razon Social <input type="text" value="Nombre Apellido"/>	Documento/CUIT <input type="text" value="00000000000 (Sin espacios)"/>	Persona <input type="text" value="Seleccione"/>
Nacionalidad/Pais <input type="text" value="Nombre Pais"/>	Nacimiento/Orig <input type="text" value="00/00/0000 (Seleccione)"/>	Provincia/Distrito <input type="text" value="Seleccione"/>
Codigo Postal <input type="text" value="Seleccione Provincia"/>	Localidad <input type="text" value="Seleccione Codigo Postal"/>	Teléfono <input type="text" value="00000000000 (Sin espacios)"/>
Calle <input type="text" value="Nombre/Numero Calle"/>	Género <input type="text" value="Seleccione"/>	Correo <input type="text" value="correo@ejemplo.com (Opcional)"/>

Debe ingresar el Nombre
Debe ingresar el Documento
Debe ingresar el tipo de Persona
Debe ingresar la Nacionalidad

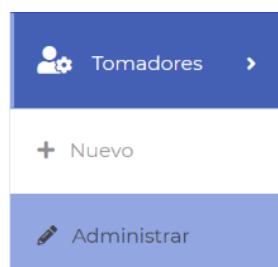
✗ Cancelar Grabar

En caso de no existir errores, se mostrará un mensaje de éxito, y en caso de haberse producido un problema interno, el mensaje será de error.



Para deshacer todos los cambios y no seguir con el proceso de registro, presione en “Cancelar”, seguido de esto se limpiarán todos los campos que hayan sido completado con información.

➔ 3.3.5.2 Administrar tomadores



Primero: Nos dirigimos al menú principal y presionamos en “Tomadores”, una vez desplegado el listado de las acciones, presionar en la opción “Administrar”, seguido de esto, se mostrará en el contenedor principal la siguiente tabla con el listado de todos los tomadores registrados en el sistema:

Campo para buscar un tomador específico.

Administrador Tomadores

Nro Tom	Nombre	Provincia	Localidad	Acciones
32549852	Juan Rodriguez	Catamarca	Los Guaytmas	
85935423	Federico Izquierdoz	La Pampa	Santa Rosa	
13947637	Esteban Clara	La Pampa	Santa Rosa	
98496322	Erasmo Jesus Jordan	La Pampa	Winifreda	
95656746	Bautista Romero	La Pampa	Santa Rosa	

Listado de todos los tomadores registrados.

Opción para modificar la información del tomador. Inst. 3.2.3.5.2.1

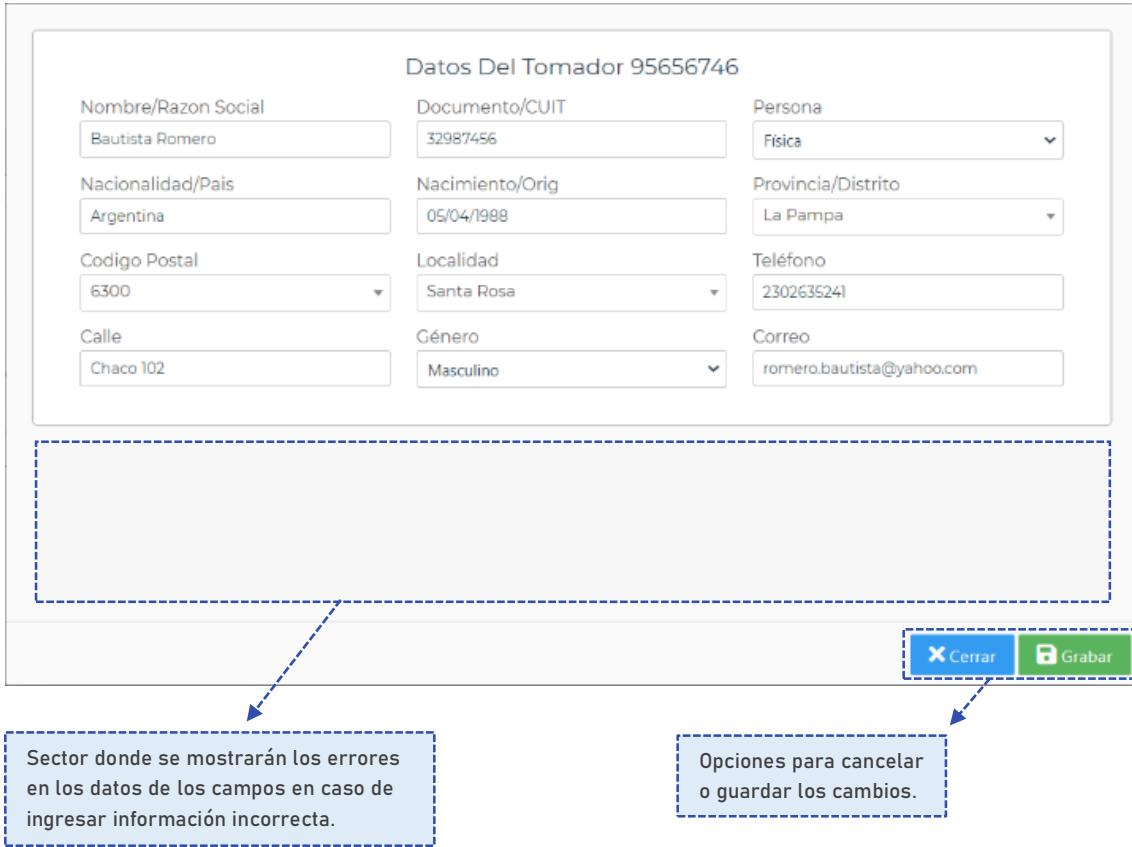
Opción para consultar la información del tomador. Inst. 3.2.3.5.2.2

Opción para dar de baja del sistema a ese tomador. Inst. 3.2.3.5.2.3

Cada tomador es representado por una fila de la tabla, mientras que en las columnas se representa su información general.

* 3.3.5.2.1 Modificar información

Una vez ubicado el tomador al que deseé modificar sus datos, presione en la opción ubicada a la izquierda de la columna “Acciones” (), y seguido de esto se abrirá la siguiente ventana:



Datos Del Tomador 95656746

Nombre/Razon Social Bautista Romero	Documento/CUIT 32987456	Persona Física
Nacionalidad/Pais Argentina	Nacimiento/Orig 05/04/1988	Provincia/Distrito La Pampa
Código Postal 6300	Localidad Santa Rosa	Teléfono 2302635241
Calle Chaco 102	Género Masculino	Correo romero.bautista@yahoo.com

Cerrar **Grabar**

Sector donde se mostrarán los errores en los datos de los campos en caso de ingresar información incorrecta.

Opciones para cancelar o guardar los cambios.

En el formulario, estarán todos los campos completos con la información del tomador, para poder modificarlos, debe reemplazarla por los nuevos datos, en caso de que estos sean incorrectos o no se encuentren permitidos por el sistema, se notificara los errores en el recuadro ubicado en la parte inferior del formulario. Una vez reemplazados los campos deseados presione en “Grabar” para aplicar los cambios, en caso de haber finalizado el proceso correctamente, se mostrará un mensaje de éxito, de lo contrario, el mensaje será de error.

Usaremos como ejemplo el tomador registrado en el apartado anterior, y modificaremos el campo “Documento/CUIT” de “32987456” a “32987459”.

Para cancelar el proceso de modificación, y cerrar la ventana, presionen en “Cancelar”.



* 3.3.5.2.2 Ver información

Una vez ubicado el tomador al que deseé consultar sus datos, presione en la opción ubicada en el centro de la columna “Acciones” (), y seguido de esto se abrirá la siguiente ventana, en la que podrá visualizar la información:

Consultar Datos Tomador #95656746	
Nombre Completo	Documento
Bautista Romero	32987459
Tipo de Persona	Nacionalidad
Física	Argentina
Provincia	Localidad
La Pampa	Santa Rosa
Código Postal	Calle
6300	Chaco 102
Género	Teléfono
Masculino	2302635241
 Cerrar	
Correo	Fecha de Nacimiento
romero.bautista@yahoo.com	05/04/1988

Presione en “Cerrar” para salir de la ventana y volver al listado de tomadores.

* 3.3.5.2.3 Dar de baja del sistema

Una vez ubicado el tomador al que deseé dar de baja, presione en la opción ubicada a la derecha de la columna “Acciones” (), y seguido de esto se mostrara el siguiente mensaje de confirmación:

Eliminar cliente

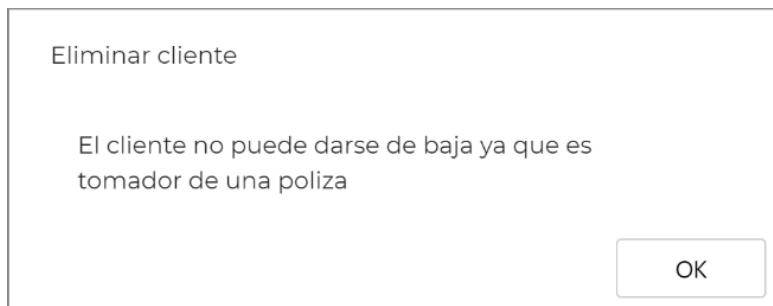
¿Dar de baja el cliente? Se eliminara por completo del sistema, y sus datos no podran recuperarse.

Presione en la opción “Atrás” para cancelar los cambios y regresar al listado de tomadores.

Para dar de baja al tomador, presione en la opción “Si”, pero antes se debe tener en cuenta que, al realizar esta acción, se estaría **eliminando por completo esa persona registrada y toda su información, sin la opción de poder recuperarla**, a no ser que se registre nuevamente en el futuro.

Al momento de presionar en “Si”, pueden presentarse dos situaciones:

Acceso denegado para dar de baja:



Si al intentar dar de baja a un cliente, se presenta este mensaje informativo, se debe a que el sistema no permite que esta persona se elimine, ya

que esta posee un rol de tomador de una o más pólizas, es decir, en caso de llevarse a cabo el proceso generaría un grave error interno. Solo es posible dar de baja aquellas personas que se encuentren excluidas de cualquier cargo o responsabilidad a su nombre, de cualquiera de los componentes del sistema.

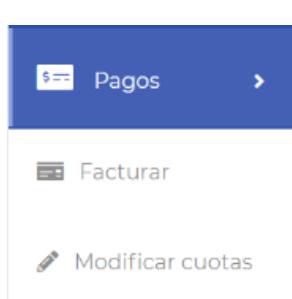
Proceso de eliminación finalizado correctamente:

Dado el caso de que la persona que se intenta dar de baja no forme parte de ningún componente del sistema (Ser tomador de una póliza), el proceso se llevará a cabo sin problemas, y una vez este finalice correctamente, se mostrara un mensaje de éxito, o de lo contrario, si ocurre algún problema interno, se mostrara un mensaje de error.



3.3.6 Pagos

*Se recomienda leer el instructivo 2.2.2 (Patrón de funcionamiento de pagos) para comprender correctamente todo el contenido de este apartado.

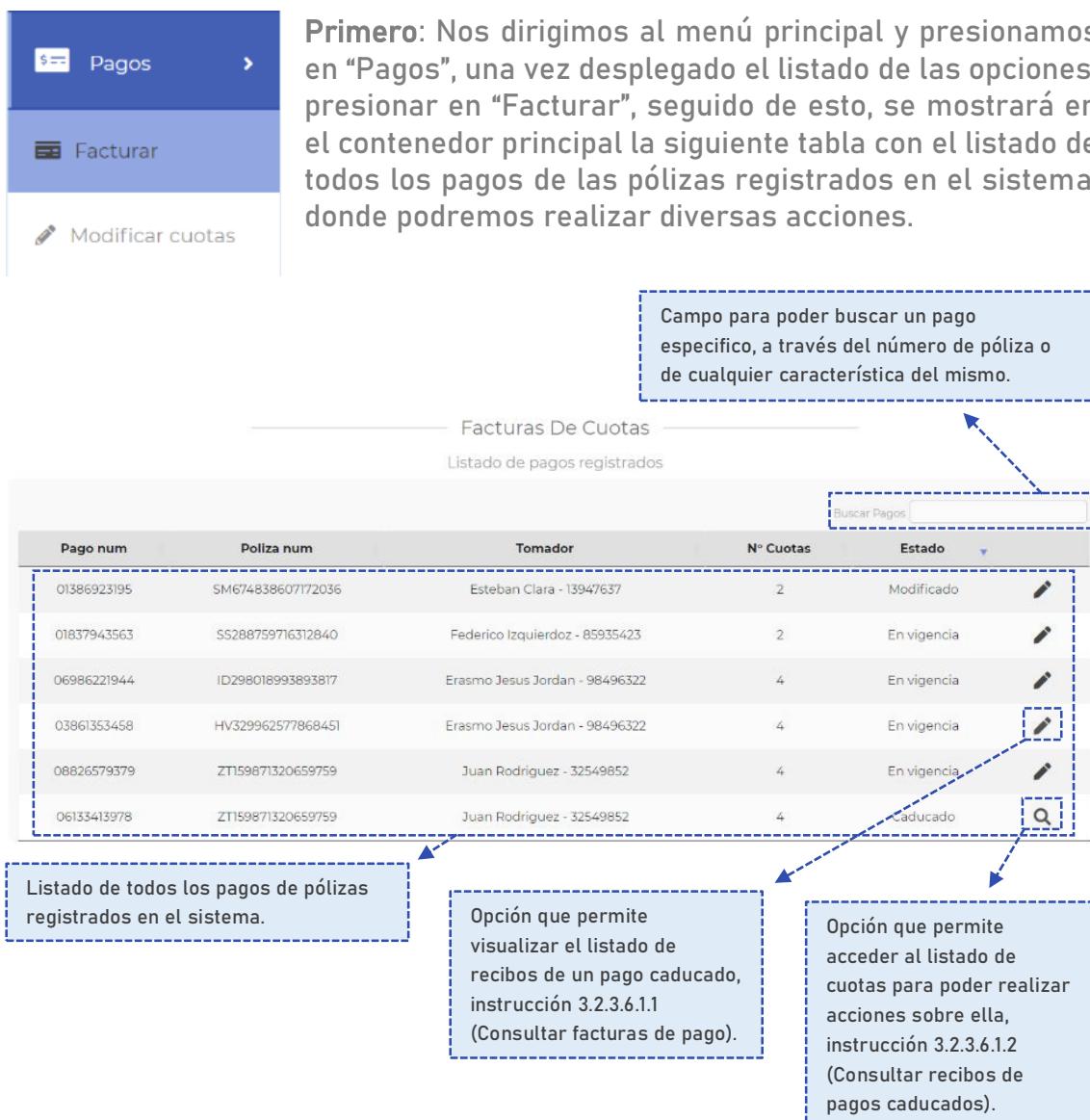


En esta sección podremos llevar a cabo todas las actividades relacionadas a los pagos relacionados a cada una de las pólizas registradas.

Dentro de las opciones disponibles podremos realizar el proceso de abonar las cuotas disponibles, visualizar, enviar e imprimir aquellas ya efectuadas, y también modificar las condiciones de pago ya una vez establecidos.

➔ 3.3.6.1 Facturar

Primero: Nos dirigimos al menú principal y presionamos en “Pagos”, una vez desplegado el listado de las opciones, presionar en “Facturar”, seguido de esto, se mostrará en el contenedor principal la siguiente tabla con el listado de todos los pagos de las pólizas registrados en el sistema, donde podremos realizar diversas acciones.



Pago num	Poliza num	Tomador	Nº Cuotas	Estado
01386923195	SM674838607172036	Esteban Clara - 13947637	2	Modificado
01837943563	SS288759716312840	Federico Izquierdoz - 85935423	2	En vigencia
06986221944	ID298018993893817	Erasmo Jesus Jordan - 98496322	4	En vigencia
03861353458	HV329962577868451	Erasmo Jesus Jordan - 98496322	4	En vigencia
08826579379	ZT159871320659759	Juan Rodriguez - 32549852	4	En vigencia
06133413978	ZT159871320659759	Juan Rodriguez - 32549852	4	Caducado

Cada pago relacionado a una póliza se verá representado en una fila, mientras que, en las columnas de cada una, se podrá visualizar la principal información del mismo.

Pago númer.: Es el numero único que asigna el sistema al pago que se asigna a una póliza al ser registrada.

Póliza númer.: Es el numero único de póliza a la que se relaciona el pago.

Tomador: Nombre y número del tomador de la póliza a la que se relaciona el pago.

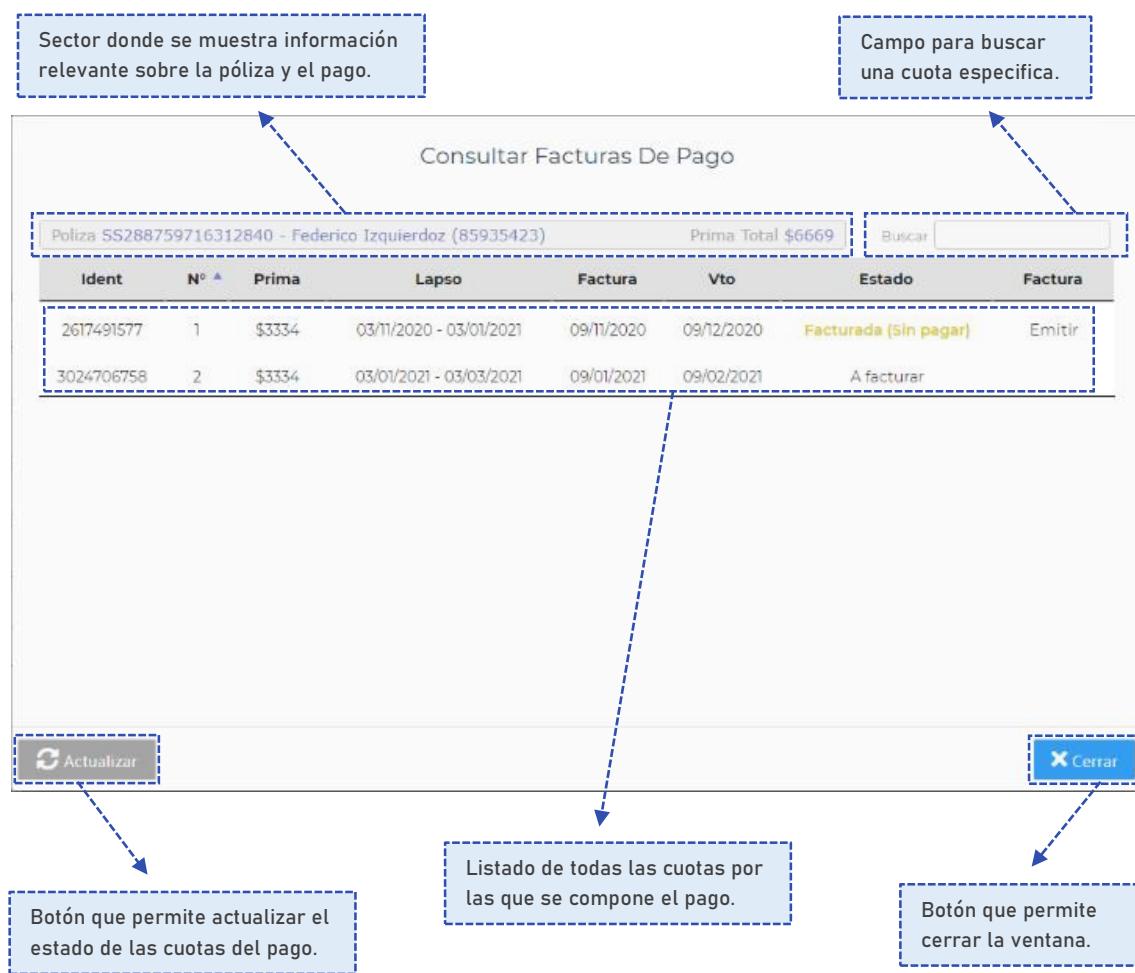
Nº Cuotas: Cantidad de cuotas por las que se compone el pago.

Estado: Es el estado actual en el que se encuentra el pago, pueden existir tres tipos, cuando se encuentra “Vigente” significa que la póliza a la que se relacione se encuentra actualmente en vigencia o en espera; si es “Modificado” significa

que las condiciones de ese pago fueron alteradas en algún momento; y si se encuentra “Caducado”, quiere decir que la póliza a la que se relaciona ese pago, se encuentra vencida, anulada o renovada, por lo que ya no podrá ser modificado y debería de tener todas sus cuotas facturadas y con dicho monto abonado (en estado “Pago aprobado”).

* 3.3.6.1.1 Consultar facturas de pago

Una vez ubicado el pago de la póliza deseado, presione en la opción ubicada al final de la fila (), y seguido de esto se abrirá la siguiente ventana:



Para volver al listado de pagos de pólizas y salir de la ventana actual, presione en “Cerrar”.

Para poder actualizar en tiempo real el estado de todas las cuotas de los pagos registrados en el sistema, presione en la opción “Actualizar”, y después de aproximadamente dos segundos, las cuotas actualizaran su ultimo estado.

Aquí se mostrarán todas las cuotas por las que se compone el pago de la póliza, cada una es representada por una fila de la tabla, mientras que en cada columna se aprecia la información general:

Ident.: El identificador es el numero único asignado a cada cuota al momento de establecer las condiciones del pago de la póliza al ser registrada.

Nº: Es el orden en relación a la fecha de facturación en el que se muestran las cuotas, apareciendo en primer lugar aquellas que se facturen antes.

Prima: Es el monto de la prima mensual que representa esa cuota.

Lapso: Es Intervalo de tiempo que cubre la facturación de esa cuota.

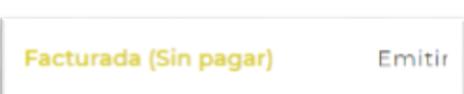
Factura: Es el día en que se genera el recibo de esa cuota, y por ende a partir de esa fecha estará disponible para abonar el pago.

Vto.: El vencimiento es la fecha en que esa cuota pasa a estar en deuda.

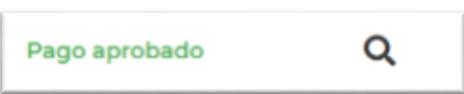
Estado: Es el actual estado en el que se encuentra la cuota, la cual puede presentarse en cinco tipos diferentes:



A facturar: Cuando aún no se alcanzó la fecha de inicio del lapso de facturación, no habrá ninguna acción disponible para realizar en la cuota al tener este estado.



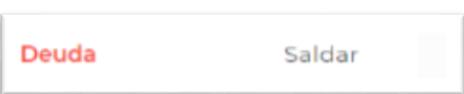
podrá efectuar el pago desde la opción "Emitir".



Pago aprobado: Cuando el pago ya fue abonado. En este estado ya podremos visualizar el recibo de la cuota desde la opción



Pago rechazado: Cuando ocurrió un problema con el método de pago utilizado (Cuando se asocia una tarjeta de crédito), este puede ser saldado posteriormente en efectivo a través de la opción "Aprobar".



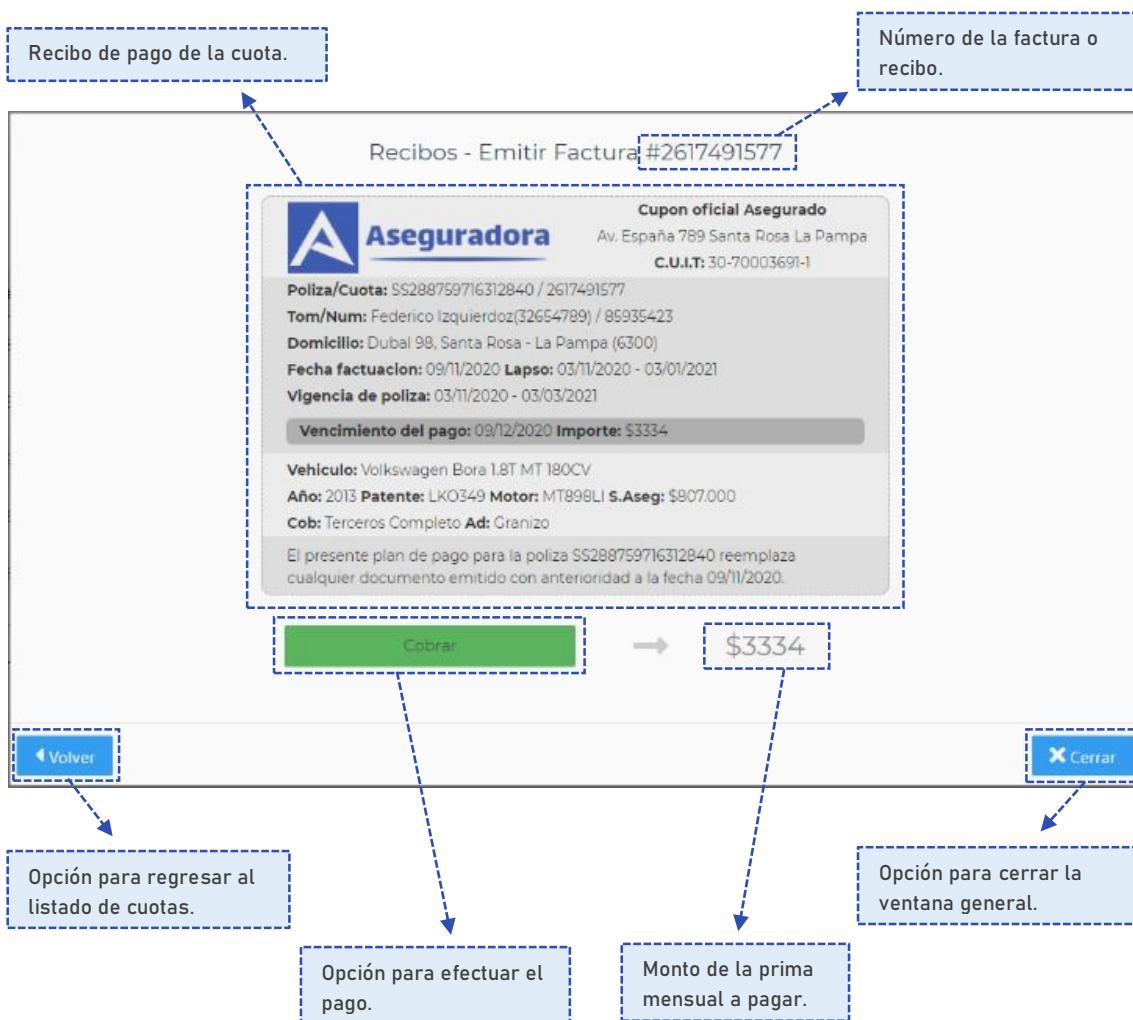
Deuda: Cuando se superó la fecha de fin del lapso de facturación sin aun efectuar el pago. Una vez alcanzado este estado en una cuota, provoca que la póliza se encuentre en deuda, impidiendo realizar algunas actividades cruciales con

esta, como puede ser emitir un nuevo siniestro. Podremos emitir el recibo y aprobar el pago en cualquier momento desde la opción “Saldar”.

> 3.3.6.1.1.1 Emitir recibo

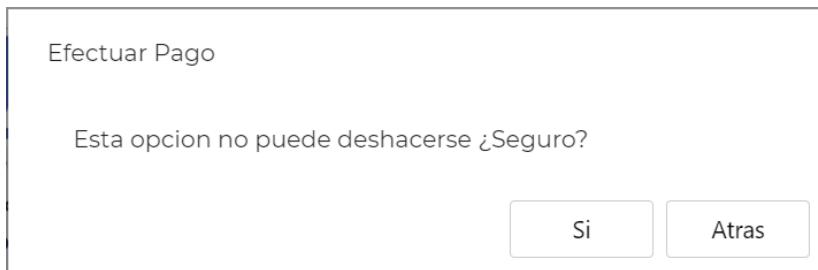
El proceso de efectuar el pago es exactamente el mismo tanto para las distintas opciones en las que se puede presentar una cuota dependiendo de su estado, denominadas “Emitir” cuando la cuota se encuentra “Facturada (Sin pagar)”, “Aprobar” cuando la cuota se encuentra con “Pago rechazado” y “Saldar” cuando el estado es en “Deuda”.

Primero: Una vez haya ubicado la cuota que desee, siempre y cuando su correspondiente pago no haya sido efectuado aun, debe presionar en la opción que se ubica al final de la fila sobre la columna “Factura” (**Emitir**), (**Saldar**), (**Aprobar**) y seguido de esto se mostrara la siguiente interfaz:



Presione en la opción “Volver” si desea regresar al listado de cuotas.

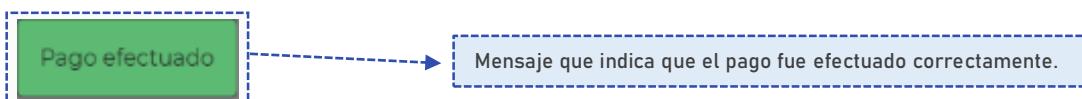
Segundo: Una vez se muestre esta interfaz, presione en la opción “Cobrar”, seguido de esto se mostrará el siguiente mensaje de confirmación:



Presione “Atrás” si desea regresar a la anterior interfaz.

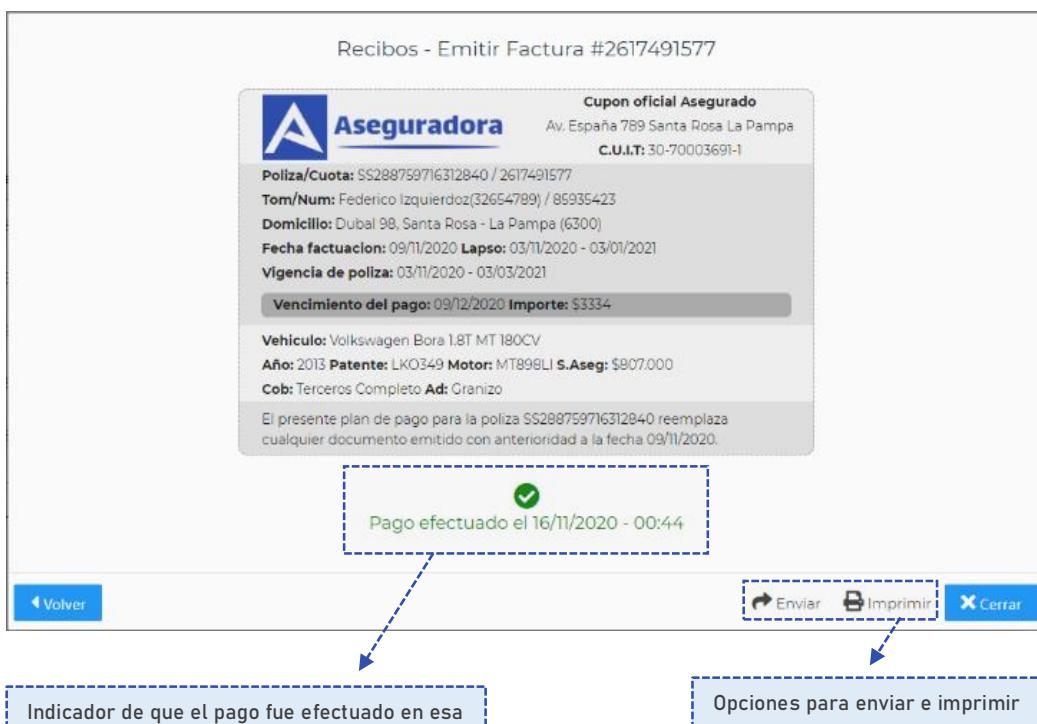
Presione en la opción “Si” para efectuar el pago, una vez realizado este proceso ya no podrá ser revertido. En caso de haber finalizado la operación correctamente, se mostrará un mensaje de éxito, en caso contrario el mensaje será de error.

Una vez se haya efectuado el pago, en lugar de la opción “Cobrar”, quedara establecida el día, mes, año, hora y minuto en que se realizó el proceso, además de que se habilitaran las opciones para enviar (Instrucción 3.2.3.6.1.1.3) e imprimir (Instrucción 3.2.3.6.1.1.4) el recibo de pago.



> 3.3.6.1.1.2 Ver recibo

Una vez ubicada la cuota cuyo pago se encuentre efectuado (Estado “Pago aprobado”), presione en la opción ubicada al final de la fila sobre la columna “Factura” (), seguido de esto, se mostrara la siguiente interfaz:



Indicador de que el pago fue efectuado en esa fecha y horario.

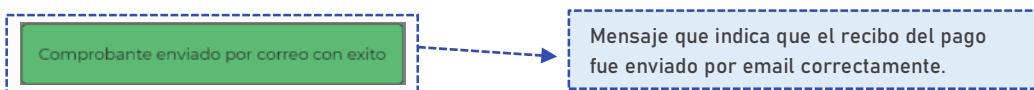
Opciones para enviar e imprimir el recibo habilitadas.

> 3.3.6.1.1.3 Enviar recibo

Primero: Una vez se encuentre en la ventana para visualizar el recibo de la cuota (Instrucción 3.2.3.6.1.1.2), presione en la opción para enviarlo por correo electrónico (), seguido de esto, se abrirá una ventana la cual contendrá un campo de texto con el email del tomador, el cual el sistema traerá automáticamente, pero puede ser modificado por el usuario.



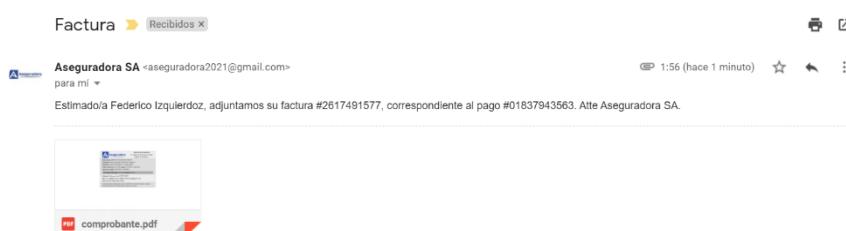
Segundo: Una vez se el campo se encuentre completo con el email del tomador, presione en la opción “Enviar”, seguido el formulario comenzara a procesar él envío, y una vez finalice correctamente, se mostrará un mensaje de éxito, en caso contrario, se mostrará un mensaje de error.



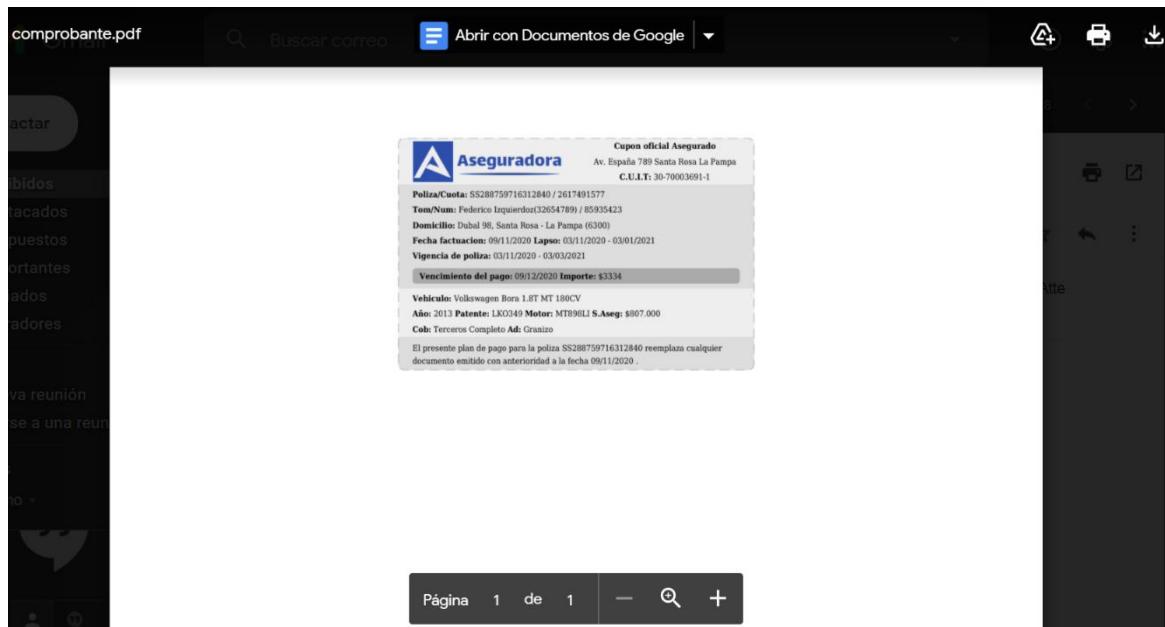
Una vez finalice el proceso de envío, se podrá visualizar el siguiente email en la bandeja de entrada del tomador:



El correo llegara con el asunto “Factura” de la dirección “aseguradora2021@gmail.com”.

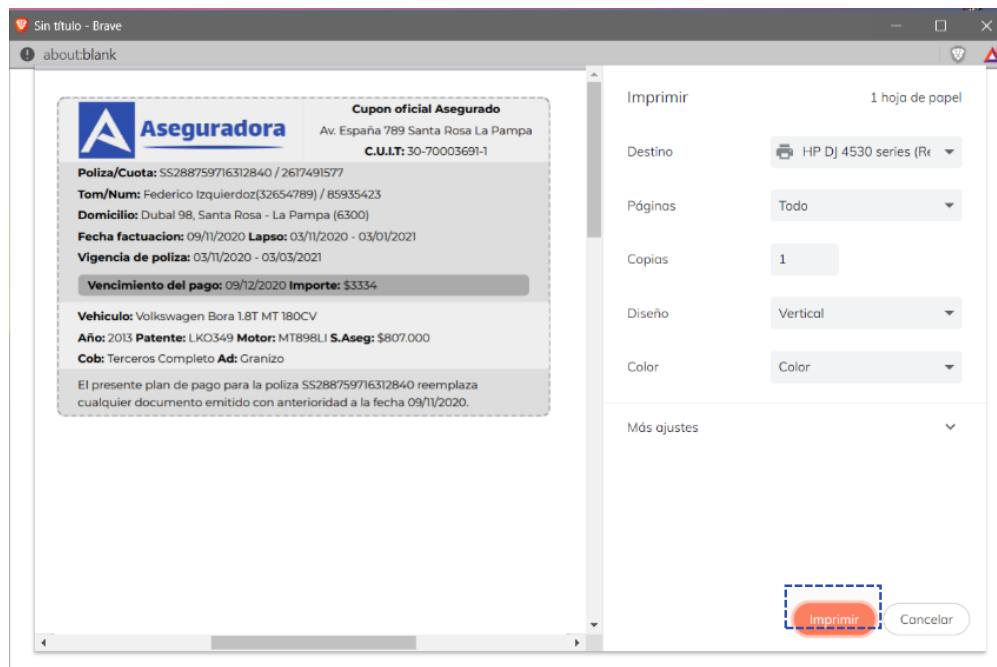


El archivo enviado será en formato PDF, con lo que podrá descargarse e imprimirse si así lo desea.



> 3.3.6.1.1.4 Imprimir recibo

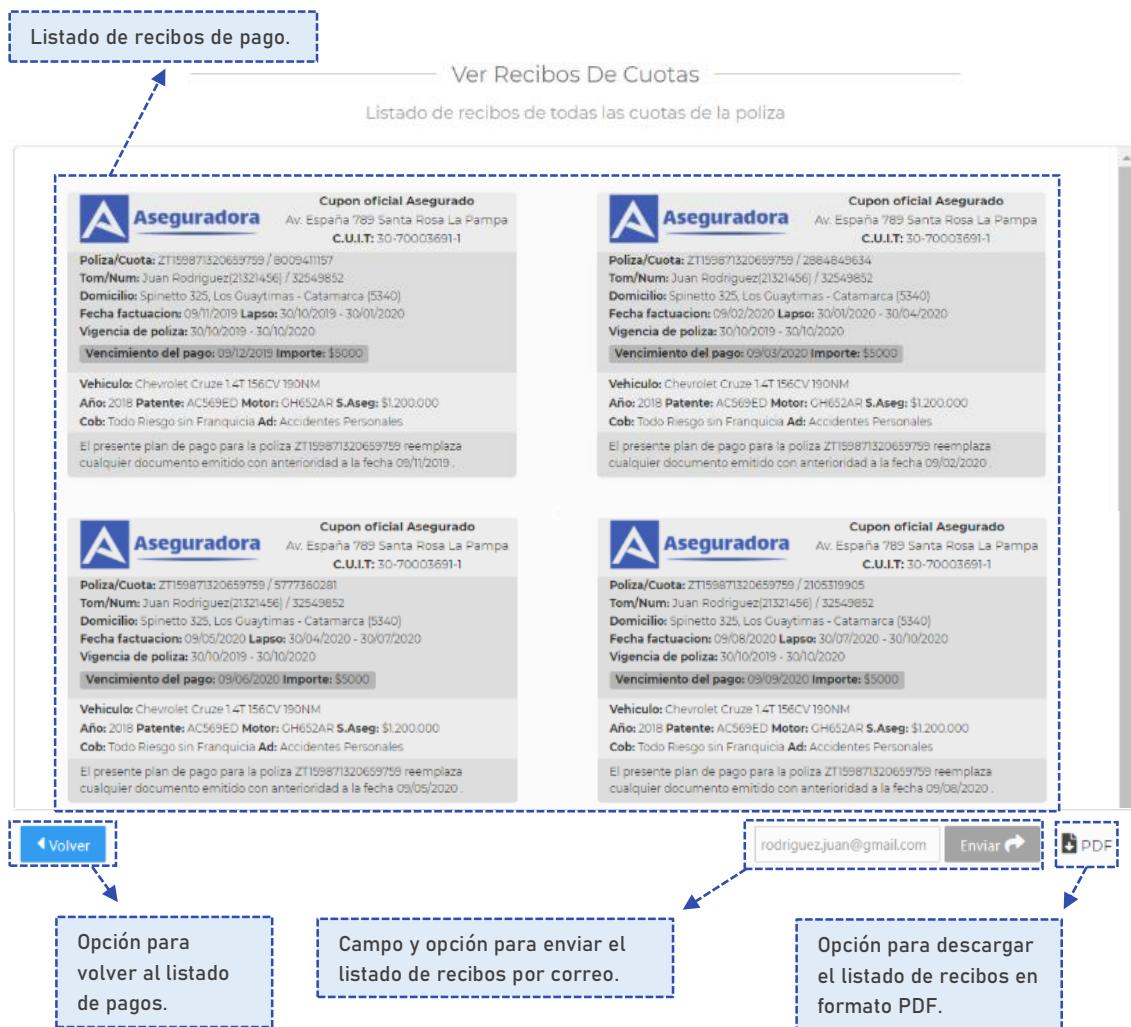
Primero: Una vez se encuentre en la ventana para visualizar el recibo de la cuota (Instrucción 3.2.3.6.1.1.2), presione en la opción para imprimirlo (), seguido de esto, se abrirá una ventana la cual contendrá las herramientas para poder llevar a cabo el proceso, la interfaz puede variar dependiendo del navegador en el que se encuentre operando, en el ejemplo se está utilizando Brave Browser, pero la navegabilidad suele ser similar en el resto.



Una vez se encuentre en la ventana de impresión, presione en la opción "Imprimir". El archivo se imprimirá en tamaño de 10cmx7cm.

* 3.3.6.1.2 Consultar recibos de pagos caducados

Una vez se encuentre en el listado de todos los pagos de pólizas del sistema (Instrucción 3.2.3.6.1 Facturar) y haya ubicado el pago deseado, cuyo estado sea “Caducado”, presione en la opción de consultar pagos caducados ubicada a la derecha de la fila (), y seguido de esto se mostrara la siguiente información:



The screenshot shows a list of four payment receipts for a policy. Each receipt contains the following information:

Aseguradora		Cupon oficial Asegurado
Poliza/Cuota:	ZT159871320659759 / 8009411197	Av: España 789 Santa Rosa La Pampa
Tom/Num:	Juan Rodriguez[21321456] / 32549852	C.U.I.T: 30-70003691-1
Domicilio:	Spinetto 325, Los Guaytomas - Catamarca (5340)	
Fecha facturacion:	09/11/2019	Lapso: 30/10/2019 - 30/10/2020
Vigencia de poliza:	30/10/2019 - 30/10/2020	
Vencimiento del pago:	09/12/2019 Importe: \$5000	
Vehiculo:	Chevrolet Cruze 1.4T 156CV 190NM	
Año:	2018 Patente: AC569ED Motor: GH652AR S.Aseg: \$1.200.000	
Cob:	Todo Riesgo sin Franquicia Ad: Accidentes Personales	
El presente plan de pago para la poliza ZT159871320659759 reemplaza cualquier documento emitido con anterioridad a la fecha 09/11/2019.		

Below the table, there are four identical sections for each receipt, each with a different date and time range:

Aseguradora		Cupon oficial Asegurado
Poliza/Cuota:	ZT159871320659759 / 2984849634	Av: España 789 Santa Rosa La Pampa
Tom/Num:	Juan Rodriguez[21321456] / 32549852	C.U.I.T: 30-70003691-1
Domicilio:	Spinetto 325, Los Guaytomas - Catamarca (5340)	
Fecha facturacion:	09/02/2020	Lapso: 30/01/2020 - 30/04/2020
Vigencia de poliza:	30/10/2019 - 30/10/2020	
Vencimiento del pago:	09/03/2020 Importe: \$5000	
Vehiculo:	Chevrolet Cruze 1.4T 156CV 190NM	
Año:	2018 Patente: AC569ED Motor: GH652AR S.Aseg: \$1.200.000	
Cob:	Todo Riesgo sin Franquicia Ad: Accidentes Personales	
El presente plan de pago para la poliza ZT159871320659759 reemplaza cualquier documento emitido con anterioridad a la fecha 09/02/2020.		

At the bottom of the screen, there are several buttons:

- Volver (Back)
- Enviar (Send) with an envelope icon
- PDF (PDF) with a PDF icon
- Opción para volver al listado de pagos (Option to return to the list of payments)
- Campo y opción para enviar el listado de recibos por correo. (Field and option to send the list of receipts by email)
- Opción para descargar el listado de recibos en formato PDF. (Option to download the list of receipts in PDF format)

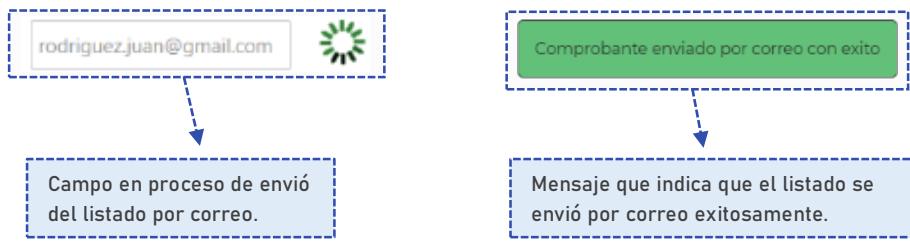
Para regresar al listado de pagos, presione en “Volver”.

En este ejemplo, se visualizan solo cuatro recibos, ya que, en las condiciones de pago de la póliza, se determinó que la prima total seria abonada en cuatro cuotas.

> 3.3.6.1.2.1 Enviar listado

Primero: Ingrese el email de tomador, o deje el ya traído por el sistema en el campo de texto ubicado en la parte inferior derecha de la pantalla.

Segundo: Presione en la opción “Enviar” y espere a que termine el proceso, y en caso de finalizar correctamente, se mostrara un mensaje de éxito.

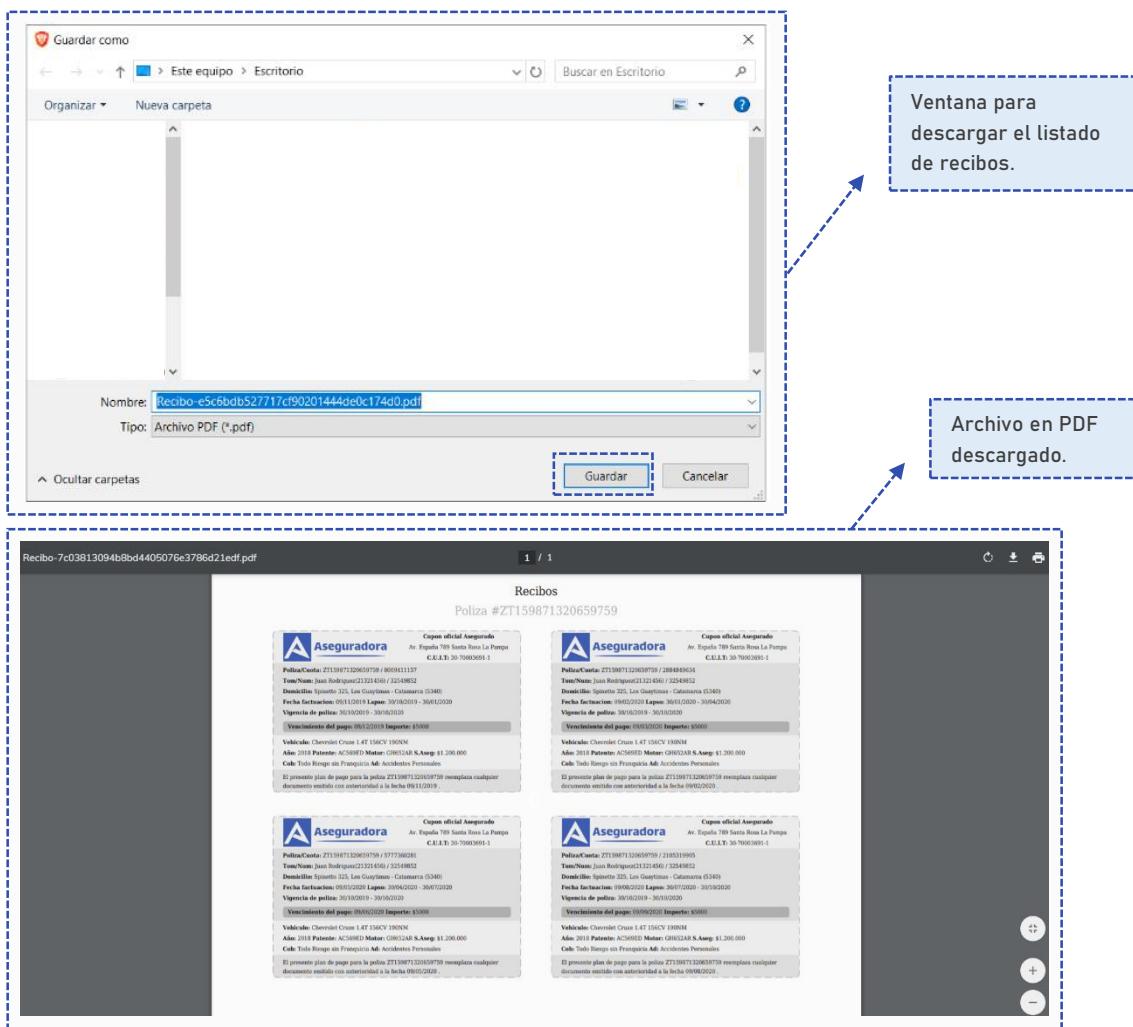


Una vez enviado, en la bandeja de entrada del correo del tomador, se mostrará un mensaje de la dirección “aseguradora2021@gmail.com” con el asunto “Listado de comprobantes de pago”, a nombre de “Aseguradora SA”.

El archivo enviado estará en formato PDF, por lo que podrá ser descargado o imprimido por el receptor, y tendrá el nombre de “Recibos.pdf”.

› 3.3.6.1.2.2 Descargar listado

Para poder descargar el listado de recibos de pago de una póliza, una vez estando en la ventana consultar recibos de pagos caducados (Instrucción 3.2.3.6.2), presione en la opción para descargar en PDF ubicada en la parte inferior derecha de la pantalla (), y seguido de esto se abrirá una ventana donde deberá seleccionar el directorio en donde deseé guardar el archivo, y presione en “Guardar”, o dependiendo la configuración, comenzara directamente la descarga.



➔ 3.3.6.2 Modificar cuotas



Primero: Nos dirigimos al menú principal y presionamos en “Pagos”, una vez desplegado el listado de las opciones, presionar en “Modificar cuotas”, seguido de esto, se mostrará en el contenedor principal la siguiente tabla con el listado de todos los pagos de las pólizas registrados en el sistema:

Establecer Cuotas

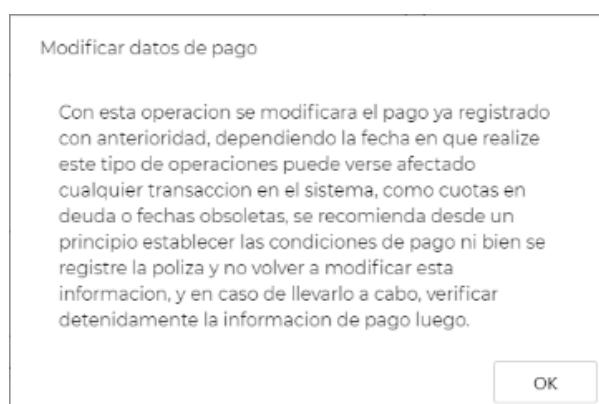
Listado de polizas las cuales se han configurado o no los parametros de pago.

Nro	Tomador	Vehiculo	Inicio Vig	Fin Vig	Registro	Cuotas
HV329962577868451	Erasmo Jesus Jordan - 98496322	Volkswagen Bora	19/11/2020	19/03/2021	09/11/2020	
ZT159871320659759	Juan Rodriguez - 32549852	Chevrolet Cruze	10/11/2020	10/03/2021	09/11/2020	
ID298018993893817	Erasmo Jesus Jordan - 98496322	Abarth Punto Evo	06/11/2020	06/03/2021	06/11/2020	
SM674838607172036	Esteban Clara - 13947637	Dacia Logan	18/11/2020	18/11/2021	06/11/2020	
SS288759716312840	Federico Izquierdoz - 85935423	Volkswagen Bora	03/11/2020	03/03/2021	02/11/2020	

Opción para acceder a la ventana para modificar las condiciones de pago.

En la tabla se puede ver el listado de todas las pólizas, cada una es representada en una fila, y en las columnas se detallan sus datos.

Segundo: Una vez ubicada la póliza la cual desea modificar el pago, presione en la opción de modificación ubicada al final de la fila sobre la columna “Cuotas” () y seguido de esto se abrirá un mensaje de notificación:



Este aviso se genera, ya que el sistema brinda la opción de poder realizar modificaciones en las condiciones de pago una vez ya registrada la póliza, pero no lo recomienda ya que, tal como dice el aviso, pueden verse afectadas dichas condiciones al existir cuotas ya efectuadas, pudiendo producir perdidas tanto a la compañía como al cliente.

¿Cuándo si es recomendable realizar esta acción?

Se recomienda modificar las condiciones en aquellos pagos cuyas pólizas aún no se encuentren vigentes, es decir con el estado “En espera”, ya que, en esta etapa, ninguna de las cuotas generadas se facturará. Sin embargo, el anterior aviso se mostrará en pagos cuyas pólizas se encuentren en cualquier estado.

Presione en “Ok” para cerrar el mensaje de notificación, y una vez hecho esto se mostrará la siguiente ventana:

Información a tener en cuenta para asignar las nuevas condiciones de pago.

Modificar Condiciones De Pago

Poliza
Número SS288759716312840
Emitida el 02/11/2020 a las 15:24

Vigencia
Desde 03/11/2020 11:59 hasta 03/03/2021 11:59 (4 mes/es)

Vehículo
Volkswagen Bora 1.8T MT 180CV - Winifreda La Pampa - 6313
Año: 2013
Es cero km: No
Kilómetros: 98.300
Equipo de gas No

Coberturas
Principal Terceros Completo TIPO C1
Adicional Granizo
Suma Asegurada \$807.000

Prima total \$	Cant cuotas	Prima mensual \$	Día a facturar
6669	2	3334	9

Ver cuotas

X Cerrar **Modificar**

Formulario para asignar las nuevas condiciones de pago.

Sector donde se mostrarán los errores en caso de ingresar información incorrecta en los campos.

Opción para visualizar el estado de las cuotas de ese pago.

Los campos admitirán exactamente los mismos datos tal cual se describen en la instrucción 3.2.3.2.1.5 (Sección pago) al emitir una nueva póliza.

Para cancelar los cambios y regresar a la ventana donde se muestran el listado de pólizas, presione en la opción “Cerrar”.

En caso de ser necesario, podrá visualizar el listado de cuotas las cuales componen al pago y todos sus detalles, presionando en la opción “Ver cuotas”

Ubicada en la zona derecha del formulario (), y seguido de esto se abrirá la siguiente ventana:

Listado De Cuotas						
Poliza: SS288759716312840						
Cuota N°	Iden Cuota	Prima	Lapso	Factura	Vto Pago	Estado
2 / 2	3024706758	\$3334	03/01/2021 - 03/03/2021	09/01/2021	09/02/2021	A facturar
1 / 2	267491577	\$3334	03/11/2020 - 03/01/2021	09/11/2020	09/12/2020	Pago aprobado

Aquí se podrá observar toda la información relacionadas a cada una de las cuotas del pago de esa póliza.

En el ejemplo, se pueden ver solo dos filas, ya que es la cantidad de cuotas por la que se compone el pago.

Presione en “Cerrar” para regresar al formulario de modificación de las condiciones de pago.

En el ejemplo, se modificaron los parámetros del pago “01837943563” correspondiente a la póliza número “SS288759716312840”, realizando las siguientes modificaciones:

Prima total: De “6669” a “6969”.

Cant. cuotas: De “2” a “4” cuotas, por lo que la prima mensual pasara de “3334” a “1742”.

Día a facturar: De “9” al “5” de cada mes.

Tercero: Una vez haya reemplazado la información de los campos deseados con los nuevos datos, presione en la opción “Modificar”, seguido de esto se mostrará un mensaje informativo, donde se detallarán todos los nuevos parámetros establecidos.

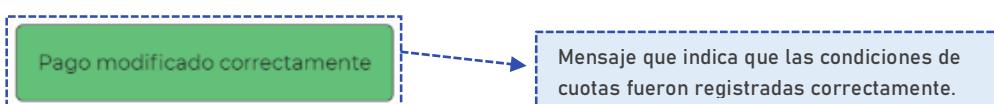
Modificar informacion de pago

Valor de la prima total de \$6969 que se realizará en 4 pagos de \$1742.
El pago de la prima mensual se facturará cada 1 mes/es, en el día 5 del mes que así corresponda, durante un plazo de 4 mes/es de vigencia.

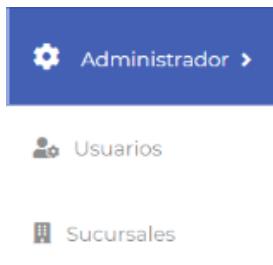
Una vez estando de acuerdo, presione en la opción “Registrar” para confirmar el proceso de modificación de pago.

Presione en la opción “Atrás” para regresar a la ventana de modificación.

Se mostrará un mensaje de éxito en caso de haber finalizado el proceso correctamente, de lo contrario el mensaje será de error.



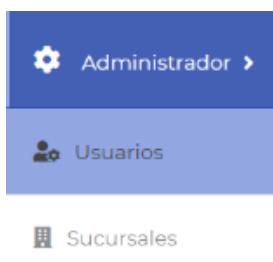
3.3.7 Administrador



En esta sección podremos llevar a cabo todas las actividades que se permiten hacer a aquellos usuarios cuyo rol en el sistema sea de administrador.

Dentro de las opciones disponibles podremos agregar, dar de baja y solicitar información de usuarios, como así modificar el rol de cada uno, también se podrá agregar, eliminar, modificar y consultar información de sucursales.

3.3.7.1 Gestionar usuarios



Vaya al menú principal y presione en “Administrador”, y una vez desplegada todas las acciones disponibles, pulse en “Usuarios”, seguido de esto, se mostrará en el contenedor general una tabla donde se listarán todos los usuarios registrados en el sistema.

The screenshot shows the 'Administrador' interface with the 'Usuarios' section selected. At the top, there are buttons for 'Nuevo' (New) and 'Bajas' (Deletions). To the right is a search bar labeled 'Buscar usuario'. The main area is a table titled 'Administrar Usuarios' with columns: 'Número', 'Nombre', 'Rol', and 'Estado'. Two rows of data are visible:

Número	Nombre	Rol	Estado
1736074609	Erasmo Jordan Usuario	Usuario	Activo
9608024074	Administrador De Ejemplo	Administrador	Activo

Annotations with dashed boxes and arrows point to specific parts of the interface:

- A box points to the 'Nuevo' and 'Bajas' buttons with the text: "Sector donde se mostrará el listado de opciones de todas las acciones a realizar sobre un usuario."
- A box points to the search bar with the text: "Campo para buscar un usuario específico."
- A box points to the table area with the text: "Sector donde se muestra el listado de usuarios registrados."

En esta sección podremos gestionar aquellos usuarios los cuales son o serán responsables de llevar a cabo todas las acciones dentro del sistema.

Cada uno se representa en una fila de la tabla, mientras que en las columnas se detalla la principal información.

Comenzando de izquierda a derecha:

- ✓ : Al presionar sobre el campo de una fila, correspondiente a la primera columna de la tabla, se seleccionará la fila y seguido de esto se mostrará el menú de opciones de todas las acciones disponibles para realizar sobre un usuario:

The screenshot shows a table titled "Administrar Usuarios" with columns: Numero, Nombre, Rol, and Estado. A user row is selected, indicated by a blue border around the first column cell. A dashed blue box labeled "Listado de acciones disponibles." covers the top navigation bar. Another dashed blue box labeled "Usuario seleccionado." covers the first column of the table. Three action buttons are visible at the top left: "Baja" (red), "Rol" (yellow), and "Info" (blue).

Al pulsar nuevamente sobre la opción, tanto la fila como el listado de acciones volverían a su estado anterior, y al presionar en otro usuario, este será quien quede seleccionado en lugar del anterior.

Numero: Número único asignado por el sistema al registrar un nuevo usuario.

Nombre: Nombre completo del usuario.

Rol: Rol que actualmente asume ese usuario.

Estado: Actual estado en el que se encuentra ese usuario. Existen dos tipos de estados, el primero es “Activo”, que se aplica cuando el usuario administrador permite que la persona ingrese al sistema, y el segundo es “Baja”, que corresponde a aquellos usuarios los cuales el administrador dio de baja en algún momento, provocando que se le prohíba el acceso al sistema temporalmente hasta que vuelva a ser dado de alta.

* 3.3.7.1 Nuevo usuario

Primero: Una vez nos encontramos en la pantalla de gestión de usuarios (Instrucción 3.2.3.7.1 Gestionar usuarios), aségrese que ninguna fila se encuentre seleccionada y presione en la opción “Nuevo” ubicada en la parte superior izquierda de la tabla (), y seguido de esto se mostrara el siguiente formulario:

The screenshot shows a form titled "Datos Del Nuevo Usuario" with fields for: Nombre completo (with placeholder "Nombre Apellido"), Documento (with placeholder "00000000000000000000 (Sin espacios)"), Nacionalidad (with placeholder "Argentina"), Provincia/Distrito (dropdown menu "Seleccionne"), Localidad (dropdown menu "Seleccionne"), Dirección (placeholder "Calle - Número"), Fecha nacimiento (placeholder "00/00/0000"), Teléfono (placeholder "00000000000000000000 (Sin espacios)"), E-Mail (placeholder "correo@ejemplo.com"), Género (dropdown menu "Seleccionne"), and Rol (dropdown menu "Seleccionne"). At the bottom are "Volver" and "Registrar" buttons.

Para regresar al listado de usuarios presione en “Volver”.

Segundo: Debe completar todos los campos del formulario con los datos personales de la persona para poder terminar el proceso de registro, siendo todos requeridos obligatoriamente:

Nombre completo: Nombre y apellido de la persona.

Documento: Numero de documento de la persona.

Nacionalidad: País de nacimiento de la persona.

Provincia: Provincia en la que reside la persona.

Localidad: Localidad en la que reside la persona.

Dirección: Dirección del domicilio de la persona, preferentemente que sea la denominación de la calle y la altura o número de vivienda.

Fecha de nacimiento: Fecha de nacimiento de la persona.

Teléfono: Número de teléfono móvil de la persona.

Email: Dirección de correo electrónico personal, este campo es de suma importancia para que la persona pueda acceder al sistema más adelante.

Género: Genero de la persona.

Rol: El tipo de rol que asumirá esa persona en el sistema.

En caso de intentar registrar el usuario sin completar alguno de los anteriores datos, se mostrará un listado de errores en la parte inferior del formulario haciendo mención de los campos cuya información no es correcta.

————— Administrar Usuarios ————

👤 Datos Del Nuevo Usuario

Nombre completo	Documento	Nacionalidad
<input type="text" value="Nombre Apellido"/>	<input type="text" value="00000000000 (Sin espacios)"/>	<input type="text" value="Argentina"/>
Provincia/Distrito	Localidad	Direccion
<input type="text" value="Seleccione"/>	<input type="text"/>	<input type="text" value="Calle - Numero"/>
Fecha nacimiento	Teléfono	E-Mail
<input type="text" value="00/00/0000"/>	<input type="text" value="00000000000 (Sin espacios)"/>	<input type="text" value="correo@ejemplo.com"/>
Género	Rol	
<input type="text" value="Seleccione"/>	<input type="text" value="Seleccione"/>	

[◀ Volver](#) **💾 Registrar**

Debe seleccionar una Localidad

Debe ingresar una Direccion

Debe seleccionar una Fecha de Nacimiento

Debe ingresar un Telefono

Debe ingresar una Direccion de E-Mail

Listado de errores cometidos en los datos de los campos.

Registro de usuarios

La información para el acceso al sistema fueron enviados por e-mail al usuario.

En caso de no haber recibido la información, debe de ponerse en contacto con el administrador.

OK

Tercero: Una vez haya completado todos los campos con la información personal de la persona, presione en la opción “Registrar”, y seguido de esto se mostrará un mensaje informativo.

Presione en “OK” para cerrarlo.

En caso de haber finalizado el proceso correctamente se mostrará un mensaje de éxito, de lo contrario se mostrará un mensaje de error.

Usuario registrado exitosamente

Mensaje que indica que el usuario ha sido registrado correctamente.

¡Importante!

Al finalizar este proceso, el sistema genera un nombre de usuario y una contraseña con números al azar, la cual será enviada a la dirección de correo electrónico que se ingrese en el formulario de registro de usuario, **sin esta información, la persona no podrá acceder al sistema**, por lo que, en caso de perderla, debe de contactarse con el administrador, para que este le reenvíe los datos nuevamente (Instrucción 3.2.3.7.1.5 Reenviar datos de acceso a usuarios).

En la bandeja de entrada del correo personal del usuario debe de llegar un mensaje desde la dirección “aseguradora2021@gmail.com”, con el nombre de “Aseguradora SA” y el asunto “Información de acceso”:



Con la siguiente estructura e información:

Datos para que el usuario pueda acceder al sistema.

Información de acceso Recibidos X

Aseguradora SA <aseguradora2021@gmail.com>
para mí ▾

Estimado/a Carlos Piniera, a continuación se detallara la información necesaria para que pueda acceder al sistema de Aseguradora SA

USUARIO: 38972-27741258
CONTRASEÑA: 1997242982

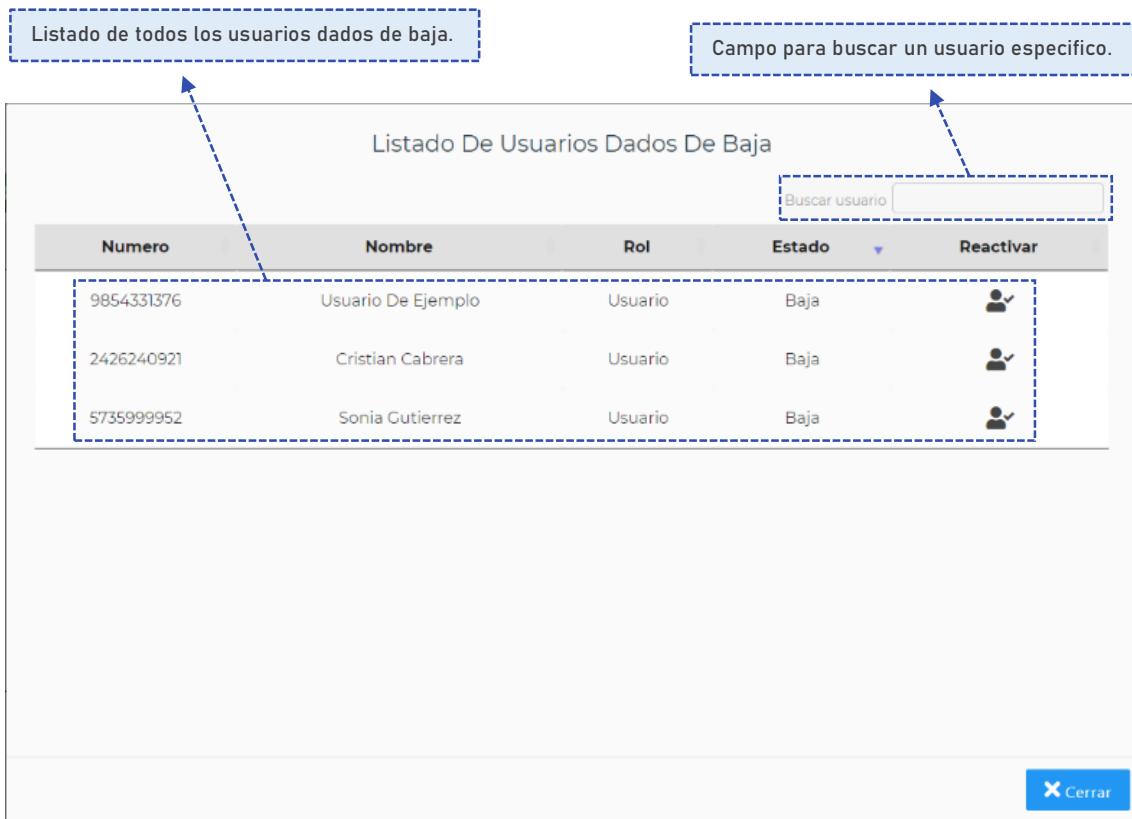
Recomendamos:
1) Bajo ninguna circunstancia comparta con nadie la anterior información.
2) Ni bien complete el inicio de sesión, modifique la mencionada información para así preservar la seguridad en el sistema.

Atte, Aseguradora SA.

Responder Reenviar

* 3.3.7.1.2 Listado de bajas

Para acceder al listado de todos los usuarios a los cuales el administrador dio de baja (Instrucción 3.2.3.7.1.3 Dar de baja) en algún momento, primero debe acceder a la pantalla de gestión de usuarios (Instrucción 3.2.3.7.1 Gestionar usuarios), aségrese que ninguna fila se encuentre seleccionada y presione en la opción “Bajas” ubicada en la parte superior izquierda de la tabla (), y seguido de esto se abrirá la siguiente ventana:



Numero	Nombre	Rol	Estado	Reactivar
9854331376	Usuario De Ejemplo	Usuario	Baja	
2426240921	Cristian Cabrera	Usuario	Baja	
5735999952	Sonia Gutierrez	Usuario	Baja	

 Cerrar

Para regresar al listado de usuarios dados de alta, presione en “Cerrar”.

En la tabla, cada usuario dado de baja se representa en una fila, mientras que en cada columna se detalla su información general.

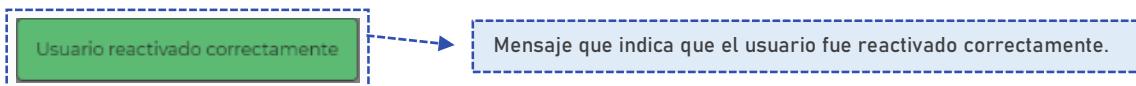
> 3.3.7.1.2.1 Reactivar usuario

Una vez ubicado el usuario que desea reactivar en la tabla, presione en la opción ubicada al final de la fila sobre la columna “Reactivar” (), seguido de esto se mostrara el siguiente mensaje de confirmación:



Para volver al listado de usuarios dados de baja, presione en la opción “Atrás”.

Para confirmar la reactivación, presione en la opción “Reactivar” y seguido de esto se mostrará un mensaje de éxito si el proceso se realizó correctamente, en caso contrario, el mensaje será de error.



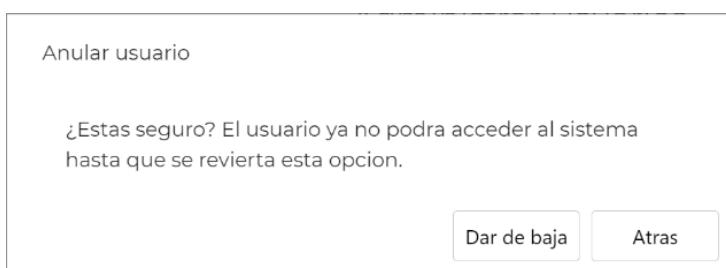
Una vez finalizado correctamente el proceso, el usuario reactivado ya podrá ingresar nuevamente al sistema con los últimos datos proporcionados, y ya podrá visualizarse en la tabla con el listado de usuarios dados de alta.

□	5735999952	Sonia Gutierrez	Usuario	Activo
---	------------	-----------------	---------	--------

En el ejemplo podemos ver el usuario “Sonia Gutiérrez” recientemente dado de alta con su actual estado “Activo”.

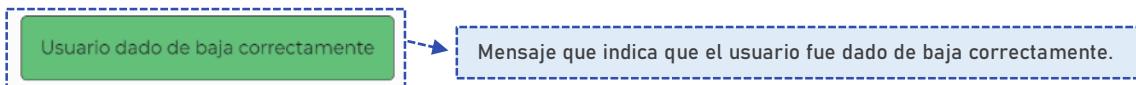
* 3.3.7.1.3 Dar de baja

Primero: Una vez se encuentre en la pantalla de gestión de usuarios del sistema, (Instrucción 3.2.3.7.1 Gestionar usuarios), debe seleccionar la fila correspondiente a la persona que desea dar de baja, y presionar en la opción “Baja” ubicada en la parte superior izquierda de la tabla (Baja) y seguido de esto se mostrara el siguiente mensaje de confirmación:



Presione en la opción “Atrás” para volver a la pantalla de gestión de usuarios.

Segundo: Presione en la opción “Dar de baja” para confirmar el proceso de anulación del usuario, seguido de esto, si la operación fue realizada correctamente, se mostrará un mensaje de éxito, en caso contrario, se mostrará un mensaje de error.



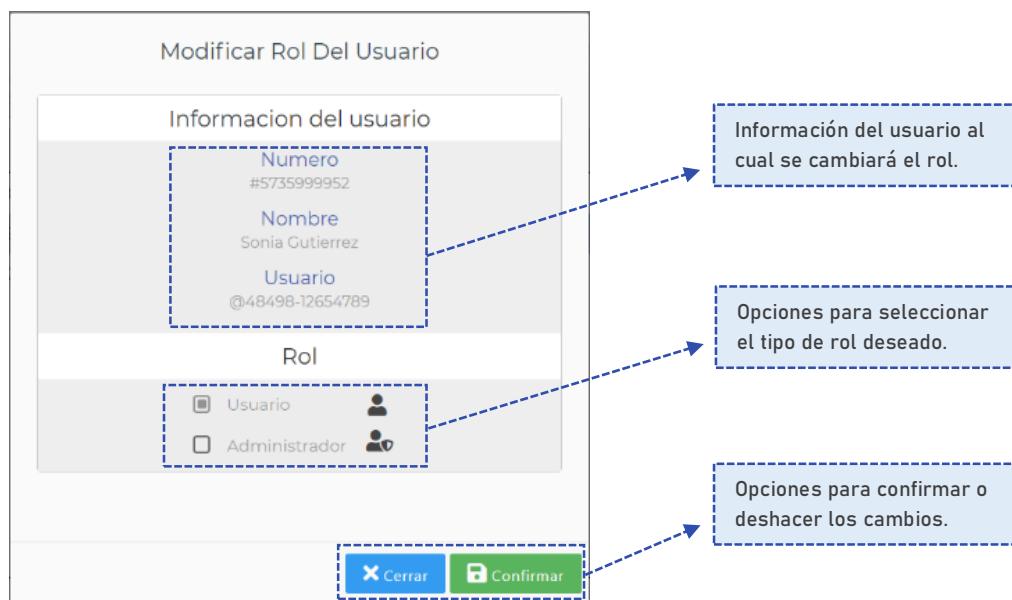
Una vez finalizado el proceso, el usuario ya se encontrará dado de baja y no tendrá autorización para ingresar al sistema.

En el ejemplo, fue dado de baja el usuario “Carlos Piniera”.

2055183625	Carlos Piniera	Usuario	Baja	
------------	----------------	---------	------	--

* 3.3.7.1.4 Modificar rol

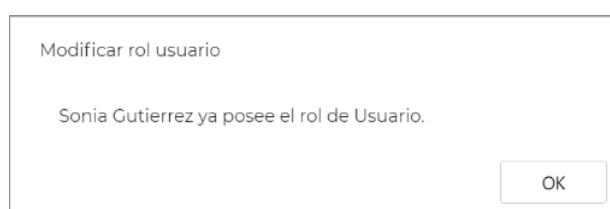
Primero: Una vez se encuentre en la pantalla de gestión de usuarios del sistema, (Instrucción 3.2.3.7.1 Gestionar usuarios), debe seleccionar la fila correspondiente a la persona que desea modificar el rol, y presionar en la opción “Rol” ubicada en la parte superior izquierda de la tabla ( Rol) y seguido de esto se abrirá la siguiente ventana:



Si desea volver a la pantalla de gestión de usuarios y deshacer los cambios, presione en “Cerrar”.

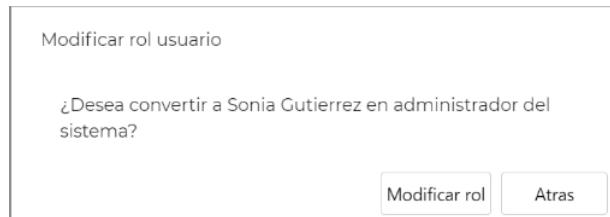
Segundo: Una vez abierta la ventana, debe seleccionar el rol deseado.

En el ejemplo se cambiará el rol a “Sonia Gutiérrez”, la cual posee actualmente el de tipo “Usuario”.



En caso de seleccionar el mismo tipo de rol y se intente guardar los cambios, el sistema mostrara un mensaje de notificación informativo.

Tercero: Una vez seleccionado el rol deseado, presione en la opción “Confirmar”, y seguido de esto se mostrará un mensaje de confirmación:



Presione en “Modificar rol” para confirmar el cambio de rol.

Presione “Atrás” para regresar a la pantalla anterior.



Una vez realizado con éxito el cambio de rol, se mostrará un mensaje informativo.

Presione en “OK” para cerrarlo.

Una vez finalizado el proceso correctamente, el usuario ya tendrá a las

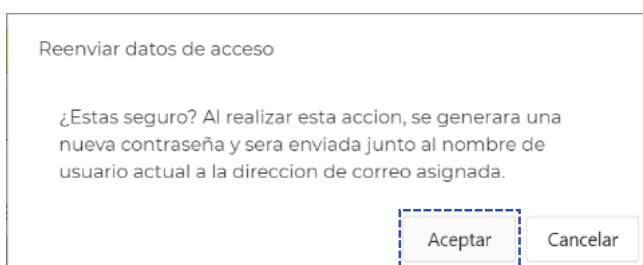
<input type="checkbox"/>	5735999952	Sonia Gutierrez	Administrador	Activo
--------------------------	------------	-----------------	---	--------

funcionalidades que ese rol específico posee.

* 3.3.7.1.5 Reenviar datos de acceso a usuarios

Al momento de registrar un nuevo usuario en el sistema, puede darse la situación en que algunos datos no sean ingresados y por ende grabados correctamente, ya sea por diferentes motivos, pero el caso particular del email significa un problema mayor, ya que, si este campo es ingresado con errores, como por ejemplo mal escrito, quiere decir que los datos de acceso generados por el sistema nunca llegarán a la casilla del email del usuario, provocando que este nunca pueda ingresar. Para este problema específico es que se creó esta función, permitiendo generar una nueva contraseña por cada vez que se lleva a cabo, la cual podrá enviarse a una dirección de correo electrónico diferente a la ya almacenada en el sistema.

Primero: Una vez se encuentre en la pantalla de gestión de usuarios, (Instrucción 3.2.3.7.1 Gestionar usuarios), debe seleccionar la fila correspondiente a la persona que desea reenviar los datos para acceso al sistema, y luego presionar en la opción “Reenviar datos de acceso” ubicada en la parte superior izquierda de la tabla (Reenviar datos de acceso) y seguido de esto se mostrara el siguiente mensaje de confirmación:



Presione en la opción “Cancelar” si desea cerrar el mensaje y regresar a la pantalla de gestión de usuarios

Segundo: Presione en “Aceptar” para continuar con el proceso.

Reenviar Informacion De Acceso

Correo electronico
sonia15gutierrez@yahoo.com

X Cerrar Enviar

Una vez haya continuado en el paso anterior, se abrirá una nueva ventana en la que habrá un campo de texto, este contendrá la dirección de correo electrónico que se ingresó al registrar el usuario, pero podrá ser reemplazado sin problemas.

Si desea cerrar la ventana y regresar a la pantalla de gestión de usuarios, presione en la opción “Cerrar”.

Tercero: Una vez ingresado el correo electrónico deseado, presione en la opción “Enviar” para procesar el envío de la información.

Reenviar Informacion De Acceso

Correo electronico
sonia15gutierrez@yahoo.com

X Cerrar

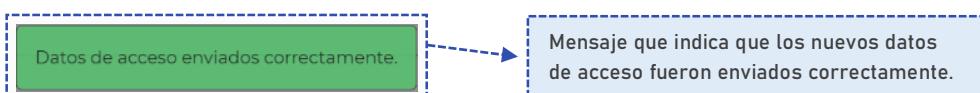
Registro de usuarios

La informacion para el acceso al sistema fueron enviados por e-mail al usuario.
En caso de no haber recibido la informacion, debe de ponerse en contacto con el administrador.

OK

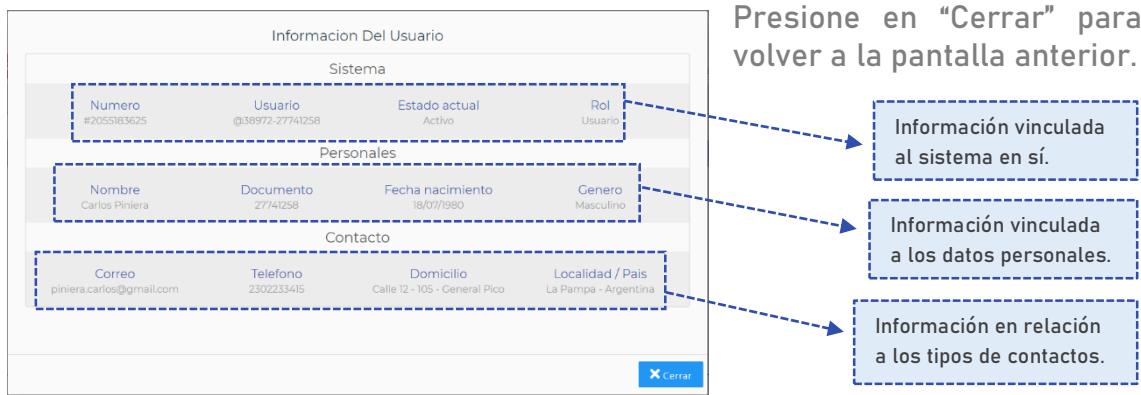
Si el proceso finalizo correctamente, se mostrará un mensaje de éxito, y el usuario podrá acceder a la información desde su bandeja de entrada de su correo electrónico personal. La estructura del email será exactamente igual que la descrita en la instrucción 3.2.3.7.1.1 (Nuevo usuario).

En caso de provocarse un fallo interno, se mostrar un mensaje de error.

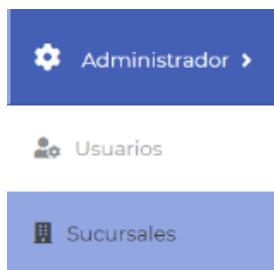


* 3.3.7.1.6 Consultar información

Una vez se encuentre en la pantalla de gestión de usuarios, (Instrucción 3.2.3.7.1 Gestión de usuarios), debe seleccionar la fila correspondiente a la persona que desea consultar su información, y luego presionar en la opción “Info” ubicada en la parte superior izquierda () y seguido de esto se abrirá la siguiente ventana:



→ 3.3.7.2 Gestionar sucursales



Vaya al menú principal y presione en “Administrador”, y una vez desplegada todas las acciones disponibles, pulse en “Sucursales”, seguido de esto, se mostrará en el contenedor general una tabla donde se listarán todas las sucursales registrados en el sistema.

Opción para agregar una nueva sucursal.

Campo para buscar una sucursal específica.

Administrador Sucursales

Identificador	Nombre	Dirección	Código Postal	Ubicación	Acciones
SR-906356	Sucursal Santa Rosa	9 de julio 566	6300	Santa Rosa - La Pampa	
BB-435476	Sucursal Bahía blanca	Gral Belgrano 112	8000	Bahía Blanca - Buenos Aires	
UQ-849673	Sucursal Buenos Aires	Avenida Gral Paz 655	0	C.A.B.A. - Ciudad Autónoma de Buenos Aires (CABA)	

Listado de todas las sucursales registradas en el sistema.

Opción para consultar los datos de la sucursal.

Opción para modificar la información de la sucursal.

Opción para eliminar la sucursal del sistema.

En la tabla se muestran todas las sucursales registradas en el sistema, de las cuales, cada una se representa en una fila específica, y en las columnas se detalla su información general.

Identificador: Es el número único asignado a cada sucursal.

Nombre:

Dirección: Es la calle y altura del domicilio en el que se encuentra la sucursal.

Código Postal: Código postal de la zona geográfica en la que se encuentra la sucursal.

Ubicación: Nombre de la ciudad y provincia en la que se encuentra la sucursal.

Acciones: Columna que hace referencia al listado de acciones disponibles a realizar sobre la sucursal:

 : Opción que permite consultar la información de la sucursal (Instrucción 3.2.3.7.2.2 Consultar informaticen de una sucursal),

 : Opción que permite modificar los datos de una sucursal (Instrucción 3.2.3.7.2.3 Modificar información de una sucursal).

 : Opción que permite eliminar por completo del sistema una sucursal (Instrucción 3.2.3.7.2.4 Eliminar una sucursal).

* 3.3.7.2.1 Agregar nueva sucursal

Primero: Una vez nos encontremos en la pantalla de gestión de sucursales (Instrucción 3.2.3.7.2 Gestionar sucursales), presione en la opción “Agregar” ubicada en la parte superior izquierda de la tabla (), y seguido de esto se mostrara el siguiente formulario:

————— Administrar Sucursales ————

 Agregar Nueva Sucursal

<p>Nombre</p> <input type="text" value="Sucursal de X ciudad"/>	<p>Cuit</p> <input type="text" value="00-00000000-0"/>	<p>Dirección</p> <input type="text" value="Calle x Numero x"/>
Provincia	Código Postal	Localidad
<input type="button" value="Seleccione"/>	<input type="button" value="Seleccione Provincia"/>	<input type="button" value="Seleccione Código Postal"/>





Opción para volver a la pantalla anterior.

Opción para confirmar el registro.

Segundo: Debe completar los siguientes campos del formulario:

Nombre: Campo de texto, debe ingresar el nombre que caracterice a la sucursal con respecto al resto, debe ser intuitivo así facilita la elección al usuario al momento de iniciar sesión en el sistema, se recomienda el formato “Sucursal + Nombre de la ciudad de origen”. Este campo es requerido obligatoriamente.

CUIT: Campo alfanumérico, debe ingresar la clave única de identificación tributaria de la empresa/sucursal, en caso de no contar, puede ingresar el DNI del responsable local. Este campo es requerido obligatoriamente.

Dirección: Campo alfanumérico, debe ingresar el domicilio de la sucursal, preferentemente el nombre o denominación de la calle seguido del número de la altura. Este campo es requerido obligatoriamente.

Provincia: Campo de selección, debe seleccionar la provincia en la que se encuentra la sucursal. Este campo es requerido obligatoriamente.

Código postal: Campo de selección, debe seleccionar el código postal de la zona geográfica en la que se ubica la sucursal. Este campo es requerido obligatoriamente una vez se haya seleccionado el campo “Provincia”.

Localidad: Campo de selección, debe seleccionar la provincia en la que se ubica la sucursal. Este campo es requerido obligatoriamente una vez se haya seleccionado el campo “Código postal”.

En caso de no completar alguno de los campos e intentar guardar los cambios de registro, se mostrará en la parte inferior del formulario un listado de todos los errores cometidos en los campos,

————— Administrar Sucursales ————

■ Agregar Nueva Sucursal

Nombre <input type="text" value="Sucursal de X ciudad"/>	Cuit <input type="text" value="00-00000000-0"/>	Direccion <input type="text" value="Calle x Numero x"/>
Provincia <input type="text" value="Seleccione"/>	Codigo Postal <input type="text" value="Seleccione Provincia"/>	Localidad <input type="text" value="Seleccione Codigo Postal"/>

◀ Volver■ Registrar

Debe ingresar el Nombre

Debe ingresar el Cuit

Debe ingresar la Direccion

Debe seleccionar una Provincia

Listado de errores cometidos en los campos requeridos.

Tercero: Una vez completada la información en los campos requeridos, presione en la opción “Registrar” para confirmar el registro de la sucursal.

Presione en la opción “Volver” para deshacer los cambios y volver a la pantalla de gestión de sucursales.

Si la operación fue realizada correctamente, se mostrará un mensaje de éxito, en caso contrario, se mostrará un mensaje de error.



* 3.3.7.2.2 Consultar información de una sucursal

Una vez nos encontremos en la pantalla de gestión de sucursales (Instrucción 3.2.3.7.2 Gestionar sucursales), y tenga ubicada la sucursal la cual desee consultar su información, presione en la opción ubicada al final de la fila sobre la izquierda en la columna “Acciones” (), y seguido se mostrara en el contenedor general la siguiente interfaz, donde se podrán apreciar todos los datos de la sucursal:

Presione en la opción “Volver” para regresar a la pantalla anterior.

Datos vinculados a la empresa.

Administrador Sucursales

Consultar Informacion

Sucursal Gral Pico

Empresa

Nombre
Sucursal Gral Pico

Identificador
HL-885363

Cuit
35-52358534-3

Ubicacion

Direccion
Calle 65 - 56

Codigo Postal
6360

Provincia
La Pampa

Localidad
General Pico

Volver

Datos vinculados a la ubicación.

* 3.3.7.2.3 Modificar información de una sucursal

Primero: Una vez nos encontremos en la pantalla de gestión de sucursales (Instrucción 3.2.3.7.2 Gestionar sucursales), y tenga ubicada la sucursal la cual desee modificar su información, presione en la opción ubicada al final de la fila, en el centro de la columna “Acciones” (), y seguido se mostrara en el contenedor general el siguiente formulario, cuyos campos se encontraran completos con los datos de la sucursal:

Presione en la opción “Volver” para regresar a la pantalla de gestión de sucursales.

————— Administrar Sucursales ————

 **Modificar Datos**
Sucursal Gral Pico - HL-885363

Nombre	Cuit	Direccion
Sucursal Gral Pico	35-52358534-3	Calle 65 - 56
Provincia	Codigo Postal	Localidad
La Pampa	6360	General Pico

 Volver Modificar

Segundo: Debe reemplazar la información de los campos requeridos, cuya validación es exactamente igual a la detallada en la instrucción 3.2.3.7.2.1 (Agregar nueva sucursal).

Tercero: Una vez complete los campos requeridos, presione en la opción “Modificar”, y si la operación finalizo correctamente, se mostrara un mensaje de éxito, en caso contrario, el mensaje será de error.

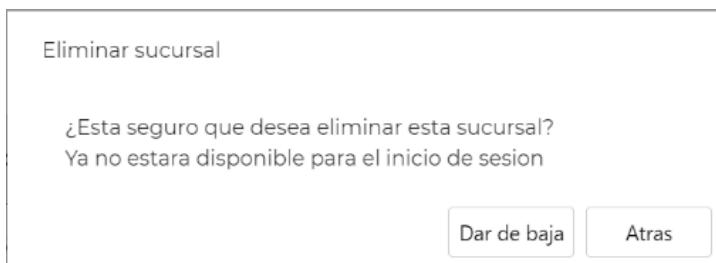
En el ejemplo, se modificó el campo “Dirección”, de “Calle 65 - 56” a “Calle 75 - 56”.



* 3.3.7.2.4 Eliminar una sucursal

Primero: Una vez nos encontremos en la pantalla de gestión de sucursales (Instrucción 3.2.3.7.2 Gestionar sucursales), y tenga ubicada la sucursal la cual desee eliminar del sistema, presione en la opción ubicada al final de la fila, a la derecha de la columna “Acciones” (), y seguido se mostrara el siguiente mensaje de confirmación:

Tenga en cuenta que, si elimina una sucursal, estará borrando por completo toda su información del sistema, y solo será posible recuperarla nuevamente registrando esa misma sucursal desde cero.



Presione en la opción "Atrás" si desea regresar a la pantalla de gestión de sucursales.

Segundo: Para confirmar la eliminación por completo de la sucursal del sistema, presione en la opción “Dar de baja”, y si el proceso finaliza correctamente, se mostrará un mensaje de éxito, y en caso contrario se mostrará un mensaje de error.



Capítulo cuarto

4º

Documentación técnica

Contenido:

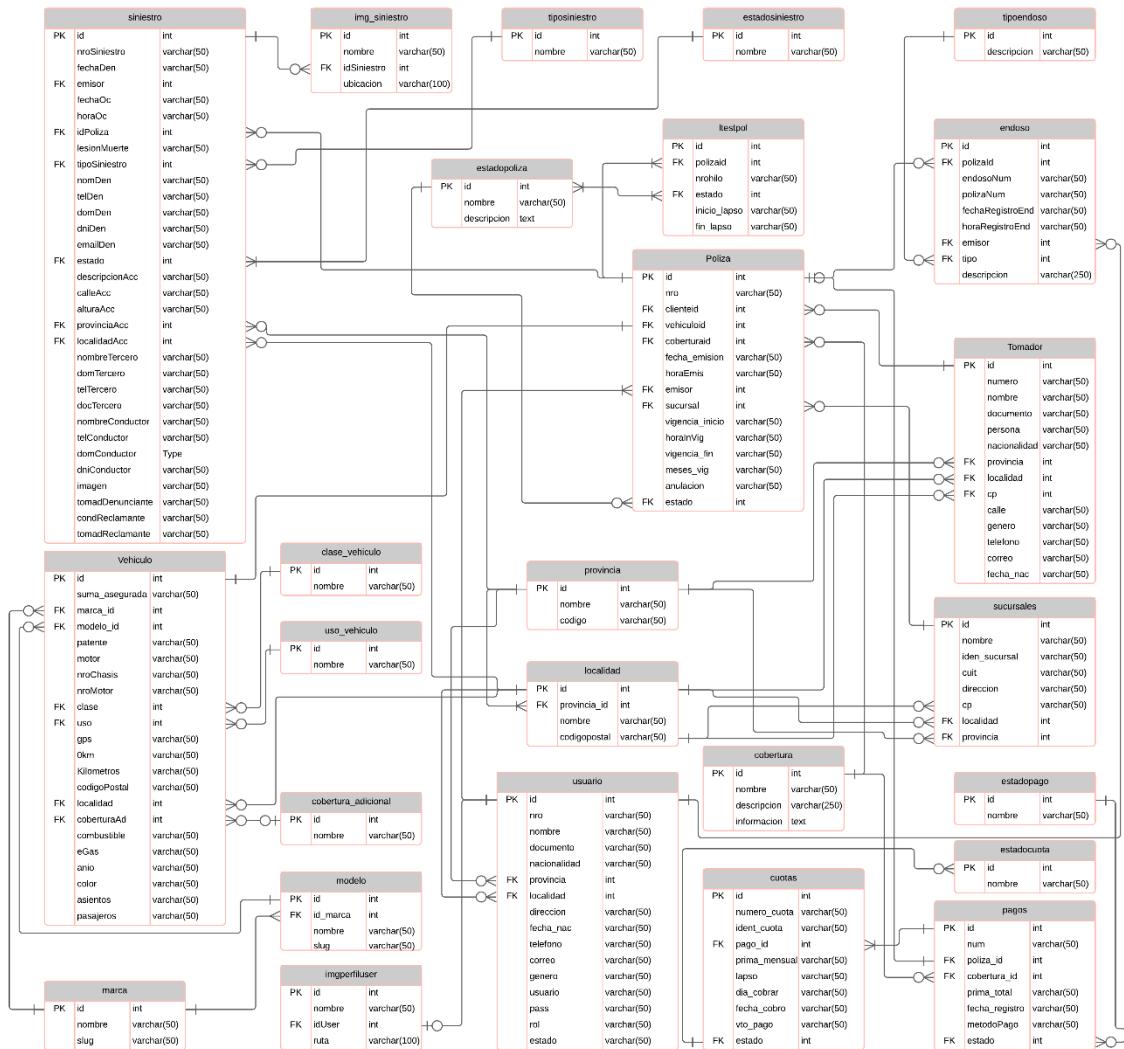
- 4.1 Diagramas del sistema
- 4.2 Listado de tablas de la base de datos
- 4.3 Código comentado

➤ 4.1 Diagramas del sistema

En esta sección se detallarán todos los gráficos o diagramas que representan la funcionalidad detallada del sistema. Los diagramas están desarrollados para ayudar a todos los intervenientes del desarrollo y modelado del sistema, logrando describir, diseñar, especificar, visualizar, construir y documentar todos los artefactos que lo componen, sirviéndose de varios tipos de diagramas. El modelado es la principal forma de visualizar el diseño de una aplicación con la finalidad de compararla con los requisitos antes de que el equipo de desarrollo comience a codificar, y cobra vital importancia para que todo funcione correctamente, permitiendo la escalabilidad, la seguridad y la ejecución.

✳ 4.1.1 Diagrama de relación de entidades

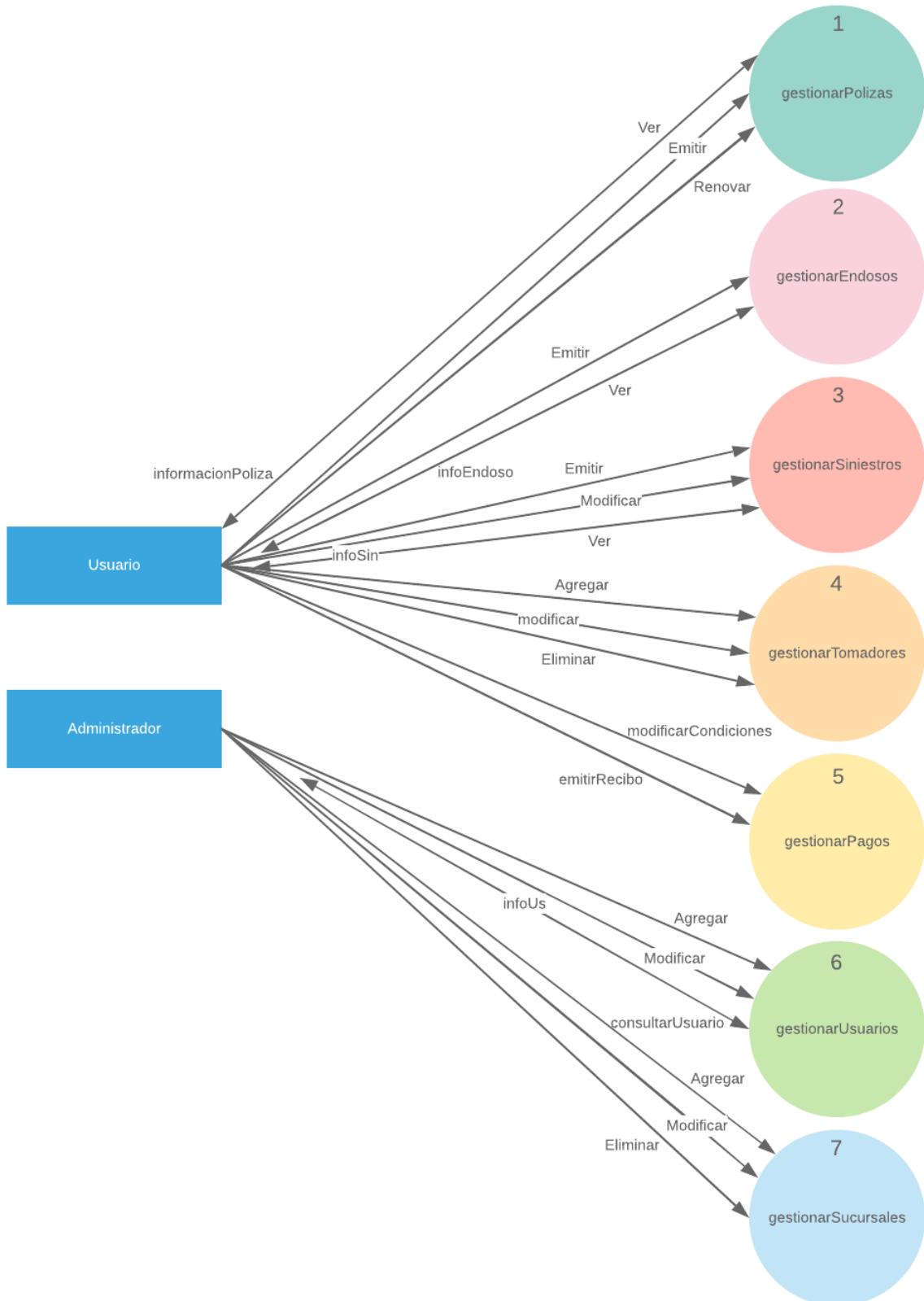
El modelo de datos de entidad-relación (ER) se basa en una percepción de un mundo real que consiste en un conjunto de objetos básicos llamados entidades y de relaciones entre estos objetos. Se desarrolló para facilitar el diseño de bases de datos permitiendo especificar un esquema empresarial. Este esquema representa la estructura lógica general de la base de datos.



4.1.2 Diagrama de flujo de datos

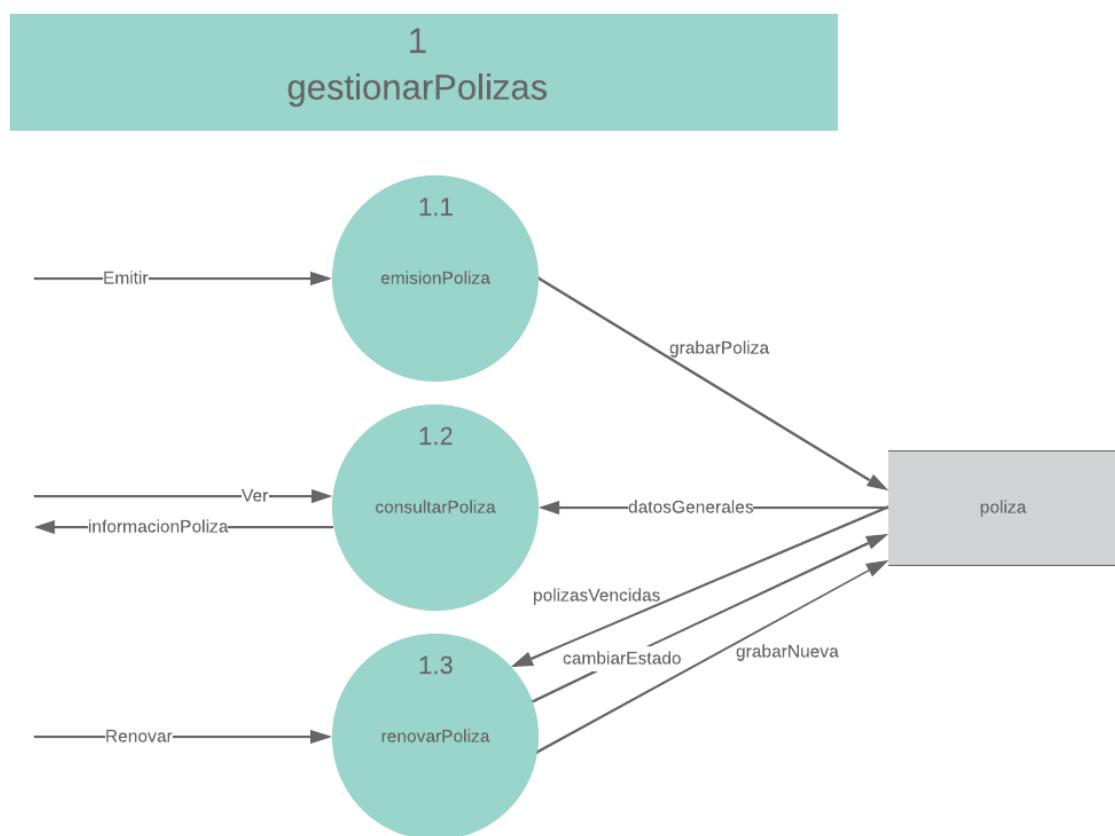
El diagrama de flujo de datos, se encarga de representar una secuencia de pasos lógicos, necesarios para realizar una tarea. Es una representación gráfica donde se desglosa de forma secuencial un proceso, se usan en cualquier actividad dentro de las empresas. Se trata de una herramienta que facilita el entendimiento de cada una de las fases de un proceso y su funcionamiento, permitiendo por lo tanto la posibilidad de estudiarlo para mejorar sus procedimientos. Los diagramas de flujo de datos han de representar el sistema de la forma más clara posible, por ello su construcción se basa en el principio de descomposición o explosión en distintos niveles de detalle. La descomposición por niveles se realiza de arriba abajo (*Top-Down*), es decir, se comienza en el nivel más general y se termina en el más detallado, pasando por los niveles intermedios necesarios. De este modo se dispondrá de un conjunto de particiones del sistema que facilitaran su estudio y su desarrollo.

A continuación, se muestra el diagrama de flujo de datos general del sistema o nivel 0:



En su nivel 0 cuenta con 7 procesos que a continuación serán nivelados a detalle:

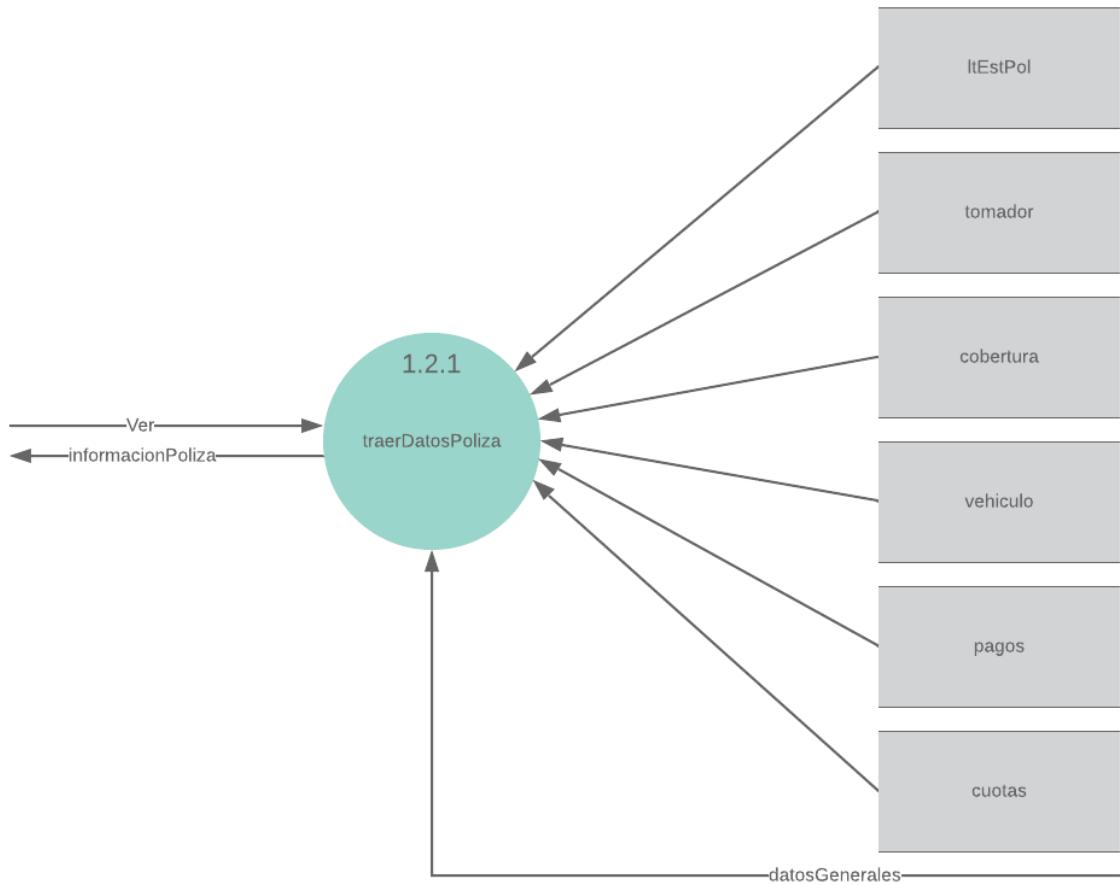
➔ 4.1.2.1 D.F.D. Nivel 1 Pólizas (1)



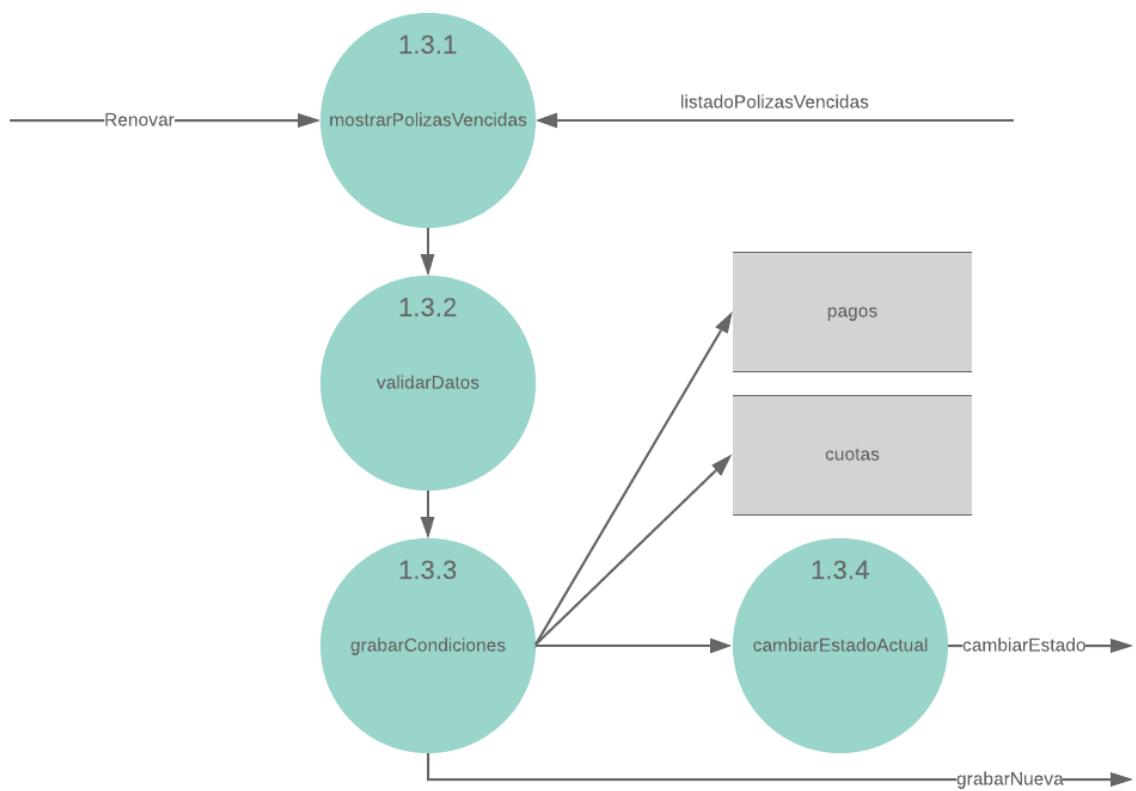
* 4.1.2.1.1 D.F.D. Nivel 2 Pólizas: Emisión de póliza (1.1)



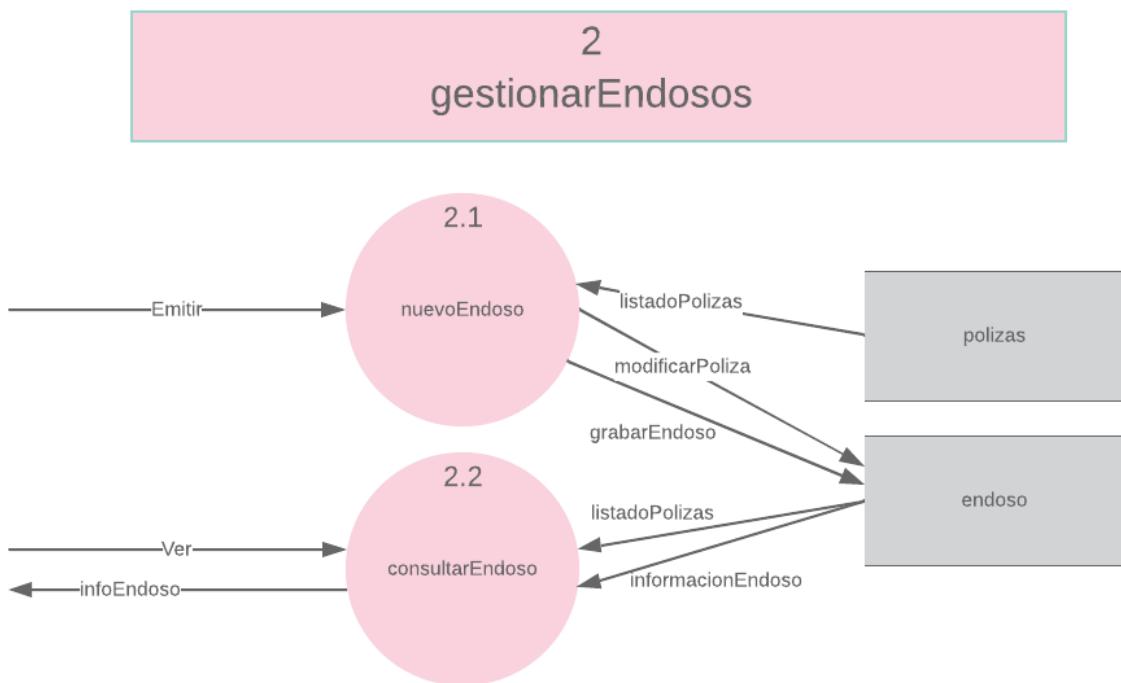
* 4.1.2.1.2 D.F.D. Nivel 2 Pólizas: Consultar póliza (1.2)



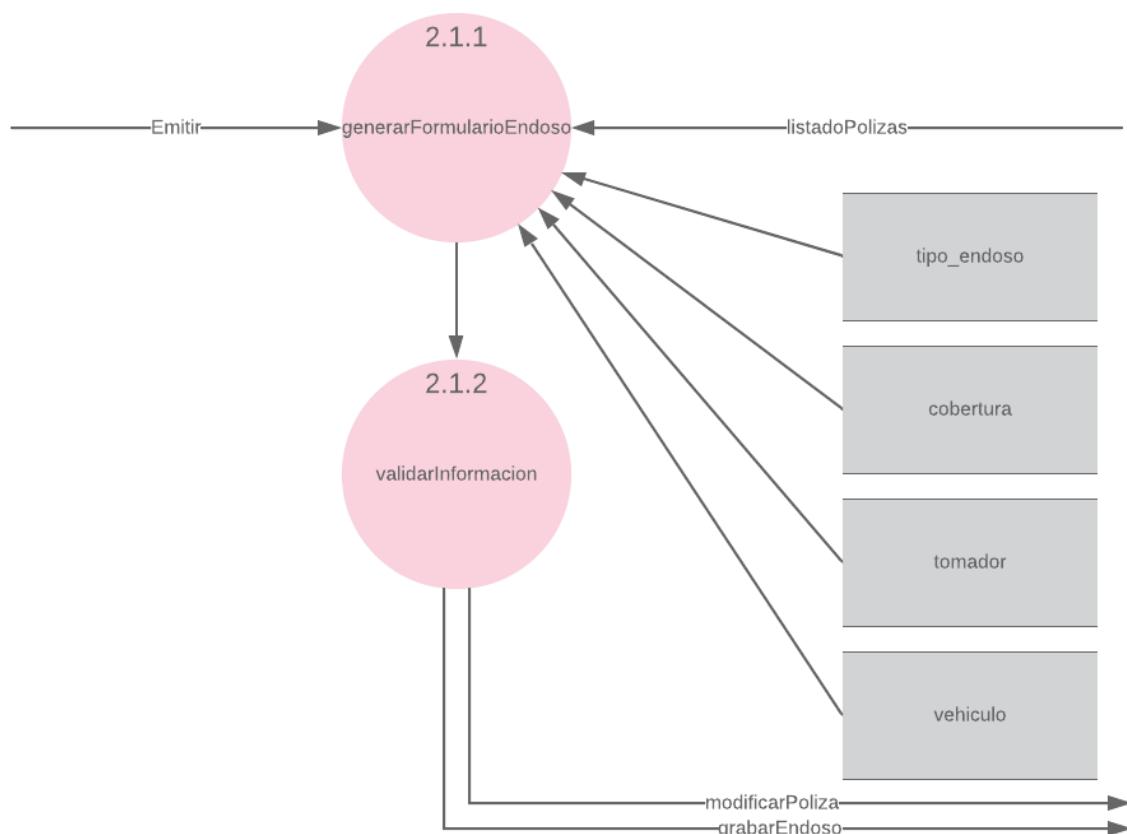
* 4.1.2.1.3 D.F.D. Nivel 2 Pólizas: Renovar póliza (1.3)



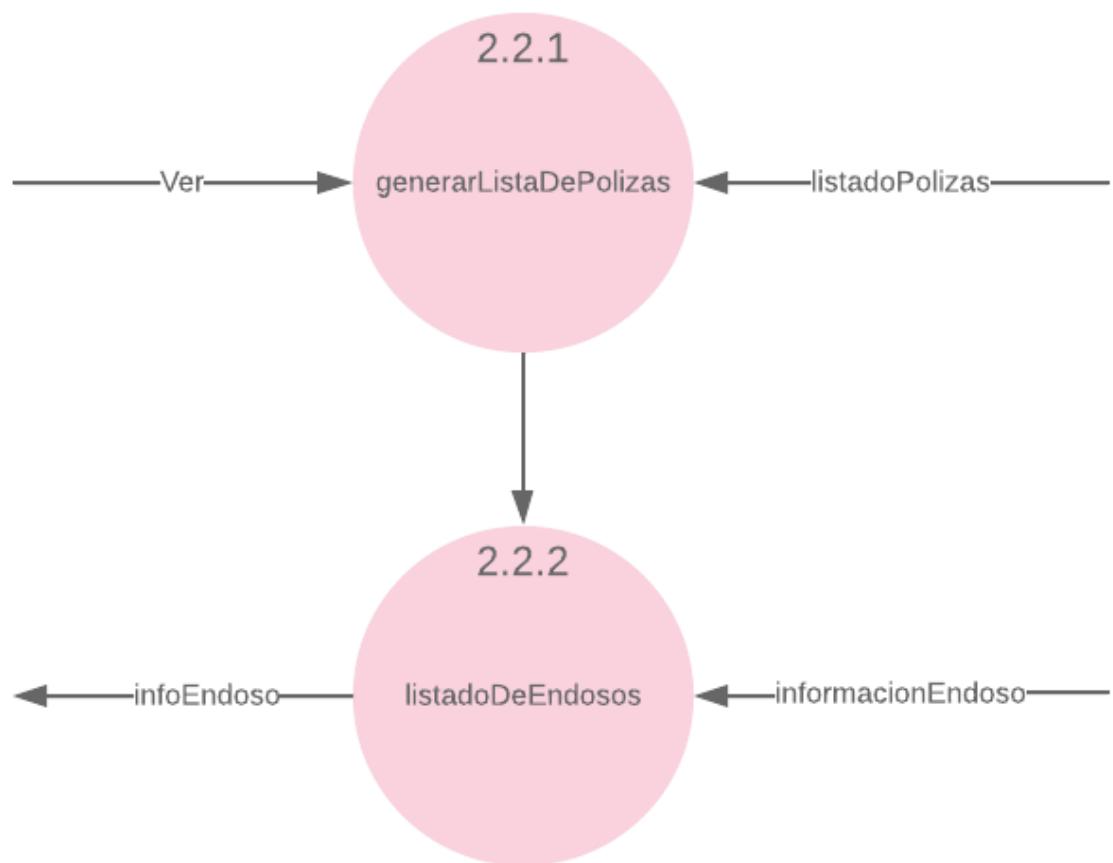
➔ 4.1.2.2 D.F.D. Nivel 1 Endosos (2)



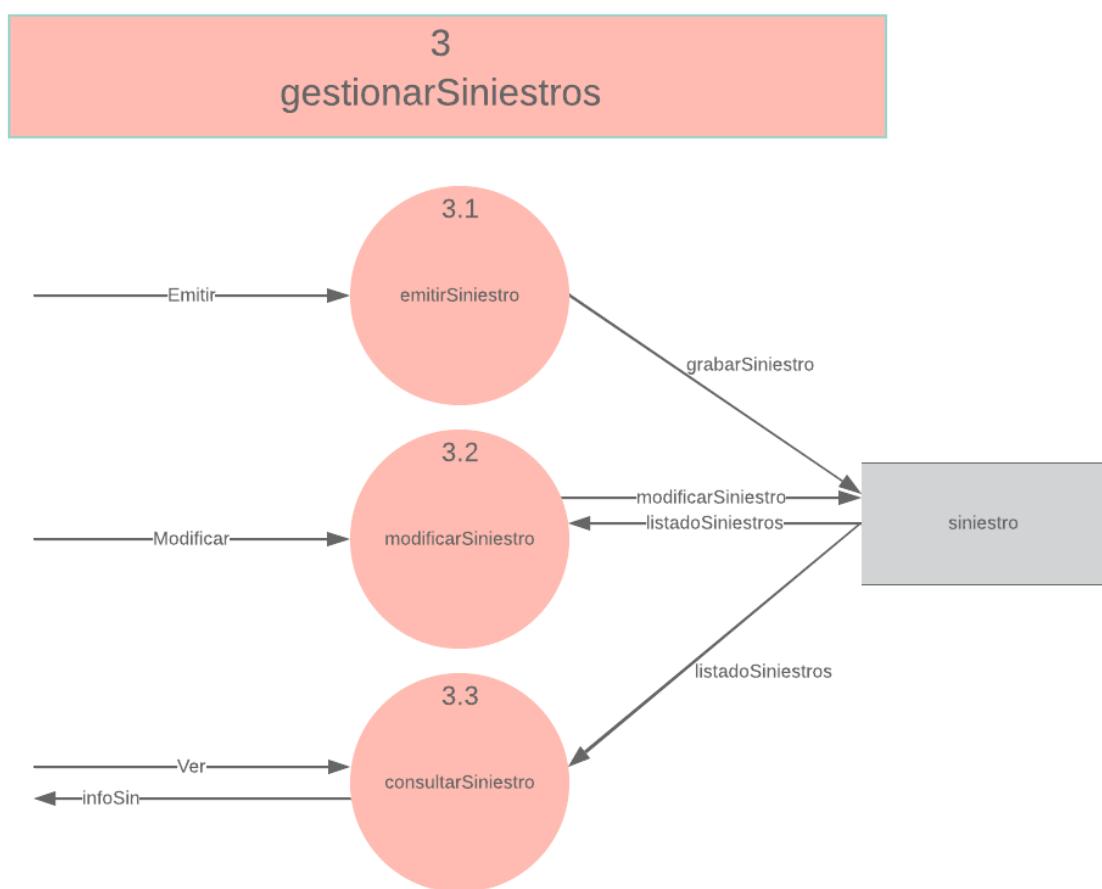
* 4.1.2.2.1 D.F.D. Nivel 2 Endosos: Emitir endosos (2.1)



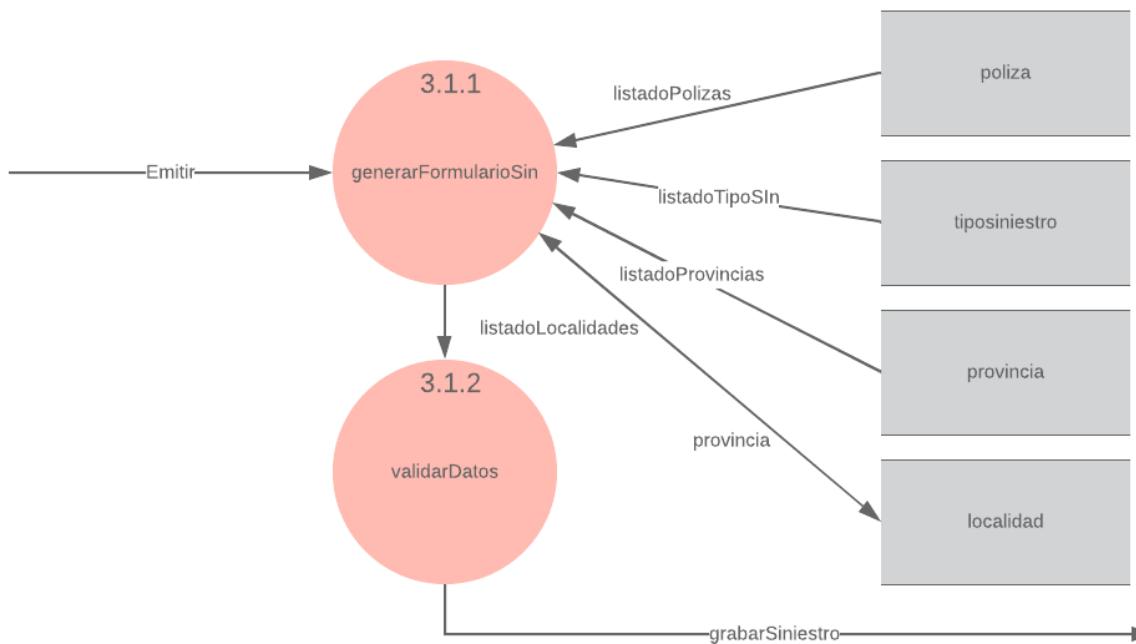
* 4.1.2.2.2 D.F.D. Nivel 2 Endosos: Consultar endosos (2.2)



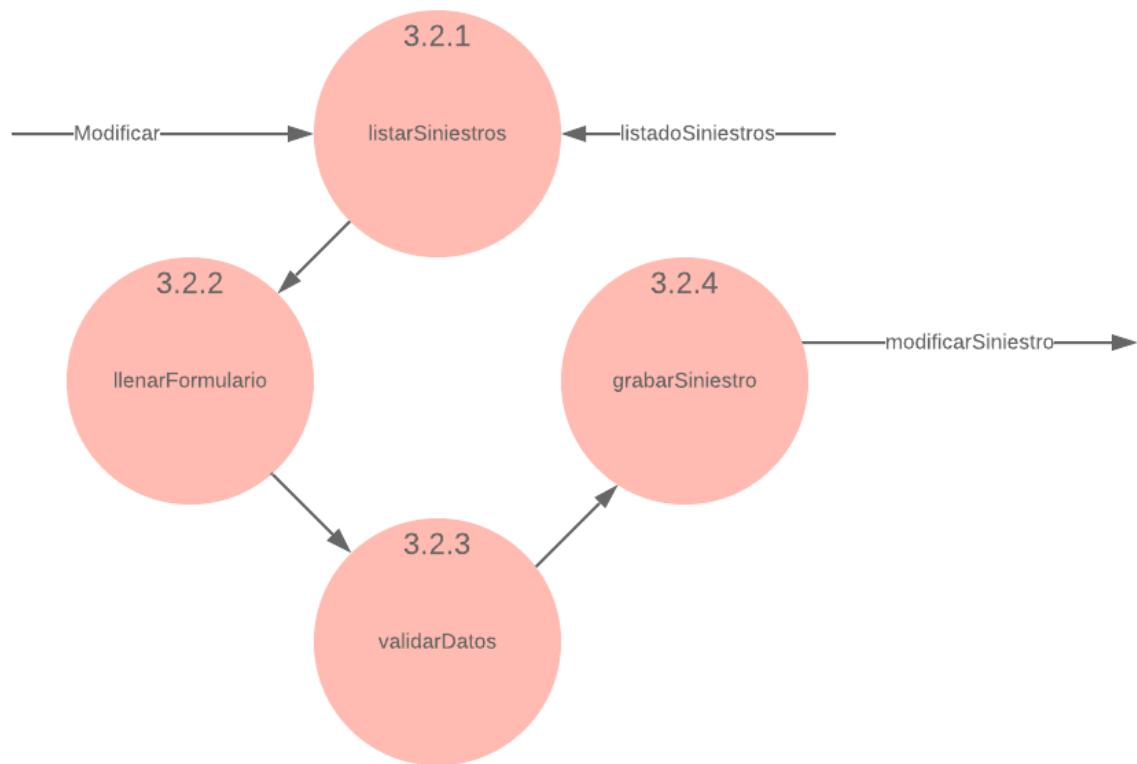
➔ 4.1.2.3 D.F.D. Nivel 1 Siniestros (3)



* 4.1.2.3.1 D.F.D. Nivel 2 Siniestros: Emitir siniestros (3.1)



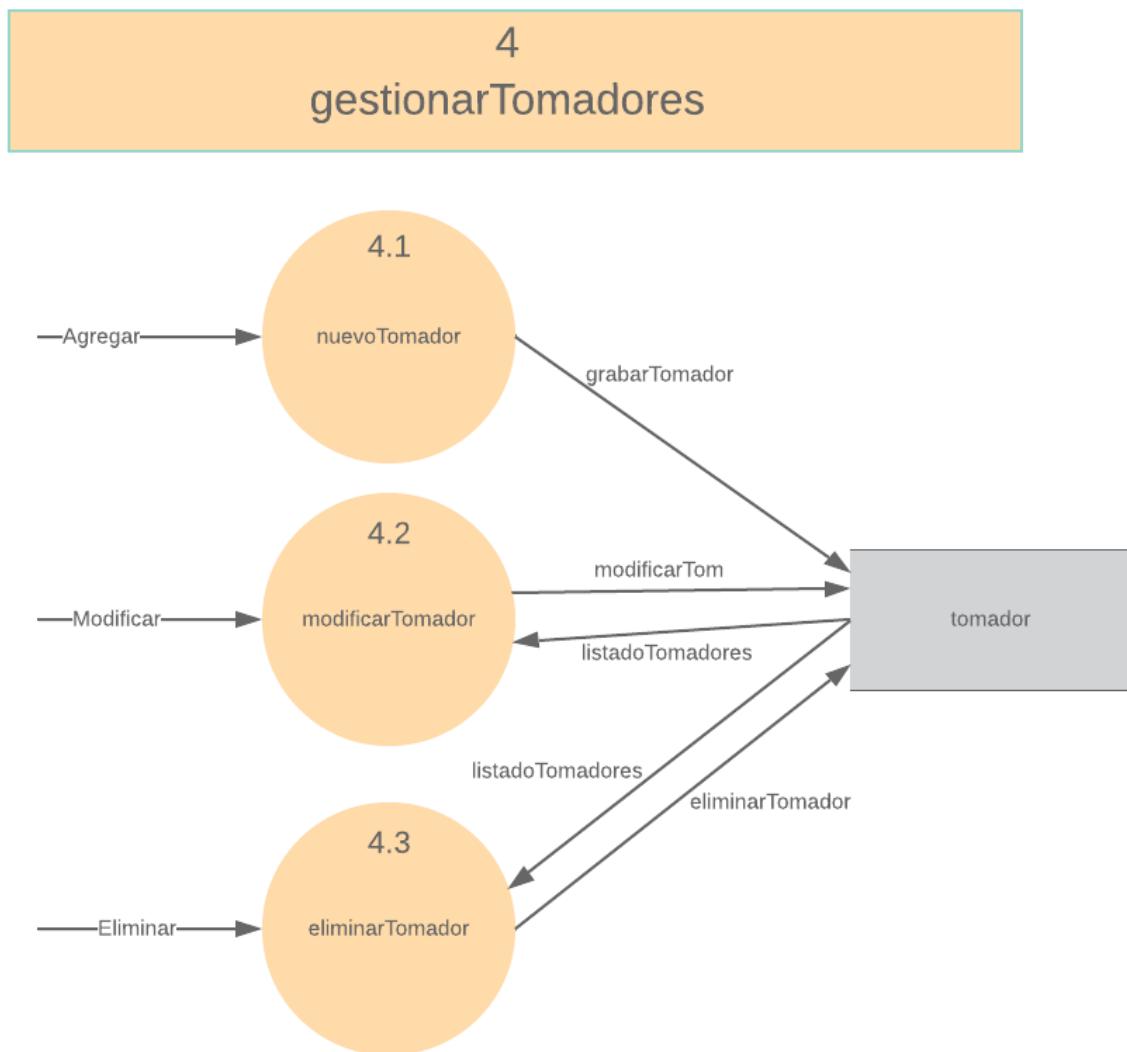
* 4.1.2.3.2 D.F.D. Nivel 2 Siniestros: Modificar siniestros
(3.2)



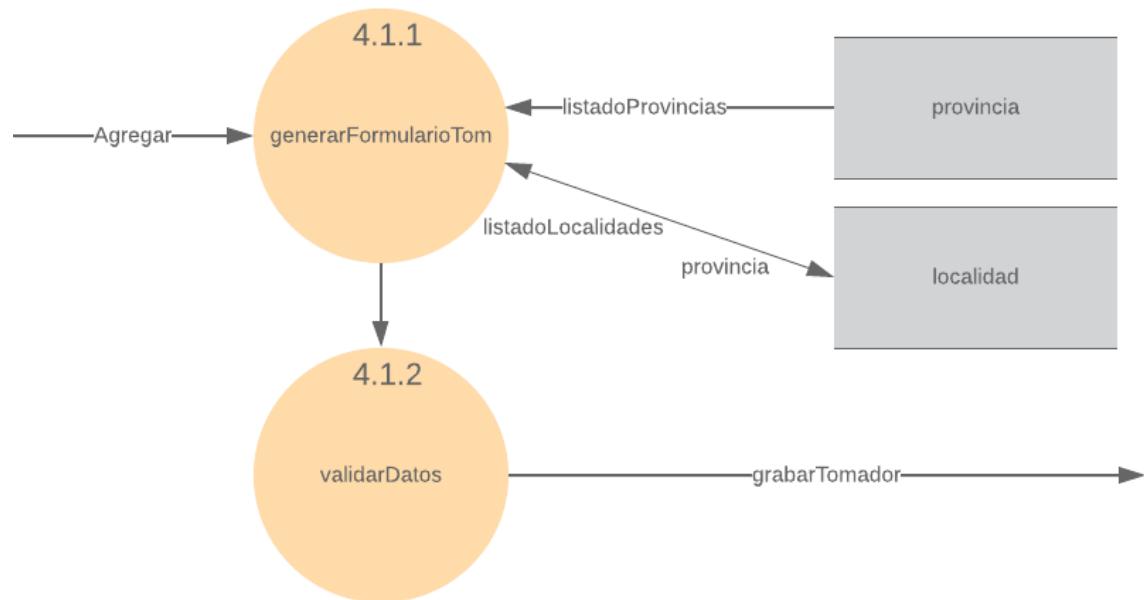
* 4.1.2.3.3 D.F.D. Nivel 2 Siniestros: Consultar siniestros
(3.3)



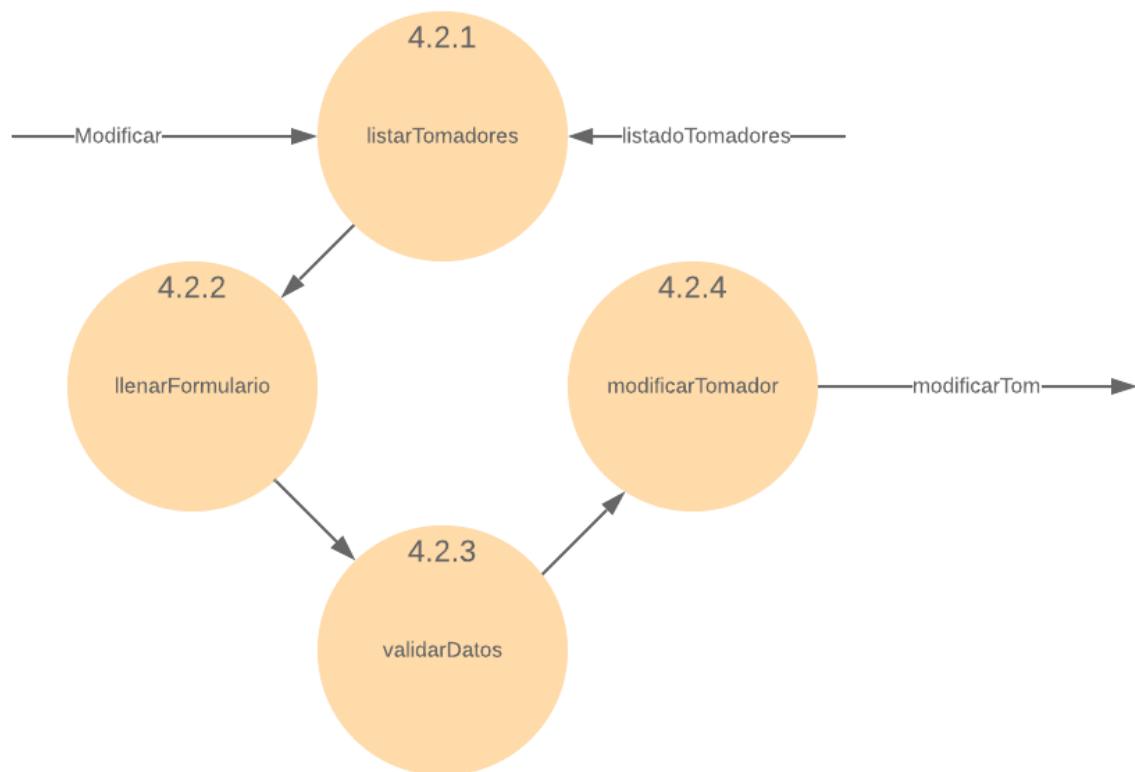
➔ 4.1.2.4 D.F.D. Nivel 1 Tomadores (4)



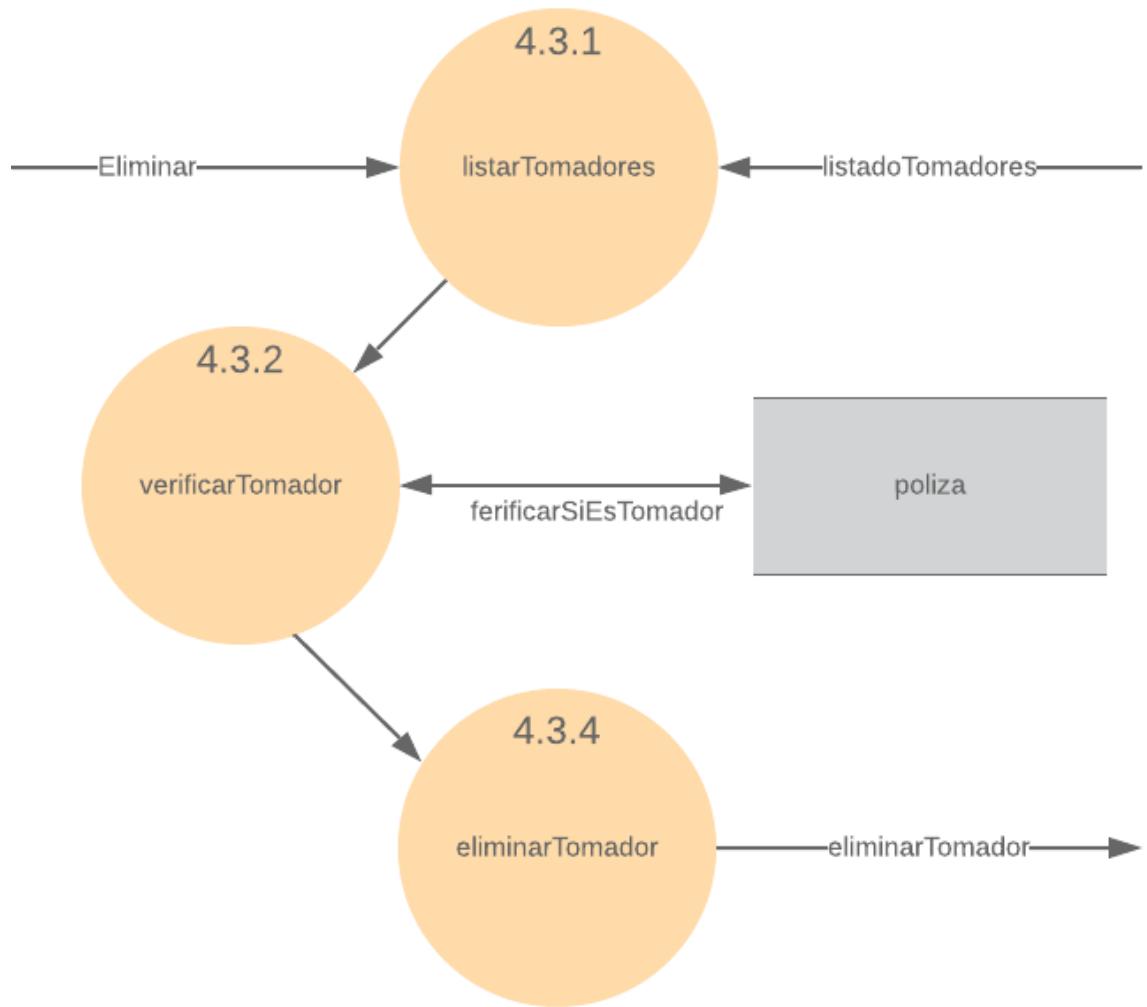
* 4.1.2.4.1 D.F.D. Nivel 2 Tomadores: Grabar tomadores (4.1)



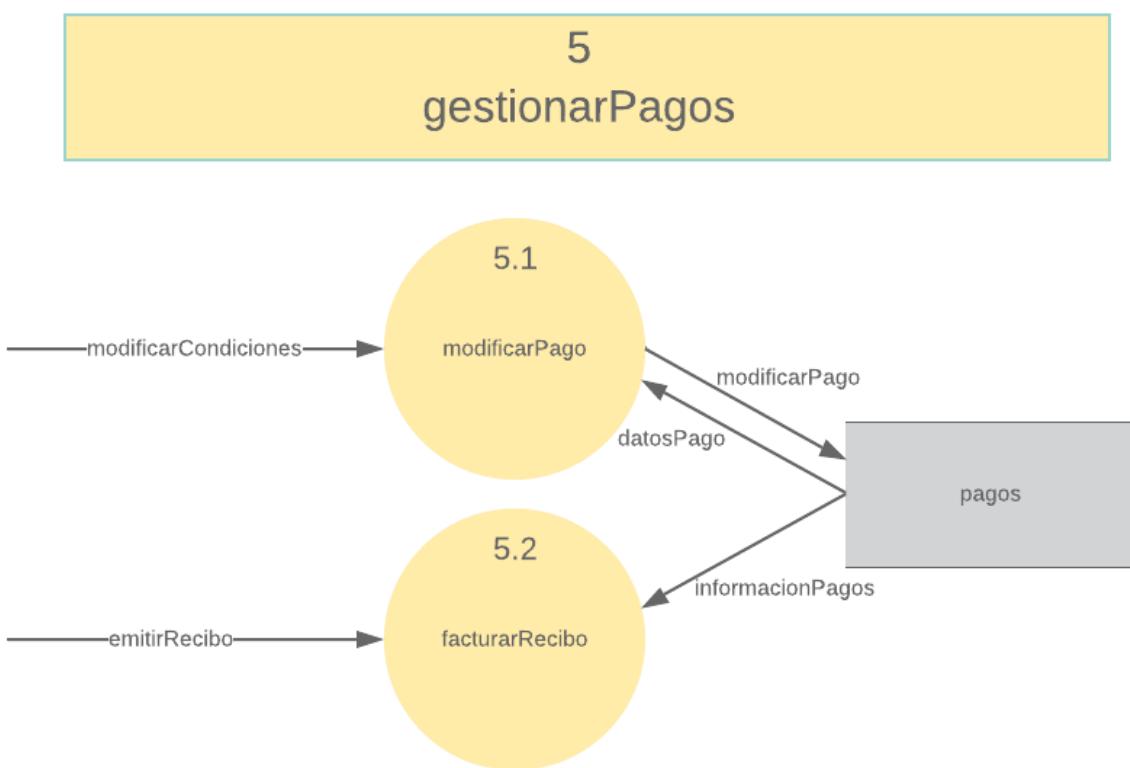
* 4.1.2.4.2 D.F.D. Nivel 2 Tomadores: Modificar tomadores
(4.2)



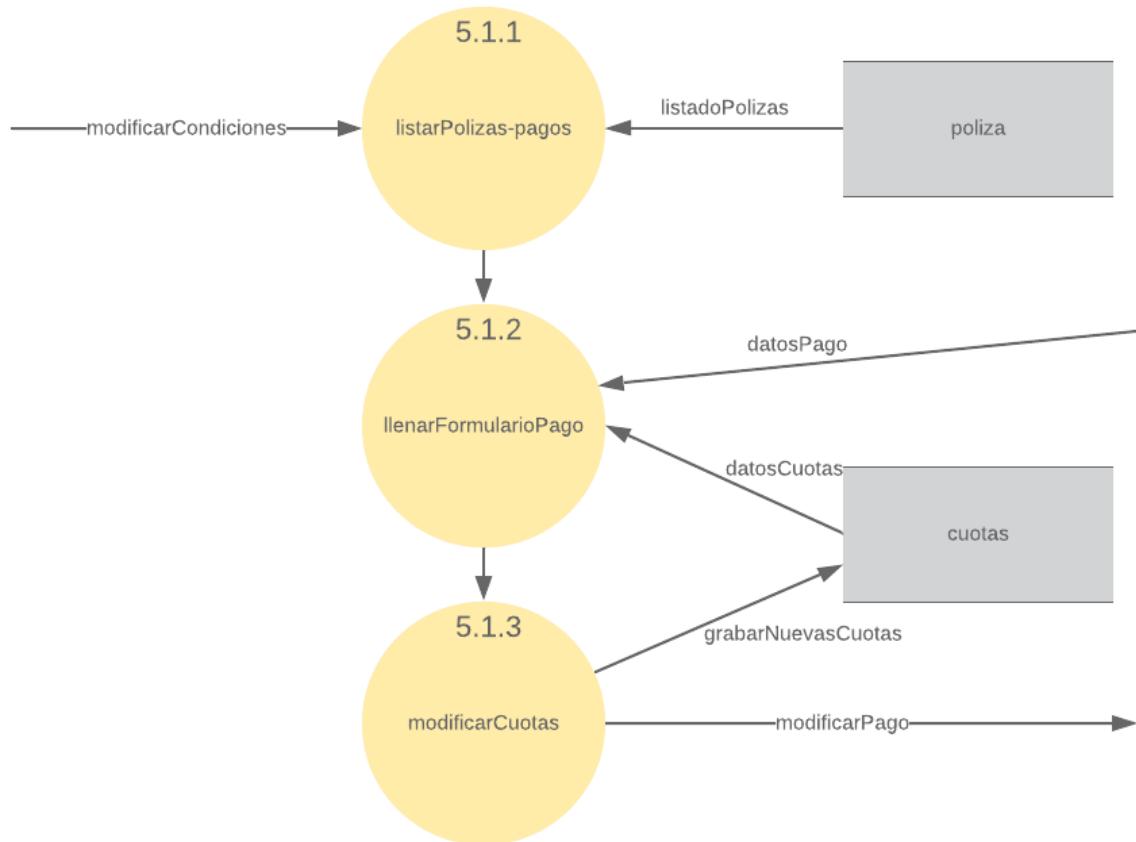
* 4.1.2.4.3 D.F.D. Nivel 2 Tomadores: Eliminar tomadores
(4.3)



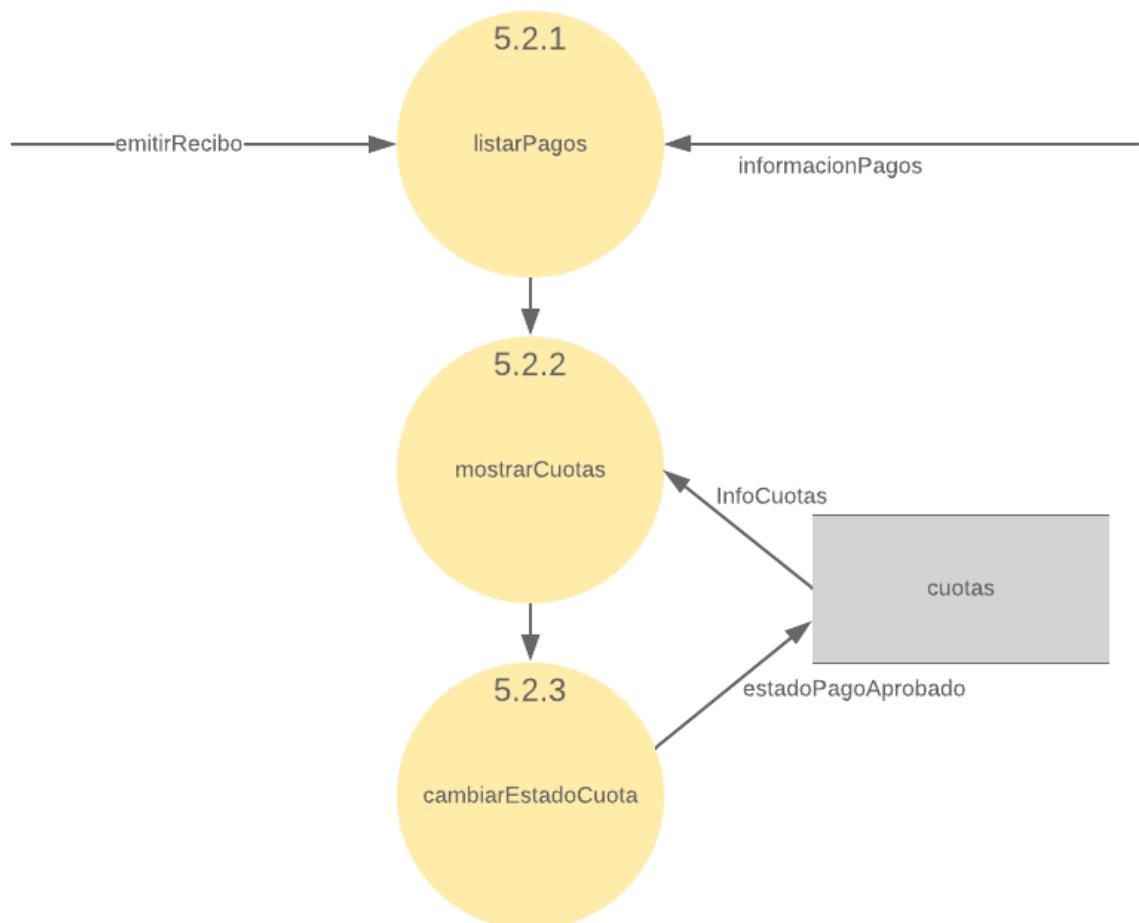
➔ 4.1.2.5 D.F.D. Nivel 1 Pagos (5)



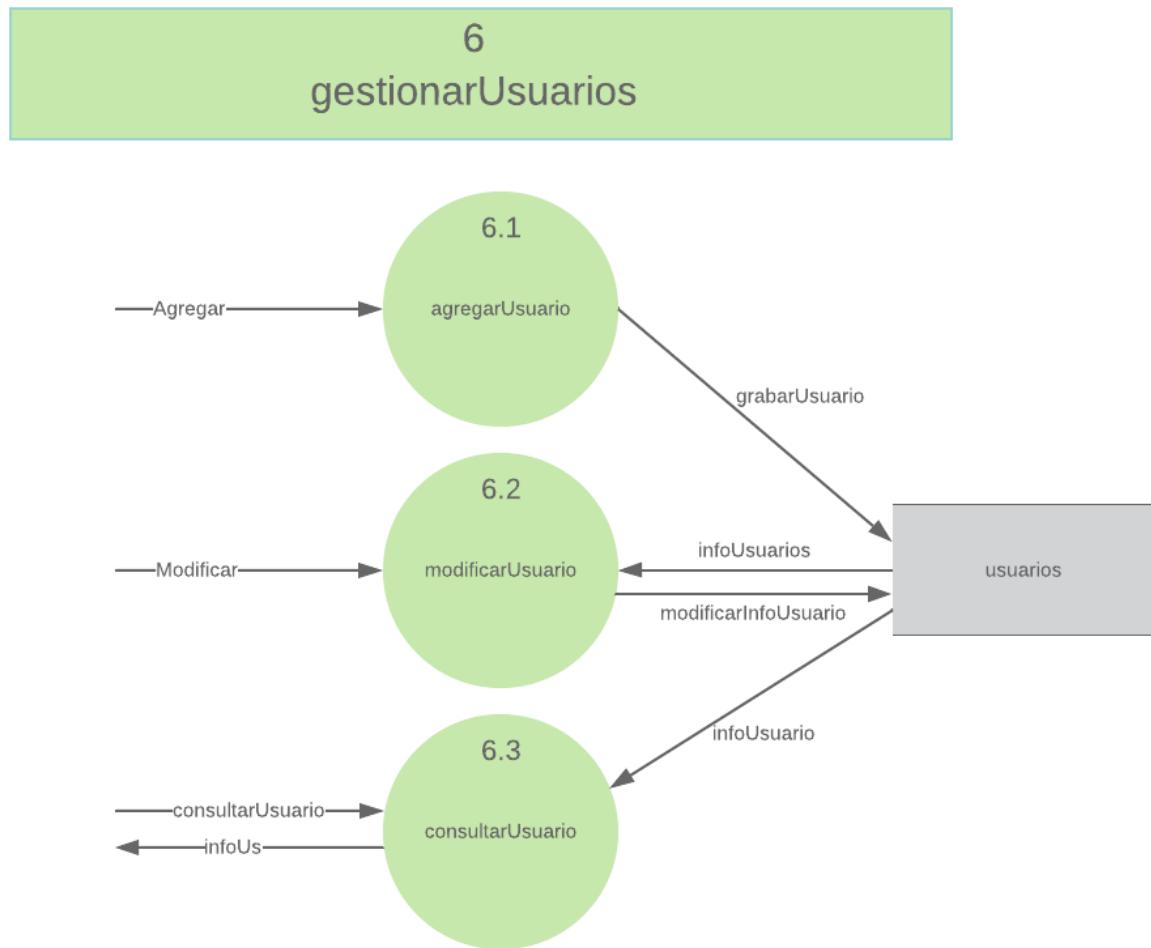
* 4.1.2.5.1 D.F.D. Nivel 2 Pagos: Modificar pagos (5.1)



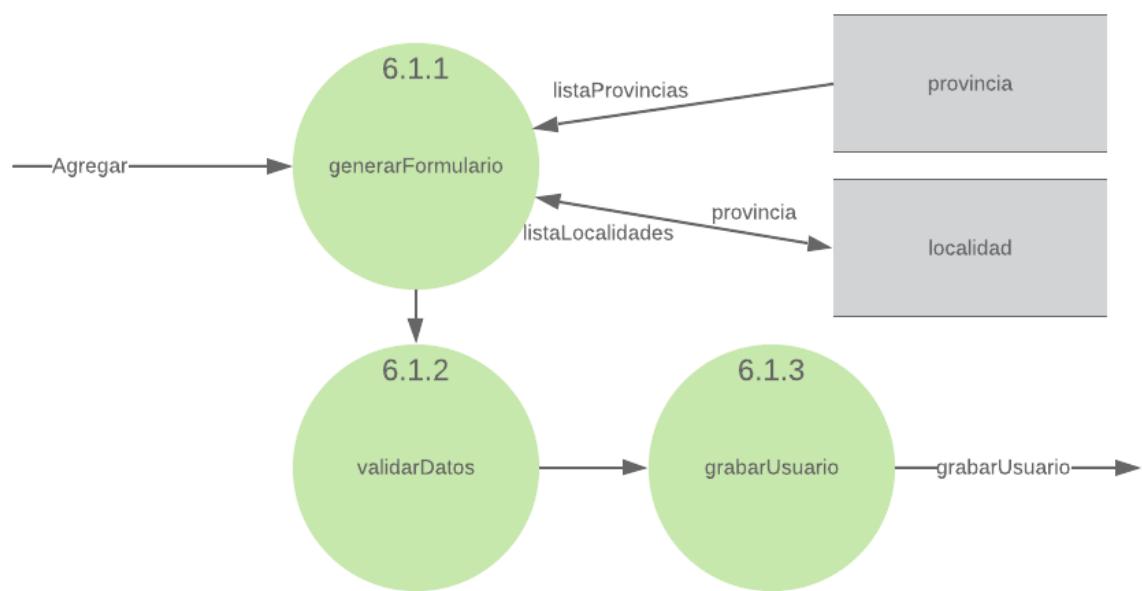
* 4.1.2.5.2 D.F.D. Nivel 2 Pagos: Emitir recibos (5.2)



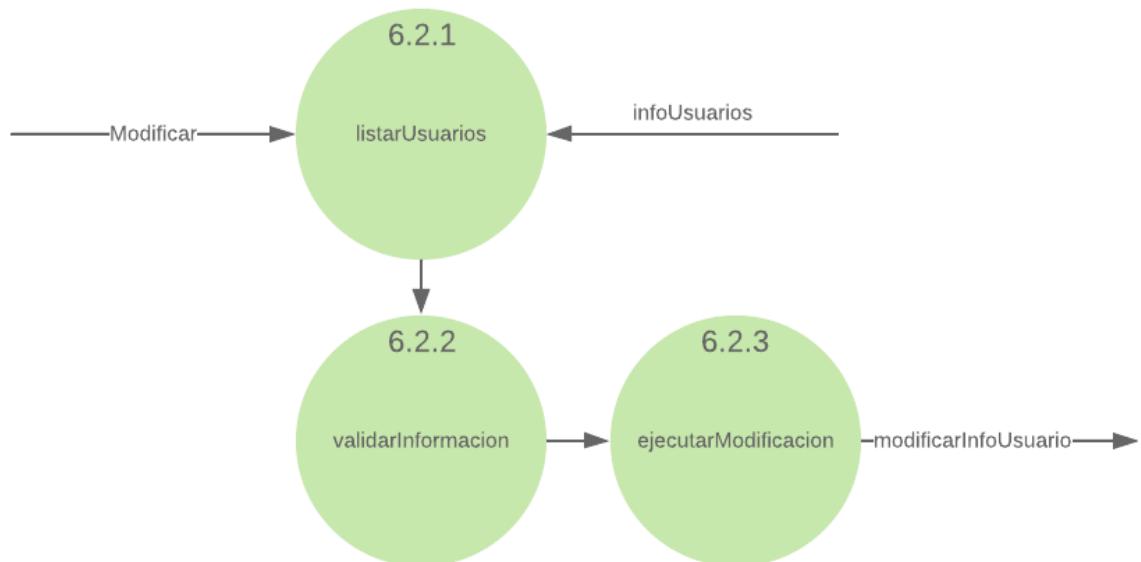
➔ 4.1.2.6 D.F.D. Nivel 1 Usuarios (6)



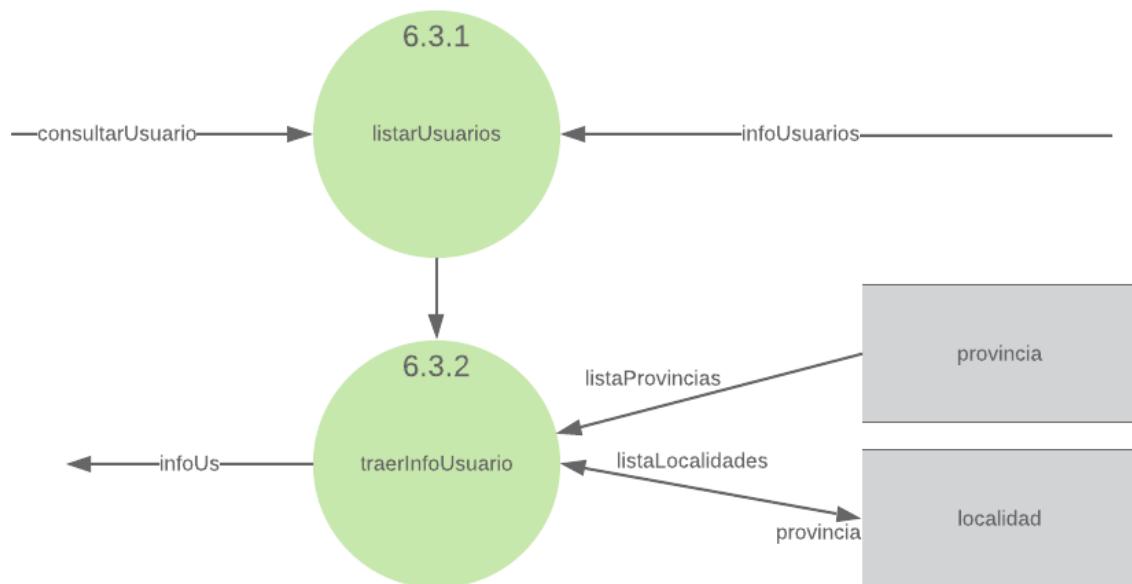
* 4.1.2.6.1 D.F.D. Nivel 2 Usuarios: Agregar usuarios (6.1)



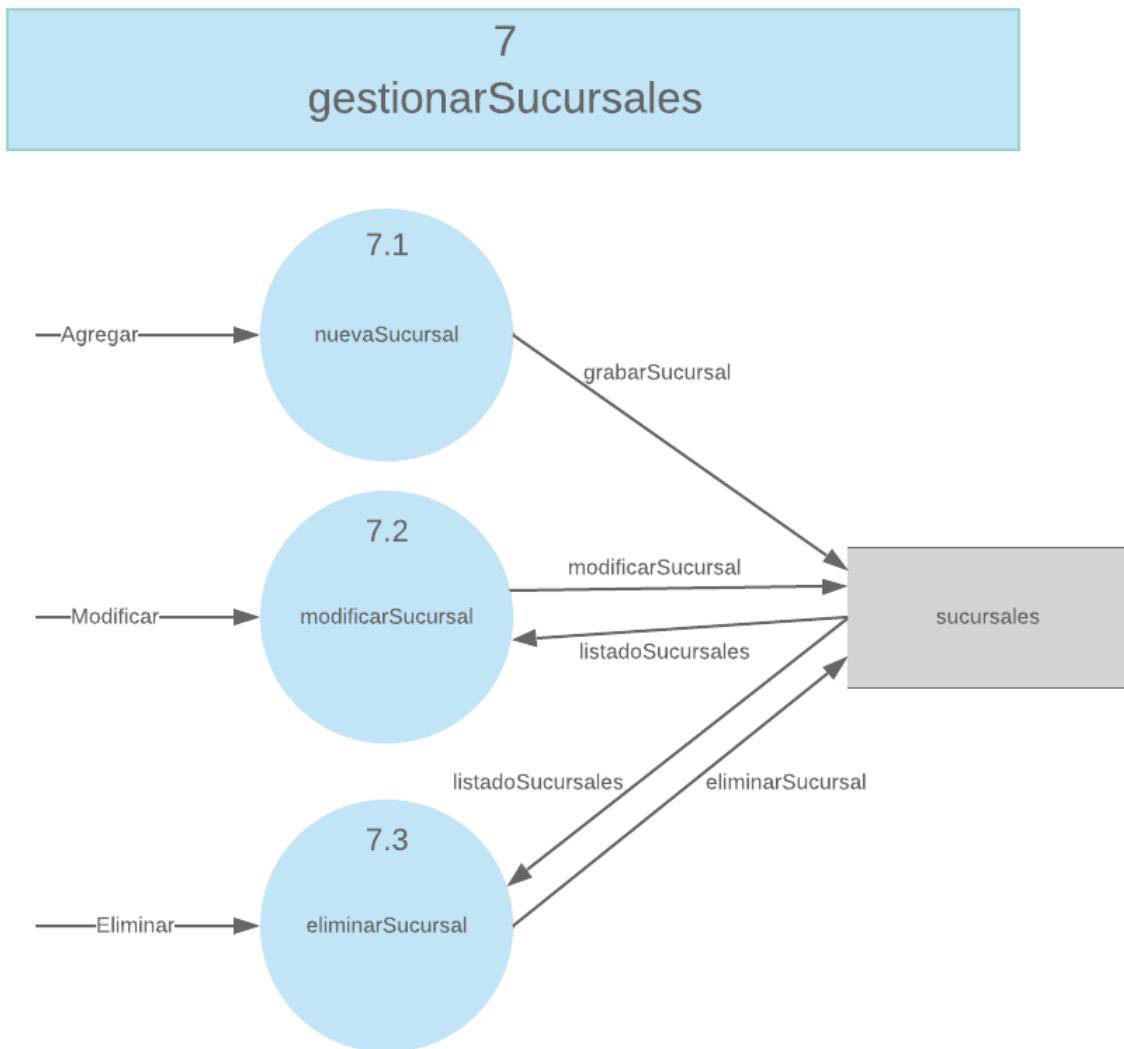
* 4.1.2.6.2 D.F.D. Nivel 2 Usuarios: Modificar usuarios (6.2)



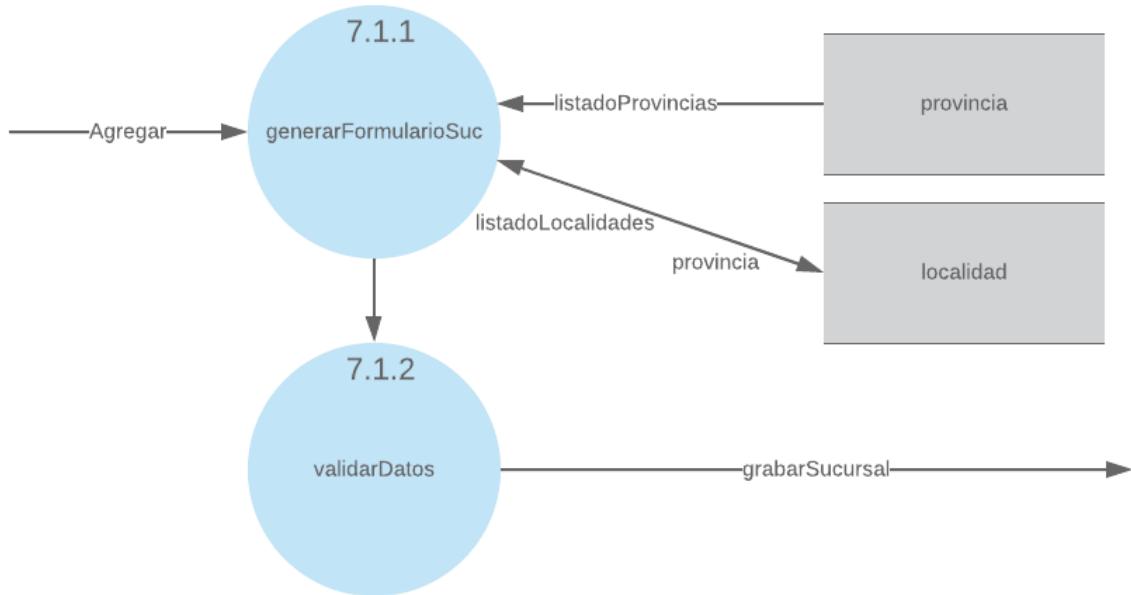
* 4.1.2.6.3 D.F.D. Nivel 2 Usuarios: Consultar usuarios (6.3)



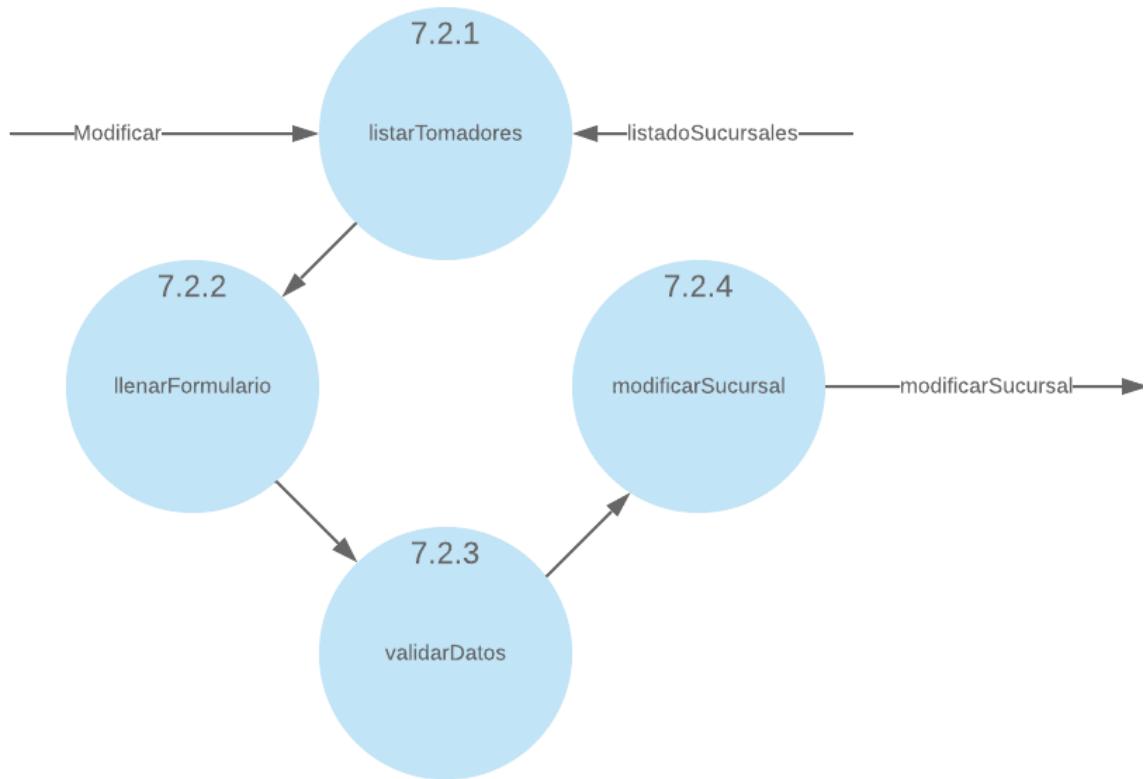
➔ 4.1.2.7 D.F.D. Nivel 1 Sucursales (7)



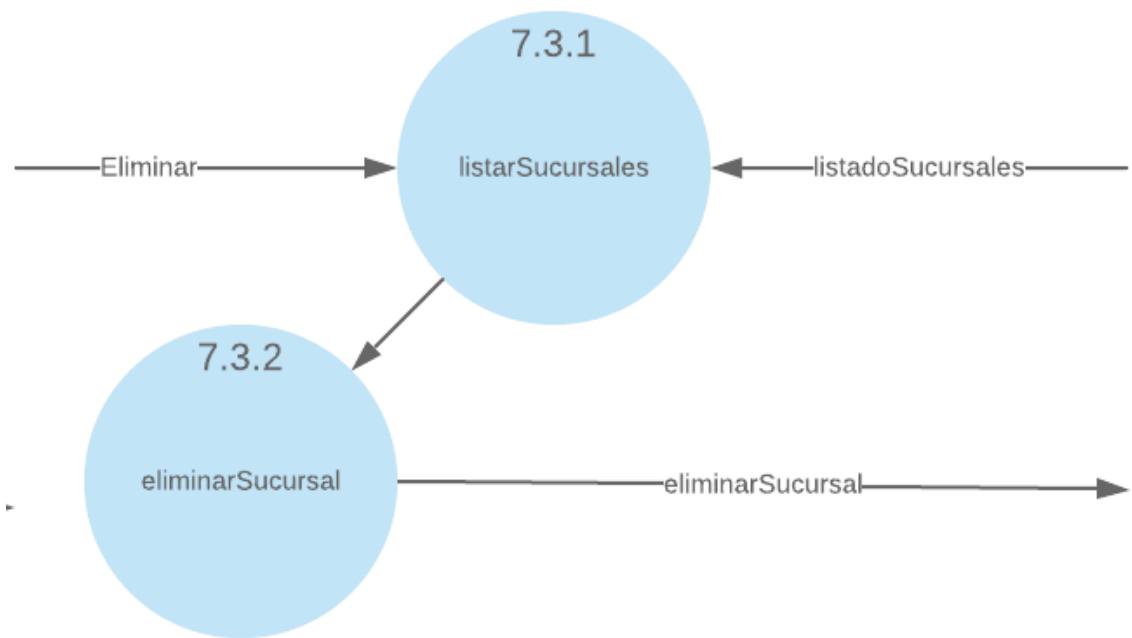
* 4.1.2.7.1 D.F.D. Nivel 2 Sucursales: Agregar sucursales (7.1)



* 4.1.2.7.2 D.F.D. Nivel 2 Sucursales: Modificar sucursales
(7.2)

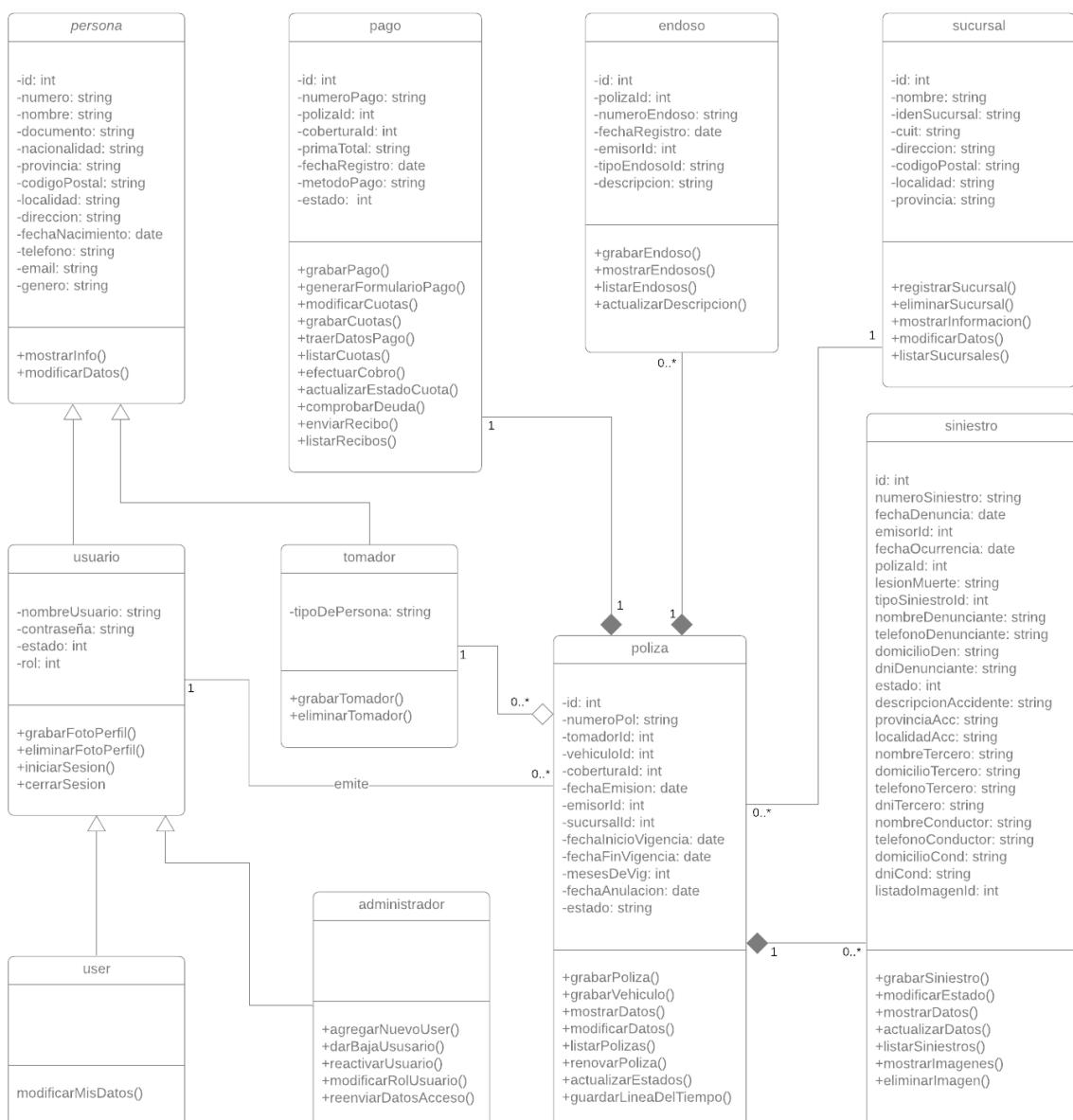


* 4.1.2.7.3 D.F.D. Nivel 2 Sucursales: Eliminar sucursales
(7.3)



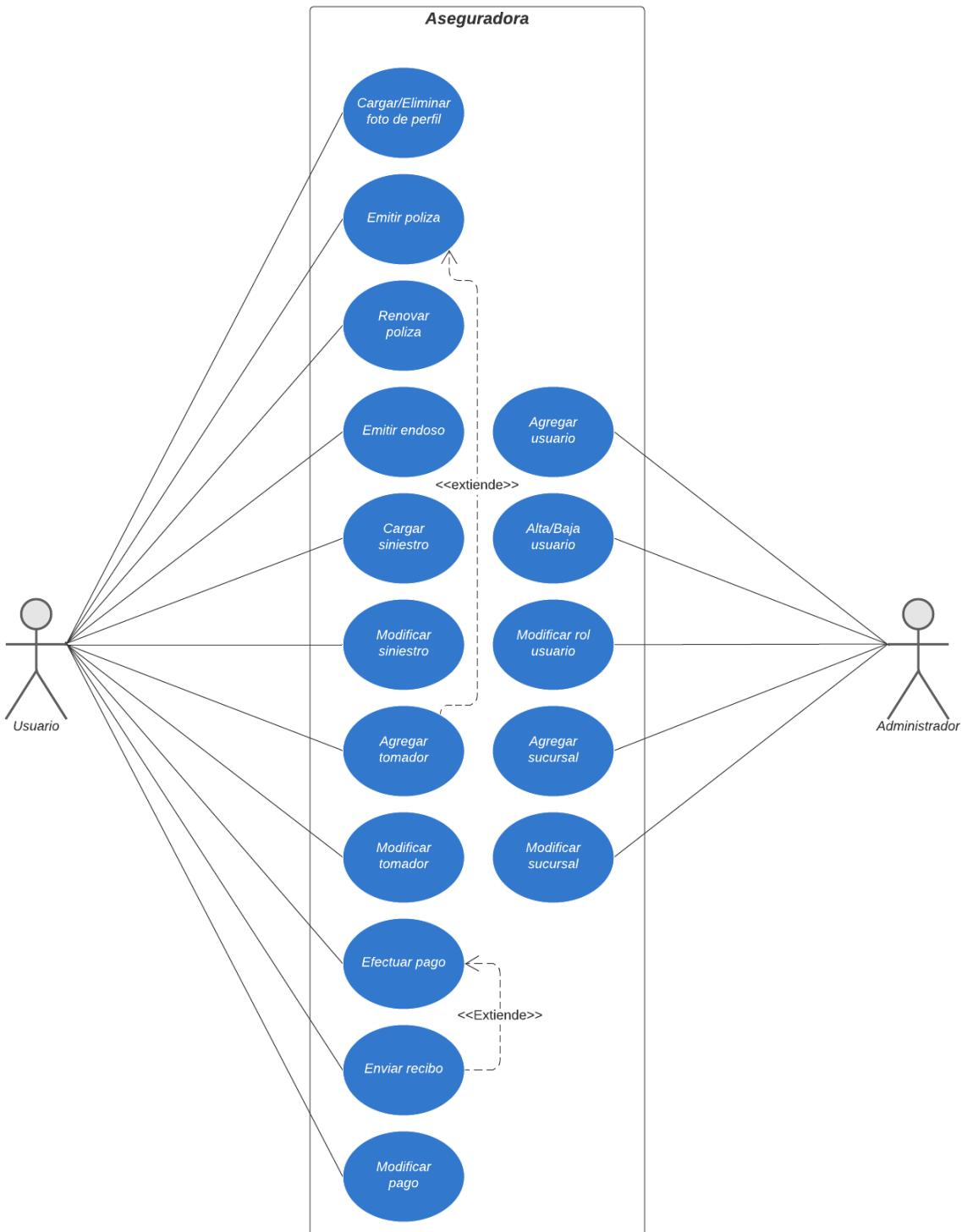
4.1.3 Diagrama de clases

Un diagrama de clases nos permitirá representar gráficamente y de manera estática la estructura general de un sistema, mostrando cada una de las clases y sus interacciones (como herencias, asociaciones, etc.), representadas en forma de bloques, los cuales son unidos mediante líneas y arcos. Los diagramas de clases son ampliamente utilizados tanto para análisis como para diseño de sistemas y software en general.



4.1.4 Diagrama de casos de uso (C.U)

El modelo de casos de uso captura los requisitos de un sistema. Los casos de uso son un medio de comunicación con los usuarios y otros interesados acerca de lo que se piensa hacer del sistema.

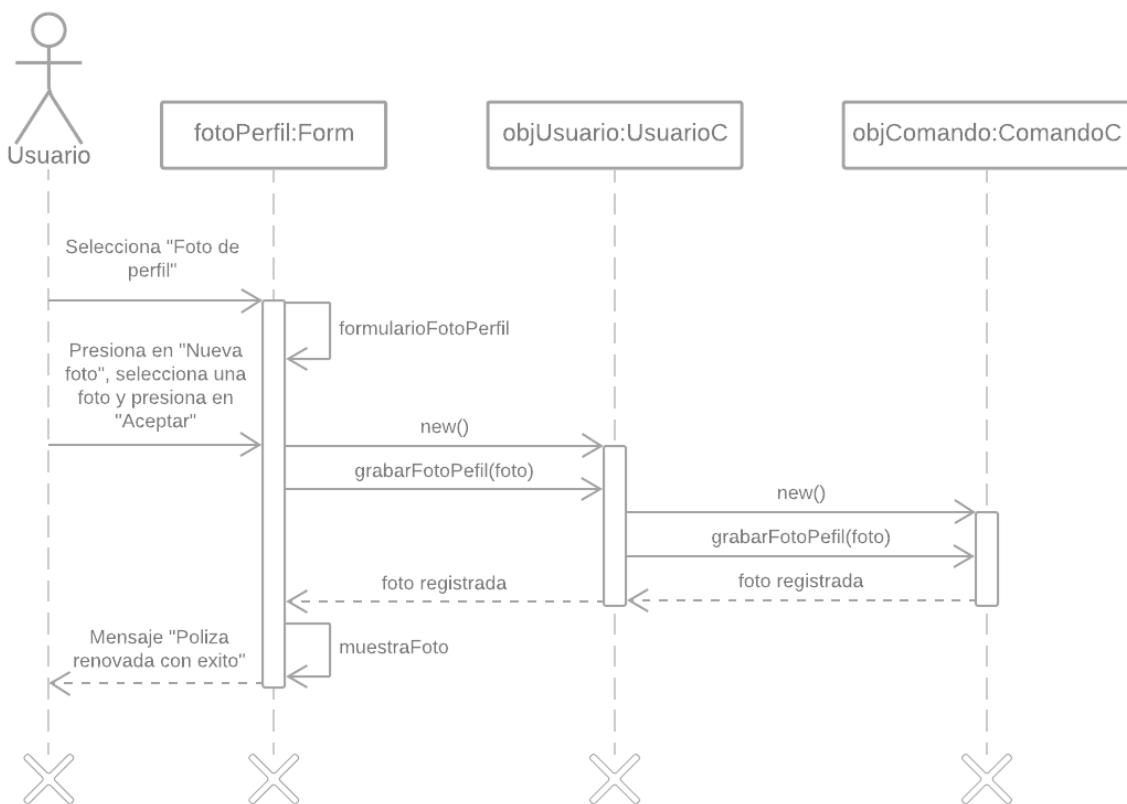


4.1.5 Diagrama de secuencias

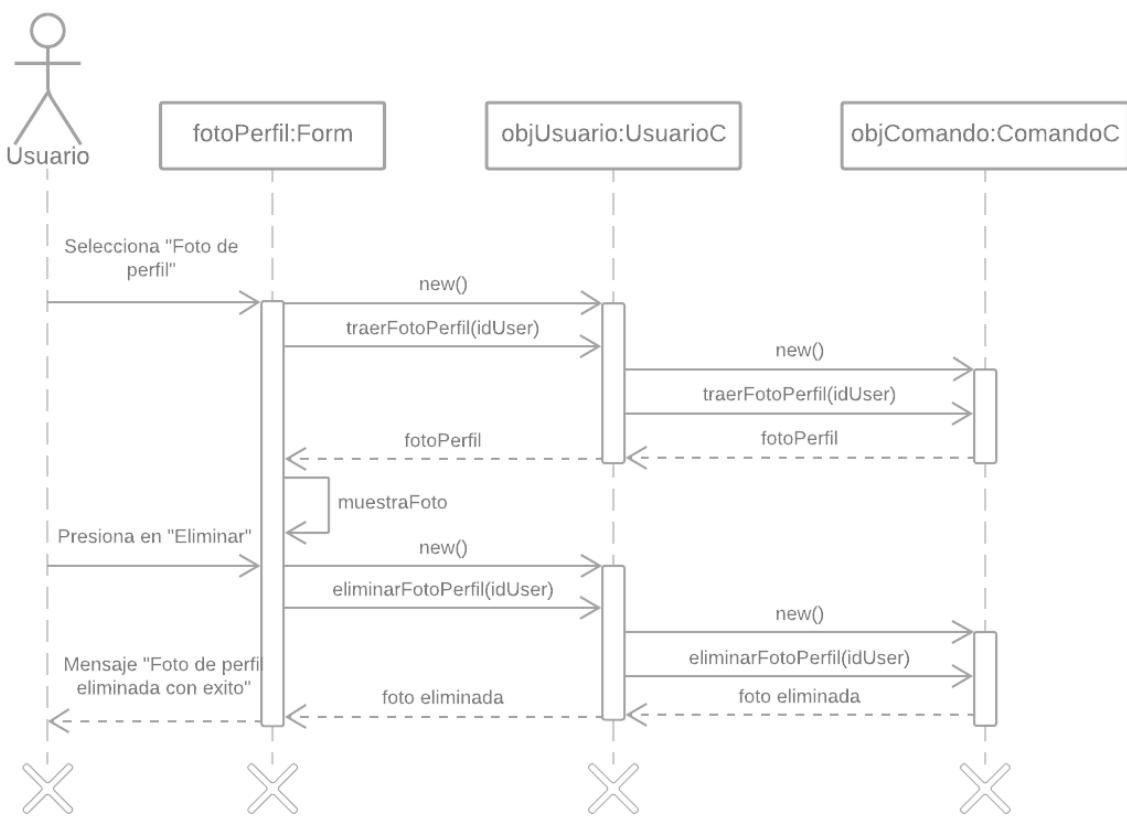
Los diagramas de secuencias describen el comportamiento del sistema y las operaciones que se realizan representando los objetos y los mensajes que se intercambian. Como se mencionó en el apartado 4.1.3 (Diagramas de clases) los diagramas de clases representan información estática en un sistema, pero ya en un sistema funcional, los objetos interactúan entre sí con el tiempo, esto se puede representar mediante un diagrama de secuencias, donde muestra la interacción de un conjunto de objetos en una aplicación a través del tiempo y **se modela para cada caso de uso** (Sección 4.1.4 - Diagrama de casos de uso), esto facilita como se distribuyen las tareas entre los componentes.

⇒ 4.1.5.1 Diagrama de secuencia para C.U. cargar/eliminar foto de perfil

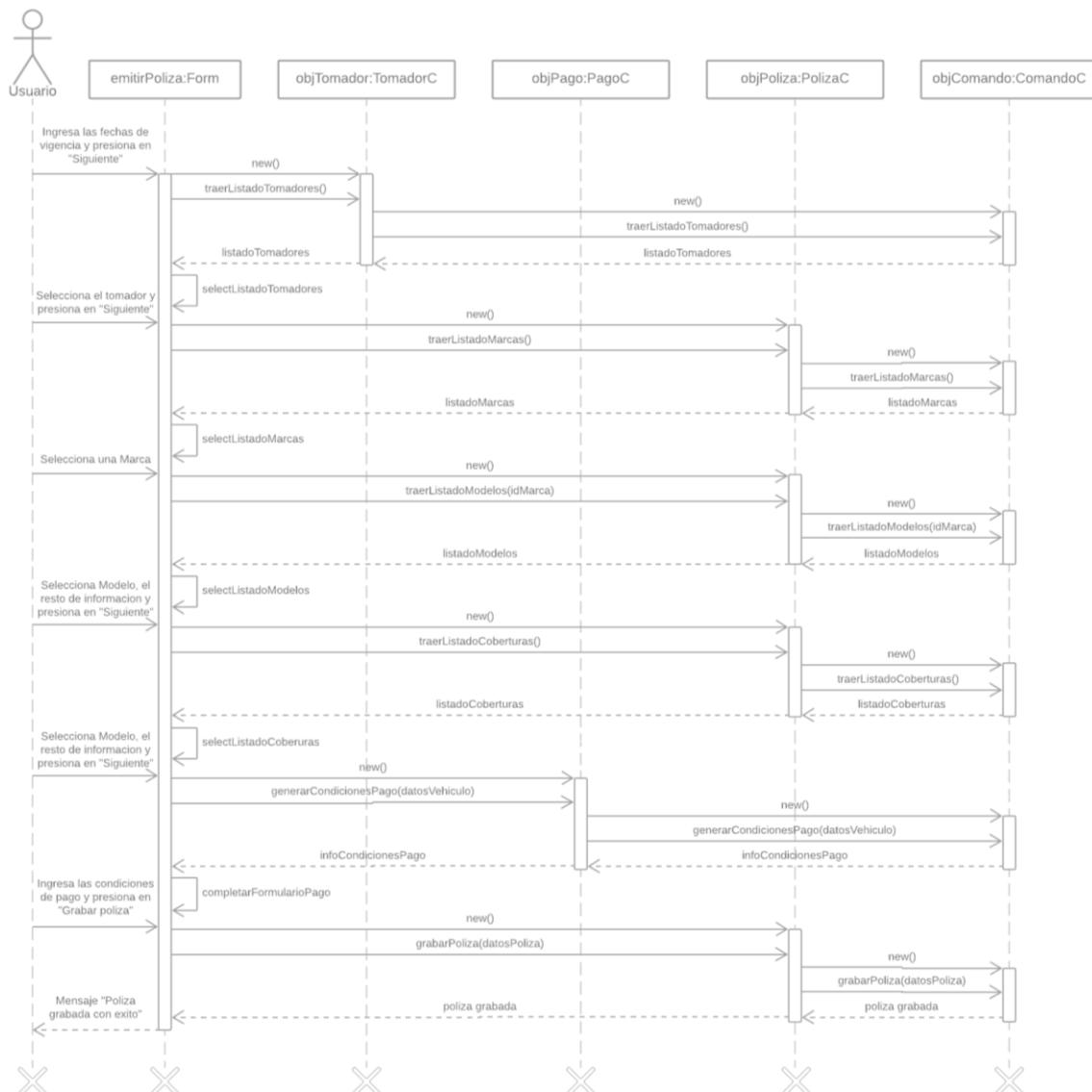
* 4.1.5.1.1 Cargar foto de perfil



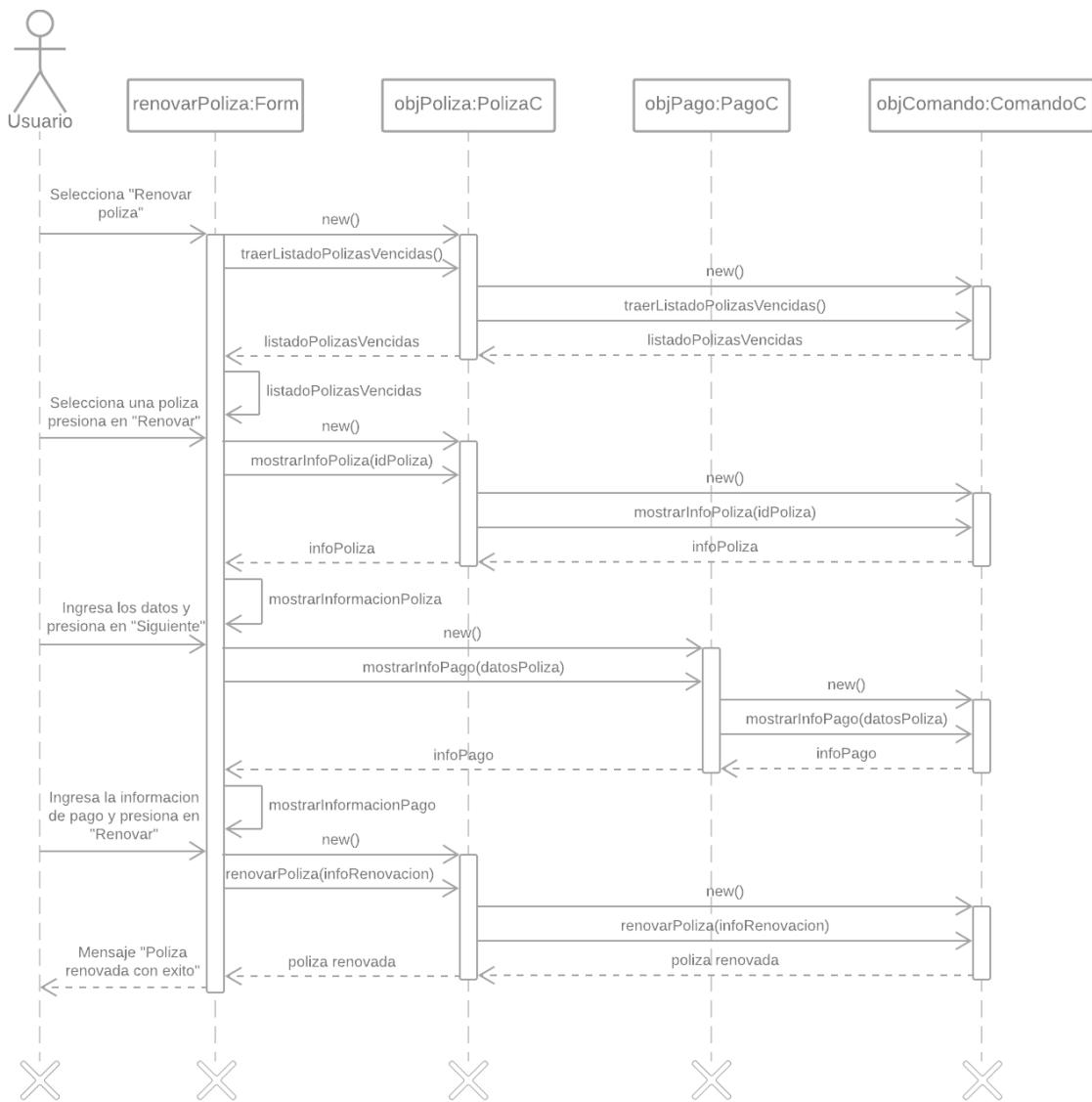
* 4.1.5.1.2 Eliminar foto de perfil



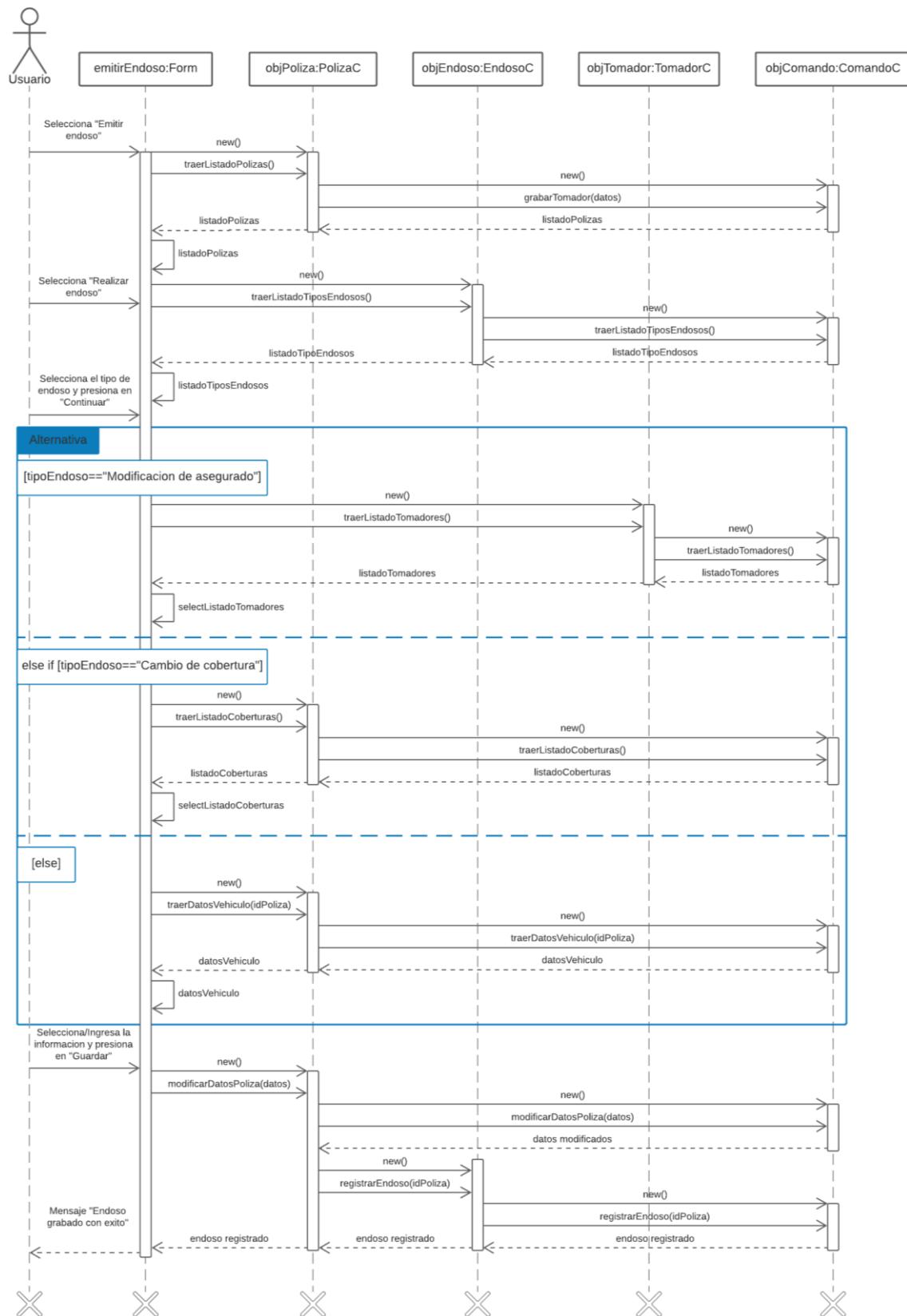
➡ 4.1.5.2 Diagrama de secuencia para C.U emitir póliza:



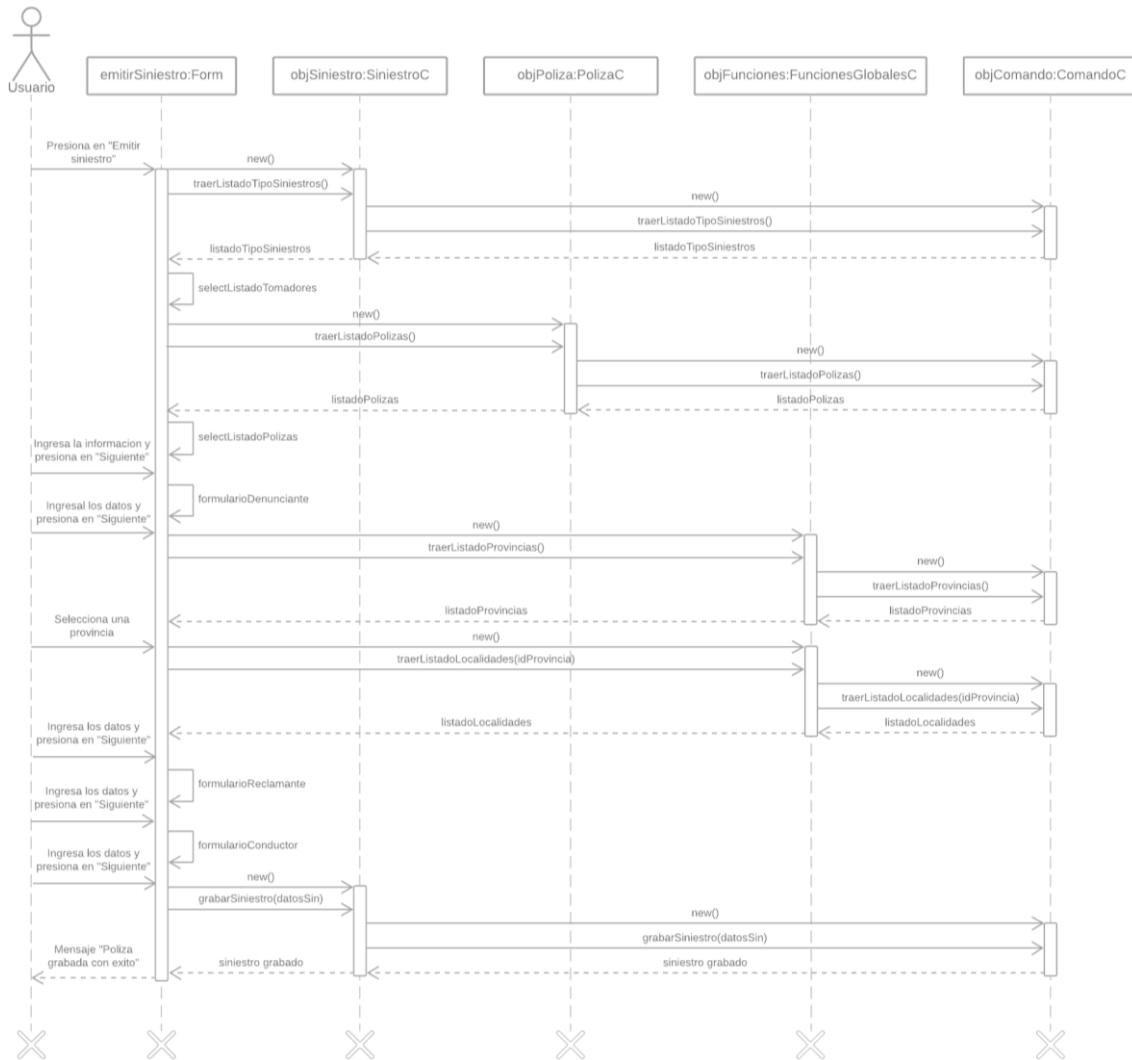
⇒ 4.1.5.3 Diagrama de secuencia para C.U renovar póliza:



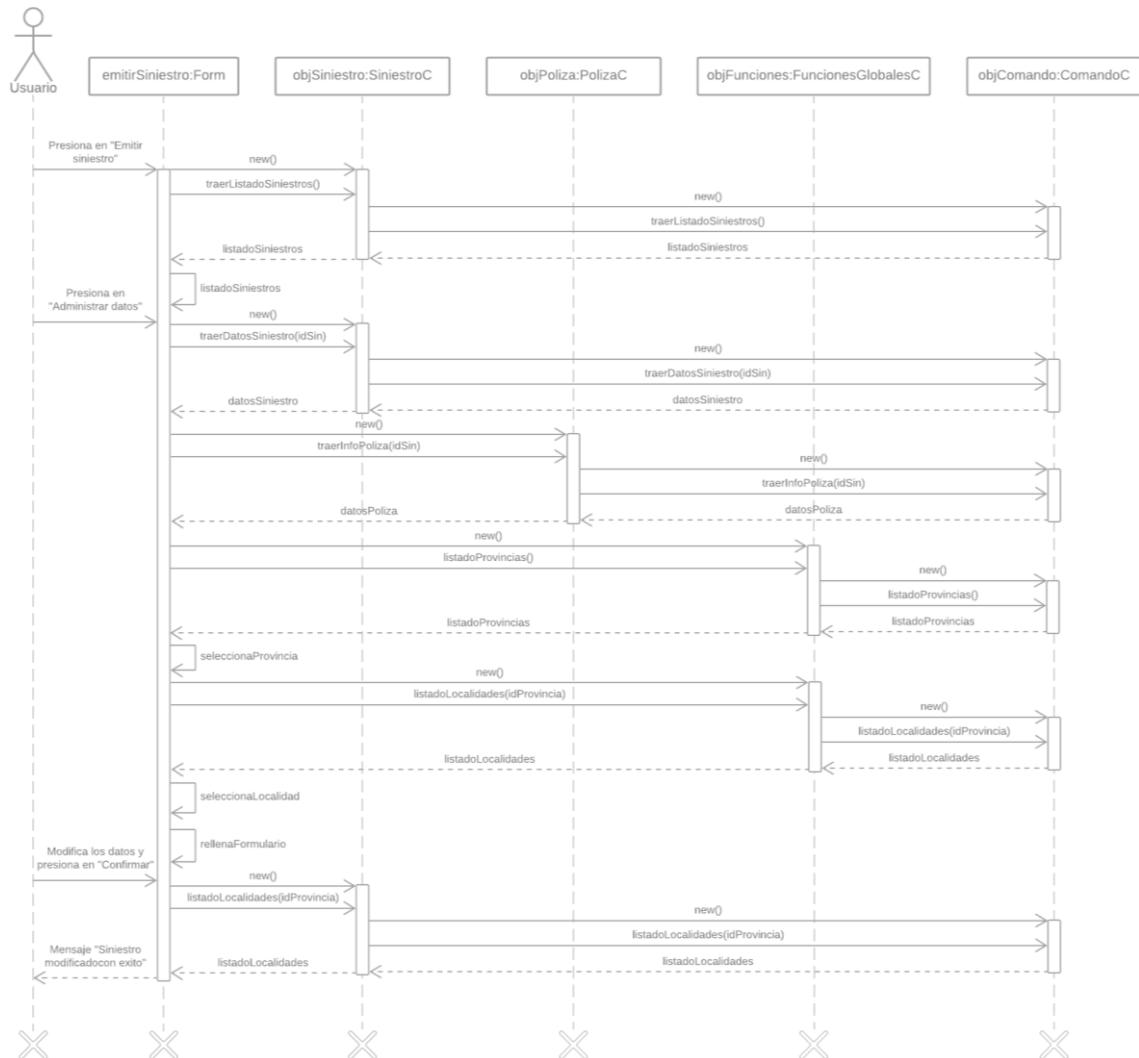
➡ 4.1.5.4 Diagrama de secuencia para C.U emitir endoso:



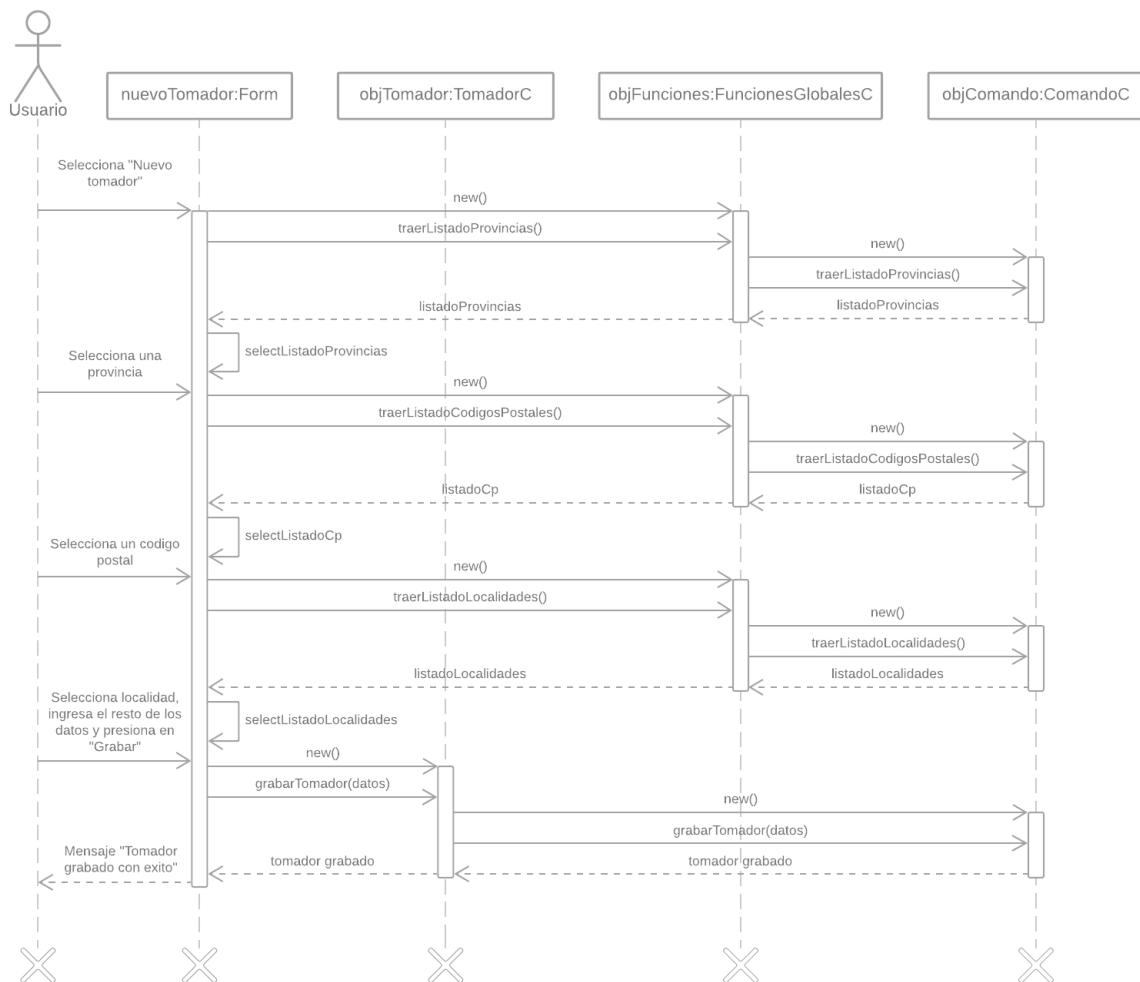
⇒ 4.1.5.5 Diagrama de secuencia para C.U cargar siniestro:



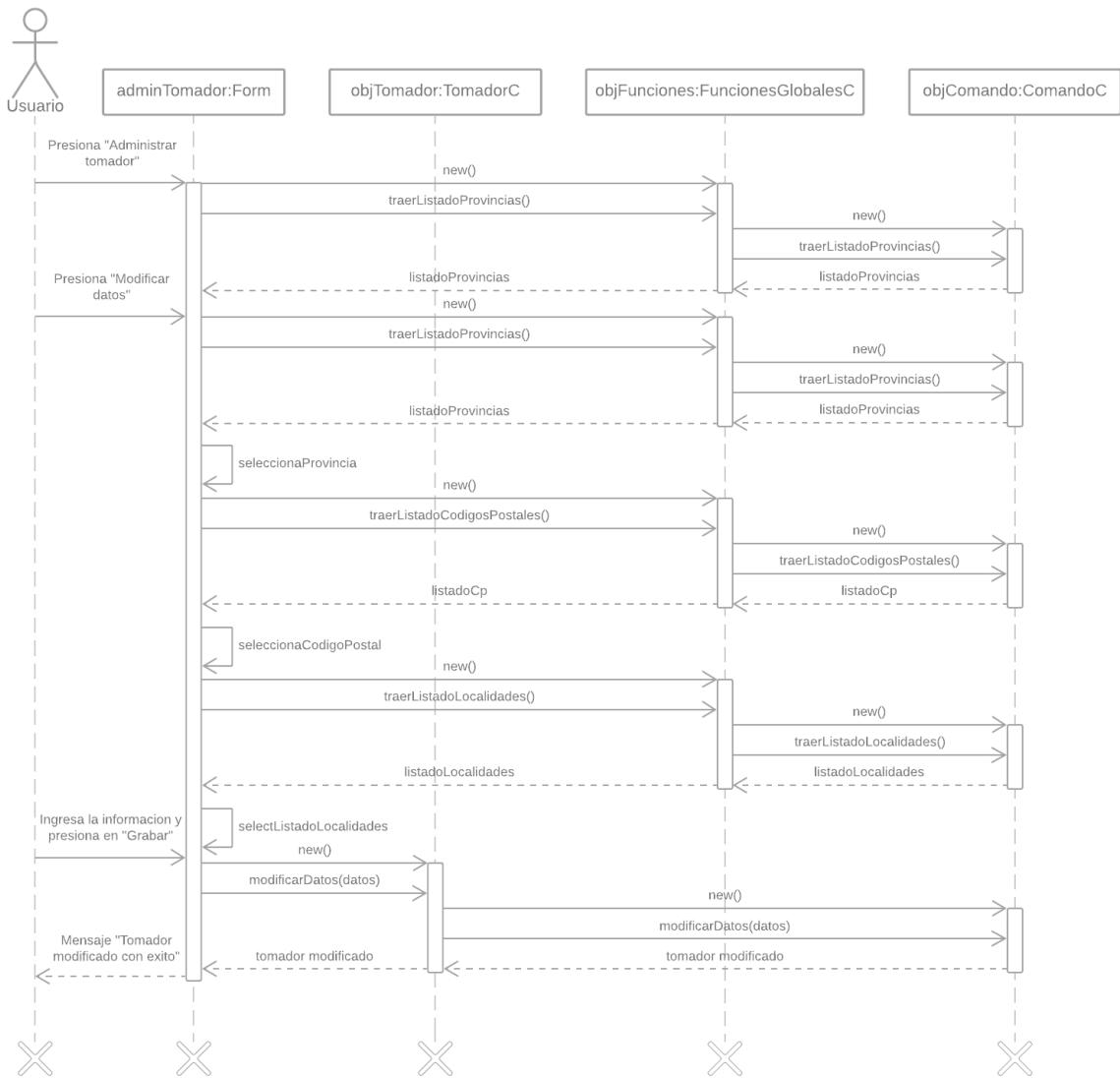
⇒ 4.1.5.6 Diagrama de secuencia para C.U modificar siniestro:



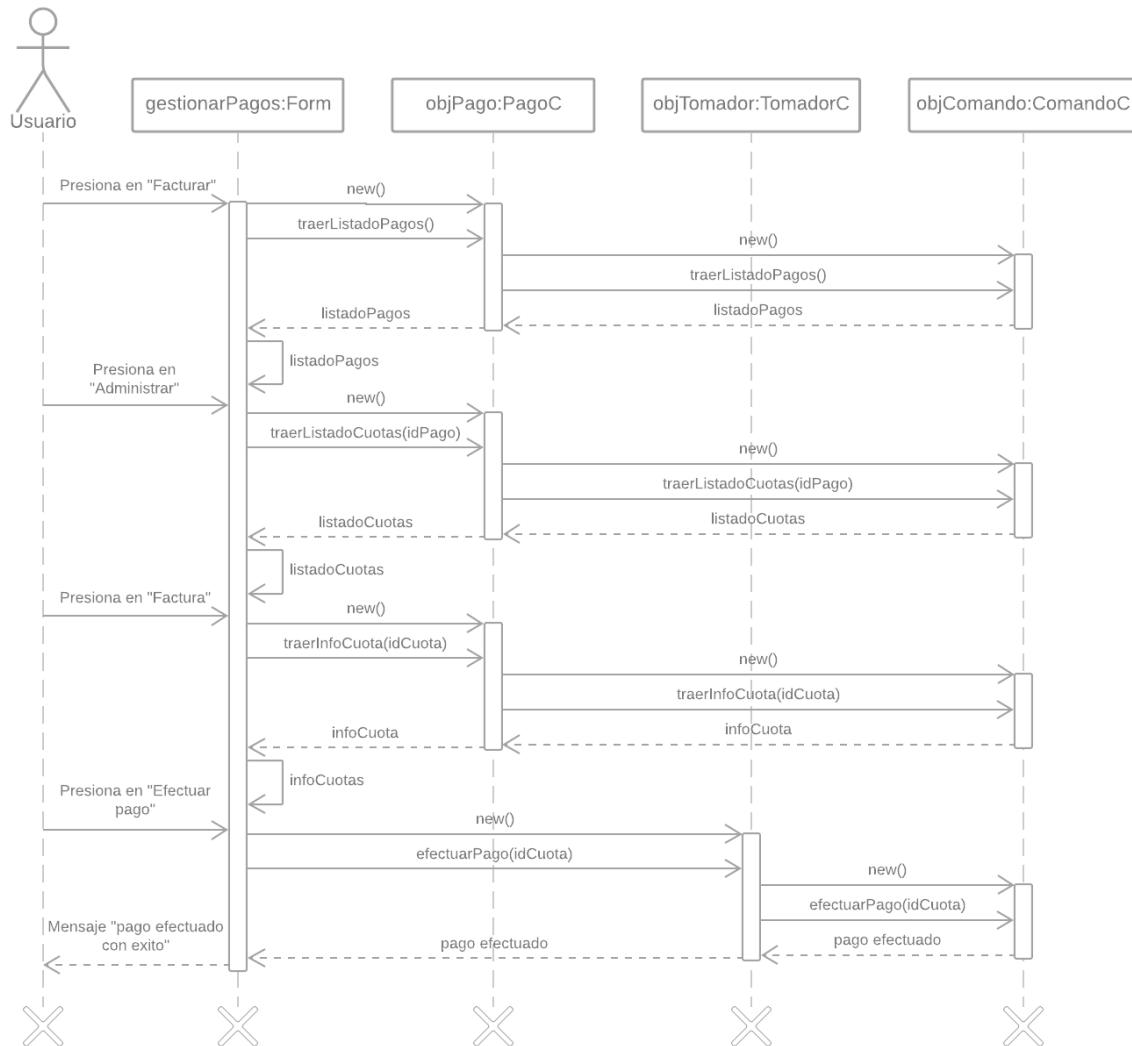
⇒ 4.1.5.7 Diagrama de secuencia para C.U agregar tomador:



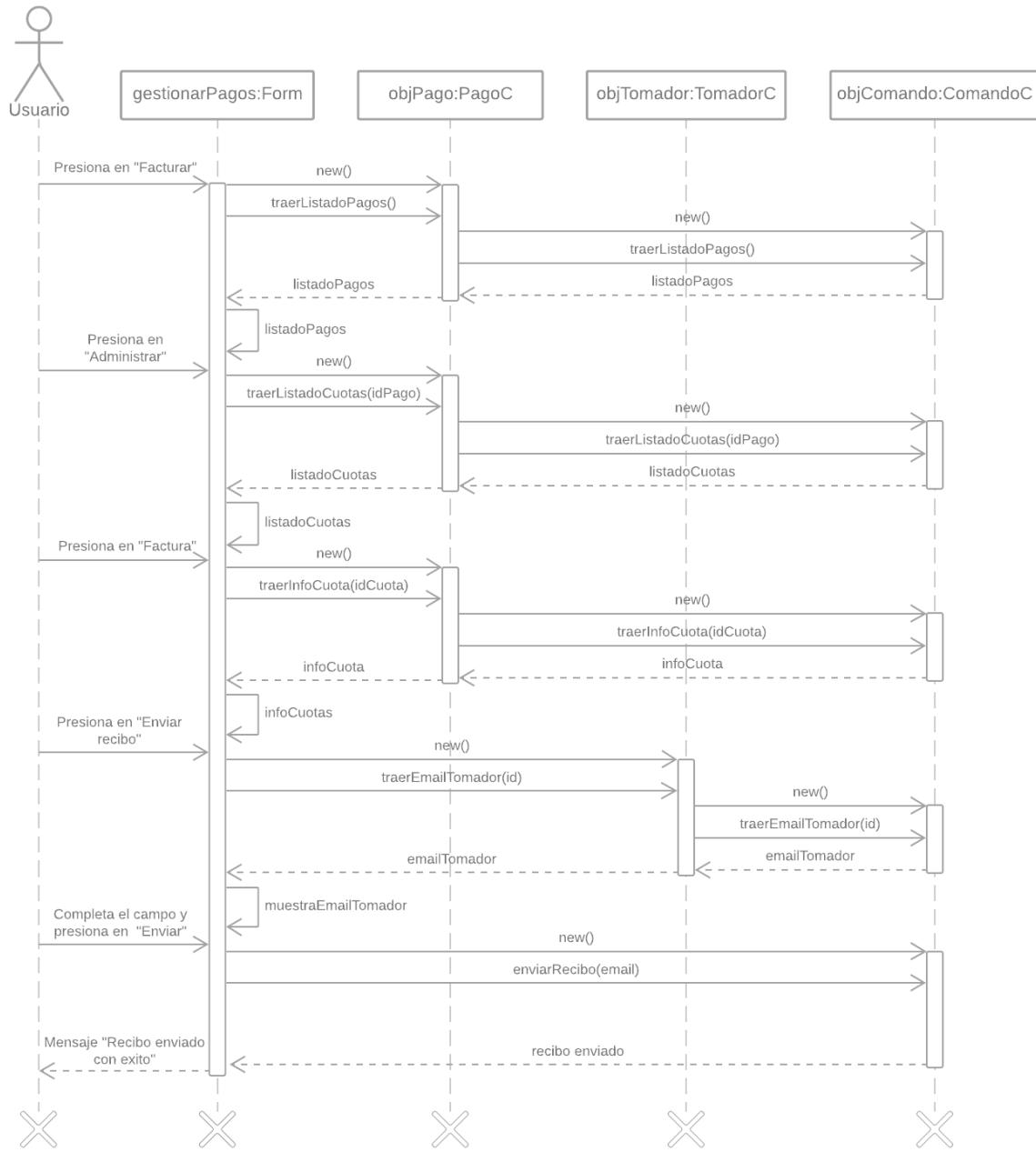
► 4.1.5.8 Diagrama de secuencia para C.U modificar tomador:



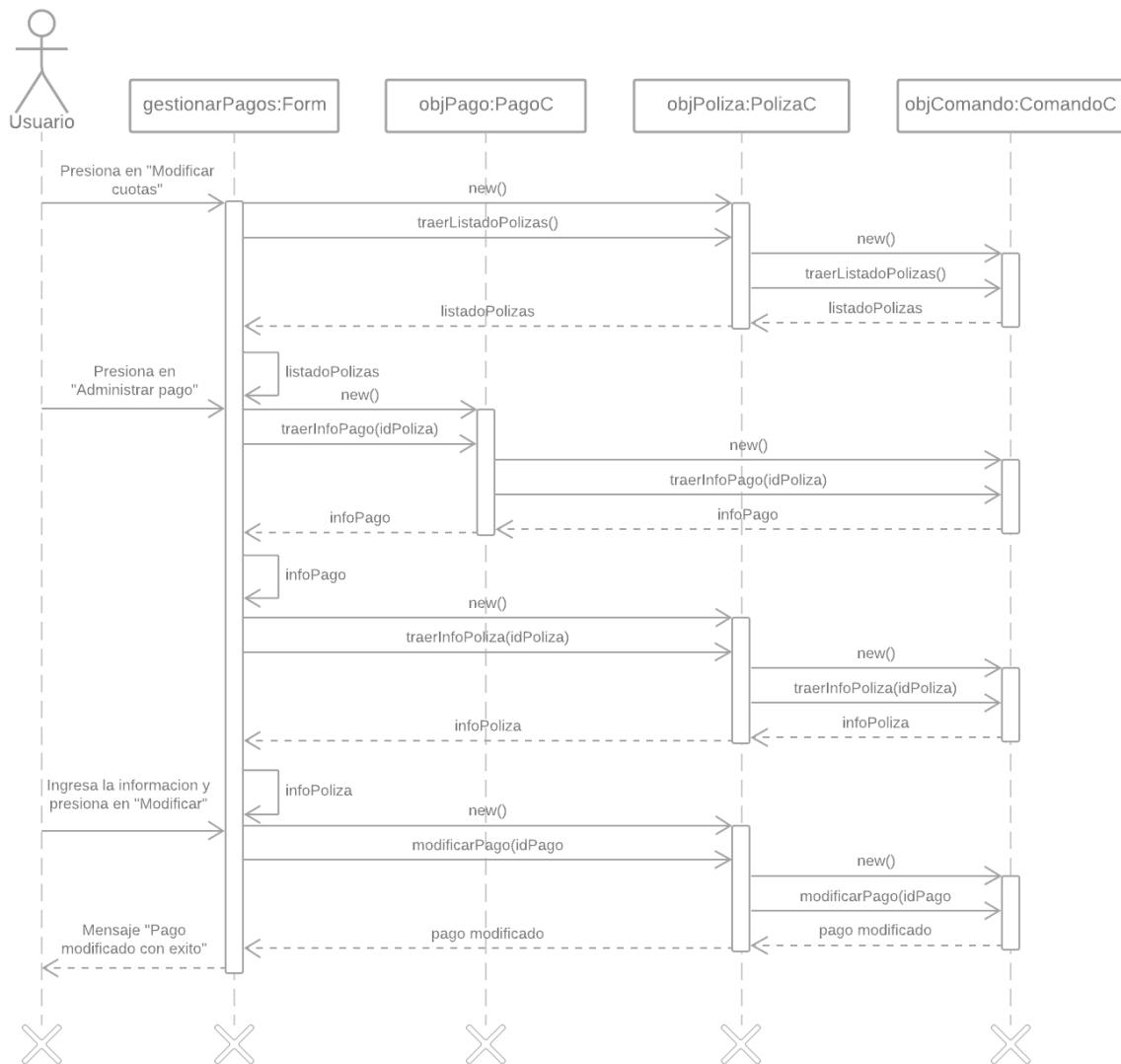
⇒ 4.1.5.9 Diagrama de secuencia para C.U efectuar pago:



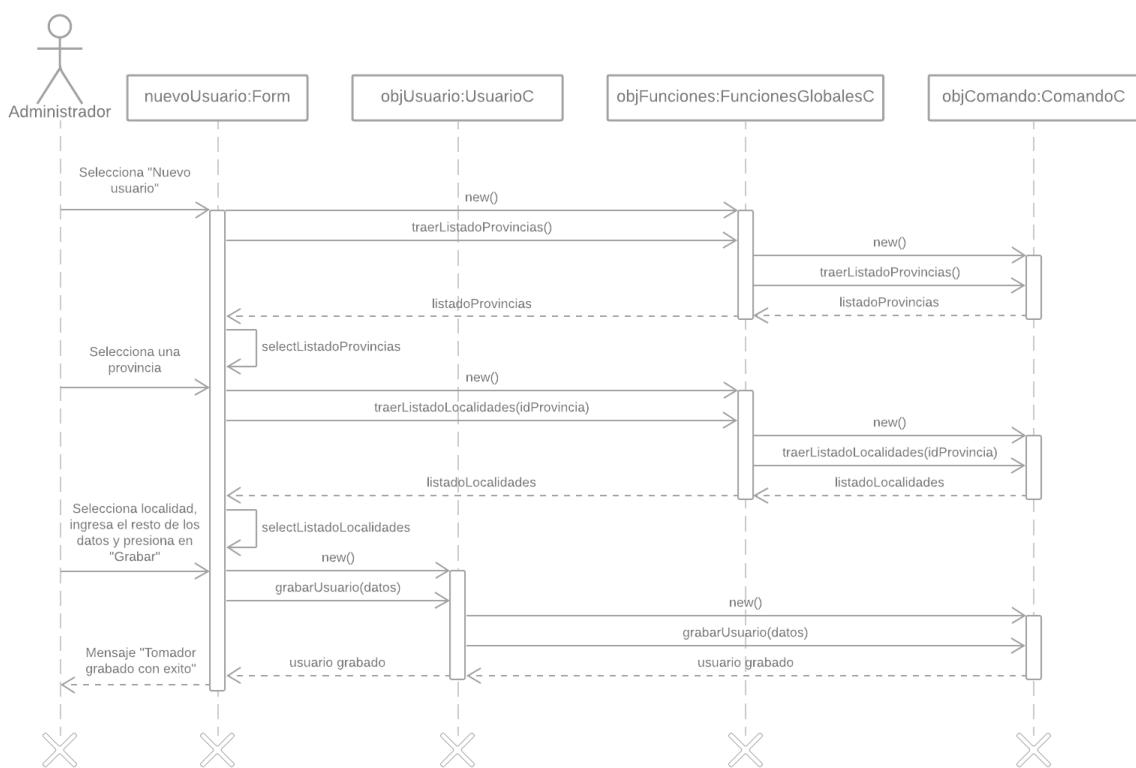
⇒ 4.1.5.10 Diagrama de secuencia para C.U enviar recibo:



➡ 4.1.5.11 Diagrama de secuencia para C.U modificar pago:

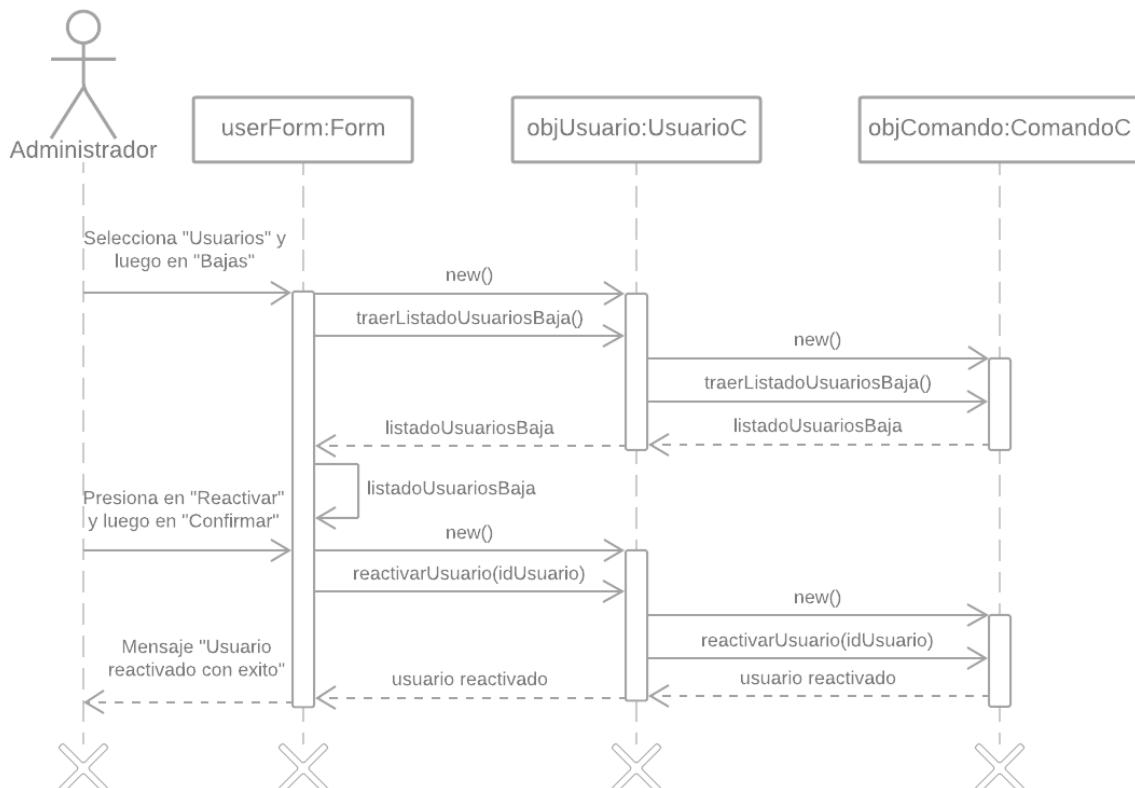


⇒ 4.1.5.12 Diagrama de secuencia para C.U agregar usuario:

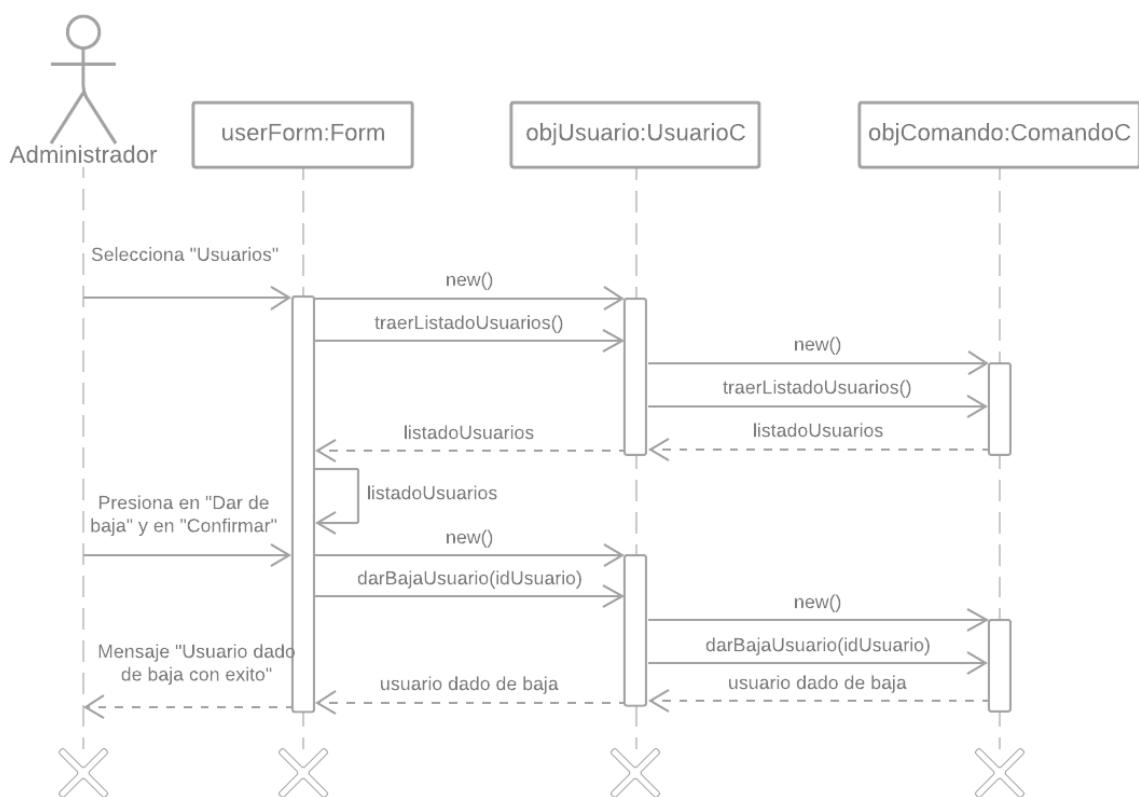


⇒ 4.1.5.13 Diagrama de secuencia para C.U alta/baja usuario:

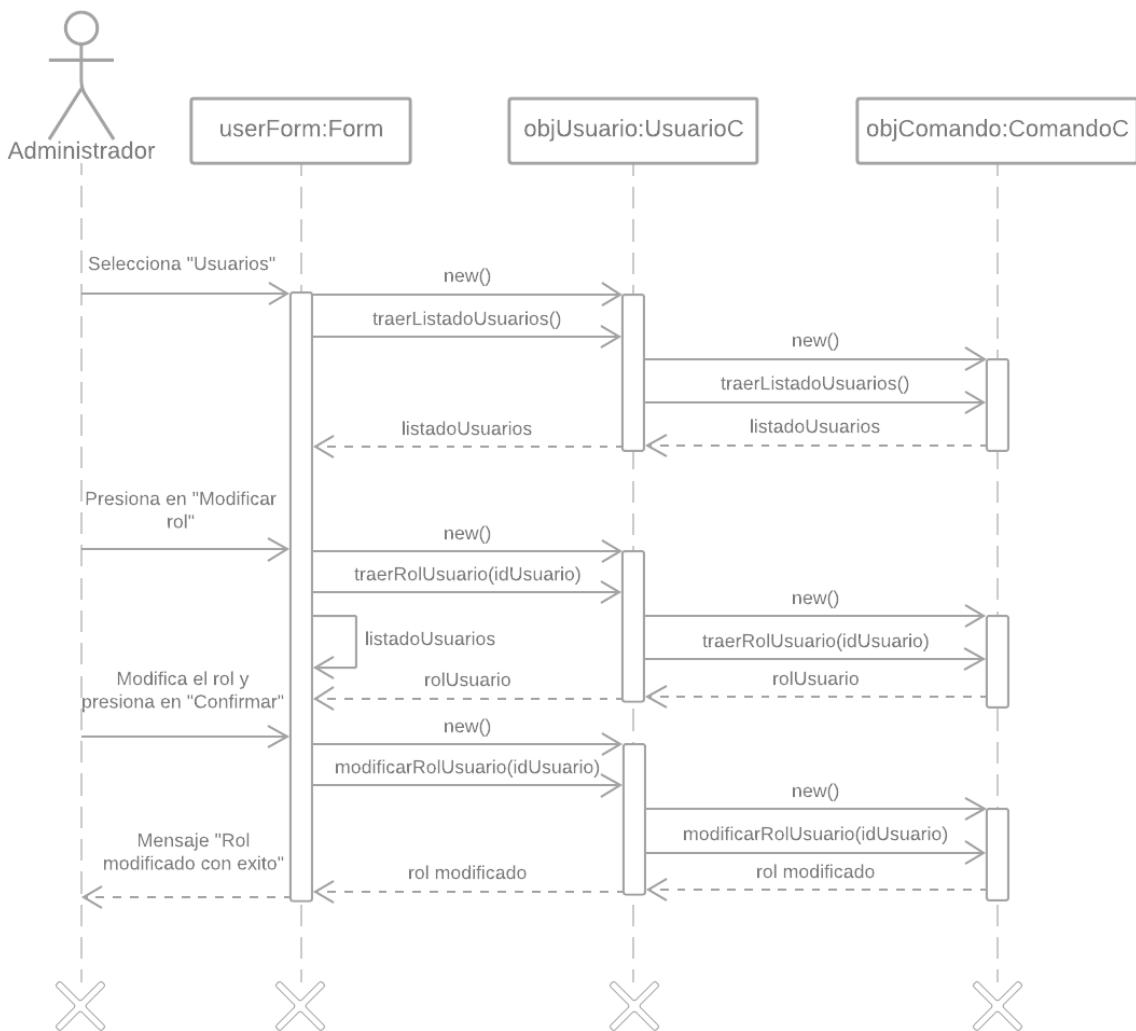
* 4.1.5.13.1 Dar de alta usuario



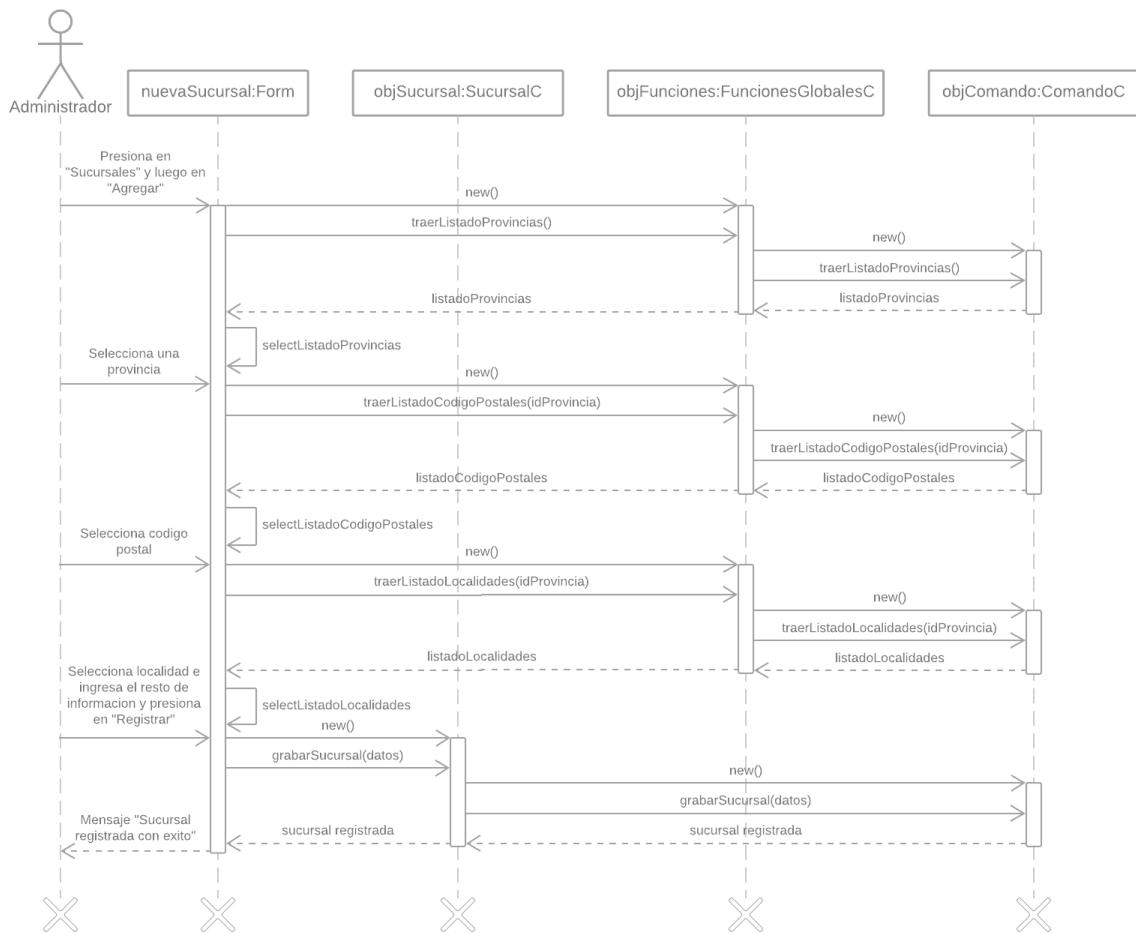
* 4.1.5.13.2 Dar de baja usuario



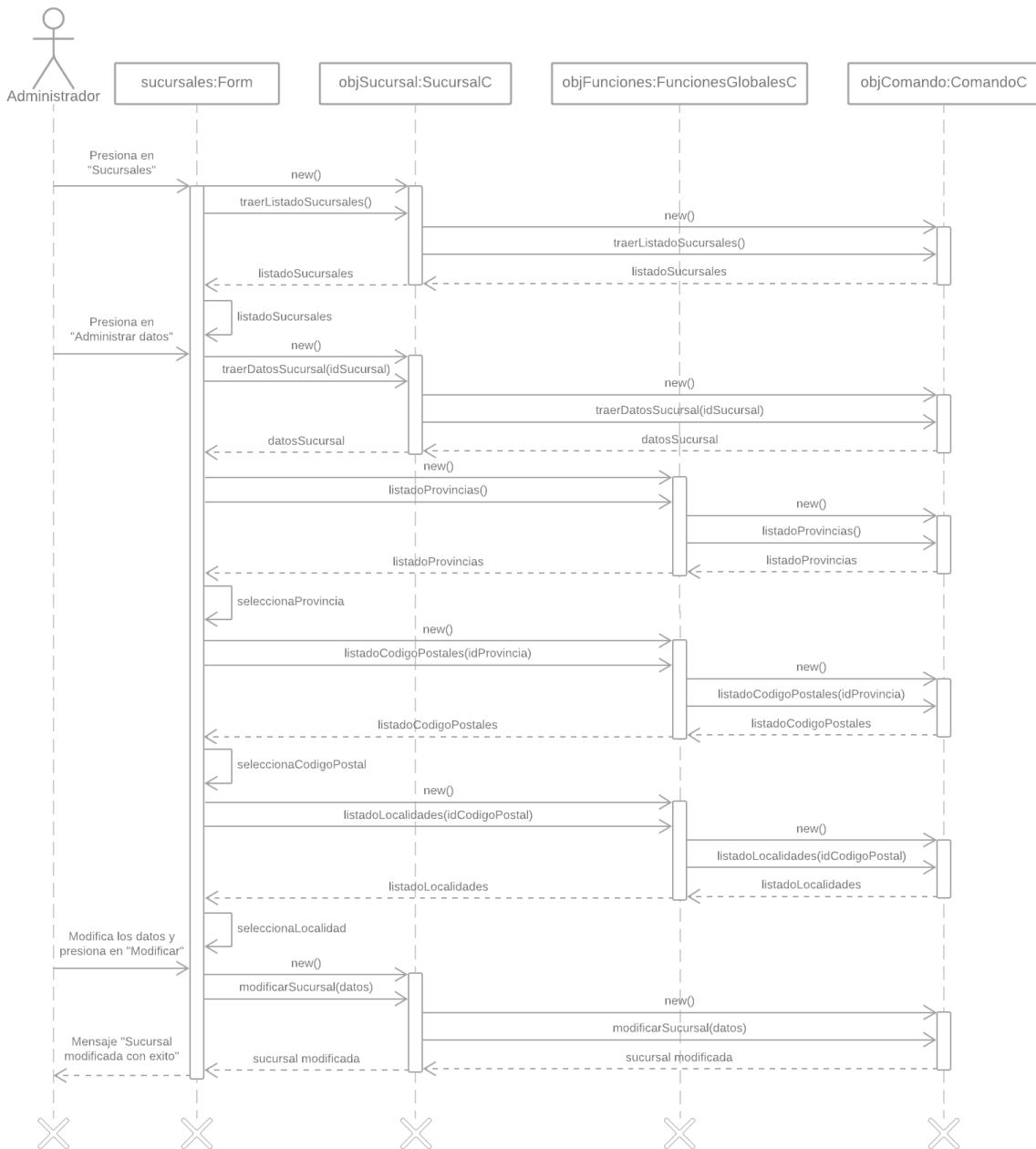
⇒ 4.1.5.14 Diagrama de secuencia para C.U modificar rol usuario:



➡ 4.1.5.15 Diagrama de secuencia para C.U agregar sucursal:



⇒ 4.1.5.16 Diagrama de secuencia para C.U modificar sucursal:



➤ 4.2 Listado de tablas de la base de datos

En esta sección es donde se realizará un listado de las tablas relacionales las cuales componen la base de datos con la que interactúa el sistema de aseguradora, también se agregará junto a cada una de ellas, una descripción con la finalidad de poder especificar detalladamente las características con las que están compuestas, cuál es su rol y cuál es la manera en que funcionan.

Las representaciones de cada una de las tablas corresponden a la base de datos denominada “aseguradora”, accedemos y mostramos su información desde el software administrador “phpmyadmin”, y estarán listadas en orden alfabético.

✳ 4.2.1 clase_vehiculo

Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
id	int(11)			No	Ninguna		AUTO_INCREMENT	Cambiar Eliminar Más
nombre	varchar(100)	utf8_general_ci		No	Ninguna			Cambiar Eliminar Más

En esta tabla se almacenan todos los registros relacionados a las clases de vehículos disponibles para seleccionar. Los campos son el id (Llave primaria) y el nombre.

✳ 4.2.2 cobertura

Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
id	int(11)			No	Ninguna		AUTO_INCREMENT	Cambiar Eliminar Más
nombre	varchar(11)	utf8_general_ci		No	Ninguna			Cambiar Eliminar Más
descripcion	varchar(255)	utf8_general_ci		No	Ninguna			Cambiar Eliminar Más
informacion	text	utf8_general_ci		No	Ninguna			Cambiar Eliminar Más

Aquí se almacenan todos los registros relacionados a las coberturas de riesgo de las pólizas disponibles para seleccionar. Los campos se representan con un id (Llave primaria), el nombre, la descripción y la información de la cobertura.

4.2.3 cobertura_adicional

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
1	id	int(10)			No	Ninguna		AUTO_INCREMENT	Cambiar Eliminar Más
2	nombre	varchar(30)	utf8_general_ci		No	Ninguna			Cambiar Eliminar Más

Aquí se almacenan todos los registros relacionados a los tipos de coberturas adicionales para agregar en un vehículo. Almacena el id (Llave primaria) y el nombre descriptivo.

4.2.4 cuotas

Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
id	int(10)			No	Ninguna		AUTO_INCREMENT	Cambiar Eliminar Más
numero_cuota	varchar(20)	utf8_general_ci		No	Ninguna			Cambiar Eliminar Más
iden_cuota	varchar(20)	utf8_general_ci		No	Ninguna			Cambiar Eliminar Más
pago_id	int(10)			No	Ninguna			Cambiar Eliminar Más
prima_mensual	varchar(10)	utf8_general_ci		No	Ninguna			Cambiar Eliminar Más
lapso	varchar(50)	utf8_general_ci		No	Ninguna			Cambiar Eliminar Más
dia_cobrar	varchar(10)	utf8_general_ci		No	Ninguna			Cambiar Eliminar Más
fecha_cobro	varchar(50)	utf8_general_ci		Sí	NULL			Cambiar Eliminar Más
vto_pago	varchar(20)	utf8_general_ci		No	Ninguna			Cambiar Eliminar Más
estado	int(10)			No	Ninguna			Cambiar Eliminar Más

Aquí se almacenan todos los registros relacionados a las coberturas de riesgo de las pólizas disponibles para seleccionar. Almacena el id (Llave primaria), el número de cuota, el número único identificador de la cuota, el id del pago al que corresponde la cuota (Llave foránea de la tabla “pagos” ap. 4.2.16), la prima mensual, las fechas del lapso de vigencia de la cuota, la fecha de facturación, la fecha de vencimiento del lapso y el id del estado actual en el que se encuentra (Llave foránea de la tabla “estado_cuota” ap. 4.2.8).

4.2.5 endoso

Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
id	int(50)			No	Ninguna		AUTO_INCREMENT	Cambiar Eliminar Más
polizaid	int(50)			No	Ninguna			Cambiar Eliminar Más
endosoNum	varchar(15)	utf8_general_ci		No	Ninguna			Cambiar Eliminar Más
polizaNum	varchar(50)	utf8_general_ci		No	Ninguna			Cambiar Eliminar Más
fechaRegistroEnd	varchar(10)	utf8_general_ci		No	Ninguna			Cambiar Eliminar Más
horaRegistroEnd	varchar(10)	utf8_general_ci		No	Ninguna			Cambiar Eliminar Más
emisor	int(10)			No	Ninguna			Cambiar Eliminar Más
tipo	int(20)			No	Ninguna			Cambiar Eliminar Más
descripcion	varchar(250)	utf8_general_ci		No	Ninguna			Cambiar Eliminar Más

Aquí se almacenan todos los registros cuya información corresponde a un endoso particular de una poliza, almacenando el id (Llave primaria), el id de la poliza correspondiente (Llave foránea de la tabla “poliza” ap. 4.2.17), el número del endoso, el número de la poliza correspondiente, la fecha en que se grabó ese endoso, la hora en que se grabó, el id del usuario quien realizo el registro, el tipo de endoso (Llave foránea de la tabla “tipo_endoso” ap. 4.2.22), y la descripción.

4.2.6 estadopoliza

Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
id	int(50)			No	Ninguna		AUTO_INCREMENT	Cambiar Eliminar Más
nombre	varchar(100)	utf8_general_ci		No	Ninguna			Cambiar Eliminar Más
descripcion	text	utf8_general_ci		No	Ninguna			Cambiar Eliminar Más

Aquí se almacenan todos los registros cuya información corresponde a todos los estados por los que puede pasar una poliza.

4.2.7 estadosiniestro

Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
id	int(100)			No	Ninguna		AUTO_INCREMENT	Cambiar Eliminar Más
nombre	varchar(100)	utf8_general_ci		No	Ninguna			Cambiar Eliminar Más

Aquí se almacenan todos los registros cuya información corresponde a todos los tipos de estados por los que puede pasar un siniestro.

✳ 4.2.8 estado_cuota

Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
id	int(10)			No	Ninguna		AUTO_INCREMENT	Cambiar Eliminar Más
nombre	varchar(30)	utf8_general_ci		No	Ninguna			Cambiar Eliminar Más

Aquí se almacenan todos los registros cuya información corresponde a todos los tipos de estados por los que puede pasar una cuota.

✳ 4.2.9 estado_pago

Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
id	int(10)			No	Ninguna		AUTO_INCREMENT	Cambiar Eliminar Más
nombre	varchar(20)	utf8_general_ci		No	Ninguna			Cambiar Eliminar Más

Aquí se almacenan todos los registros cuya información corresponde a todos los tipos de estados por los que puede pasar un pago.

✳ 4.2.10 imgperfilusers

Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
id	int(50)			No	Ninguna		AUTO_INCREMENT	Cambiar Eliminar Más
nombre	varchar(100)	utf8mb4_general_ci		No	Ninguna			Cambiar Eliminar Más
IdUser	int(50)			No	Ninguna			Cambiar Eliminar Más
ruta	varchar(100)	utf8mb4_general_ci		No	Ninguna			Cambiar Eliminar Más

Aquí se almacenan todos los registros cuya información corresponde a todas las imágenes de perfil de los usuarios. Sus campos constan del id (Llave primaria), el nombre del archivo de imagen, el id del usuario al que le corresponde la foto (Llave foránea de la tabla “usuarios” ap. 4.2.25), y la ruta donde se almacena el archivo.

4.2.11 img_siniestro

Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
id 📄	Int(11)			No	Ninguna		AUTO_INCREMENT	Cambiar Eliminar ▾ Más
nombre	varchar(50) utf8_general_ci			No	Ninguna			Cambiar Eliminar ▾ Más
idSiniestro	int(11)			No	Ninguna			Cambiar Eliminar ▾ Más
ubicacion	text	utf8_general_ci		No	Ninguna			Cambiar Eliminar ▾ Más

Aquí se almacenan todos los registros cuya información corresponde a todas las imágenes de los sucesos ocurridos en los siniestros. Sus campos constan del id (Llave primaria), el nombre del archivo de imagen, el id del siniestro al que le corresponde la foto (Llave foránea de la tabla “siniestro” ap. 4.2.19), y la ruta donde se almacena el archivo.

4.2.12 localidad

Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
id 📄	Int(10)		UNSIGNED	No	Ninguna		AUTO_INCREMENT	Cambiar Eliminar ▾ Más
provincia_id 📄	tinyint(3)		UNSIGNED	No	Ninguna			Cambiar Eliminar ▾ Más
nombre	varchar(50) utf8_general_ci			No	Ninguna			Cambiar Eliminar ▾ Más
codigopostal 📄	smallint(10)			No	Ninguna			Cambiar Eliminar ▾ Más

Aquí se almacenan todos los registros cuya información corresponde a todas las localidades disponibles para seleccionar. Sus campos constan del id (Llave primaria), el id de la provincia a la que corresponde la localidad (Llave foránea de la tabla “provincia” ap. 4.2.18), y el número de código postal correspondiente (Llave foránea de la tabla “provincia” ap. 4.2.18, pero para relacionar el campo “codigopostal”).

4.2.13 ltestpol

Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
id 📃	int(11)			No	Ninguna		AUTO_INCREMENT	Cambiar Eliminar ▾ Más
poliza_id	int(10)			No	Ninguna			Cambiar Eliminar ▾ Más
nrohilo	varchar(20) utf8_general_ci			No	Ninguna			Cambiar Eliminar ▾ Más
estado 📃	int(10)			No	Ninguna			Cambiar Eliminar ▾ Más
inicio_lapso	varchar(20) utf8_general_ci			No	Ninguna			Cambiar Eliminar ▾ Más
fin_lapso	varchar(20) utf8_general_ci			Si	NULL			Cambiar Eliminar ▾ Más

Aquí se almacenan todos los registros cuya información corresponde a la línea del tiempo de estados de las pólizas. Sus campos constan del id (Llave primaria), el id de la poliza correspondiente (Llave foránea de la tabla “poliza” ap. 4.2.17), el número de registro, el id del estado correspondiente (Llave foránea de la tabla “estadopoliza” ap. 4.2.6), la fecha de inicio del lapso, y la fecha de finalización del lapso.

4.2.14 marca

Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
id 📃	int(11)			No	Ninguna		AUTO_INCREMENT	Cambiar Eliminar ▾ Más
nombre	varchar(255) utf8_general_ci			No	Ninguna			Cambiar Eliminar ▾ Más
slug	varchar(255) utf8_general_ci			No	Ninguna			Cambiar Eliminar ▾ Más

Aquí se almacenan todos los registros cuya información corresponde a todas las marcas de vehículos disponibles para seleccionar. Sus campos constan del id (Llave primaria), el nombre de la marca y el slug o denominación.

4.2.15 modelo

Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
id_marca	int(11)			No	Ninguna			Cambiar Eliminar ▾ Más
id	int(11)			No	Ninguna		AUTO_INCREMENT	Cambiar Eliminar ▾ Más
nombre	varchar(255) utf8_general_ci			No	Ninguna			Cambiar Eliminar ▾ Más
slug	varchar(255) utf8_general_ci			No	Ninguna			Cambiar Eliminar ▾ Más

Aquí se almacenan todos los registros cuya información corresponde a todas los modelos de vehículos disponibles para seleccionar. Sus campos constan del id (Llave primaria), el id de la marca al que corresponde el modelo (Llave foránea de la tabla “marca” ap. 4.2.14), el nombre del modelo y el slug o denominación.

4.2.16 pagos

Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
id	int(11)			No	Ninguna		AUTO_INCREMENT	Cambiar Eliminar ▾ Más
num	varchar(15) utf8_general_ci			No	Ninguna			Cambiar Eliminar ▾ Más
poliza_id	int(10)			No	Ninguna			Cambiar Eliminar ▾ Más
cobertura_id	int(11)			No	Ninguna			Cambiar Eliminar ▾ Más
prima_total	varchar(10) utf8_general_ci			No	Ninguna			Cambiar Eliminar ▾ Más
fecha_registro	varchar(15) utf8_general_ci			No	Ninguna			Cambiar Eliminar ▾ Más
metodoPago	varchar(2) utf8_general_ci			No	Ninguna			Cambiar Eliminar ▾ Más
estado	int(10)			No	Ninguna			Cambiar Eliminar ▾ Más

Aquí se almacenan todos los registros cuya información corresponde a todos los pagos de las pólizas. Sus campos constan del id (Llave primaria), el número de pago, el id de la poliza al que corresponde el pago (Llave foránea de la tabla “poliza” ap. 4.2.17), el id de la cobertura que representa el pago (Llave foránea de la tabla “cobertura” ap. 4.2.2), la prima total, la fecha en que se grabó el pago (o en que se registró la poliza), el método de pago utilizado, y el id del estado en el que se encuentra el pago (Llave foránea de la tabla “estado_pago” ap. 4.2.9).

4.2.17 poliza

Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
id	int(11)			No	Ninguna		AUTO_INCREMENT	Cambiar Eliminar Más
nro	varchar(50)	utf8_general_ci		No	Ninguna			Cambiar Eliminar Más
clienteid	int(11)			No	Ninguna			Cambiar Eliminar Más
vehiculoid	int(11)			No	Ninguna			Cambiar Eliminar Más
coberturaid	int(11)			No	Ninguna			Cambiar Eliminar Más
fecha_emision	varchar(100)	utf8_general_ci		No	Ninguna			Cambiar Eliminar Más
horaEmis	varchar(100)	utf8_general_ci		No	Ninguna			Cambiar Eliminar Más
emisor	int(255)			No	Ninguna			Cambiar Eliminar Más
sucursal	int(10)			No	Ninguna			Cambiar Eliminar Más
vigencia_inicio	varchar(100)	utf8_general_ci		No	Ninguna			Cambiar Eliminar Más
horarioVig	varchar(100)	utf8_general_ci		No	Ninguna			Cambiar Eliminar Más
vigencia_fin	varchar(100)	utf8_general_ci		No	Ninguna			Cambiar Eliminar Más
meses_vig	varchar(2)	utf8_general_ci		Si	NULL			Cambiar Eliminar Más
anulacion	varchar(20)	utf8_general_ci		No	Ninguna			Cambiar Eliminar Más
estado	int(100)			No	Ninguna			Cambiar Eliminar Más

Aquí se almacenan todos los registros cuya información corresponde a todas las pólizas. Sus campos constan del id (Llave primaria), el número de poliza, el id del cliente tomador de la poliza (Llave foránea de la tabla “tomador” ap. 4.2.23), el id del vehículo que representa la poliza (Llave foránea de la tabla “vehículo” ap. 4.2.26), el id de la cobertura que representa la poliza (Llave foránea de la tabla “cobertura” ap. 4.2.2), la fecha en la que se grabó la poliza, la hora en la que se grabó, el id del usuario emisor de la poliza (Llave foránea de la tabla “usuarios” ap. 4.2.25), el id de la sucursal en la cual se grabó la poliza (Llave foránea de la tabla “sucursales” ap. 4.2.20), la fecha en la que la poliza inicia su vigencia, el horario en el que inicia y finaliza la vigencia, la fecha en que finaliza la vigencia, la cantidad de meses que dura la vigencia, la fecha en la cual se anula la poliza y el id correspondiente al estado en el que se encuentra actualmente la poliza (Llave foránea de la tabla “estadopoliza” ap. 4.2.6).

4.2.18 provincia

Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
id	tinyint(3)	UNSIGNED		No	Ninguna		AUTO_INCREMENT	Cambiar Eliminar Más
nombre	varchar(50)	utf8_spanish_ci		No	Ninguna			Cambiar Eliminar Más
codigo31662	char(4)	utf8_general_ci		No	Ninguna			Cambiar Eliminar Más

Aquí se almacenan todos los registros cuya información corresponde a todas las provincias disponibles para seleccionar. Sus campos constan del id (Llave primaria), el nombre de la provincia y el código de referencia.

4.2.19 siniestro

id	int(100)	No	Ninguna	AUTO_INCREMENT			Más
nroSiniestro	varchar(20) utf8_general_ci	No	Ninguna				Más
fechaDen	text utf8_general_ci	No	Ninguna				Más
emisor	int(10)	No	Ninguna				Más
fechaOc	text utf8_general_ci	No	Ninguna				Más
horaOc	text utf8_general_ci	No	Ninguna				Más
idPoliza	int(100)	No	Ninguna				Más
lesionMuerte	text utf8_general_ci	No	Ninguna				Más
tipoSiniestro	int(50)	No	Ninguna				Más
nomDen	text utf8_general_ci	Sí	NULL				Más
telDen	varchar(50) utf8_general_ci	Sí	NULL				Más
domDen	text utf8_general_ci	Sí	NULL				Más
dniDen	varchar(50) utf8_general_ci	Sí	NULL				Más
emailDen	varchar(100) utf8_general_ci	Sí	NULL				Más
estado	int(100)	No	Ninguna				Más
descripcionAcc	text utf8_spanish2_ci	No	Ninguna				Más
calleAcc	varchar(100) utf8_general_ci	Sí	NULL				Más
alturaAcc	varchar(100) utf8_general_ci	Sí	NULL				Más
provinciaAcc	text utf8_general_ci	No	Ninguna				Más
localidadAcc	text utf8_general_ci	No	Ninguna				Más
nombreTercero	text utf8_general_ci	No	Ninguna				Más
domTercero	text utf8_general_ci	No	Ninguna				Más
telTercero	varchar(50) utf8_general_ci	No	Ninguna				Más
docTercero	varchar(50) utf8_general_ci	No	Ninguna				Más
nombreConductor	text utf8_general_ci	Sí	NULL				Más
telConductor	varchar(50) utf8_general_ci	Sí	NULL				Más
domConductor	text utf8_general_ci	Sí	NULL				Más
dniConductor	varchar(50) utf8_general_ci	Sí	NULL				Más
imagen	varchar(50) utf8_general_ci	Sí	NULL				Más
tomadDenunciante	varchar(10) utf8_general_ci	No	Ninguna				Más
condReclamante	varchar(10) utf8_general_ci	No	Ninguna				Más
tomadReclamante	varchar(10) utf8_general_ci	No	Ninguna				Más

Aquí se almacenan todos los registros cuya información corresponde a todos los siniestros. Sus campos constan del id (Llave primaria), el número del siniestro, la fecha en que se realizó la denuncia, el id del usuario emisor del siniestro (Llave foránea de la tabla “usuarios” ap. 4.2.25), la fecha en que ocurrió el siniestro, la hora de ocurrencia, el id de la poliza que cubre el siniestro (Llave foránea de la tabla “poliza” ap. 4.2.17), lesión o muerte; el nombre, teléfono, domicilio, DNI y email del denunciante, el id del estado en el que se encuentra el siniestro (Llave foránea de la tabla “estadosiniestro” ap. 4.2.7), la descripción del accidente, calle del accidente, altura del accidente, localidad del accidente (Llave foránea de la tabla “localidad” ap. 4.2.12), provincia del accidente (Llave foránea de la tabla “provincia” ap. 4.2.18); nombre, domicilio, teléfono y documento del tercero; nombre, domicilio, teléfono y documento del conductor tercero, opción de tomador es igual al denunciante, opción de conductor es igual a reclamante y opción de tomador es igual a reclamante.

4.2.20 sucursales

Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
id	int(10)			No	Ninguna		AUTO_INCREMENT	Cambiar Eliminar Más
nombre	varchar(40)	utf8mb4_general_ci		No	Ninguna			Cambiar Eliminar Más
iden_sucursal	varchar(15)	utf8mb4_general_ci		No	Ninguna			Cambiar Eliminar Más
cuit	varchar(20)	utf8mb4_general_ci		No	Ninguna			Cambiar Eliminar Más
direccion	varchar(50)	utf8mb4_general_ci		No	Ninguna			Cambiar Eliminar Más
cp	int(10)			No	Ninguna			Cambiar Eliminar Más
localidad	int(10)			No	Ninguna			Cambiar Eliminar Más
provincia	int(10)			No	Ninguna			Cambiar Eliminar Más

Aquí se almacenan todos los registros cuya información corresponde a todas las sucursales de la compañía. Sus campos constan del id (Llave primaria), el nombre de la sucursal, el número de la sucursal, el CUIT, la dirección, el código postal de la ubicación (Llave foránea de la tabla “localidad” ap. 4.2.12 para el campo “codigopostal”), el id de la localidad en la que se encuentra la sucursal (Llave foránea de la tabla “localidad” ap. 4.2.12) y el id de la provincia en la que se encuentra la sucursal (Llave foránea de la tabla “provincia” ap. 4.2.18).

4.2.21 tiposiniestro

Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
id	int(50)			No	Ninguna		AUTO_INCREMENT	Cambiar Eliminar Más
nombre	varchar(100)	utf8_general_ci		No	Ninguna			Cambiar Eliminar Más

Aquí se almacenan todos los registros cuya información corresponde a todos los tipos de siniestros disponibles para seleccionar. Sus campos constan del id (Llave primaria) y el nombre.

4.2.22 tipo_endoso

Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
id	int(20)			No	Ninguna		AUTO_INCREMENT	Cambiar Eliminar Más
descripcion	varchar(50)	utf8_general_ci		No	Ninguna			Cambiar Eliminar Más

Aquí se almacenan todos los registros cuya información corresponde a todos los tipos de endosos disponibles para seleccionar. Sus campos constan del id (Llave primaria) y la descripción.

4.2.23 tomador

Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
id	Int(11)			No	Ninguna		AUTO_INCREMENT	Cambiar Eliminar Más
num_tom	varchar(20)	utf8_general_ci		No	Ninguna			Cambiar Eliminar Más
nombre	varchar(30)	utf8_spanish_ci		No	Ninguna			Cambiar Eliminar Más
documento	varchar(10)	utf8_general_ci		No	Ninguna			Cambiar Eliminar Más
persona	varchar(10)	utf8_general_ci		No	Ninguna			Cambiar Eliminar Más
nacionalidad	varchar(20)	utf8_general_ci		No	Ninguna			Cambiar Eliminar Más
provincia	varchar(20)	utf8_spanish_ci		No	Ninguna			Cambiar Eliminar Más
localidad	varchar(20)	utf8_spanish_ci		No	Ninguna			Cambiar Eliminar Más
cp	varchar(20)	utf8_general_ci		No	Ninguna			Cambiar Eliminar Más
calle	varchar(20)	utf8_general_ci		No	Ninguna			Cambiar Eliminar Más
genero	varchar(20)	utf8_general_ci		No	Ninguna			Cambiar Eliminar Más
telefono	varchar(20)	utf8_general_ci		No	Ninguna			Cambiar Eliminar Más
correo	varchar(50)	utf8_general_ci		Sí	NULL			Cambiar Eliminar Más
fecha_nac	varchar(20)	utf8_general_ci		No	Ninguna			Cambiar Eliminar Más

Aquí se almacenan todos los registros cuya información corresponde a todos los clientes. Sus campos constan del id (Llave primaria), el número del cliente, el nombre, el documento, el tipo de persona, la nacionalidad, el id de la provincia de residencia (Llave foránea de la tabla "provincia" ap. 4.2.18), el id de la localidad de residencia (Llave foránea de la tabla "localidad" ap. 4.2.12), el código postal de la ubicación (Llave foránea de la tabla "localidad" ap. 4.2.12 para el campo "codigopostal"), la calle, el género, el número de teléfono, el correo y la fecha de nacimiento.

4.2.24 uso_vehiculo

Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
id	Int(100)			No	Ninguna		AUTO_INCREMENT	Cambiar Eliminar Más
nombre	varchar(250)	utf8_general_ci		No	Ninguna			Cambiar Eliminar Más

Aquí se almacenan todos los registros cuya información corresponde a todos los tipos de uso que se le pueden dar a un vehículo. Sus campos constan del id (Llave primaria) y el nombre descriptivo.

4.2.25 usuarios

Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
id	Int(20)			No	Ninguna		AUTO_INCREMENT	Cambiar Eliminar Más
nro	varchar(20)	utf8_general_ci		No	Ninguna			Cambiar Eliminar Más
nombre	varchar(50)	utf8_general_ci		No	Ninguna			Cambiar Eliminar Más
documento	varchar(20)	utf8_general_ci		No	Ninguna			Cambiar Eliminar Más
nacionalidad	varchar(20)	utf8_general_ci		No	Ninguna			Cambiar Eliminar Más
provincia	int(50)			No	Ninguna			Cambiar Eliminar Más
localidad	int(50)			No	Ninguna			Cambiar Eliminar Más
direccion	varchar(50)	utf8_general_ci		No	Ninguna			Cambiar Eliminar Más
fecha_nac	varchar(20)	utf8_general_ci		No	Ninguna			Cambiar Eliminar Más
telefono	varchar(20)	utf8_general_ci		No	Ninguna			Cambiar Eliminar Más
correo	varchar(50)	utf8_general_ci		No	Ninguna			Cambiar Eliminar Más
genero	varchar(20)	utf8_general_ci		No	Ninguna			Cambiar Eliminar Más
usuario	varchar(20)	utf8_general_ci		No	Ninguna			Cambiar Eliminar Más
pass	varchar(100)	utf8_general_ci		No	Ninguna			Cambiar Eliminar Más
rol	varchar(20)	utf8_general_ci		No	Ninguna			Cambiar Eliminar Más
estado	varchar(20)	utf8_general_ci		No	Ninguna			Cambiar Eliminar Más

Aquí se almacenan todos los registros cuya información corresponde a todos los usuarios del sistema. Sus campos constan del id (Llave primaria), el número del usuario, el nombre, el documento, la nacionalidad, el id de la provincia de residencia (Llave foránea de la tabla “provincia” ap. 4.2.18), el id de la localidad de residencia (Llave foránea de la tabla “localidad” ap. 4.2.12), la dirección, la fecha de nacimiento, el género, el número de teléfono, el correo, el género, el nombre de usuario, la pass (o contraseña), el tipo de rol y el estado en el que se encuentra actualmente el usuario.

4.2.26 vehiculo

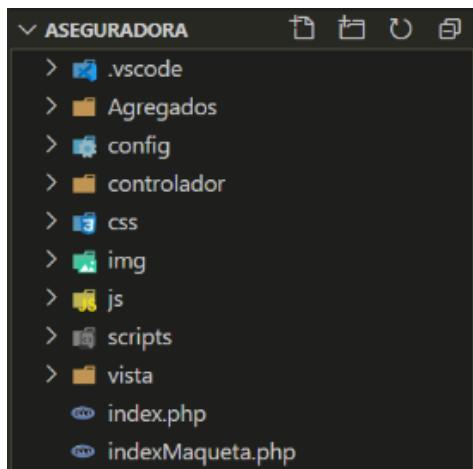
Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
id	int(11)			No	Ninguna		AUTO_INCREMENT	Cambiar Eliminar Más
suma_asegurada	varchar(100)	utf8_general_ci		No	Ninguna			Cambiar Eliminar Más
marca_id	int(11)			No	Ninguna			Cambiar Eliminar Más
modelo_id	int(11)			No	Ninguna			Cambiar Eliminar Más
patente	varchar(100)	utf8_general_ci		No	Ninguna			Cambiar Eliminar Más
motor	varchar(50)	utf8_general_ci		No	Ninguna			Cambiar Eliminar Más
nroChasis	varchar(50)	utf8_general_ci		No	Ninguna			Cambiar Eliminar Más
nroMotor	varchar(50)	utf8_general_ci		No	Ninguna			Cambiar Eliminar Más
clase	int(50)			No	Ninguna			Cambiar Eliminar Más
uso	int(50)			No	Ninguna			Cambiar Eliminar Más
gps	varchar(10)	utf8_general_ci		No	Ninguna			Cambiar Eliminar Más
0km	varchar(10)	utf8_general_ci		No	Ninguna			Cambiar Eliminar Más
Kilometros	varchar(10)	utf8_general_ci		No	Ninguna			Cambiar Eliminar Más
codigoPostal	int(10)			No	Ninguna			Cambiar Eliminar Más
localidad	int(10)			No	Ninguna			Cambiar Eliminar Más
coberturaAd	int(10)			No	Ninguna			Cambiar Eliminar Más
combustible	varchar(50)	utf8_general_ci		No	Ninguna			Cambiar Eliminar Más
eGas	varchar(10)	utf8_general_ci		No	Ninguna			Cambiar Eliminar Más
anio	varchar(50)	utf8_general_ci		No	Ninguna			Cambiar Eliminar Más
color	varchar(50)	utf8_general_ci		Sí	NULL			Cambiar Eliminar Más
asientos	int(50)			Sí	NULL			Cambiar Eliminar Más
pasajeros	int(50)			Sí	NULL			Cambiar Eliminar Más

Aquí se almacenan todos los registros cuya información corresponde a todos los vehículos. Sus campos constan del id (Llave primaria), la suma asegurada, el id de su marca (Llave foránea de la tabla “marca” ap. 4.2.14), el id de su modelo (Llave foránea de la tabla “modelo” ap. 4.2.15), el número de matrícula, el motor, el número de chasis, el número de motor, el id de la clase de vehículo al que corresponde (Llave foránea de la tabla “clase_vehiculo” ap. 4.2.1), el id del tipo de uso que se asignó al vehículo (Llave foránea de la tabla “uso_vehiculo” ap. 4.2.24), si requiere GPS, si es cero kilómetro, el número de código postal de la ubicación de residencia del vehículo (Llave foránea de la tabla “localidad” ap. 4.2.12 para el campo “codigopostal”), el id de la localidad en la que se encuentra el vehículo (Llave foránea de la tabla “localidad” ap. 4.2.12), el id del tipo de cobertura adicional que posee el vehículo (Llave foránea de la tabla “cobertura_adicional” ap. 4.2.3), si lleva equipo de gas, el año de fabricación, el número o nombre del color, la cantidad de asientos y la cantidad de pasajeros.

➤ 4.3 Código comentado

En esta sección del manual es donde se describirá todo el código fuente de programación del sistema. En el proceso se recorrerá toda la estructura lógica y funcional de cada una de las operaciones realizadas, así como el funcionamiento y el objetivo de cada una de las clases y métodos que componen el proyecto final. Se mostrará una captura de pantalla de la codificación y será comentada al principio de los procesos más importantes o relevantes. Toda la visualización será representada desde el editor de código visual studio code.

Hay que tener en cuenta que no se realizaran comentarios descriptivos en aquellos sectores de los ficheros donde la lógica del código sea muy repetitiva o fácil de predecir para cualquier programador, ya que resultaría innecesario y en ocasiones, provocaría mayor dificultad para la legibilidad y comprensión del mismo.



La carpeta principal del proyecto se denomina “Aseguradora”.

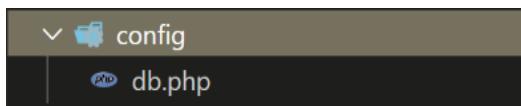
Dentro de la carpeta principal, el proyecto se subdivide en varias carpetas más, siendo las siguientes:

agregados: Aquí se almacenan todas las librerías de backend que sean agregadas al sistema.

css: Carpeta donde se guardan los ficheros que contienen las hojas de estilos del sistema.

img: Carpeta destinada a guardar todas las imágenes con las que interactúe el sistema.

✿ 4.3.1 Carpeta “config”



En esta carpeta se almacenan todos los ficheros que contienen alguna configuración general del sistema, en este caso solo se configura la base de datos.

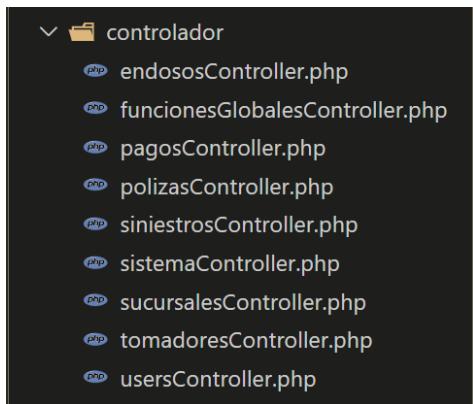
➔ 4.3.1.1 fichero “db.php”

```
<?php

//Hacemos la conexión a la base de datos y la retornamos.

class Database
{
    public static function connect()
    {
        $conexion = new mysqli('localhost', 'root', '', 'aseguradora');
        $conexion->query("SET NAMES 'utf8'");
        return $conexion;
    }
}
```

4.3.2 Carpeta “controlador”



En esta carpeta, se encuentran todos los ficheros que contienen las clases principales del sistema, las cuales funcionan como controlador en el paradigma MVC, y son las encargadas de ejecutar consultas a la base de datos y realizar todas las funciones del lado del servidor.

➔ 4.3.2.1 fichero “endososController.php”

Todas las acciones relacionadas a los endosos se realizaran en este fichero.

```
<?php

class EndososController{

    private $conexion;

    // Función constructora que realiza la conexión a la base de datos.
    public function __construct()
    {
        $this->conexion = Database::connect();
        date_default_timezone_set("America/Argentina/Buenos_Aires");
    }

    // Funcion para grabar un endoso
    public function grabarEndoso($info)
    {
        $mysqli = $this->conexion;

        $arr = array('success'=>false);

        $error = false;

        $idPoliza = $info['idPoliza'];

        date_default_timezone_set("America/Argentina/Buenos_Aires");

        $tipoEnd = filter_var($info['tipoEnd'], FILTER_VALIDATE_INT);
        if($tipoEnd==FALSE || is_null($tipoEnd)) {$error = true;}

        $idPoliza = filter_var($idPoliza, FILTER_VALIDATE_INT);
        if($idPoliza==FALSE || is_null($idPoliza)) {$error = true;}

        $fechaEmisEnd = date("d/m/Y");

        $horaEmisEnd = date("H:i");

        if(!isset($_SESSION)){
            session_start();
        }
        if(isset($_SESSION['user'])){
            $emis = $_SESSION['user']['id'];
        }

        // Obtenemos el numero de poliza para guardarla en el endoso.
        $nroPoliza =
        "SELECT nro FROM poliza
        WHERE id = ?";
```

```

$stmt = $mysqli->prepare($nroPoliza);
if($stmt!==FALSE){
    $stmt->bind_param('i', $idPoliza);
    $stmt->execute();

    $rs = $stmt->get_result();
    if ($rs->num_rows > 0) {
        while ($fila = $rs->fetch_array()) {
            $nroPoliza = $fila[0];
        }
    }
}

if($tipoEnd==1){
    $descripcion = "Modificacion de riesgo poliza $nroPoliza.";
}else if($tipoEnd==2){
    $descripcion = "Modificacion de datos nominales poliza $nroPoliza.";
}else if($tipoEnd==3){
    $descripcion = "Modificacion de datos del vehiculo poliza $nroPoliza.";
}else if($tipoEnd==4){
    $descripcion = "Modificacion de asegurado poliza $nroPoliza.";
};
}else if($tipoEnd==5){
    $descripcion = "Cambio de cobertura poliza $nroPoliza.";
}else if($tipoEnd==6){
    $descripcion = "Modificacion de Estado de Poliza $nroPoliza."
};

// Generamos un numero aleatorio para asignarle al endoso
$numeroEnd = rand(999999999,999999999);

// Ejecutamos la consulta.
if(!$error){

    $mysqli->set_charset("utf8");
    $sql = "INSERT INTO endoso (polizaId, endosoNum, polizaNum,
fechaRegistroEnd, horaRegistroEnd, emisor, tipo, descripcion) VALUES (?,?,?,?,?,?,
?,?)";
    $stmt = $mysqli->prepare($sql);
    if ($stmt!==FALSE) {
        $stmt-
>bind_param('issssi', $idPoliza, $numeroEnd, $nroPoliza, $fechaEmisEnd, $ho
raEmisEnd, $emis, $tipoEnd, $descripcion);
        $stmt->execute();
        $stmt->close();

        $arr = array('success'=>true);
    }
}

```

```

        return json_encode($arr);
    }
    // Funcion para mostrar el listado de endosos de una poliza.
    public function mostrarEndososPorPoliza($id)
    {
        $mysqli = $this->conexion;
        $arr = array('success'=>false);
        $id = $id['id'];
        $mysqli->set_charset("utf8");
        $sql = "SELECT E.polizaNum AS polizaNum, T.nombre AS tomador, E.fechaRegistroEnd, E.horaRegistroEnd, TE.descripcion AS tipo, E.descripcion AS descripcion, E.id, E.endosoNum, P.id
                FROM endoso E
                INNER JOIN tipo_endoso TE ON TE.id = E.tipo
                INNER JOIN poliza P ON E.polizaId = P.id
                INNER JOIN tomador T ON T.id = P.clienteid
                WHERE E.polizaId = ?
                ORDER BY E.fechaRegistroEnd DESC";
        $stmt = $mysqli->prepare($sql);
        if ($stmt!==FALSE) {
            $stmt->bind_param('i', $id);
            $stmt->execute();
            $rs = $stmt->get_result();
            if ($rs->num_rows > 0)
            {
                while ($mostrar = mysqli_fetch_row($rs)){
                    $arr = array(
                        $mostrar[7],
                        $mostrar[2],
                        $mostrar[3],
                        $mostrar[4],
                        $mostrar[5],
                        '<a class="editarDescripcion" value="'. $mostrar[6] .'"><i class="fas fa-pencil-alt"></i></a>';
                    );
                    $numPol = $mostrar[0];
                    $tom = $mostrar[1];
                    $idPol = $mostrar[8];
                    $arr1[] = $arr;
                }
            }
            echo json_encode(
                array('success'=>true, 'numPol'=>$numPol, 'tom'=>$tom)
            );
        }
        $stmt->close();
    }
    // Función donde traeremos la descripción de un endoso particular.
    public function traerDescripcionEndoso($idEndoso)
    {
        $mysqli = $this->conexion;

```

```

$arr = array('success'=>false);

$idEndoso = $idEndoso['idEndoso'];

mysqli->set_charset("utf8");

$sql = "SELECT descripcion, endosoNum
        FROM endoso
        WHERE id = ? ";

$stmt = $mysqli->prepare($sql);

if ($stmt!==FALSE) {

    $stmt->bind_param('i', $idEndoso);
    $stmt->execute();
    $rs = $stmt->get_result();

    if ($rs->num_rows > 0) {
        while ($endoso = $rs->fetch_array()) {

            $arr['descripcion'] = $endoso[0];
            $arr['endosoNum'] = $endoso[1];
            $arr['idEndoso'] = $idEndoso;
            $arr['success'] = true;
        }
    }
}

$stmt->close();
echo json_encode(array('data' => $arr));
}

// Función donde actualizaremos la descripción de un endoso.
public function actualizarDescripcion($post)
{
    $arr = array('success'=>false);

    $error = false;

    $idEndoso = $post['idEndoso'];

    $nuevaDesc = filter_var($post['nuevaDesc'], FILTER_SANITIZE_FULL_
SPECIAL_CHARS);
    if($nuevaDesc==FALSE || is_null($nuevaDesc)) {$error = true;}

    if(!$error){

        $mysqli = $this->conexion;

        $mysqli->set_charset("utf8");

        $sql = "UPDATE endoso
                SET descripcion = ?

```

```

        WHERE id      = ?";

$stmt = $mysqli->prepare($sql);
$stmt->bind_param('si', $nuevaDesc, $idEndoso);
$stmt->execute();
$rs = $stmt->get_result();

if ($stmt!==FALSE){
    $arr = array('success'=>true);
}
$stmt->close();
}

echo json_encode($arr);
}

// Función donde traemos el listado de los tipos de endosos para el
campo de selección.
public function listadoTipoEndosos()
{
    $mysqli = $this->conexion;
    $tiposEnd = '';

$sql = "SELECT id, descripcion FROM tipo_endoso";
$stmt = $mysqli->prepare($sql);
if($stmt!=FALSE){
    $stmt->execute();
    $rs = $stmt->get_result();
    if($rs->num_rows > 0){
        $tiposEnd .= '<option value="0" selected disabled>Selección</option>';
        foreach($rs as $end){
            $tiposEnd .= '<option value="'. $end['id'] .'>' . $end['
descripcion']. '</option>';
        }
    }
    $stmt->close();
}

return $tiposEnd;
}
}

```

➔ 4.3.2.2 fichero “funcionesGlobalesController.php”

Todas las funcionalidades que serán utilizadas en todo el sistema se encuentran en este fichero.

```
<?php

// Clase principal
class FuncionesGlobalesController{

    private $conexion;

    // Función constructora donde realizamos la conexión a la base de datos.
    public function __construct()
    {
        $this->conexion = Database::connect();
        date_default_timezone_set("America/Argentina/Buenos_Aires");
    }

    // Función devolvemos la lista de provincias para los selects.
    public function listadoProvincias()
    {
        $provincias = '';
        $mysqli = $this->conexion;

        $sql = "SELECT id, nombre FROM provincia";

        $stmt = $mysqli->prepare($sql);
        if($stmt!==FALSE){
            $stmt->execute();

            $rs = $stmt->get_result();
            if ($rs->num_rows > 0) {
                $provincias .= '<option value="n" selected disabled>Seleccion
e</option>';
                foreach($rs as $pro){
                    $provincias .= '<option value="'. $pro['id'] .'>' . $pro['no
mbre'] . '</option>';
                }
            }
            $stmt->close();
            return $provincias;
        }
    }

    // Función devolvemos la lista de localidades para los selects.
    public function listadoLocalidadesPorCp($cp)
    {
        $cp = $cp['cp'];
        $arr1 = array();

        $sql =
```

```

" SELECT id, proper(nombre) FROM localidad
WHERE codigopostal = ? ";

$mysqli = $this->conexion;

$stmt = $mysqli->prepare($sql);
if($stmt!==FALSE){
    $stmt->bind_param('s', $cp);
    $stmt->execute();

    $rs = $stmt->get_result();
    if ($rs->num_rows > 0) {
        while ($fila = $rs->fetch_array()) {
            $arr = array();
            $arr['localidadId'] = $fila[0];
            $arr['localidadNombre'] = $fila[1];
            $arr1[] = $arr;
        }
    }

    $stmt->close();

    echo json_encode(array('data' => $arr1));
}

// Función devolvemos la lista de provincias a través de un código postal
para los selects.
public function listadoLocalidadesPorProvincia($post)
{
    $id = $post['id'];
    $arr1 = array();

    $mysqli = $this->conexion;

    $sql =
"SELECT id, proper(nombre) FROM localidad
WHERE provincia_id = ?";

$stmt = $mysqli->prepare($sql);
if($stmt!==FALSE){
    $stmt->bind_param('i', $id);
    $stmt->execute();

    $rs = $stmt->get_result();
    if ($rs->num_rows > 0) {
        while ($fila = $rs->fetch_array()) {
            $arr = array();
            $arr['localidadId'] = $fila[0];
            $arr['localidadNombre'] = $fila[1];
            $arr1[] = $arr;
        }
    }
}

```

```

        $stmt->close();
        echo json_encode(array('data' => $arr1));
    }
}

// Función devolvemos la lista de código postales a través de una
// provincia para los selects.
public function listadoCpPorProvincia($id)
{
    $id = $id['id'];

    $arr1 = array();

    $mysqli = $this->conexion;

    $sql =
"SELECT id, codigopostal FROM localidad
WHERE provincia_id = ?
GROUP BY codigopostal";

    $stmt = $mysqli->prepare($sql);
    if($stmt!==FALSE){
        $stmt->bind_param('i', $id);
        $stmt->execute();

        $rs = $stmt->get_result();
        if ($rs->num_rows > 0) {
            while ($fila = $rs->fetch_array()) {
                $arr = array();
                $arr['cpId'] = $fila[0];
                $arr['cpNum'] = $fila[1];
                $arr1[] = $arr;
            }
        }

        $stmt->close();
        echo json_encode(array('data' => $arr1));
    }
}

// Función devolvemos la lista de código postales para los selects.
public function listadoCp()
{
    $mysqli = $this->conexion;
    $cps = '';

    $sql = "SELECT DISTINCT codigopostal FROM localidad";
    $stmt = $mysqli->prepare($sql);
    if($stmt!=FALSE){
        $stmt->execute();
        $rs = $stmt->get_result();
        if($rs->num_rows > 0){

```

```
        $cps .= '<option value="n" selected disabled>Seleccione</option>';
        foreach($rs as $cp){
            $cps .= '<option value="'. $cp['codigopostal'] .'">' . $cp['codigopostal'] . '</option>';
        }
    }

$stmt->close();
return $cps;
}

}
```

➔ 4.3.2.3 fichero “pagosController.php”

Este fichero contiene todas las funciones que realizan las acciones relacionadas a los pagos y las cuotas en el sistema.

```
<?php

// Clase principal
class PagosController{

    private $conexion;

    // Función constructora donde realizamos la conexión a la base de datos.
    public function __construct()
    {
        $this->conexion = Database::connect();
        date_default_timezone_set("America/Argentina/Buenos_Aires");
    }

    // Función donde grabamos un nuevo pago
    public function grabarPago($idPol, $cob, $sumAs, $mesesVig, $datosPago)
    {
        $mysqli = $this->conexion;
        $respuesta = false;

        // Extraemos los datos del pago
        $datosPago = $datosPago;

        // Eliminamos el punto de la suma asegurada (.)
        $sumAs = str_replace('.', '', $sumAs);

        // Generamos un numero aleatorio de 10 dígitos para el pago.
        $iden_pago = mt_rand(999999999, 999999999);

        $idPol = $idPol;

        $cob = $cob;

        $prima = $datosPago[0];

        // Eliminamos los decimales de la prima
        $prima = number_format($prima, '0', ',', '.');
        // Eliminamos el punto(.)
        $prima = str_replace('.', '', $prima);

        $d = date("d");
        $m = date("m");
        $a = date("Y");

        $fecha_registro = $d.'.'.$m.'/'.$a;
```

```

$metodoPago = $datosPago[7];

$estado = '1';

// Ejecutamos la consulta
$sql = "INSERT INTO pagos (num, poliza_id, cobertura_id, prima_total,
fecha_registro, metodoPago, estado) VALUES (?,?,?,?,?,?,?,?)";
$stmt = $mysqli->prepare($sql);
$stmt-
>bind_param('siisssi', $iden_pago, $idPol, $cob, $prima, $fecha_registro, $me
todoPago, $estado);
$r = $stmt->execute();

if($r){

    $conIdPago = "SELECT MAX(id) AS id FROM pagos";
    $stmt = $mysqli->prepare($conIdPago);
    $stmt->execute();
    $rs = $stmt->get_result();
    if ($rs->num_rows > 0) {

        //obtenemos el id del pago grabado
        while ($fila = $rs->fetch_array()) {
            $pagoId = $fila[0];
        }

        $idPago      = $pagoId;
        $primaTotal  = $prima;
        $cantCuotas = $datosPago[1];
        $primaMensual = $datosPago[2];
        $diaCobrar   = $datosPago[3];
        $inicioVig   = $datosPago[4];
        $finVig     = $datosPago[5];
        $tiempoAcobrar = $datosPago[6];

        $datosPago = [$pagoId, $primaTotal, $cantCuotas, $primaMensual,
                     $diaCobrar, $inicioVig, $finVig, $tiempoAcobrar];

        // Grabamos las cuotas del pago desde la funcion
        $cuotas = $this->grabarCuotas($datosPago);

        if($cuotas){
            $respuesta = true;
        }else{
            $respuesta = false;
        }
    }

}else{
    $respuesta = false;
}

$stmt->close();

```

```

        return $respuesta;
    }

    // Funcion para generar los datos de pago en el formulario.
    public function generarFormularioPago($post)
    {
        $mysqli = $this->conexion;

        $idpoliza = $post['idpoliza'];

        $arr = array('success'=>false);

        $sql =
        " SELECT
            P.id,
            P.nro,
            P.fecha_emision,
            P.horaEmis,
            P.horaInVig,
            P.vigencia_inicio,
            P.vigencia_fin,
            EP.nombre AS estado,
            P.meses_vig,

            V.suma_asegurada,
            MA.nombre AS marca,
            M.nombre AS modelo,
            V.motor,
            V.0km,
            V.kilometros,
            V.codigoPostal, proper(L.nombre) AS localidad,
            PR.nombre AS provincia,
            V.combustible,
            V.eGas,
            V.anio,

            C.nombre,
            C.Descripcion,
            CA.nombre AS coberturaAd,

            PA.prima_total,

            PA.id,
            PA.estado

        FROM poliza P
        INNER JOIN vehiculo V ON V.id = P.vehiculoid
        INNER JOIN marca MA ON MA.id = V.marca_id
        INNER JOIN modelo M ON M.id = V.modelo_id
        INNER JOIN localidad L ON V.localidad = L.id
        INNER JOIN provincia PR ON PR.id = L.provincia_id
        INNER JOIN cobertura_adicional CA ON V.coberturaAd = CA.id
        INNER JOIN cobertura C ON C.id = P.coberturaid
        INNER JOIN clase_vehiculo CV ON CV.id = V.clase
    }

```

```

    INNER JOIN uso_vehiculo UV ON UV.id = V.uso
    INNER JOIN estadopoliza EP ON EP.id = P.estado
    INNER JOIN pagos PA ON PA.poliza_id = P.id

    WHERE P.id = ? ";

$stmt = $mysqli->prepare($sql);
if($stmt!==FALSE){
    $stmt->bind_param('i', $idpoliza);
    $stmt->execute();

    $rs = $stmt->get_result();
    if ($rs->num_rows > 0) {
        while ($fila = $rs->fetch_array())
        {
            $arr = array();
            $arr['polizaId'] = $fila[0];
            $arr['polizaNum'] = $fila[1];
            $arr['polizaEmis'] = $fila[2];
            $arr['polizaEmisHora'] = $fila[3];
            $arr['polizaHorInVig'] = $fila[4];
            $arr['polizaInVig'] = $fila[5];
            $arr['polizaFinVig'] = $fila[6];
            $arr['polizaEstado'] = $fila[7];
            $arr['polizaMesesVig'] = $fila[8];

            $arr['sumaAsegurada'] = $fila[9];
            $arr['marca'] = $fila[10];
            $arr['modelo'] = $fila[11];
            $arr['motor'] = $fila[12];
            $arr['ceroKm'] = $fila[13];
            $arr['kilometros'] = $fila[14];
            $arr['cpVehiculo'] = $fila[15];
            $arr['vehiculoLocalidad'] = $fila[16];
            $arr['vehiculoProvincia'] = $fila[17];
            $arr['combustible'] = $fila[18];
            $arr['eGas'] = $fila[19];
            $arr['anio'] = $fila[20];

            $arr['cobertura'] = $fila[21];
            $arr['coberturaDesc'] = $fila[22];
            $arr['coberturaAd'] = $fila[23];

            $arr['primaTotal'] = $fila[24];

            $arr['pagoId'] = $fila[25];
            $arr['pagoEstado'] = $fila[26];

            $arr['success'] = true;
        }
    }
}

$stmt->close();

```

```

        echo json_encode(array('data' => $arr));
    }

    // Funcion para modificar las cuotas de un pago.
    public function modificarCuotas($post)
    {

        $datos = $post['datos'];
        $respuesta = false;
        $error = false;
        $mysqli = $this->conexion;

        $idPago = filter_var($datos[0], FILTER_VALIDATE_INT);
        if($idPago==FALSE || is_null($idPago)) {$error = true; }

        $primaTotal = filter_var($datos[1], FILTER_SANITIZE_FULL_SPECIAL_CHARS);
        if($primaTotal==FALSE || is_null($primaTotal)) {$error = true; }

        if(!$error){

            // Modificamos la informacion del pago.
            $sql = "UPDATE pagos
                    SET prima_total = ?,
                        estado      = ?
                    WHERE id       = ?";
            $estado = 3; // Estado 3 = Modificado.

            $stmt = $mysqli->prepare($sql);
            $stmt->bind_param('sii', $primaTotal, $estado, $idPago);

            if($stmt->execute()){

                // Si se ejecuta la consulta, pasamos a eliminar las cuotas.
                $elCuota = "DELETE FROM cuotas WHERE pago_id = ?";
                $stmt = $mysqli->prepare($elCuota);
                $stmt->bind_param('i', $idPago);

                if($stmt->execute()){

                    // Pasamos a grabar las nuevas cuotas desde la función.
                    $modCuotas = $this->grabarCuotas($datos);

                    if($modCuotas){
                        $respuesta = true;
                    }
                }
            }
        }

        $stmt->close();
        return $respuesta;
    }
}

```

```

//Funcion destinada solo a generar y grabar las cuotas en la BBDD.
public function grabarCuotas($datos)
{
    $respuesta = false;

    $error = false;

    $mysqli = $this->conexion;

    $datos = $datos;

    $idPago = filter_var($datos[0], FILTER_VALIDATE_INT);
    if($idPago==FALSE || is_null($idPago)) {$error = true; }

    $primaTotal = filter_var($datos[1], FILTER_SANITIZE_FULL_SPECIAL_CHARS);
    if($primaTotal==FALSE || is_null($primaTotal)) {$error = true; }

    $cantCuotas = filter_var($datos[2], FILTER_VALIDATE_INT);
    if($cantCuotas==FALSE || is_null($cantCuotas)) {$error = true; }

    $primaMensual = filter_var($datos[3], FILTER_SANITIZE_FULL_SPECIAL_CHARS);
    if($primaMensual==FALSE || is_null($primaMensual)) {$error = true; }

    $diaCobrar = filter_var($datos[4], FILTER_VALIDATE_INT);
    if($diaCobrar==FALSE || is_null($diaCobrar)) {$error = true; }

    $inicioVig = filter_var($datos[5], FILTER_SANITIZE_FULL_SPECIAL_CHARS);
    if($inicioVig==FALSE || is_null($inicioVig)) {$error = true; }

    $finVig = filter_var($datos[6], FILTER_SANITIZE_FULL_SPECIAL_CHARS);
    if($finVig==FALSE || is_null($finVig)) {$error = true; }

    $tiempoAcobrar = $datos[7];

    // Obtenemos el día, mes y año de inicio de vigencia (substrayendo
    // información) para poder sumarlos en los lapsos después.
    $diaDeVigencia = substr($inicioVig, 0, -8);
    $mesDeVigencia = substr($inicioVig, 3, -5);
    $anioDeVigencia = substr($inicioVig, 6);

    $mesCobro = substr($inicioVig, 3, -5);
    $anioCobro = substr($inicioVig, 6);

    if($diaCobrar<10){
        $diaCobrar = "0$diaCobrar";
    }
}

```

```

// Si el día a cobrar es menor al de inicio de vigencia, se suma un
mes más a la facturación, y se agrega un cero por delante para seguir con la
notación de escritura de los números.

if($diaCobrar<=$diaDeVigencia){
    $mesCobro = $mesCobro + 1;
    if($mesCobro<10){
        $mesCobro = "0$mesCobro";
    }
}

$insCuotas = "INSERT INTO cuotas (numero_cuota, iden_cuota, pago_id,
prima_mensual, lapso, dia_cobrar, vto_pago, estado) VALUES (?,?,?,?,?,?,$mesCobro,?)"
;

for ($i=1; $i<=$cantCuotas ; $i++) {

    //-----LAPSOS DE COBRO-----

    // Primera fecha del lapso, es la original
    $lapso1 = "$diaDeVigencia/$mesDeVigencia/$anioDeVigencia";

    // Sumamos un mes a la fecha por cada ciclo para hacer los lapsos
    de tiempo
    $mesDeVigencia = $mesDeVigencia + $tiempoAcobrar;

    // Si colocamos el numero de mes correcto
    if($mesDeVigencia>12){

        if($mesDeVigencia==13){
            $mesDeVigencia = '1';
        }else if($mesDeVigencia==14){
            $mesDeVigencia = '2';
        }else if($mesDeVigencia==15){
            $mesDeVigencia = '3';
        }else if($mesDeVigencia==16){
            $mesDeVigencia = '4';
        }else if($mesDeVigencia==17){
            $mesDeVigencia = '5';
        }else if($mesDeVigencia==18){
            $mesDeVigencia = '6';
        }else if($mesDeVigencia==19){
            $mesDeVigencia = '7';
        }else if($mesDeVigencia==20){
            $mesDeVigencia = '8';
        }else if($mesDeVigencia==21){
            $mesDeVigencia = '9';
        }else if($mesDeVigencia==22){
            $mesDeVigencia = '10';
        }else if($mesDeVigencia==23){
            $mesDeVigencia = '11';
        }
    }
}

```

```

$anioDeVigencia = $anioDeVigencia + 1;

}

// Agregamos el cero cuando el mes es menor a 10
if($mesDeVigencia<10){
    $mesDeVigencia = "0$mesDeVigencia";
}

// Segunda fecha del lapso, donde ya se sumo un mes
$lapso2 = "$diaDeVigencia/$mesDeVigencia/$anioDeVigencia";

$lapso = "$lapso1 - $lapso2";

-----  

FECHA DE VENCIMIENTO DEL PAGO (1 mes despues de la facturacion)-----  

$mesVto = ($mesCobro+1);
$anioVto = $anioCobro;

if($mesVto>12){

    if($mesVto==13){
        $mesVto = '1';
    }else if($mesVto==14){
        $mesVto = '2';
    }else if($mesVto==15){
        $mesVto = '3';
    }else if($mesVto==16){
        $mesVto = '4';
    }else if($mesVto==17){
        $mesVto = '5';
    }else if($mesVto==18){
        $mesVto = '6';
    }else if($mesVto==19){
        $mesVto = '7';
    }else if($mesVto==20){
        $mesVto = '8';
    }else if($mesVto==21){
        $mesVto = '9';
    }else if($mesVto==22){
        $mesVto = '10';
    }else if($mesVto==23){
        $mesVto = '11';
    }

    $anioVto = $anioVto + 1;
}

if($mesVto<10){
    $mesVto = "0$mesVto";
}

$vto_pago = "$diaCobrar/$mesVto/$anioVto";

```

```

//-----FECHA DE FACTURACION DE CADA LAPSO-----

// Armamos la fecha separada en partes
$fechaCobrarCuota = "$diaCobrar/$mesCobro/$anioCobro";

// Sumamos un mes a la fecha de facturacion por cada lapso
$mesCobro = $mesCobro + $tiempoAcobrar;

// Si colocamos el numero de mes correcto
if($mesCobro>12){

    if($mesCobro==13){
        $mesCobro = '1';
    }else if($mesCobro==14){
        $mesCobro = '2';
    }else if($mesCobro==15){
        $mesCobro = '3';
    }else if($mesCobro==16){
        $mesCobro = '4';
    }else if($mesCobro==17){
        $mesCobro = '5';
    }else if($mesCobro==18){
        $mesCobro = '6';
    }else if($mesCobro==19){
        $mesCobro = '7';
    }else if($mesCobro==20){
        $mesCobro = '8';
    }else if($mesCobro==21){
        $mesCobro = '9';
    }else if($mesCobro==22){
        $mesCobro = '10';
    }else if($mesCobro==23){
        $mesCobro = '11';
    }
}

$anioCobro = $anioCobro + 1;
}

// Agregamos el cero cuando el mes es menor a 10
if($mesCobro<10){
    $mesCobro = "0$mesCobro";
}

//-----RESTO DE INFORMACION-----

// Generamos el numero de cuota
$iden_cuota = mt_rand(999999999,999999999);

// Guardamos en una variable el dia a cobrar las cuotas o la
prima total
if($cantCuotas==1){
    $fecha = $fechaCobrar;
}

```

```

}else{
    $fecha = $fechaCobrarCuota;
}

$estado = 0;

//-----INSERION BBDD-----

$stmt = $mysqli->prepare($insCuotas);
$exe = $stmt-
>bind_param('ssissssi', $i, $iden_cuota, $idPago, $primaMensual, $lapso, $fec
ha, $vto_pago, $estado);

if($stmt->execute()){
    $respuesta = true;
}else{
    echo $stmt->error;
}
$stmt->close();

return $respuesta;
}

// Función para mostrar la información de un pago.
public function mostrarInfoPago($post)
{
    $arr = array();
    $idPol = $post['idPol'];
    $mysqli = $this->conexion;

    $sql = "SELECT P.id, P.prima_total, C.prima_mensual, C.dia_cobrar
        FROM pagos P
        INNER JOIN cuotas C ON C.pago_id = P.id
        WHERE P.poliza_id = ?";
    $stmt = $mysqli->prepare($sql);
    $stmt->bind_param('i', $idPol);

    if($stmt->execute()){
        $resultado = $stmt->get_result();
        if($resultado->num_rows > 0){

            $cantCuotas = $resultado->num_rows;

            while($dato = $resultado->fetch_array()){

                $arr = array();
                $arr['id'] = $dato[0];
                $arr['primaTotal'] = $dato[1];
                $arr['primaMensual'] = $dato[2];
                $arr['diaCobrar'] = $dato[3];

                $arr['cantCuotas'] = $cantCuotas;
                $arr['success'] = true;
            }
        }
    }
}

```

```

        }

    }

$stmt->close();
echo json_encode($arr);
}

// Función para mostrar el listado de cuotas de un pago.
public function mostrarCuotas($post)
{
    $idPago = $post['idPago'];
    $arr = array();
    $tbody = array();
    $mysqli = $this->conexion;

    $sql = "SELECT C.numero_cuota, C.iden_cuota, C.prima_mensual, C.lapso
, C.dia_cobrar, C.vto_pago, EC.nombre, PO.nro
    FROM pagos P
    INNER JOIN cuotas C ON C.pago_id = P.id
    INNER JOIN poliza PO ON PO.id = P.poliza_id
    INNER JOIN estado_cuota EC ON EC.id = C.estado
    WHERE P.id = ?";

    $stmt = $mysqli->prepare($sql);
    if($stmt!==FALSE){

        $stmt->bind_param('i', $idPago);
        $stmt->execute();
        $re = $stmt->get_result();

        if($re->num_rows > 0){
            while ($cuota = $re->fetch_array() ) {
                $arr['numCuota'] = $cuota[0];
                $arr['idenCuota'] = $cuota[1];
                $arr['priMen'] = $cuota[2];
                $arr['lapso'] = $cuota[3];
                $arr['diaCobrar'] = $cuota[4];
                $arr['vtoPago'] = $cuota[5];
                $arr['estado'] = $cuota[6];
                $arr['numPol'] = $cuota[7];

                $arr['success'] = true;

                // Devolvemos los datos con la estructura de una tabla.
                $infoTabla =
                    '<tr>
                        <td>' . $arr['numCuota'] . ' / ' . $re->num_rows . '</td>
                        <td>' . $arr['idenCuota'] . '</td>
                        <td>' . $arr['priMen'] . '</td>
                        <td>' . $arr['lapso'] . '</td>
                        <td>' . $arr['diaCobrar'] . '</td>
                        <td>' . $arr['vtoPago'] . '</td>
                        <td>' . $arr['estado'] . '</td>
                    
```

```

        </tr>';

        $tbody[] = $infoTabla;
    }
}

$stmt->close();
return json_encode(array('data' => $arr, 'tbody' => $tbody));
}

// Función para mostrar generar el formulario al facturar una cuota.
public function facturarPago($post)
{
    $idPago = $post[' pagoId'];
    $arr['success'] = false;
    $mysqli = $this->conexion;

    $sql = "SELECT PO.nro, CONCAT(T.nombre, ' (' ,T.num_tom, ')'), P.prima_total
            FROM pagos P
            INNER JOIN poliza PO ON PO.id = P.poliza_id
            INNER JOIN tomador T ON T.id = PO.clienteid
            WHERE P.id = ?";

    $stmt = $mysqli->prepare($sql);
    if($stmt!==FALSE){

        $stmt->bind_param('i', $idPago);
        $stmt->execute();
        $re = $stmt->get_result();

        if($re->num_rows > 0){
            while ($cuota = $re->fetch_array() ) {
                $arr['numPol'] = $cuota[0];
                $arr['tomador'] = $cuota[1];
                $arr['primaTot'] = $cuota[2];

                $arr['success'] = true;
            }
        }
    }
    $stmt->close();
    return json_encode($arr);
}

// Función para traer la información de un recibo de pago.
public function traerDatosReciboCuota($post)
{
    $idCuota = $post['idCuota'];
    $arr['success'] = false;
    $mysqli = $this->conexion;
}

```

```

// Almacenamos la información en una sesión y si existe, la
eliminamos para evitar mostrar siempre los mismos datos.
session_start();

if(isset($_SESSION['recibo'])){
    unset($_SESSION["recibo"]);
}

$sql = "SELECT CU.iden_cuota,
            POL.nro,
            CONCAT(CLIENTE.nombre, '(',CLIENTE.documento,') / ',CLIENTE.num_tom
),
            CONCAT(CLIENTE.calle, ' ',proper(LOCALIDAD.nombre), ' - ',PROVINCIA.nom
bre, ' (' ,LOCALIDAD.codigopostal,')'),
            CU.dia_cobrar,
            CONCAT(POL.vigencia_inicio, ' - ',POL.vigencia_fin),
            CU.vto_pago,
            CU.prima_mensual,
            CONCAT(MARQUE.nombre, ' ',MODELO.nombre, ' ',VEHICULO.motor),
            VEHICULO.anio,
            VEHICULO.patente,
            VEHICULO.nroMotor,
            VEHICULO.suma_asegurada,
            COBERTURA.descripcion,
            COBAD.nombre,
            PA.num,
            ESCU.nombre,
            CU.lapso,
            CU.numero_cuota,
            (SELECT COUNT(*) FROM cuotas CUOT WHERE CUOT.pago_id =
(SELECT pago_id FROM cuotas WHERE cuotas.id = $idCuota)),
            CU.fecha_cobro
        FROM poliza POL
        INNER JOIN vehiculo VEHICULO ON VEHICULO.id = POL.vehiculoid
        INNER JOIN tomador CLIENTE ON CLIENTE.id = POL.clienteid
        INNER JOIN provincia PROVINCIA ON PROVINCIA.id = CLIENTE.provincia
        INNER JOIN localidad LOCALIDAD ON LOCALIDAD.id = CLIENTE.localidad AND LOCALIDAD.co
digopostal = CLIENTE.cp
        INNER JOIN pagos PA ON PA.poliza_id = POL.id
        INNER JOIN cuotas CU ON CU.pago_id = PA.id
        INNER JOIN marca MARQUE ON MARQUE.id = VEHICULO.marca_id
        INNER JOIN modelo MODELO ON MODELO.id = VEHICULO.modelo_id
        INNER JOIN cobertura COBERTURA ON COBERTURA.id = POL.coberturaid
        INNER JOIN cobertura_adicional COBAD ON COBAD.id = VEHICULO.cobertu
raAd
        INNER JOIN estado_cuota ESCU ON ESCU.id = CU.estado
        WHERE CU.id = ?";

$stmt = $mysqli->prepare($sql);
$stmt->bind_param('i', $idCuota);
$ejecutar = $stmt->execute();

if($ejecutar){
    $resultado = $stmt->get_result();
}

```

```

        if($r->num_rows > 0){
            while($dato = $r->fetch_array()){
                $arr['nroCu'] = $dato[0];
                $arr['numPol'] = $dato[1];
                $arr['tom'] = $dato[2];
                $arr['dom'] = $dato[3];
                $arr['fact'] = $dato[4];
                $arr['vig'] = $dato[5];
                $arr['vtoPa'] = $dato[6];
                $arr['priMen'] = $dato[7];
                $arr['veh'] = $dato[8];
                $arr['anioVe'] = $dato[9];
                $arr['pat'] = $dato[10];
                $arr['nroMot'] = $dato[11];
                $arr['sumAs'] = $dato[12];
                $arr['cob'] = $dato[13];
                $arr['cobAd'] = $dato[14];
                $arr['nroPa'] = $dato[15];
                $arr['estCu'] = $dato[16];
                $arr['lapCu'] = $dato[17];
                $arr['cuotNro'] = $dato[18];
                $arr['cantCuo'] = $dato[19];
                $arr['fechCob'] = $dato[20];
                $arr['success'] = true;
            }
        }
    }

    $stmt->close();

    $_SESSION['recibo'] = $arr;
    return json_encode($arr);
}

// Función para efectuar el pago de una cuota.
public function efectuarPago($post)
{
    $cuotaId = $post['cuotaId'];
    $arr['success'] = false;
    $arr['permitido'] = false;
    $mysqli=$this->conexion;
    $estado = 2; // Estado 2 = Pago aprobado.

    // Comprobamos si el estado de la cuota es “A facturar” o “Pago aprobado”.
    $comprobar = "SELECT estado FROM cuotas WHERE id = ?";
    $stmtC = $mysqli->prepare($comprobar);
    $stmtC->bind_param('i', $cuotaId);
    if($stmtC->execute()){
        $r = $stmtC->get_result();
        $data = $r->fetch_assoc();
        if($data['estado']==0 || $data['estado']==2){
            $arr['permitido'] = true;
        }
    }
}

```

```

        }else{
            $arr['permitido'] = true;
        }
    }
$stmtC->close();

// Si el pago es permitido
if($arr['permitido']){

    $fechaActualCobro = Date("d/m/Y").' - '.Date("H:i"); //Obtenemos
la fecha actual

    $sql = "UPDATE cuotas
            SET estado      = ?,
                fecha_cobro = ?
            WHERE id       = ?";
    $stmt = $mysqli->prepare($sql);
    $stmt->bind_param('isi', $estado, $fechaActualCobro, $cuotaId);

    if($stmt->execute()){
        $arr['success'] = true;
        $arr['fechaCobro'] = $fechaActualCobro;
    }else{
        echo $stmt->error;
    }

    $stmt->close();
}

return json_encode($arr);
}

// Función que actualiza los estados de las cuotas.
public function actualizarEstadoCuota()
{
    $arr['actCuota'] = false;
    $mysqli = $this->conexion;
    $fechaActual = new DateTime( str_replace("/", "-", Date("d/m/Y")) );

    // Obtenemos todas las cuotas de todos los pagos.
    $sql = "SELECT C.id as cuotaId, C.dia_cobrar as dia_cobrar, C.vto_pag
o as vto_pago, P.fecha_registro as fecha_registro, C.estado as estado, P.estad
o as estadoPago
        FROM cuotas C
        INNER JOIN pagos P ON P.id = C.pago_id";
    $stmt = $mysqli->prepare($sql);
    if($stmt){
        $stmt->execute();
        $r = $stmt->get_result();

        // Obtenemos todas las fechas de cada una.
        foreach($r as $var){
            $diaCobrar = new DateTime(str_replace("/", "-",
", $var['dia_cobrar']));
        }
    }
}

```

```

        $vtoPago = new DateTime(str_replace("/", " - ", $var['vto_pago']));
        $fechaRegistro = new DateTime(str_replace("/", " - ", $var['fecha_registro']));

        //Si el estado de la cuota es diferente de aprobado y rechazado, y el pago es distinto a caducado.
        if($var['estado']!=2 && $var['estado']!=3 && $var['estadoPago']!=2){

            if($fechaActual>=$fechaRegistro && $fechaActual<$diaCobrar){
                //A facturar
                $estadoCuota = 0;
                $idCuota = $var['cuotaId'];
                var_dump($idCuota.'= A facturar');

                $uCuota = $mysqli->prepare("UPDATE cuotas SET estado = $estadoCuota WHERE id = $idCuota");

                }else if($fechaActual>=$diaCobrar && $fechaActual<$vtoPago){

                    //Facturada (Sin pagar)
                    $estadoCuota = 1;
                    $idCuota = $var['cuotaId'];
                    var_dump($idCuota.'=Facturada sin pagar');

                    $uCuota = $mysqli->prepare("UPDATE cuotas SET estado = $estadoCuota WHERE id = $idCuota");

                    }else if($fechaActual>=$vtoPago){
                        //Deuda
                        $estadoCuota = 4;
                        $idCuota = $var['cuotaId'];
                        var_dump($idCuota.'= Deuda');

                        $uCuota = $mysqli->prepare("UPDATE cuotas SET estado = $estadoCuota WHERE id = $idCuota");

                        }else{
                            $estadoCuota = NULL;
                        }

                //Por cada ciclo ejecutamos la consulta y actualizamos su estado.
                if($uCuota){
                    $uCuota->execute();
                    $uCuota->close();
                    $arr['actCuota'] = true;
                }else{
                    echo $uCuota->error;
                }
            }
    
```

```

        }
        $stmt->close();
    }

    return json_encode($arr);
}

//Funcion para renovar una poliza.
public function comprobarDeuda($post)
{
    $mysqli = $this->conexion;
    $info = $post['data'];
    $idPoliza = intval($info[0]);

    $arr['success'] = false;
    $r = false;
    $fact = 0;
    $apro = 0;
    $rech = 0;
    $deud = 0;

    //Comprobamos si la poliza a renovar se encuentra con deudas.
    $sql = "SELECT C.iden_cuota as ideCuo, C.estado as estado, EC.nombre
as estadoCuota, P.num as nroPago
FROM cuotas C
INNER JOIN estado_cuota EC ON EC.id = C.estado
INNER JOIN pagos P ON P.id = C.pago_id
INNER JOIN poliza PO ON PO.id = P.poliza_id
WHERE PO.id = ?";

    $stmt10 = $mysqli->prepare($sql);
    if($stmt10->bind_param('i', $idPoliza)){
        $stmt10->execute();
        $re = $stmt10->get_result();

        // Sumamos un contador por cada tipo de estado posible.
        foreach($re as $cuota){
            if($cuota['estado'] == 1){
                $fact = $fact + 1;

            }else if($cuota['estado'] == 2){
                $apro = $apro + 1;

            }else if($cuota['estado'] == 3){
                $rech = $rech + 1;

            }else if($cuota['estado'] == 4){
                $deud = $deud + 1;
            }
        }

        $nroPago = $cuota['nroPago'];
    }
}

```

```

        $arr[ 'fact' ]      = $fact;
        $arr[ 'apro' ]       = $apro;
        $arr[ 'rech' ]       = $rech;
        $arr[ 'deud' ]       = $deud;
        $arr[ 'cantCuotas' ] = $re->num_rows;
        $arr[ 'nroPago' ]     = $nroPago;
        $arr[ 'success' ]    = true;
    }

    $stmt10->close();
    return json_encode($arr);
}

// Función para enviar un recibo de pago.
public function enviarRecibo($post)
{
    if(!isset($_SESSION))
    {
        session_start();
    }

    if(isset($_SESSION['urlImg'])){
        unset($_SESSION['urlImg']);
    }

    $cuotaId = $post['cuotaId'];
    $arr[ 'success' ] = false;

    $sql = "SELECT T.correo AS correo, T.nombre AS tomador, PA.num AS nro
Pago, CU.iden_cuota AS nroCuota
    FROM tomador T
    INNER JOIN poliza P ON P.clienteid = T.id
    INNER JOIN pagos PA ON PA.poliza_id = P.id
    INNER JOIN cuotas CU ON CU.pago_id = PA.id AND CU.id = ?";
}

$mysqli = $this->conexion;
$stmt = $mysqli->prepare($sql);
if($stmt != FALSE){
    $stmt->bind_param('i', $cuotaId);
    $stmt->execute();
    $r = $stmt->get_result();

    if($r->num_rows > 0){
        foreach($r as $var){

            // Si existe un correo o no.
            $correo = $var['correo'];
            if($correo != '' || $correo != NULL){
                $arr[ 'existeCorreo' ] = 'si';
            }else{
                $arr[ 'existeCorreo' ] = 'no';
            }
            $arr[ 'correo' ] = $correo;
        }
    }
}

```

```

        $arr['tomador'] = $var['tomador'];
        $arr['nroPago'] = $var['nroPago'];
        $arr['nroCuota'] = $var['nroCuota'];
        $arr['success'] = true;

        // Guardamos el principio de la ruta en una sesión para
        // usarla posteriormente.
        $_SESSION['urlImg'] = '../../../../../';
    }
}

$stmt->close();
return json_encode($arr);
}

// Funcion para mostrar el listado de recibos de una poliza.
public function mostrarListadoRecibos($post)
{
    /*Truncamos la sesión de recibo para evitar cargar esa información*/
    session_start();
    if(isset($_SESSION['recibo'])){
        unset($_SESSION["recibo"]);
    }
    /*Truncamos la sesión de info para poder modificar la información sol
    icitada*/
    if(isset($_SESSION['infoListadoRecibos'])){
        unset($_SESSION["infoListadoRecibos"]);
    }

    $pagoId = $post['pagoId'];
    $mysqli = $this->conexion;
    $r['success'] = false;

    $sql = "SELECT CU.iden_cuota,
                  POL.nro,
                  CONCAT(CLIENTE.nombre, '(', CLIENTE.documento, ') / ', CLIENTE.num_tom
    ),
                  CONCAT(CLIENTE.calle, ' ', proper(LOCAL.nombre), ' - ', PROPIETARIO.nombre, ' (', LOCAL.codigopostal, ')'),
                  CU.dia_cobrar,
                  CONCAT(POL.vigencia_inicio, ' - ', POL.vigencia_fin),
                  CU.vto_pago,
                  CU.prima_mensual,
                  CONCAT(MAR.nombre, ' ', MODELO.nombre, ' ', VEHICULO.motor),
                  VEHICULO.anio,
                  VEHICULO.patente,
                  VEHICULO.nroMotor,
                  VEHICULO.suma_asegurada,
                  COB.descripcion,
                  COBAD.nombre,
                  CU.lapso,
                  CLIENTE.correo
    FROM poliza POL
    INNER JOIN vehiculo VEHICULO ON VEHICULO.id = POL.vehiculoid

```

```

        INNER JOIN tomador CLI ON CLI.id = POL.clienteid
        INNER JOIN provincia PRO ON PRO.id = CLI.provincia
        INNER JOIN localidad LOC ON LOC.id = CLI.localidad AND LOC.co
digopostal = CLI.cp
        INNER JOIN pagos PA ON PA.poliza_id = POL.id
        INNER JOIN cuotas CU ON CU.pago_id = PA.id
        INNER JOIN marca MAR ON MAR.id = VE.marca_id
        INNER JOIN modelo MO ON MO.id = VE.modelo_id
        INNER JOIN cobertura COB ON COB.id = POL.coberturaid
        INNER JOIN cobertura_adicional COBAD ON COBAD.id = VE.cobertu
raAd

        WHERE PA.id = ?";

$stmt = $mysqli->prepare($sql);

if($stmt->bind_param('i', $pagoId)){
    $stmt->execute();
    $rs    = $stmt->get_result();
    $filas = $rs->num_rows;
    if ($filas > 0){
        while ($info = $rs->fetch_array()) {
            $arr           = array();
            $arr['nroCu']   = $info[0];
            $arr['numPol']  = $info[1];
            $arr['tom']     = $info[2];
            $arr['dom']     = $info[3];
            $arr['fact']    = $info[4];
            $arr['vig']     = $info[5];
            $arr['vtoPa']   = $info[6];
            $arr['priMen']  = $info[7];
            $arr['veh']      = $info[8];
            $arr['anioVe']  = $info[9];
            $arr['pat']      = $info[10];
            $arr['nroMot']   = $info[11];
            $arr['sumAs']    = $info[12];
            $arr['cob']      = $info[13];
            $arr['cobAd']    = $info[14];
            $arr['lapCu']    = $info[15];
            $arr['email']    = $info[16];

            $r['success'] = true;

            $arr1[]          = $arr;
        }

        $arr1['filas'] = $filas;
        $arr1['urlImg'] = ' ../../';

        // Guardamos la información en una sesión para usarla
posteriormente.
        $_SESSION['infoListadoRecibos'] = $arr1;
    }
}

```

```
        return json_encode(array('resp'=>$r, 'info'=>$arr1));
    }
}
```

➔ 4.3.2.4 fichero “polizasController.php”

Este fichero contiene todas las funciones que realizan las acciones relacionadas a las pólizas en el sistema.

```
<?php

// Clase principal.

class PolizasController
{
    private $conexion;

    // Funcion constructora donde realizamos la conexión a la base de datos.
    public function __construct()
    {
        $this->conexion = Database::connect();
        date_default_timezone_set("America/Argentina/Buenos_Aires");
    }

    // Función donde traemos el listado de todas las marcas de autos para mostrar en el select.
    public function listadoMarcas()
    {
        $marcas = '';
        $mysqli = $this->conexion;

        $sql = "SELECT id, nombre FROM marca";

        $stmt = $mysqli->prepare($sql);
        if($stmt !== FALSE){
            $stmt->execute();

            $rs = $stmt->get_result();
            if ($rs->num_rows > 0) {

                // Mostramos el listado concatenado en una variable y en formato HTML para un select.
                $marcas .= '<option value="0" selected disabled>Seleccione</option>';
                foreach($rs as $marca){
                    $marcas .= '<option value="'. $marca['id'] .'>' . $marca['nombre'] . '</option>';
                }
            }
        }
    }
}
```

```

        $stmt->close();

        return $marcas;
    }
}

// Función para mostrar el listado de modelos de una marca.
public function buscarmodelos($post)
{
    $idMarca = $post['marca'];
    $modelos = array();
    $mysqli = $this->conexion;

    $sql = "SELECT id, nombre FROM modelo WHERE id_marca = ?";
    $stmt = $mysqli->prepare($sql);
    if($stmt !== FALSE){
        $stmt->bind_param('i', $idMarca);
        $stmt->execute();

        $rs = $stmt->get_result();
        if ($rs->num_rows > 0) {
            while ($modelo = $rs->fetch_array()) {
                $arr = array();
                $arr['id'] = $modelo[0];
                $arr['nombre'] = $modelo[1];
                $modelos[] = $arr;
            }
        }
        $stmt->close();
        echo json_encode(array('data' => $modelos));
    }
}

// Función para mostrar el listado de clases de vehiculos.
public function listadoClases()
{
    $mysqli = $this->conexion;
    $clases = '';

    $sql = "SELECT id, nombre FROM clase_vehiculo";
    $stmt = $mysqli->prepare($sql);
    if($stmt != FALSE){
        $stmt->execute();
        $rs = $stmt->get_result();
        if($rs->num_rows > 0){

```

```

        // Lo concatenamos en una variable en formato HTML.
        $clases .= '<option value="0" selected disabled>Selección</option>';
        foreach($rs as $clase){
            $clases .= '<option value="'. $clase['id'].'">' . $clase['nombre'] . '</option>';
        }
    }
    $stmt->close();

    return $clases;
}

// Función donde mostramos el listado de tipos de uso de un
vehículo.
public function listadoUsos()
{
    $mysqli = $this->conexion;
    $usos = '';

    $sql = "SELECT id, nombre FROM uso_vehiculo";
    $stmt = $mysqli->prepare($sql);
    if($stmt!=FALSE){
        $stmt->execute();
        $rs = $stmt->get_result();
        if($rs->num_rows > 0){

            // Concatenamos los resultados en una variable y lo
            mostramos en formato HTML para mostrarlo en un select directamente.
            $usos .= '<option value="0" selected disabled>Selección</option>';
            foreach($rs as $uso){
                $usos .= '<option value="'. $uso['id'].'">' . $uso['nombre'] . '</option>';
            }
        }
    }
    $stmt->close();

    return $usos;
}

// Función para mostrar el listado de coberturas adicionales.
public function listadoCoberturasAd()
{
    $mysqli = $this->conexion;
    $cobAd = '';

```

```

$sql = "SELECT id, nombre FROM cobertura_adicional";
$stmt = $mysqli->prepare($sql);
if($stmt!=FALSE){
    $stmt->execute();
    $rs = $stmt->get_result();
    if($rs->num_rows > 0){

        // Concatenamos los resultados en una variable en
        formato HTML para mostrarlo directamente en un select.
        $cobAd .= '<option value="0" selected disabled>Selecci
        one</option>';
        foreach($rs as $coA){
            $cobAd .= '<option value="'. $coA['id'].'">' . $coA['
        nombre'] . '</option>';
        }
    }
    $stmt->close();
}

return $cobAd;
}

// Función para mostrar el listado de coberturas para una poliza.
public function listadoCoberturas()
{
    $info = array();
    $mysqli = $this->conexion;

    $sql = "SELECT id, nombre, descripcion, informacion FROM cober
tura";

    $stmt = $mysqli->prepare($sql);
    if($stmt==FALSE){
        $stmt->execute();
        $rs = $stmt->get_result();

        if ($rs->num_rows>0){
            foreach($rs as $cob){
                $arr = array();
                $arr['id'] = $cob['id'];
                $arr['nombre'] = $cob['nombre'];
                $arr['descripcion'] = $cob['descripcion'];
                $arr['informacion'] = $cob['informacion'];
                $info[] = $arr;
            }
        }
    }

    $stmt->close();
}

```

```

        return json_encode($info);
    }

}

// Función donde grabamos el vehiculo de una poliza.
public function grabarVehiculo($post)
{
    try{

        $mysqli = $this->conexion;

        $error = false;

        $arr = array('success'=>false);

        $datos = $post['datosVehiculo'];

        $datosPoliza = $post['datosPoliza'];

        // Validamos todos los datos.
        $sumAs = filter_var($datos[0], FILTER_SANITIZE_FULL_SPECI
AL_CHARS);
        if($sumAs==FALSE || is_null($sumAs)) {$error = true;}

        $mar = filter_var($datos[1], FILTER_VALIDATE_INT);
        if($mar==FALSE || is_null($mar)) {$error = true;}

        $mod = filter_var($datos[2], FILTER_VALIDATE_INT);
        if($mod==FALSE || is_null($mod)) {$error = true;}

        $patente = filter_var($datos[3], FILTER_SANITIZE_FULL_SPEC
IAL_CHARS);
        if($patente==FALSE || is_null($patente)) {$error = true;}

        $motor = filter_var($datos[4], FILTER_SANITIZE_FULL_SPECIA
L_CHARS);
        if($motor==FALSE || is_null($motor)) {$error = true;}

        $nroChasis = filter_var($datos[5], FILTER_SANITIZE_FULL_SP
ECIAL_CHARS);
        if($nroChasis==FALSE || is_null($nroChasis)) {$error = tr
ue;}


        $nroMotor = filter_var($datos[6], FILTER_SANITIZE_FULL_SPE
CIAL_CHARS);
        if($nroMotor==FALSE || is_null($nroMotor)) {$error = true
;}}
    }
}

```

```

$clase = filter_var($datos[7], FILTER_VALIDATE_INT);
if($clase === FALSE || is_null($clase)) {$error = true;}

$uso = filter_var($datos[8], FILTER_VALIDATE_INT);
if($uso === FALSE || is_null($uso)) {$error = true;}

$gps = filter_var($datos[9], FILTER_SANITIZE_FULL_SPECIAL_CHARS);
if($gps === FALSE || is_null($gps)) {$error = true;}

$ceroKm = filter_var($datos[10], FILTER_SANITIZE_FULL_SPECIAL_CHARS);
if($ceroKm === FALSE || is_null($ceroKm)) {$error = true;}

$kmS = filter_var($datos[11], FILTER_SANITIZE_FULL_SPECIAL_CHARS);
if($kmS === FALSE || is_null($kmS)) {$error = true;}

$codigoPostal = filter_var($datos[12], FILTER_VALIDATE_INT);
if($codigoPostal === FALSE || is_null($codigoPostal)) {$error = true; }

$localidad = filter_var($datos[13], FILTER_VALIDATE_INT);
if($localidad === FALSE || is_null($localidad)) {$error = true; }

$coberturaAd = filter_var($datos[14], FILTER_VALIDATE_INT);
if($coberturaAd === FALSE || is_null($coberturaAd)) {$error = true; }

$combustible = filter_var($datos[15], FILTER_SANITIZE_FULL_SPECIAL_CHARS);
if($combustible === FALSE || is_null($combustible)) {$error = true; }

$eGas = filter_var($datos[16], FILTER_SANITIZE_FULL_SPECIAL_CHARS);
if($eGas === FALSE || is_null($eGas)) {$error = true; }

$anio = filter_var($datos[17], FILTER_VALIDATE_INT);
if($anio === FALSE || is_null($anio)) {$error = true; }

$pasajeros = filter_var($datos[18], FILTER_SANITIZE_FULL_SPECIAL_CHARS);
if($pasajeros === FALSE) {$error = true; }

```



```

        $rs = $stmt->get_result();
        if ($rs->num_rows > 0) {
            while ($fila = $rs->fetch_array()) {
                $idVehiculo = $fila[0];
            }

            // Grabamos el pago de la poliza desde la
            // función.
            $datosPago = $post['datosPago'];
            $r = $this-
>grabarPoliza($datosPoliza, $idVehiculo, $sumAs, $datosPago);
            if($r==true){
                $arr = array('success'=>true);
            }
        }

        $stmt->close();
    }

    return json_encode($arr);

}catch(Exception $e){
    return $e->getMessage();
}
}

// Función donde traemos los datos de la poliza y la mostramos en
el apartado “Consultar poliza”.
public function mostrarInformacion($post)
{
    $mysqli = $this->conexion;

    $idpoliza = intval($post['idpoliza']);

    $arr = array('success'=>false);

    $sql = "SELECT PO.nro, PO.fecha_emision, PO.horaEmis, PO.vige
nacia_inicio, PO.vigencia_fin, PO.meses_vig, PO.anulacion,
/*INFO POLIZA*/
    (SELECT COUNT(id) FROM endoso WHERE polizaId = ?) A
S num_endosos,
    (SELECT COUNT(id) FROM siniestro WHERE idPoliza = ?) A
S num_siniestros,
    CONCAT(CO.nombre, ' - ', CO.descripcion) A
S cobertura,
    COAD.nombre A
S cobAdicionalM,

```

```

        USU.nombre AS emisor,
        USU.nro AS nroEmisor,
        SUC.nombre AS nombreSuc,
        SUC.iden_sucursal AS idenSuc,

        /*INFO TOMADOR*/
        CONCAT(TOM.calle, ', ',(SELECT proper(nombre) FROM localidad WHERE id = TOM.localidad),',',(SELECT nombre FROM provincia WHERE id = TOM.provincia)) AS domicilio,
        TOM.num_tom, TOM.nombre, TOM.documento, TOM.persona, TOM.nacionalidad, TOM.cp, TOM.fecha_nac, TOM.genero, TOM.telefono, TOM.correo,

        /*INFO PAGO*/
        PA.num, PA.fecha_registro, PA.prima_total, PA.metodoPago, (SELECT nombre from estado_pago WHERE id = PA.estado) AS estadoPago,

        /*INFO VEHICULO*/
        (SELECT nombre FROM marca WHERE id = VE.marca_id)
        AS marca,
        (SELECT nombre FROM modelo WHERE id = VE.modelo_id)
        AS modelo,
        (SELECT nombre FROM clase_vehiculo WHERE id = VE.clase
)
        AS clase,
        (SELECT nombre FROM uso_vehiculo WHERE id = VE.uso)
        AS uso,
        (SELECT proper(nombre) FROM localidad WHERE id = VE.localidad) AS veLocalidad,
        VE.suma_asegurada, VE.motor, VE.patente, VE.nroChasis,
        VE.nroMotor, VE.gps, VE.0Km, VE.Kilometros, VE.codigoPostal, VE.combustible, VE.eGas, VE.anio, VE.color, VE.asientos, VE.pasajeros
        FROM poliza PO
        INNER JOIN cobertura CO          ON CO.id = PO.coberturaId
        INNER JOIN cobertura_adicional COAD ON COAD.id = (SELECT coberturaAd FROM vehiculo WHERE id = PO.vehiculoid)
        INNER JOIN tomador TOM           ON TOM.id = PO.clienteId
        INNER JOIN pagos PA             ON PA.poliza_id = PO.id
        INNER JOIN vehiculo VE           ON VE.id = PO.vehiculoid
        INNER JOIN sucursales SUC         ON SUC.id = PO.sucursal
        INNER JOIN usuarios USU          ON USU.id = PO.emisor
        WHERE PO.id = ?";

```

```

$sqlPago = "SELECT numero_cuota, iden_cuota, prima_mensual, (S
ELECT nombre FROM estado_cuota WHERE id = CU.estado) AS estadoCuota
                    FROM cuotas CU
                    WHERE pago_id = (SELECT id FROM pagos WHERE poliza
_id = ?)";

$sqlEstados = "SELECT CONCAT (EP.nombre,': Desde ',LT.inicio_l
apso,' hasta ',LT.fin_lapso) as estadoPoliza
                    FROM ltestpol LT
                    INNER JOIN estadopoliza EP ON EP.id = LT.estado
                    WHERE polizaid = ?
                    ORDER BY nrohilo DESC";

$stmt1 = $mysqli->prepare($sql);

$stmt2 = $mysqli->prepare($sqlPago);

$stmt3 = $mysqli->prepare($sqlEstados);

if($stmt1&&$stmt2&&$stmt3){

    $stmt1-
>bind_param('iii', $idpoliza, $idpoliza, $idpoliza);

    if($stmt1->execute()){

        $r1 = $stmt1->get_result();
        foreach($r1 as $v1){
            $info[] = '';
            //poliza
            $info[0] = $v1['nro'];
            $info[1] = $v1['fecha_emision'];
            $info[2] = $v1['horaEmis'];
            $info[3] = $v1['vigencia_inicio'];
            $info[4] = $v1['vigencia_fin'];
            $info[5] = $v1['meses_vig'];
            $info[6] = $v1['anulacion'];
            $info[7] = $v1['num_endosos'];
            $info[8] = $v1['num_siniestros'];
            $info[9] = $v1['cobertura'];
            $info[10] = $v1['cobAdicionalM'];
            $info['ne'] = $v1['emisor'];
            $info['nroe'] = $v1['nroEmisor'];
            $info['ns'] = $v1['nombreSuc'];
            $info['is'] = $v1['idenSuc'];
            //tomador
            $info[11] = $v1['domicilio'];
            $info[12] = $v1['num_tom'];
            $info[13] = $v1['nombre'];
    }
}

```

```

$info[14] = $v1['documento'];
$info[15] = $v1['persona'];
$info[16] = $v1['nacionalidad'];
$info[17] = $v1['cp'];
$info[18] = $v1['fecha_nac'];
$info[19] = $v1['genero'];
$info[20] = $v1['telefono'];
$info[21] = $v1['correo'];
//pago
$info[22] = $v1['num'];
$info[23] = $v1['fecha_registro'];
$info[24] = $v1['prima_total'];
$info[25] = $v1['metodoPago'];
$info[26] = $v1['estadoPago'];
//vehiculo
$info[27] = $v1['marca'];
$info[28] = $v1['modelo'];
$info[29] = $v1['clase'];
$info[30] = $v1['uso'];
$info[31] = $v1['veLocalidad'];
$info[32] = $v1['suma_asegurada'];
$info[33] = $v1['motor'];
$info[34] = $v1['patente'];
$info[35] = $v1['nroChasis'];
$info[36] = $v1['nroMotor'];
$info[37] = $v1['gps'];
$info[38] = $v1['0Km'];
$info[39] = $v1['Kilometros'];
$info[40] = $v1['codigoPostal'];
$info[41] = $v1['combustible'];
$info[42] = $v1['eGas'];
$info[43] = $v1['anio'];
$info[44] = $v1['color'];
$info[45] = $v1['asientos'];
$info[46] = $v1['pasajeros'];

$arr['info'] = $info;

$stmt2->bind_param('i', $idpoliza);

//Listado de cuotas
if($stmt2->execute()){
    $r2 = $stmt2->get_result();
    if($r2->num_rows > 0){
        while($v2 = $r2->fetch_array()){
            $cuota['numero_cuota'] = $v2[0];
            $cuota['iden_cuota'] = $v2[1];
            $cuota['prima_mensual'] = $v2[2];
            $cuota['estadoCuota'] = $v2[3];
        }
    }
}

```

```

                $cuotas[] = $cuota;
            }
        }

        $arr['cuotas'] = $cuotas;

        //Listado de Estados de la poliza
        $stmt3->bind_param('i', $idpoliza);
        if($stmt3->execute()){
            $r3 = $stmt3->get_result();
            if($r3->num_rows > 0){
                while($v3 = $r3->fetch_array()){
                    $estado['estado'] = $v3[0];
                    $estados[] = $estado;
                }
            }

            $arr['estados'] = $estados;
            $arr['success'] = true;
        }
    }

    $stmt1->close();
    $stmt2->close();
    $stmt3->close();
}

return json_encode(array('info'=>$arr['info'], 'cuotas'=>$arr['cuotas'], 'estados'=>$arr['estados'], 'success'=>$arr['success']));
}

// Función donde grabamos la poliza.
public function grabarPoliza($datosPoliza, $idVehiculo, $sumAs, $datosPago)
{
    try{

        $mysqli = $this->conexion;

        $error = false;

        $res = false;

        $datosPoliza = $datosPoliza;

        $idVehiculo = $idVehiculo;
    }
}

```

```

        // Validamos los datos.
        $nroPol = filter_var($datosPoliza[0], FILTER_SANITIZE_FULL
_SPECIAL_CHARS);
        if($nroPol === FALSE || is_null($nroPol)) {$error = true;}

        $vig_ini = filter_var($datosPoliza[1], FILTER_SANITIZE_FUL
L_SPECIAL_CHARS);
        if($vig_ini === FALSE || is_null($vig_ini)) {$error = true;}

        $vig_fin = filter_var($datosPoliza[2], FILTER_SANITIZE_FUL
L_SPECIAL_CHARS);
        if($vig_fin === FALSE || is_null($vig_fin)) {$error = true;}

        $cli = filter_var($datosPoliza[3], FILTER_VALIDATE_INT);
        if($cli === FALSE || is_null($cli)) {$error = true;}

        $cob = filter_var($datosPoliza[4], FILTER_VALIDATE_INT);
        if($cob === FALSE || is_null($cob)) {$error = true;}

        $mesesVig = filter_var($datosPoliza[5], FILTER_SANITIZE_FU
LL_SPECIAL_CHARS);
        if($mesesVig === FALSE || is_null($mesesVig)) {$error = true
;}

        // Iniciamos la sesión y traemos los datos del usuario y
la sucursal desde sesiones existentes.
        if(!isset($_SESSION)){
            session_start();
        }
        if(isset($_SESSION['user'])){
            $emis = $_SESSION['user']['id'];
            $sucursal = $_SESSION['user']['sucursal'];
        }

        //Reemplazamos los "/" por "-"
"" en la fecha para poder sumarle un mes.
        $finVig = str_replace("/", "-", $vig_fin);
        $f = new DateTime($finVig);
        $f->modify('next month');
        $anulacion = $f->format('d/m/Y');

        if(!$error){

            $estado = 5;
            $fechaRegistro = Date("d/m/Y");
            $horaRegistro = Date("H:i");
            $horaInVig = '11:59';

            $mysqli->set_charset("utf8");

```

```

        $sql = "INSERT INTO poliza (nro, clienteid, vehiculo
id, coberturaid, fecha_emision, horaEmis, emisor, sucursal, vigencia_in
icio, horaInVig, vigencia_fin, meses_vig, anulacion, estado) VALUES (?,
?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)";
        $stmt = $mysqli->prepare($sql);
        if ($stmt!==FALSE) {
            $stmt-
>bind_param('siiissiissssi', $nroPol, $cli, $idVehiculo, $cob, $fecha
Registro, $horaRegistro, $emis, $sucursal, $vig_ini, $horaInVig, $vig_
fin, $mesesVig, $anulacion, $estado);
            $ej = $stmt->execute();

            if($ej){

                $conIdPol = "SELECT MAX(id) AS id FROM poliza"
;

                $stmt = $mysqli->prepare($conIdPol);
                $stmt->execute();
                $rs = $stmt->get_result();
                if ($rs->num_rows > 0) {
                    while ($fila = $rs->fetch_array()) {
                        $polizaId = $fila[0];
                    }
                }

                include_once('pagosController.php');
                $pagosController = new PagosController;
                $pago = $pagosController-
>grabarPago($polizaId, $cob, $sumAs, $mesesVig, $datosPago);
                $fechaRegistroLapso = $fechaRegistro.' - '
.$horaRegistro;
                $lineaTiempo = $mysqli-
>prepare("INSERT INTO ltestpol (polizaid, nrhilo, estado, inicio_laps
o, fin_lapso) VALUES ('$polizaId', '1', '$estado', '$fechaRegistroLaps
o', 'Actualidad')");
                if($pago){
                    if($lineaTiempo->execute()){
                        $lineaTiempo->close();
                        $res=true;
                    }
                }
            }
            $stmt->close();
        }
    }

    return $res;
}catch(Exception $e){

```

```

        return $e->getMessage();
    }

}

// Función para mostrar los datos de una poliza.
public function traerDatosPoliza($id)
{
    $mysqli = $this->conexion;

    $arr = array('success'=>false);

    $id = $id['id'];

    $mysqli->set_charset("utf8");

    $sql = "SELECT
            P.nro, P.fecha_emision, P.horaEmis, P.vigencia_inicio,
            P.vigencia_fin, P.horaInVig, P.clienteid, P.coberturaid,
            V.suma_asegurada, V.marca_id, V.modelo_id, V.patente,
            V.motor, V.nroChasis, V.nroMotor,
            V.clase, V.uso, V.gps, V.0km, V.Kilometros, V.codigoPostal,
            V.localidad, V.coberturaAd,
            V.combustible, V.eGas, V.anio, V.color, V.asientos, V.
pasajeros
            FROM poliza P
            INNER JOIN vehiculo V ON P.vehiculoid = V.id
            WHERE P.id = ? ";

    $stmt = $mysqli->prepare($sql);

    if ($stmt!==FALSE) {

        $stmt->bind_param('i', $id);
        $stmt->execute();
        $rs = $stmt->get_result();

        if ($rs->num_rows > 0) {
            while ($po = $rs->fetch_array()) {

                $arr['nro'] = $po[0];
                $arr['fecha_emision'] = $po[1];
                $arr['horaEmis'] = $po[2];
                $arr['vigencia_inicio'] = $po[3];
                $arr['vigencia_fin'] = $po[4];
                $arr['horaInVig'] = $po[5];
                $arr['clienteid'] = $po[6];
                $arr['coberturaid'] = $po[7];
                $arr['suma_asegurada'] = $po[8];
                $arr['marca_id'] = $po[9];
            }
        }
    }
}

```

```

        $arr[ 'modelo_id' ]      = $po[10];
        $arr[ 'patente' ]        = $po[11];
        $arr[ 'motor' ]          = $po[12];
        $arr[ 'nroChasis' ]      = $po[13];
        $arr[ 'nroMotor' ]       = $po[14];
        $arr[ 'clase' ]          = $po[15];
        $arr[ 'uso' ]             = $po[16];
        $arr[ 'gps' ]             = $po[17];
        $arr[ '0km' ]             = $po[18];
        $arr[ 'Kilometros' ]      = $po[19];
        $arr[ 'codigoPostal' ]     = $po[20];
        $arr[ 'localidad' ]       = $po[21];
        $arr[ 'coberturaAd' ]      = $po[22];
        $arr[ 'combustible' ]      = $po[23];
        $arr[ 'eGas' ]             = $po[24];
        $arr[ 'anio' ]             = $po[25];
        $arr[ 'color' ]             = $po[26];
        $arr[ 'asientos' ]         = $po[27];
        $arr[ 'pasajeros' ]        = $po[28];

        $arr[ 'success' ]          = true;
    }
}

$stmt->close();
echo json_encode(array('data' => $arr));
}

// Función para modificar la información de una poliza (Endoso).
public function modificarDatosPoliza($nuevosDatos)
{
    $datosActEnd = $nuevosDatos['datos'];

    $arr = array('success'=>false, 'accion'=>0);

    $error = false;

    // Validamos los datos.
    $fechEmPoliza = filter_var($datosActEnd[0], FILTER_SANITIZE_FULL_SPECIAL_CHARS);
    if($fechEmPoliza==FALSE || is_null($fechEmPoliza)) {$error = true;}

    $horEmisPol = filter_var($datosActEnd[1], FILTER_SANITIZE_FULL_SPECIAL_CHARS);
    if($horEmisPol==FALSE || is_null($horEmisPol)) {$error = true}
}

```

```

$iniVigPol = filter_var($datosActEnd[2], FILTER_SANITIZE_FULL_
SPECIAL_CHARS);
if($iniVigPol === FALSE || is_null($iniVigPol)) {$error = true;}

$finVigPol = filter_var($datosActEnd[3], FILTER_SANITIZE_FULL_
SPECIAL_CHARS);
if($finVigPol === FALSE || is_null($finVigPol)) {$error = true;}

$horaInVig = filter_var($datosActEnd[4], FILTER_SANITIZE_FULL_
SPECIAL_CHARS);
if($horaInVig === FALSE || is_null($horaInVig)) {$error = true;}

$sumaAs = filter_var($datosActEnd[5], FILTER_SANITIZE_FULL_SPE_
CIAL_CHARS);
if($sumaAs === FALSE || is_null($sumaAs)) {$error = true;}

$marca = filter_var($datosActEnd[6], FILTER_VALIDATE_INT);
if($marca === FALSE || is_null($marca)) {$error = true;}

$modelo = filter_var($datosActEnd[7], FILTER_VALIDATE_INT);
if($modelo === FALSE || is_null($modelo)) {$error = true;}

$patente = filter_var($datosActEnd[8], FILTER_SANITIZE_FULL_SP_
ECIAL_CHARS);
if($patente === FALSE || is_null($patente)) {$error = true;}

$motor = filter_var($datosActEnd[9], FILTER_SANITIZE_FULL_SPEC_
IAL_CHARS);
if($motor === FALSE || is_null($motor)) {$error = true;}

$nroChasis = filter_var($datosActEnd[10], FILTER_SANITIZE_FULL_
_SPECIAL_CHARS);
if($nroChasis === FALSE || is_null($nroChasis)) {$error = true;}

$nroMotor = filter_var($datosActEnd[11], FILTER_SANITIZE_FULL_
SPECIAL_CHARS);
if($nroMotor === FALSE || is_null($nroMotor)) {$error = true;}

$clase = filter_var($datosActEnd[12], FILTER_VALIDATE_INT);
if($clase === FALSE || is_null($clase)) {$error = true;}

$uso = filter_var($datosActEnd[13], FILTER_VALIDATE_INT);
if($uso === FALSE || is_null($uso)) {$error = true;}

$gps = filter_var($datosActEnd[14], FILTER_SANITIZE_FULL_SPECI_
AL_CHARS);
if($gps === FALSE || is_null($gps)) {$error = true;}

```

```

$ceroKm = filter_var($datosActEnd[15], FILTER_SANITIZE_FULL_SPECIAL_CHARS);
if($ceroKm === FALSE || is_null($ceroKm)) {$error = true;}

$cantKms = filter_var($datosActEnd[16], FILTER_SANITIZE_FULL_SPECIAL_CHARS);
if($cantKms === FALSE || is_null($cantKms)) {$error = true;}

$cp = filter_var($datosActEnd[17], FILTER_VALIDATE_INT);
if($cp === FALSE || is_null($cp)) {$error = true;}

$vehiculoLocalidad = filter_var($datosActEnd[18], FILTER_SANITIZE_FULL_SPECIAL_CHARS);
if($vehiculoLocalidad === FALSE || is_null($vehiculoLocalidad))
{$error = true;}

$cobAd = filter_var($datosActEnd[19], FILTER_VALIDATE_INT);
if($cobAd === FALSE || is_null($cobAd)) {$error = true;}

$combustible = filter_var($datosActEnd[20], FILTER_SANITIZE_FULL_SPECIAL_CHARS);
if($combustible === FALSE || is_null($combustible)) {$error = true; }

$eGas = filter_var($datosActEnd[21], FILTER_SANITIZE_FULL_SPECIAL_CHARS);
if($eGas === FALSE || is_null($eGas)) {$error = true; }

$anio = filter_var($datosActEnd[22], FILTER_VALIDATE_INT);
if($anio === FALSE || is_null($anio)) {$error = true; }

$pasajeros = filter_var($datosActEnd[23], FILTER_SANITIZE_FULL_SPECIAL_CHARS);
if($pasajeros === FALSE) {$error = true; }

$asientos = filter_var($datosActEnd[24], FILTER_SANITIZE_FULL_SPECIAL_CHARS);
if($asientos === FALSE) {$error = true; }

$color = filter_var($datosActEnd[25], FILTER_SANITIZE_FULL_SPECIAL_CHARS);
if($color === FALSE) {$error = true; }

$cliente = filter_var($datosActEnd[26], FILTER_VALIDATE_INT);
if($cliente === FALSE || is_null($cliente)) {$error = true; }

$cobertura = filter_var($datosActEnd[27], FILTER_VALIDATE_INT)
;
if($cobertura === FALSE || is_null($cobertura)) {$error = true; }

```

```

$idPoliza = $datosActEnd[28];

if(!$error){

    // Asignamos mayúsculas y formateamos los puntos.
    $patente = strtoupper($patente);
    $nroChasis = strtoupper($nroChasis);
    $nroMotor = strtoupper($nroMotor);
    $motor = strtoupper($motor);
    $color = ucwords($color);

    $sig[ ]='.';
    $sig[ ]=',';

    $sumaAs = str_replace($sig,'', $sumaAs);
    $cantKms = str_replace($sig,',', $cantKms);

    $sumaAs = number_format($sumaAs, 0, ',', '.');
    $cantKms = number_format($cantKms, 0, ',', '.');

    $sql =
"SELECT
P.clienteid, P.vehiculoid, P.coberturaid, P.vigencia_inicio, P.vigencia_fin,
V.suma_asegurada, V.marca_id, V.modelo_id, V.patente, V.motor, V.nroChasis, V.nroMotor, V.clase, V.uso,
V.gps, V.0km, V.Kilometros, V.codigoPostal, V.localidad, V.coberturaAd, V.combustible, V.eGas, V.anio,
V.pasajeros, V.asientos, V.color
FROM poliza P
INNER JOIN vehiculo V
WHERE
P.clienteid      = ? AND
P.coberturaid    = ? AND
P.vigencia_inicio = ? AND
P.vigencia_fin    = ? AND
V.suma_asegurada = ? AND
V.marca_id        = ? AND
V.modelo_id       = ? AND
V.patente         = ? AND
V.motor           = ? AND
V.nroChasis       = ? AND
V.nroMotor         = ? AND
V.clase            = ? AND
V.uso              = ? AND
V.gps              = ? AND
V.0km              = ? AND
V.Kilometros       = ? AND
V.codigoPostal     = ? AND

```

```

V.localidad      = ? AND
V.coberturaAd   = ? AND
V.combustible    = ? AND
V.eGas           = ? AND
V.anio            = ? AND
V.pasajeros      = ? AND
V.asientos        = ? AND
V.color           = ? ";

$mysqli = $this->conexion;
$stmt = $mysqli->prepare($sql);
$stmt-
>bind_param('iissiiissssiissiiiis', $cliente, $cobertura, $iniVig
Pol, $finVigPol, $sumaAs, $marca, $modelo, $patente, $motor, $nroChasi
s, $nroMotor, $clase, $uso, $gps, $ceroKm, $cantKms, $cp, $vehiculoLoc
alidad, $cobAd, $combustible, $eGas, $anio, $pasajeros, $asientos, $co
lor);
$stmt->execute();
$rs = $stmt->get_result();

// Si no existe ninguna poliza con exactamente los mismos
datos, ejecutamos la consulta.
if($rs->num_rows==0){

    $sql = "UPDATE poliza P
            INNER JOIN vehiculo V ON P.vehiculoid = V.id
            SET
                P.clienteid      = ?,
                P.coberturaid    = ?,
                P.vigencia_inicio = ?,
                P.vigencia_fin    = ?,
                V.suma_asegurada = ?,
                V.marca_id       = ?,
                V.modelo_id      = ?,
                V.patente         = ?,
                V.motor           = ?,
                V.nroChasis       = ?,
                V.nroMotor         = ?,
                V.clase            = ?,
                V.uso              = ?,
                V.gps              = ?,
                V.0km              = ?,
                V.Kilometros       = ?,
                V.codigoPostal     = ?,
                V.localidad        = ?,
                V.coberturaAd      = ?,
                V.combustible       = ?,
                V.eGas              = ?,
                V.anio              = ?,
}

```

```

        V.pasajeros      = ?,
        V.asientos        = ?,
        V.color          = ?
    WHERE
        P.id              = ?  ";

$stmt = $mysqli->prepare($sql);
if ($stmt!==FALSE) {
    $stmt-
>bind_param('iisssiissssiissiisssssi', $cliente, $cobertura, $iniVi
gPol, $finVigPol, $sumaAs, $marca, $modelo, $patente, $motor, $nroChas
is, $nroMotor, $clase, $uso, $gps, $ceroKm, $cantKms, $cp, $vehiculoLo
calidad, $cobAd, $combustible, $eGas, $anio, $pasajeros, $asientos, $c
olor, $idPoliza);
    $stmt->execute();
    $arr = array('success'=>true, 'accion'=>1);
    $stmt->close();
}

}else
{
    $arr = array('success'=>true, 'accion'=>2);
    $stmt->close();
}
}

echo json_encode($arr);
}

// Función para listar las polizas en un select.
public function listarPolizas()
{
    $mysqli = $this->conexion;
    $arr1 = array();
    $sql = "SELECT P.id as id, P.nro as nro, T.nombre as tomador,
T.documento as doc, P.estado as estado
    FROM poliza P
    INNER JOIN tomador T ON T.id = P.clienteid";
    $stmt = $mysqli->prepare($sql);
    $stmt->execute();
    $r = $stmt->get_result();

    if($r->num_rows > 0){
        while($pol = $r->fetch_array()){
            $arr = array();
            $arr['id']      = $pol[0];
            $arr['nro']     = $pol[1];
            $arr['tomador'] = $pol[2];
            $arr['doc']     = $pol[3];
        }
    }
}

```

```

        $arr['estado'] = $pol[4];
        $arr1[] = $arr;
    }
}

$stmt->close();

echo json_encode(array('data' => $arr1));
}

// Función para renovar una poliza.
public function renovarPoliza($post)
{
    $mysqli = $this->conexion;
    $arr['renPoliza'] = false;

    $info = $post['data'];

    $idPoliza = $info[0];
    $idPoliza = $idPoliza[0];
    $vig_ini = $info[1];
    $tipVig = $info[2];
    $vig_fin = $info[3];
    $primaTotal = $info[4];
    $cantCuotas = $info[5];
    $primaMensual = $info[6];
    $diaCobrar = $info[7];
    $tipoPag = $info[8];

    // Calculamos el lapso de cobro = cantidad de meses de
    // vigencia dividido la cantidad de cuotas seleccionado.
    $lapso = strval($tipVig/$cantCuotas);

    // Generamos las fecha y horas.
    $fechaRegistro = Date("d/m/Y");
    $horaRegistro = Date("H:i");
    $fechaRegLT = $fechaRegistro.' - '.$horaRegistro;

    // Formateamos el fin de vigencia y le sumamos un mes.
    $finVig = str_replace("/", "-", $vig_fin);
    $f = new DateTime($finVig);
    $f->modify('next month');
    $anulacion = $f->format('d/m/Y');

    $estadoEsp = 5; // Espera
    $estadoRen = 4; // Renovada
    $estPagoAn = 2; // Anulada
}

```

```

// 1) Insertamos en poliza un clon de la poliza
$insertarClon = "INSERT INTO poliza (nro, clienteid, vehiculoid, coberturaid, fecha_emision, horaEmis, emisor, vigencia_inicio, horaInVig, vigencia_fin, meses_vig, anulacion, estado)

SELECT nro, clienteid, vehiculoid, coberturaid, fecha_emision, horaEmis, emisor, vigencia_inicio, horaInVig, vigencia_fin, meses_vig, anulacion, estado
FROM poliza
WHERE id = ?";

$stmt1 = $mysqli->prepare($insertarClon);

// 2) Modificamos ese clon con la informacion que corresponde
// (lo buscamos con el maximo id ingresado)
$modificarClon = "UPDATE poliza
SET fecha_emision = ?,
horaEmis = ?,
vigencia_inicio = ?,
vigencia_fin = ?,
meses_vig = ?,
anulacion = ?,
estado = ?
WHERE id = ?";

$stmt2 = $mysqli->prepare($modificarClon);

// 3) Modificamos el estado de la poliza original a 'renovada'
$modificarEstPoliza = "UPDATE poliza SET estado = ? WHERE id = ?";

$stmt3 = $mysqli->prepare($modificarEstPoliza);

// 4) Modificamos el estado del pago original a 'anulado'
$modificarEstadoPago = "UPDATE pagos SET estado = ? WHERE id = 
(SELECT DISTINCT id
FROM pagos PA
WHERE PA.poliza_id = ?)";

$stmt4 = $mysqli->prepare($modificarEstadoPago);

//Si todas las consultas existen, pasamos todos los parametros
//y realizamos las ejecuciones en cadena
if($stmt1 && $stmt2 && $stmt3 && $stmt4){

```

```

    // Clonacion
    $stmt1->bind_param('i', $idPoliza);
    $stmt1->execute();
    $stmt1->close();

    // Obtenemos el id de la poliza renovada
    $stmtId= $mysqli-
>prepare("SELECT MAX(id) AS id FROM poliza");
    $stmtId->execute();
    $r = $stmtId->get_result();
    foreach($r as $id){
        $idNuevaPoliza = $id['id'];
    }
    $stmtId->close();

    // Modificacion poliza renovada
    $stmt2-
>bind_param('ssssssii', $fechaRegistro, $horaRegistro, $vig_ini, $vig_
fin, $tipVig, $anulacion, $estadoEsp, $idNuevaPoliza);
    $stmt2->execute();
    $stmt2->close();

    // Estado de poliza original a 'renovado'
    $stmt3->bind_param('ii', $estadoRen, $idPoliza);
    $stmt3->execute();
    $stmt3->close();
    //Cuando modificamos el estado de la poliza original a 're
novado', insertamos ese cambio de estado en la linea del tiempo de est
ados de la poliza
    $this-
>guardarLineaTiempoEstadoPoliza($idPoliza, $estadoRen);

    //Estado de pago original a 'anulado'
    $stmt4->bind_param('ii', $estPagoAn, $idPoliza);
    $stmt4->execute();
    $stmt4->close();

    //Traemos la informacion de la poliza renovada para genera
r el pago
    $con = $mysqli-
>prepare("SELECT P.coberturaid as cob, V.suma_asegurada as sumAs, P.me
ses_vig as mVig
                FROM poliza P
                INNER JOIN vehiculo V ON V.id = P
                .vehiculoid
                WHERE P.id = ?");
    if($con->bind_param('i', $idNuevaPoliza)){
        $con->execute();

```

```

        foreach($con->get_result() as $info){
            $cob      = $info['cob'];
            $sumAs   = $info['sumAs'];
            $mesesVig = $info['mVig'];
        }

        $datosPago = [$primaTotal, $cantCuotas, $primaMensual,
        $diaCobrar, $vig_ini, $vig_fin, $lapso, $tipoPag];
        $con->close();
    }
    // $grabarPago = $this-
>grabarPago($idNuevaPoliza, $cob, $sumAs, $mesesVig, $datosPago);

    include_once('pagosController.php');
    $pagosController = new PagosController;
    $grabarPago = $pagosController-
>grabarPago($idNuevaPoliza, $cob, $sumAs, $mesesVig, $datosPago);

    if($grabarPago){
        // Insercion de una nueva linea del tiempo de estados

        $ins = "INSERT INTO ltestpol (polizaid, nrohilo, estado, inicio_lapso, fin_lapso) VALUES (?,?,?,?,?,?)";
        $lineaTiempo = $mysqli->prepare($ins);
        $numeroHilo = '1';
        $fecFinLap = 'Actualidad';
        $lineaTiempo-
>bind_param('isiss', $idNuevaPoliza, $numeroHilo, $estadoEsp, $fechaRe
gLت, $fecFinLap);
        if($lineaTiempo!=FALSE){
            $lineaTiempo->execute();
            $lineaTiempo->close();
        }

        $arr['renPoliza'] = true;
    }
}
return json_encode($arr);
}

// Función para actualizar los estados de las polizas.
public function actualizarEstadosPoliza()
{
    //1) corroboramos las fechas en relacion a la fecha actual par
a asi declarar la poliza en un estado determinado.

    //2) si la poliza esta vigente, corroboramos si posee deuda o
no, en caso de que alguna cuota se encuentre en deuda, modificamos el
estado a vigenteConDeuda.

```

```

    //En caso de que la poliza se encuentre vencida o anulada, NO
SE CORROBORA SI POSEE DEUDA.
    $mysqli = $this->conexion;
    $arr['actEstPol'] = false;
    $fecha_actual = new DateTime( str_replace("/", "-",
", Date("d/m/Y")) );
    $sql = "SELECT id, estado, fecha_emision, vigencia_inicio, vig
encia_fin, anulacion FROM poliza";

    $stmt = $mysqli->prepare($sql);
    $stmt->execute();
    $r = $stmt->get_result();

    foreach($r as $resp){

        $idPol      = $resp['id'];
        $estado     = $resp['estado'];
        $emision    = $resp['fecha_emision'];
        $inicioVig = $resp['vigencia_inicio'];
        $finVigenc = $resp['vigencia_fin'];
        $anulacion = $resp['anulacion'];

        $emision    = new DateTime( str_replace("/", "-",
", $resp['fecha_emision'])) );
        $inicioVig = new DateTime( str_replace("/", "-",
", $resp['vigencia_inicio'])) );
        $finVigenc = new DateTime( str_replace("/", "-",
", $resp['vigencia_fin'])) );
        $anulacion = new DateTime( str_replace("/", "-",
", $resp['anulacion'])) );

        //Se aplica la logica a toda poliza que no este renovada.
        if($estado != 4){

            //Si la fecha actual es mayor o igual a la emision y me
nor al inicio de vigencia = Espera.
            if($fecha_actual>=$emision && $fecha_actual<$inicioVig
){
                $estActual = 5;
                //Si la fecha actual es mayor o igual al inicio de vig
encia y menor a fin de vigencia = Vigente.
                }else if($fecha_actual>=$inicioVig && $fecha_actual<$f
inVigenc){
                    $estActual = 0;
                    //Si la fecha actual es mayor a fin de vigencia y meno
r a anulacion = Vencida.
                    }else if($fecha_actual>=$finVigenc && $fecha_actual<$a
nulacion){
                        $estActual = 1;

```

```

        //Si la fecha actual es mayor o igual a la fecha de anulacion = Anulada.
    }else if($fecha_actual >= $anulacion){
        $estActual = 2;
    }

    if($estActual != $estado){

        $stmt = $mysqli->prepare("UPDATE poliza SET estado = $estActual WHERE id = $idPol");
        if($stmt->execute()){

            //Insertamos el cambio de estado en la linea del tiempo de estados de esa poliza.
            $lineaTiempo = $this->guardarLineaTiempoEstadoPoliza($idPol, $estActual);

            //Comprobamos la deuda en todas las polizas vivientes, si existen, sumamos un contador, en caso de que este sea mayor a 0, cambiamos el estado de la poliza a deuda, o viceversa.
            if($estado == 0){
                $con = "SELECT C.estado as estadoCuota, P.estado as estadoPoliza
                        FROM cuotas C
                        INNER JOIN poliza P ON P.id = $idPol
                        WHERE pago_id = (SELECT id FROM pagos WHERE poliza_id = $idPol)";

                $comp = $mysqli->prepare($con);

                if($comp->execute()){
                    $contDeud = 0;

                    foreach($comp->get_result() as $v){
                        if($v['estadoCuota'] == 4){
                            $contDeud++;
                        }
                    }

                    if($contDeud >= 1){ //Si existe una o mas deudas, cambiamos el estado y agregamos la linea del tiempo.
                        $estPol = 3;
                        $arr['accion'] = 'deuda';
                        $stmt2 = $mysqli->prepare("UPDATE poliza SET estado = $estPol WHERE id = $idPol");
                        if($stmt2->execute()){

```

```

        $this-
>guardarLineaTiempoEstadoPoliza($idPol, $estPol); //Cuando se cambia el estado a 'deuda', agregamos a la linea del tiempo.
    }
    $stmt2->close();
}else{
    $arr['accion'] = 'sin deuda';
}
$arr['actEstPol'] = true;
}
}else{
    $arr['accion'] = 'sin polizas vig';
}
}else{
    echo $stmt->error;
}

$stmt->close();
}else{
    $arr['accion'] = 'Mismo estado';
}
}
}

return json_encode($arr);
}

// Función donde guardamos o actualizamos la línea del tiempo de estados de una poliza.
public function guardarLineaTiempoEstadoPoliza($polizaId, $nuevoEstado)
{
    $mysqli = $this->conexion;
    $arr['LtEstPoliza'] = false;
    $fechaActual = Date("d/m/Y - H:i");
    $polizaId = $polizaId;
    $nuevoEstado = $nuevoEstado;

    //Obtenemos el numero de hilo mayor de la linea del tiempo de estados de esa poliza.
    $con = "SELECT MAX(nrohilo) as nroHilo FROM ltestpol WHERE polizaId = '$polizaId'";
    $stmtNroHilo = $mysqli->prepare($con);

    if($stmtNroHilo){

        $stmtNroHilo->execute();
        $r = $stmtNroHilo->get_result();
        foreach($r as $var){

```

```

        $nroHilo = $var['nroHilo'];
    }
}
$stmtNroHilo->close();

//Asignamos una fecha de finalizacion del lapso al ultimo hilo
de la linea del tiempo de estados de la poliza.
$stmtAsFech = $mysqli-
>prepare("UPDATE ltestpol SET fin_lapso = '$fechaActual' WHERE polizai
d = '$polizaId' AND nrohilo = '$nroHilo'");
if($stmtAsFech){
    $stmtAsFech->execute();
}
$stmtAsFech->close();

//Insertamos un nuevo hilo a la linea del tiempo de estados de
la poliza (Sumando el numero de hilo).
$nroHiloNuevo = strval($nroHilo+1);
$finLapso = "Actualidad";
$sql = "INSERT INTO ltestpol (polizaid, nrohilo, estado, inici
o_lapso, fin_lapso) VALUES (?,?,?,?,?)";
$stmt = $mysqli->prepare($sql);
if($stmt){
    $stmt-
>bind_param('isiss', $polizaId, $nroHiloNuevo, $nuevoEstado, $fechaAct
ual, $finLapso);
    $stmt->execute();
    $arr['LtEstPoliza'] = true;
}
$stmt->close();

return json_encode($arr);
}
}

```

⇒ 4.3.2.5 fichero “siniestrosController.php”

Este fichero contiene todas las funciones que realizan las acciones relacionadas a los siniestros en el sistema.

```
<?php

// Clase principal
class SiniestrosController
{
    private $conexion;

    // Función constructora donde se realiza la conexión a base de datos.
    public function __construct()
    {
        $this->conexion = Database::connect();
        date_default_timezone_set("America/Argentina/Buenos_Aires");
    }

    // Función donde traemos los datos del denunciante.
    public function traerDatosDenunciante($post)
    {

        $idpoliza = $post['data'];

        $arr = array('success'=>false);

        $query = "SELECT T.nombre, T.documento, T.telefono, T.correo,
                  CONCAT(T.calle, ' ', T.localidad, ' ', T.provincia) AS domicilio
                FROM poliza P
                INNER JOIN tomador T ON T.id = P.clienteid
                WHERE P.id = '$idpoliza';

        $resultado = $this->conexion->query($query);

        while ($dato = mysqli_fetch_row($resultado)) {
            $arr['nombre'] = $dato[0];
            $arr['documento'] = $dato[1];
            $arr['telefono'] = $dato[2];
            $arr['correo'] = $dato[3];
            $arr['domicilio'] = $dato[4];
            $arr['success'] = true;
        }
    }
}
```

```

        echo json_encode(array('data' => $arr));
    }

    // Función donde grabamos el siniestro.
    public function grabarSiniestro($post)
    {
        try{

            $mysqli = $this->conexion;
            $arr = array('success'=>false);
            $datosPoliza = $post['data'];
            $error = false;

            // Validamos los datos
            $nroSin = filter_var($datosPoliza[0], FILTER_SANITIZE_FULL_SPECIAL_CHARS);
            if($nroSin==FALSE || is_null($nroSin)) {$error = true;}

            $fechOcu = filter_var($datosPoliza[1], FILTER_SANITIZE_FULL_SPECIAL_CHARS);
            if($fechOcu==FALSE || is_null($fechOcu)) {$error = true; }

            $hOcu = filter_var($datosPoliza[2], FILTER_SANITIZE_FULL_SPECIAL_CHARS);
            if($hOcu==FALSE || is_null($hOcu)) {$error = true; }

            $IdPoliza = filter_var($datosPoliza[3], FILTER_VALIDATE_INT);
            if($IdPoliza==FALSE || is_null($IdPoliza)) {$error = true; }

            $lesMu = filter_var($datosPoliza[4], FILTER_SANITIZE_FULL_SPECIAL_CHARS);
            if($lesMu==FALSE || is_null($lesMu)) {$error = true; }

            $tipSin = filter_var($datosPoliza[5], FILTER_VALIDATE_INT);
            if($tipSin==FALSE || is_null($tipSin)) {$error = true; }

            $nomDen = filter_var($datosPoliza[6], FILTER_SANITIZE_FULL_SPECIAL_CHARS);
            if($nomDen==FALSE || is_null($nomDen)) {$error = true; }

            $arr['success'] = true;
            echo json_encode($arr);
        }
    }
}

```

```

        $telDen = filter_var($datosPoliza[7], FILTER_SANITIZE_
FULL_SPECIAL_CHARS);
        if($telDen==FALSE || is_null($telDen)) {$error = true
;}

        $dniDen = filter_var($datosPoliza[8], FILTER_SANITIZE_
FULL_SPECIAL_CHARS);
        if($dniDen==FALSE || is_null($dniDen)) {$error = true
;}

        $emailDen = filter_var($datosPoliza[9], FILTER_SANITIZE_
E_FULL_SPECIAL_CHARS);
        if($emailDen==FALSE) {$error = true; }

        $domicDen = filter_var($datosPoliza[10], FILTER_SANITIZE_
ZE_FULL_SPECIAL_CHARS);
        if($domicDen==FALSE || is_null($domicDen)) {$error =
true; }

        $descAcc = filter_var($datosPoliza[11], FILTER_SANITIZE_
E_FULL_SPECIAL_CHARS);
        if($descAcc==FALSE || is_null($descAcc)) {$error = tr
ue; }

        $callAcc = filter_var($datosPoliza[12], FILTER_SANITIZE_
E_FULL_SPECIAL_CHARS);
        if($callAcc==FALSE){$error = true; }

        $altAcc = filter_var($datosPoliza[13], FILTER_SANITIZE_
_FULLSCREEN_CHARS);
        if($altAcc==FALSE){$error = true; }

        $locAcc = filter_var($datosPoliza[14], FILTER_SANITIZE_
_FULLSCREEN_CHARS);
        if($locAcc==FALSE || is_null($locAcc)) {$error = true
; }

        $proAcc = filter_var($datosPoliza[15], FILTER_SANITIZE_
_FULLSCREEN_CHARS);
        if($proAcc==FALSE || is_null($proAcc)) {$error = true
; }

        $nomRec = filter_var($datosPoliza[16], FILTER_SANITIZE_
_FULLSCREEN_CHARS);
        if($nomRec==FALSE) {$error = true; }

        $domRec = filter_var($datosPoliza[17], FILTER_SANITIZE_
_FULLSCREEN_CHARS);
        if($domRec==FALSE) {$error = true; }

```

```

        $telRec = filter_var($datosPoliza[18], FILTER_SANITIZE
_FULL_SPECIAL_CHARS);
        if($telRec==FALSE) {$error = true;}

        $dniRec = filter_var($datosPoliza[19], FILTER_SANITIZE
_FULL_SPECIAL_CHARS);
        if($dniRec==FALSE) {$error = true;}

        $nomCon = filter_var($datosPoliza[20], FILTER_SANITIZE
_FULL_SPECIAL_CHARS);
        if($nomCon==FALSE) {$error = true;}

        $domCon = filter_var($datosPoliza[21], FILTER_SANITIZE
_FULL_SPECIAL_CHARS);
        if($domCon==FALSE) {$error = true;}

        $telCon = filter_var($datosPoliza[22], FILTER_SANITIZE
_FULL_SPECIAL_CHARS);
        if($telCon==FALSE) {$error = true;}

        $dniCon = filter_var($datosPoliza[23], FILTER_SANITIZE
_FULL_SPECIAL_CHARS);
        if($dniCon==FALSE) {$error = true;}

        $tomadDenunciante = $datosPoliza[24];
        $condReclamante = $datosPoliza[25];
        $tomadReclamante = $datosPoliza[26];

        //Si el tomador es el denunciante, se asigna el valor
NULL, si no es, guardamos los valores NO NULOS ingresados.
        if($tomadDenunciante=='NO'){
            if( is_null($nomDenNuev) ||
                is_null($dniDenNuev) ||
                is_null($domDenNuev) ||
                is_null($telDenNuev)
            ){
                $error = true;
            }else{
                $nomDenNuev = $nomDenNuev;
                $dniDenNuev = $dniDenNuev;
                $domDenNuev = $domDenNuev;
                $telDenNuev = $telDenNuev;
                $emailDenNuev = $emailDenNuev;
            }
        }else if($tomadDenunciante=='SI'){
            $nomDenNuev = '';
            $dniDenNuev = '';
            $domDenNuev = '';

```

```

        $telDenNuev = '';
        $emailDenNuev = '';
    }

    /*Si el siniestro fue un choque y se involucra una tercera persona, asignamos los datos NO NULOS ingresados, de lo contrario se asigna por defecto NULL*/
    if($tipSin==5){
        if(is_null($nomRec) ||
        is_null($domRec) ||
        is_null($telRec) ||
        is_null($dniRec)
        ){
            $error = true;
        }else{
            $nomRec = $nomRec;
            $domRec = $domRec;
            $telRec = $telRec;
            $dniRec = $dniRec;
        }
    }else{
        $nomRec = '';
        $domRec = '';
        $telRec = '';
        $dniRec = '';
        $nomCon = '';
        $domCon = '';
        $telCon = '';
        $dniCon = '';
        $condReclamante = null;
    }

    /*Si el conductor es la misma persona que la tercera persona (reclamante), se asigna NULL
    de lo contrario se asignan los valores NO NULOS que ingrese el usuario*/
    if($condReclamante=='NO'){
        if( is_null($nomCon) ||
        is_null($domCon) ||
        is_null($telCon) ||
        is_null($dniCon)
        ){
            $error = true;
        }else{
            $nomCon = $nomCon;
            $domCon = $domCon;
            $telCon = $telCon;
            $dniCon = $dniCon;
        }
    }
}

```

```

}else if($condReclamante=='SI'){
    $nomCon = '';
    $domCon = '';
    $telCon = '';
    $dniCon = '';
}else if($condReclamante==''){
    $condReclamante = '';
}

if(!isset($_SESSION)){
    session_start();
}
if(isset($_SESSION['user'])){
    $emis = $_SESSION['user']['id'];
}

if(!$error){

    $fechDen = date('d/m/Y').' - '.date("H:i");

    $lesMu           = ucwords($lesMu);
    $nomDen          = ucwords($nomDen);
    $domicDen        = ucwords($domicDen);
    $callAcc         = ucwords($callAcc);
    $proAcc          = ucwords($proAcc);
    $locAcc          = ucwords($locAcc);
    $nomRec          = ucwords($nomRec);
    $domRec          = ucwords($domRec);
    $nomCon          = ucwords($nomCon);
    $domCon          = ucwords($domCon);
    $estado           = '1';
    $tomadDenunciante = ucwords($tomadDenunciante);
    $condReclamante   = ucwords($condReclamante);
    $tomadReclamante = ucwords($tomadReclamante);

    $sql = "INSERT INTO siniestro (nroSiniestro, fech
aDen, emisor, fechaOc, horaOc, idPoliza, lesionMuerte, tipoSiniestro,
nomDen, telDen, domDen
,dniDen, emailDen, estado, descripcionAcc, calleAc
c, alturaAcc, provinciaAcc, localidadAcc, nombreTercero, domTercero, t
elTercero, docTercero, nombreConductor, telConductor, domConductor, dn
iConductor, tomadDenunciante, condReclamante, tomadReclamante) VALUES
(?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?)";
}

$stmt = $mysqli->prepare($sql);

if($stmt!==FALSE) {

```

```

        $stmt->bind_param('ssisisisssssssssssssssss', $nroSin, $fechDen, $emis,
, $fechOcu, $hOcu, $IdPoliza, $lesMu, $tipSin, $nomDen, $telDen, $domi
cDen, $dniDen, $emailDen, $estado, $descAcc, $callAcc, $altAcc, $proAc
c, $locAcc, $nomRec, $domRec, $telRec, $dniRec, $nomCon, $telCon, $dom
Con, $dniCon, $tomadDenunciante, $condReclamante, $tomadReclamante);
        if($stmt->execute()){
            $arr = array('success'=>true);
            $stmt->close();
        }
    }

    return json_encode($arr);

}catch(Exception $e)
{
    return $e->getMessage();
}

// Función donde modificamos el estado del siniestro.
public function modificarEstadoSiniestro($post)
{
    try{

        $mysqli = $this->conexion;

        $arr = array('success'=>false);

        $datos = $post['data'];
        $idSiniestro = $datos[0];
        $idNuevoEstSin = $datos[1];

        $sql = "UPDATE siniestro
                SET estado = ?
                WHERE id = '$idSiniestro'";
        $stmt = $mysqli->prepare($sql);
        if ($stmt!==FALSE) {
            $stmt->bind_param('i', $idNuevoEstSin);
            $stmt->execute();
            $stmt->close();

            $arr = array('success'=>true);
        }

        return json_encode($arr);

    }catch(Exception $e){

```

```

        return $e->getMessage();
    }
}

// Función donde traemos los datos de un siniestro.
public function traerDatosSiniestro($post)
{
    $arr['success'] = false;
    $idSin = $post['data'];
    $mysqli = $this->conexion;

    $sql = "SELECT S.idPoliza, P.nro, S.nroSiniestro, S.fechaDen,
    S.fecha0c, S.hora0c, S.lesionMuerte, S.tipoSiniestro, S.nomDen, S.
    telDen, S.dniDen,
    S.domDen, S.emailDen, S.descripcionAcc, S.calleAcc
    , S.alturaAcc, S.provinciaAcc,
    S.localidadAcc, S.nombreTercero, S.domTercero, S.t
    elTercero, S.docTercero, S.nombreConductor,
    S.telConductor, S.domConductor, S.dniConductor, S.
    imagen, S.tomadDenunciante,
    S.condReclamante, S.tomadReclamante
    FROM siniestro S
    INNER JOIN poliza P ON P.id = S.idPoliza
    WHERE S.id = ?";

    $stmt = $mysqli->prepare($sql);
    if($stmt!=FALSE){
        $stmt->bind_param('i', $idSin);
        if($stmt->execute()){
            $r = $stmt->get_result();

            foreach($r as $dato){

                $arr['idPoliza'] = $dato['idPoliza'];
                $arr['nroPol'] = $dato['nro'];
                $arr['nroSiniestro'] = $dato['nroSiniestro']
            ];
                $arr['fechaDen'] = $dato['fechaDen'];
                $arr['fecha0c'] = $dato['fecha0c'];
                $arr['hora0c'] = $dato['hora0c'];
                $arr['lesionMuerte'] = $dato['lesionMuerte'
            ];
                $arr['tipoSiniestro'] = $dato['tipoSiniestr
            o'];

                $arr['nomDen'] = $dato['nomDen'];
                $arr['telDen'] = $dato['telDen'];
                $arr['dniDen'] = $dato['dniDen'];
                $arr['domDen'] = $dato['domDen'];
                $arr['emailDen'] = $dato['emailDen'];
            }
        }
    }
}

```

```

        $arr['descripcionAcc'] = $dato['descripcionA
cc'];
        $arr['calleAcc'] = $dato['calleAcc'];
        $arr['alturaAcc'] = $dato['alturaAcc'];
        $arr['provinciaAcc'] = $dato['provinciaAcc
'];
        $arr['localidadAcc'] = $dato['localidadAcc
'];
        $arr['nombreTercero'] = $dato['nombreTercer
o'];
        $arr['domTercero'] = $dato['domTercero'];
        $arr['telTercero'] = $dato['telTercero'];
        $arr['docTercero'] = $dato['docTercero'];
        $arr['nombreConductor'] = $dato['nombreConduc
tor'];
        $arr['telConductor'] = $dato['telConductor'];
        $arr['domConductor'] = $dato['domConductor'];
        $arr['dniConductor'] = $dato['dniConductor'];
        $arr['imagen'] = $dato['imagen'];
        $arr['tomadDenunciante'] = $dato['tomadDenunci
ante'];
        $arr['condReclamante'] = $dato['condReclaman
te'];
        $arr['tomadReclamante'] = $dato['tomadReclama
nte'];

        $arr['success'] = true;
    }
}

if($stmt){
    $stmt->close();
}

echo json_encode($arr);
}

// Función donde se actualiza la informacion del siniestro
public function actualizarDatosSin($post)
{
    $datosActSin = $post['datosSin'];

```

```

$arr = array('success'=>false, 'accion'=>0);

$error = false;

// Validamos los datos
$fechaOcNuev = filter_var($datosActSin[0], FILTER_SANITIZE_FULL_SPECIAL_CHARS);
if($fechaOcNuev==FALSE || is_null($fechaOcNuev)) {$error = true;}

$horaOcNuev = filter_var($datosActSin[1], FILTER_SANITIZE_FULL_SPECIAL_CHARS);
if($horaOcNuev==FALSE || is_null($horaOcNuev)) {$error = true;}

$lesionMuerteNuev = filter_var($datosActSin[2], FILTER_SANITIZE_FULL_SPECIAL_CHARS);
if($lesionMuerteNuev==FALSE || is_null($lesionMuerteNuev)) {$error = true;}

$tipoSiniestroNuev = filter_var($datosActSin[3], FILTER_VALIDATE_INT);
if($tipoSiniestroNuev==FALSE || is_null($tipoSiniestroNuev)) {$error = true;}

$nomDenNuev = filter_var($datosActSin[4], FILTER_SANITIZE_FULL_SPECIAL_CHARS);
if($nomDenNuev==FALSE) {$error = true;}

$dniDenNuev = filter_var($datosActSin[5], FILTER_SANITIZE_FULL_SPECIAL_CHARS);
if($dniDenNuev==FALSE) {$error = true;}

$domDenNuev = filter_var($datosActSin[6], FILTER_SANITIZE_FULL_SPECIAL_CHARS);
if($domDenNuev==FALSE) {$error = true;}

$telDenNuev = filter_var($datosActSin[7], FILTER_SANITIZE_FULL_SPECIAL_CHARS);
if($telDenNuev==FALSE) {$error = true;}

$emailDenNuev = filter_var($datosActSin[8], FILTER_SANITIZE_FULL_SPECIAL_CHARS);
if($emailDenNuev==FALSE) {$error = true;}

$descripcionAccNuev = filter_var($datosActSin[9], FILTER_SANITIZE_FULL_SPECIAL_CHARS);
if($descripcionAccNuev==FALSE || is_null($descripcionAccNuev)) {$error = true;}

```

```

        $provinciaAccNuev = filter_var($datosActSin[10], FILTER_SANITIZE_FULL_SPECIAL_CHARS);
        if($provinciaAccNuev === FALSE || is_null($provinciaAccNuev))
) {$error = true;}

        $localidadAccNuev = filter_var($datosActSin[11], FILTER_SANITIZE_FULL_SPECIAL_CHARS);
        if($localidadAccNuev === FALSE || is_null($localidadAccNuev))
) {$error = true;}

        $calleAccNuev = filter_var($datosActSin[12], FILTER_SANITIZE_FULL_SPECIAL_CHARS);
        if($calleAccNuev === FALSE || is_null($calleAccNuev)) {$error = true; }

        $alturaAccNuev = filter_var($datosActSin[13], FILTER_SANITIZE_FULL_SPECIAL_CHARS);
        if($alturaAccNuev === FALSE || is_null($alturaAccNuev)) {$error = true; }

        $nombreTerceroNuev = filter_var($datosActSin[14], FILTER_SANITIZE_FULL_SPECIAL_CHARS);
        if($nombreTerceroNuev === FALSE) {$error = true; }

        $domTerceroNuev = filter_var($datosActSin[15], FILTER_SANITIZE_FULL_SPECIAL_CHARS);
        if($domTerceroNuev === FALSE) {$error = true; }

        $telTerceroNuev = filter_var($datosActSin[16], FILTER_SANITIZE_FULL_SPECIAL_CHARS);
        if($telTerceroNuev === FALSE) {$error = true; }

        $docTerceroNuev = filter_var($datosActSin[17], FILTER_SANITIZE_FULL_SPECIAL_CHARS);
        if($docTerceroNuev === FALSE) {$error = true; }

        $nombreConductorNuev = filter_var($datosActSin[18], FILTER_SANITIZE_FULL_SPECIAL_CHARS);
        if($nombreConductorNuev === FALSE) {$error = true; }

        $domConductorNuev = filter_var($datosActSin[19], FILTER_SANITIZE_FULL_SPECIAL_CHARS);
        if($domConductorNuev === FALSE) {$error = true; }

        $telConductorNuev = filter_var($datosActSin[20], FILTER_SANITIZE_FULL_SPECIAL_CHARS);
        if($telConductorNuev === FALSE) {$error = true; }

```

```

        $dniConductorNuev = filter_var($datosActSin[21], FILTER_SANITIZE_FULL_SPECIAL_CHARS);
        if($dniConductorNuev==FALSE) {$error = true;}

        $idSiniestro = $datosActSin[22];

        $tomadReclamante = filter_var($datosActSin[23], FILTER_SANITIZE_FULL_SPECIAL_CHARS);
        if($tomadReclamante==FALSE || is_null($tomadReclamante))
{$error = true; }

        $tomadDenunciante = filter_var($datosActSin[24], FILTER_SANITIZE_FULL_SPECIAL_CHARS);
        if($tomadDenunciante==FALSE || is_null($tomadDenunciante))
) {$error = true; }

        $condReclamante = filter_var($datosActSin[25], FILTER_SANITIZE_FULL_SPECIAL_CHARS);
        if($condReclamante==FALSE || is_null($condReclamante)) {$error = true; }

        //Si el tomador es el denunciante, se asigna el valor NULL
, si no es, guardamos los valores NO NULOS ingresados.
        if($tomadDenunciante=='NO'){
            if( is_null($nomDenNuev) || is_null($dniDenNuev) || is_null($domDenNuev) || is_null($telDenNuev)){
                $error = true;
            }else{
                $nomDenNuev = $nomDenNuev;
                $dniDenNuev = $dniDenNuev;
                $domDenNuev = $domDenNuev;
                $telDenNuev = $telDenNuev;
                $emailDenNuev = $emailDenNuev;
            }
        }else if($tomadDenunciante=='SI'){
            $nomDenNuev = '';
            $dniDenNuev = '';
            $domDenNuev = '';
            $telDenNuev = '';
            $emailDenNuev = '';
        }

        /*Si el siniestro fue un choque y se involucra una tercera
persona, asignamos los daatos NO NULOS ingresados,
de lo contrario se asigna por defecto NULL*/
        if($tipoSiniestroNuev==5){
            if(is_null($nombreTerceroNuev) || is_null($domTerceroNuev) || is_null($telTerceroNuev) || is_null($docTerceroNuev)){

```

```

        $error = true;
    }else{
        $nombreTerceroNuev = $nombreTerceroNuev;
        $domTerceroNuev = $domTerceroNuev;
        $telTerceroNuev = $telTerceroNuev;
        $docTerceroNuev = $docTerceroNuev;
    }
}else{
    $nombreTerceroNuev = '';
    $domTerceroNuev = '';
    $telTerceroNuev = '';
    $docTerceroNuev = '';
    $nombreConductorNuev = '';
    $domConductorNuev = '';
    $telConductorNuev = '';
    $dniConductorNuev = '';
    $condReclamante = '';
}

/*Si el conductor es la misma persona que la tercera persona (reclamante), se asigna NULL,
de lo contrario se asignan los valores NO NULOS que ingresó el usuario*/
if($condReclamante=='NO'){
    if( is_null($nombreConductorNuev) || is_null($domConductorNuev) || is_null($telConductorNuev) || is_null($dniConductorNuev)){
        $error = true;
    }else{
        $nombreConductorNuev = $nombreConductorNuev;
        $domConductorNuev = $domConductorNuev;
        $telConductorNuev = $telConductorNuev;
        $dniConductorNuev = $dniConductorNuev;
    }
}else if($condReclamante=='SI'){
    $nombreConductorNuev = '';
    $domConductorNuev = '';
    $telConductorNuev = '';
    $dniConductorNuev = '';
}else if($condReclamante==NULL){
    $condReclamante = '';
}

if(!$error){

    // Se formatean los datos necesarios con mayúsculas.
    $lesionMuerteNuev = ucwords($lesionMuerteNuev);
    $nomDenNuev = ucwords($nomDenNuev);
    $domDenNuev = ucwords($domDenNuev);
}

```

```

$calleAccNuev = ucwords($calleAccNuev);
$provinciaAccNuev = ucwords($provinciaAccNuev);
$localidadAccNuev = ucwords($localidadAccNuev);
$nombreTerceroNuev = ucwords($nombreTerceroNuev);
$domTerceroNuev = ucwords($domTerceroNuev);
$nombreConductorNuev = ucwords($nombreConductorNuev);
$domConductorNuev = ucwords($domConductorNuev);

$sql =
"SELECT id, fechaOc, horaOc, lesionMuerte, tipoSiniestro,
nomDen, telDen, dniDen, domDen, emailDen, descripcionAcc, calleAcc,
alturaAcc, provinciaAcc, localidadAcc, nombreTercero, domTercero, telTercero,
docTercero, nombreConductor, telConductor, domConductor, dniConductor,
tomadDenunciante, condReclamante, tomadReclamante
FROM siniestro
WHERE
fechaOc      = ? AND
horaOc       = ? AND
lesionMuerte = ? AND
tipoSiniestro = ? AND
nomDen        = ? AND
telDen        = ? AND
dniDen        = ? AND
domDen        = ? AND
emailDen      = ? AND
descripcionAcc = ? AND
calleAcc       = ? AND
alturaAcc     = ? AND
provinciaAcc  = ? AND
localidadAcc  = ? AND
nombreTercero = ? AND
domTercero    = ? AND
telTercero    = ? AND
docTercero    = ? AND
nombreConductor = ? AND
telConductor  = ? AND
domConductor  = ? AND
dniConductor  = ? AND
tomadDenunciante = ? AND
condReclamante = ? AND
tomadReclamante = ? AND
id            = ? ";

```

\$mysqli = \$this->conexion;

\$stmt = \$mysqli->prepare(\$sql);
\$stmt->bind_param('sssssssssssssssssssssi', \$fechaOcNuev, \$horaOcNuev, \$lesionMuerteNuev, \$tipoSiniestroNuev, \$nomDenNuev, \$telDenNuev, \$dniDe

```

nNuev, $domDenNuev, $emailDenNuev, $descripcionAccNuev, $calleAccNuev,
$alturaAccNuev, $provinciaAccNuev, $localidadAccNuev, $nombreTerceroN
uev, $domTerceroNuev, $telTerceroNuev, $docTerceroNuev, $nombreConduct
orNuev, $telConductorNuev, $domConductorNuev, $dniConductorNuev, $toma
dDenunciante, $condReclamante, $tomadReclamante, $idSiniestro);
$stmt->execute();
$rs = $stmt->get_result();

// Si los datos no son iguales a los ya grabados,
realizamos la consulta.
if($rs->num_rows == 0){
    $sql = "UPDATE siniestro
        SET fechaOc          = ?,
            horaOc           = ?,
            lesionMuerte     = ?,
            tipoSiniestro    = ?,
            nomDen           = ?,
            telDen           = ?,
            dniDen           = ?,
            domDen           = ?,
            emailDen         = ?,
            descripcionAcc   = ?,
            calleAcc          = ?,
            alturaAcc         = ?,
            provinciaAcc     = ?,
            localidadAcc      = ?,
            nombreTercero     = ?,
            domTercero         = ?,
            telTercero         = ?,
            docTercero         = ?,
            nombreConductor   = ?,
            telConductor       = ?,
            domConductor       = ?,
            dniConductor       = ?,
            tomadDenunciante = ?,
            condReclamante    = ?,
            tomadReclamante   = ?
        WHERE id             = ?";
}

$stmt = $mysqli->prepare($sql);
if ($stmt!==FALSE) {
    $stmt-
>bind_param('sssisssssssssssssssi', $fechaOcNuev, $horaOcNuev, $lesionMuerteNuev, $tipoSiniestroNuev, $nomDenNuev, $telDenNuev, $dniDe
nNuev, $domDenNuev, $emailDenNuev, $descripcionAccNuev, $calleAccNuev,
$alturaAccNuev, $provinciaAccNuev, $localidadAccNuev, $nombreTerceron
uev, $domTerceroNuev, $telTerceroNuev, $docTerceroNuev, $nombreConduct
orNuev, $telConductorNuev, $domConductorNuev, $dniConductorNuev, $toma
dDenunciante, $condReclamante, $tomadReclamante, $idSiniestro);

```

```

        $stmt->execute();
        $arr = array('success'=>true, 'accion'=>1);
        $stmt->close();
    }

} else{
    $arr = array('success'=>true, 'accion'=>2);
    $stmt->close();
}
echo json_encode($arr);
}

}

// Función donde traemos la informacion del siniestro.
public function verSiniestro($post)
{
    $arr = array('success'=>false);

    $idSiniestro = $post['id'];

    $query =
        "SELECT S.nroSiniestro, S.fechaDen, S.fechaOc, TS.nombre a
s tipoSiniestro, S.horaOc, S.lesionMuerte, S.tomadDenunciante, S.condR
eclamante, S.tomadReclamante, ES.nombre as estadoSiniestro,
        P.nro, CONCAT(T.nombre, ' - #',T.num_tom) as tomado
r, P.vigencia_inicio, P.vigencia_fin, EP.nombre as estadoPoliza,
        S.nomDen, S.telDen, S.domDen, S.dniDen, S.emailDen
        ,
        S.descripcionAcc, S.calleAcc, S.alturaAcc, PRO.nom
bre as provinciaAcc, proper(LOC.nombre) as localidadAcc,
        S.nombreTercero, S.domTercero, S.telTercero, S.doc
Tercero,
        S.nombreConductor, S.telConductor, S.domConductor,
        S.dniConductor
        FROM siniestro S
        INNER JOIN tiposiniestro TS ON TS.id = S.tipoSiniestro
        INNER JOIN estadoSiniestro ES ON ES.id = S.estado
        INNER JOIN poliza P ON P.id = S.idPoliza
        INNER JOIN tomador T ON T.id = P.clienteId
        INNER JOIN estadopoliza EP ON EP.id = P.estado
        INNER JOIN provincia PRO ON PRO.id = S.provinciaAcc
        INNER JOIN localidad LOC ON LOC.id = S.localidadAcc

        WHERE S.id = '$idSiniestro'";

    $resultado = $this->conexion->query($query);

    while ($sin = mysqli_fetch_row($resultado)) {

```

```

        $arr['nroSiniestro']      = $sin[0];
        $arr['fechaDen']          = $sin[1];
        $arr['fechaOc']           = $sin[2];
        $arr['tipoSiniestro']     = $sin[3];
        $arr['horaOc']            = $sin[4];
        $arr['lesionMuerte']      = $sin[5];
        $arr['tomadDenunciante']  = $sin[6];
        $arr['condReclamante']    = $sin[7];
        $arr['tomadReclamante']   = $sin[8];
        $arr['estadoSiniestro']   = $sin[9];
        $arr['nroPol']             = $sin[10];
        $arr['tomador']            = $sin[11];
        $arr['vigencia_inicio']   = $sin[12];
        $arr['vigencia_fin']       = $sin[13];
        $arr['estadoPoliza']       = $sin[14];
        $arr['nomDen']              = $sin[15];
        $arr['telDen']              = $sin[16];
        $arr['domDen']              = $sin[17];
        $arr['dniDen']              = $sin[18];
        $arr['emailDen']            = $sin[19];
        $arr['descripcionAcc']     = $sin[20];
        $arr['calleAcc']            = $sin[21];
        $arr['alturaAcc']           = $sin[22];
        $arr['provinciaAcc']        = $sin[23];
        $arr['localidadAcc']        = $sin[24];
        $arr['nombreTercero']       = $sin[25];
        $arr['domTercero']           = $sin[26];
        $arr['telTercero']           = $sin[27];
        $arr['docTercero']           = $sin[28];
        $arr['nombreConductor']     = $sin[29];
        $arr['telConductor']         = $sin[30];
        $arr['domConductor']         = $sin[31];
        $arr['dniConductor']        = $sin[32];

        $arr['success']             = true;
    }

    echo json_encode($arr);
}

// Función donde mostramos las imágenes del siniestro.
public function mostrarImgSin($post)
{
    $idSin = $post['idSin'];
    $imagenes = array();
    $info = array();
    $mysqli = $this->conexion;

```

```

        $sql = "SELECT IMG.id, IMG.nombre, IMG.idSiniestro, IMG.ub
acion,
                SIN.nroSiniestro nroSin,
                CONCAT(MAR.nombre, ' ', MO.nombre) AS marModVeh
iculo

                FROM img_siniestro IMG
                INNER JOIN siniestro SIN ON SIN.id = IMG.idSiniest
ro
                INNER JOIN poliza POL ON POL.id = SIN.idPoliza
                INNER JOIN vehiculo VE ON VE.id = POL.vehiculoid
                INNER JOIN marca MAR ON MAR.id = VE.marca_id
                INNER JOIN modelo MO ON MO.id = VE.modelo_id
                WHERE idSiniestro = ?";

$stmt = $mysqli->prepare($sql);

if($stmt!==FALSE){
    $stmt->bind_param('i', $idSin);
    $stmt->execute();
    $resultado = $stmt->get_result();

    if($resultado->num_rows > 0){

        while($dato = $resultado->fetch_array()){
            $arr['id']           = $dato[0];
            $arr['nombre']       = $dato[1];
            $arr['idSin']        = $dato[2];
            $arr['ubicacion']   = $dato[3];
            $info['nroSin']      = $dato[4];
            $info['marcaModVe'] = $dato[5];

            $arr['success']     = true;

            // Mostramos los datos en formato HTML para
            // imprimirlos directamente.
            $arr1 = '<div class="gallery-card">
                        <a href="'. $dato[3] .'" data-
lightbox="roadtrip"></a>
                                <button type="button" title="Elimi
nar imagen" class="eliminarImg" value="'. $dato[0] .'> <i class="fas fa
-trash-alt"></i> </button>
                    </div>';

            $imagenes[] = $arr1;
        }
    }
}

```

```

        $stmt->close();
        echo json_encode(array('data' => $imagenes, 'info' => $info));
    }

    // Función donde eliminamos una imagen del siniestro.
    public function eliminarImgSin($post)
    {
        $idImg = $post['idImg'];
        $arr = array('success'=>false, 'accion'=>0);
        $mysqli = $this->conexion;

        // Obtenemos la informacion de la imagen
        $sqlInfo = "SELECT idSiniestro, nombre
                    FROM img_siniestro
                    WHERE id = ?";
        $stmt = $mysqli->prepare($sqlInfo);
        if($stmt!==FALSE){
            $stmt->bind_param('i', $idImg);
            $stmt->execute();
            $resultado = $stmt->get_result();

            if($resultado->num_rows > 0){
                while($dato = $resultado->fetch_array()){
                    $idSiniestro = $dato[0];
                    $nombre = $dato[1];
                }
            }

            // Consultas para eliminar los datos
            $sqlEliminar = "DELETE FROM img_siniestro WHERE id
= ?";
            $stmt = $mysqli->prepare($sqlEliminar);

            if($stmt !== FALSE)
            {
                $stmt->bind_param('i', $idImg);
                $ejec = $stmt->execute();
                $ubicacion = '../img/ImgSin/'.$nombre;
                unlink($ubicacion); //eliminamos el archivo de
la ruta

                if($ejec){
                    $arr = array('success'=>true, 'idSin'=>$id
Siniestro);
                }
            }
        }
    }
}

```

```

$stmt->close();

echo json_encode($arr);
}

// Función donde listamos los tipos de siniestros para un
select.
public function listadoTipoSiniestro()
{
    $mysqli = $this->conexion;
    $tiposSin = '';

    $sql = "SELECT id, nombre FROM tiposiniestro";
    $stmt = $mysqli->prepare($sql);
    if($stmt!=FALSE){
        $stmt->execute();
        $rs = $stmt->get_result();
        if($rs->num_rows > 0){

            // Concatenamos los datos en una variable en
formato HTML.
            $tiposSin .= '<option value="0" selected disabled>
Selecciona</option>';
            foreach($rs as $TS){
                $tiposSin .= '<option value="'. $TS['id'] .'>' .
$TS['nombre'] . '</option>';
            }
        }
        $stmt->close();
    }

    return $tiposSin;
}

// Función donde listamos los estados de los siniestros para
un select.
public function listadoEstadosSiniestro()
{
    $mysqli = $this->conexion;
    $estadosSin = '';

    $sql = "SELECT id, nombre FROM estadosiniestro";
    $stmt = $mysqli->prepare($sql);
    if($stmt!=FALSE){
        $stmt->execute();
        $rs = $stmt->get_result();
        if($rs->num_rows > 0){

```

```
// Concatenamos los resultados en una variable en
formato HTML.
    $estadosSin .= '<option value="0" selected disable
d>Seleccione</option>';
    foreach($rs as $estado){
        $estadosSin .= '<option value="'. $estado['id']
.'">' . $estado['nombre'] . '</option>';
    }
}
$stmt->close();

return $estadosSin;
}
}
```

➔ 4.3.2.6 fichero “sistemaController.php”

Este fichero contiene todas las funciones que realizan las acciones relacionadas a las acciones directas del sistema. En este caso solo tenemos las funciones para iniciar y cerrar sesión.

```
<?php

// Clase principal.
class SistemaController{

    private $conexion;

    // Función constructora donde realizamos la conexión a base de datos.
    public function __construct()
    {
        $this->conexion = Database::connect();
        date_default_timezone_set("America/Argentina/Buenos_Aires");
    }

    // Función para validar el inicio de sesión en el sistema.
    public function loginSistema($post)
    {
        $info = $_POST['info'];
        $user = $info[0];
        $pass = $info[1];
        $idSu = $info[2];

        $arr['existe'] = false;
        $arr['rol'] = false;
        $arr['passCorr'] = '';
        $arr['sucDisp'] = false;
        $arr['activo'] = '';

        $mysqli = $this->conexion;

        // Abrimos una sesión y buscamos el correo o usuario.
        session_start();

        $sql = "SELECT id, nombre, usuario, correo, pass, rol, estado
                FROM usuarios
                WHERE usuario = ? OR correo = ? ";

        $stmt = $mysqli->prepare($sql);
```

```
if($stmt){
    $stmt->bind_param('ss', $user, $user);
    $stmt->execute();
    $r = $stmt->get_result();

    // Si existe un registro que coincida con el usuario o
correo, pasamos.
    if($r->num_rows == 1){

        $arr['existe'] = true;
        $datos = $r->fetch_assoc();

        // Verificamos si el hash de la contraseña coincide
con el grabado en la base de datos.
        if(password_verify($pass, $datos['pass'])){

            $arr['passCorr'] = 'si';

            //Consultamos si la sucursal se encuentra disponib
le, corroborando la existencia del id.
            $sqlSuc = "SELECT id FROM sucursales WHERE id = ?"
;

            $stmtSuc = $mysqli->prepare($sqlSuc);
            if($stmtSuc){
                $stmtSuc->bind_param('i', $idSu);
                $stmtSuc->execute();
                $rSuc = $stmtSuc->get_result();

                if($rSuc->num_rows > 0){
                    $arr['sucDisp'] = true;

                    // Guardamos en una variable de sesión los
datos del usuario.
                    $datos['sucursal'] = $idSu;
                    $_SESSION['user'] = $datos;

                    // Comprobamos si el estado actual es
Activo.
                    if($datos['estado'] == 'Activo'){

                        $arr['activo'] = 'si';

                        // Verificamos el rol del usuario.
                        if($datos['rol'] == 'Administrador'){
                            $arr['rol'] = 'admin';

                        }else if($datos['rol'] == 'Usuario'){
                            $arr['rol'] = 'user';
                        }
                    }
                }
            }
        }
    }
}
```

```

        }else if($datos['estado'] == 'Baja'){
            $arr['activo'] = 'no';
        }
    }else{//La sucursal no se encuentra registrada
en el sistema
        $arr['sucDisp'] = false;
    }

    if($stmtSuc){
        $stmtSuc->close();
    }

}else{//La contraseña es incorrecta
    $arr['passCorr'] = 'no';
}

}else{
    $arr['existe'] = false;
}
}

$stmt->close();
return json_encode($arr);
}

// Función para cerrar la sesión del sistema.
public function logoutSistema()
{
    session_start();
    session_destroy();
    $arr['success'] = true;

    return json_encode($arr);
}
}

```

⇒ 4.3.2.7 fichero “sucursalesController.php”

Este fichero contiene todas las funciones que realizan las acciones relacionadas con las sucursales del sistema.

```
<?php

// Clase general.
class SucursalesController
```

```

$localidad = filter_var($info[5], FILTER_VALIDATE_INT);
if($localidad === FALSE || is_null($localidad)) {$error = true;}

if(!$error){

    // Generamos un numero aleatorio para la sucursal.
    $ident = chr(rand(ord('A'), ord('Z'))).chr(rand(ord('A'),
ord('Z'))). '-' . rand(111111, 999999);

    // Formateamos los datos con mayúsculas.
    $nombre = ucwords($nombre);
    $ident = ucwords($ident);
    $direccion = ucwords($direccion);

    $sql = "INSERT INTO sucursales (nombre, iden_sucursal, cui
t, direccion, cp, localidad, provincia)
        VALUES (?, ?, ?, ?, ?, ?, ?)";
    $stmt = $mysqli->prepare($sql);
    if($stmt!=FALSE){
        $stmt-
>bind_param('ssssiii', $nombre, $ident, $cuit, $direccion, $cp, $local
idad, $provincia);
        if($stmt->execute()){
            $arr['success'] = true;
        }
    }

    $stmt->close();
}

return json_encode($arr);
}

// Función para eliminar una sucursal.
public function eliminarSucursal($post)
{
    $idSuc = $post['idSuc'];
    $arr['success'] = false;
    $mysqli = $this->conexion;

    $sql = "DELETE FROM sucursales WHERE id = ?";
    $stmt = $mysqli->prepare($sql);
    if($stmt!=FALSE){
        $stmt->bind_param('i', $idSuc);
        if($stmt->execute()){
            $arr['success'] = true;
        }
    }
}

```

```

$stmt->close();

return json_encode($arr);
}

// Función para traer la información de una sucursal.
public function traerInfoSucursal($post)
{
    $arr['success'] = false;
    $mysqli = $this->conexion;
    $idSuc = $post['idSuc'];
    $info = ' ';

    $sql = "SELECT S.nombre      AS nombre,
                  S.iden_sucursal AS iden,
                  S.cuit          AS cuit,
                  S.direccion     AS direccion,
                  S.cp            AS cp,
                  S.localidad     AS localidadId,
                  S.provincia     AS provinciaId,
                  proper(L.nombre) AS localidad,
                  P.nombre        AS provincia
        FROM sucursales S
        INNER JOIN localidad L ON L.id = S.localidad
        INNER JOIN provincia P ON P.id = S.provincia
        WHERE S.id = ?";

    $stmt = $mysqli->prepare($sql);
    if($stmt){
        $stmt->bind_param('i', $idSuc);
        if($stmt->execute()){
            $r = $stmt->get_result();
            $info = $r->fetch_assoc();
            $arr['info'] = $info;
            $arr['success'] = true;
        }
    }
    $stmt->close();

    return json_encode($arr);
}

// Función para modificar la información de una sucursal.
public function modificarSucursal($post)
{
    $info = $post['data'];
    $arr['success'] = false;
    $arr['repetido'] = false;
    $mysqli = $this->conexion;
}

```

```

        // Validamos los datos.
        $nombre = filter_var($info[0], FILTER_SANITIZE_FULL_SPECIAL_CHARS);
        if($nombre === FALSE || is_null($nombre)) {$error = true;}

        $cuit = filter_var($info[1], FILTER_SANITIZE_FULL_SPECIAL_CHARS);
        if($cuit === FALSE || is_null($cuit)) {$error = true;}

        $direccion = filter_var($info[2], FILTER_SANITIZE_FULL_SPECIAL_CHARS);
        if($direccion === FALSE || is_null($direccion)) {$error = true;}

        $provincia = filter_var($info[3], FILTER_VALIDATE_INT);
        if($provincia === FALSE || is_null($provincia)) {$error = true;}

        $cp = filter_var($info[4], FILTER_VALIDATE_INT);
        if($cp === FALSE || is_null($cp)) {$error = true;}

        $localidad = filter_var($info[5], FILTER_VALIDATE_INT);
        if($localidad === FALSE || is_null($localidad)) {$error = true;}

        $idSuc = $info[6];

        $sql = "SELECT nombre, cuit, direccion, cp, localidad, provincia
                FROM sucursales
                WHERE nombre      = ? AND
                      cuit        = ? AND
                      direccion   = ? AND
                      cp          = ? AND
                      localidad   = ? AND
                      provincia   = ? ";

```

```

$stmt = $mysqli->prepare($sql);
if($stmt){
    $stmt-
>bind_param('ssssii', $nombre, $cuit, $direccion, $cp, $localidad, $provincia);
    if($stmt->execute()){
        $r = $stmt->get_result();

        // Si los datos ingresados son diferentes a los
        // grabados ejecutamos la consulta.
        if($r->num_rows == 0){
            $arr['repetido'] = false;

            if($arr['repetido'] == false){

```

```

$sql1 = "UPDATE sucursales
          SET nombre      = ?,
              cuit        = ?,
              direccion   = ?,
              cp          = ?,
              localidad   = ?,
              provincia   = ?
          WHERE id       = ? ";
$stmt1 = $mysqli->prepare($sql1);
if($stmt1){
    $stmt1-
>bind_param('ssssiiii', $nombre, $cuit, $direccion, $cp, $localidad, $p
rovincia, $idSuc);
    if($stmt1->execute()){
        $arr[ 'success' ] = true;
    }

    $stmt1->close();
}
}else{
    $arr[ 'repetido' ] = true;
}
}

$stmt->close();

return json_encode($arr);
}

// Función para listar todas las sucursales en el select de inicio
de sesión.
public function listadoSucursalesSelect()
{
    $mysqli = $this->conexion;
    $arr[ 'success' ] = false;
    $data = array();

    $sql = "SELECT id, nombre, iden_sucursal FROM sucursales";
    $stmt = $mysqli->prepare($sql);
    if($stmt->execute()){
        $r = $stmt->get_result();

        foreach($r as $suc){
            $info[ 'id' ]      = $suc[ 'id' ];
            $info[ 'nombre' ] = $suc[ 'nombre' ];
            $info[ 'iden' ]   = $suc[ 'iden_sucursal' ];
        }
    }
}

```

```
        $data[]      = $info;
    }
    if(isset($data)){
        $arr['success'] = true;
        $arr['data']   = $data;
    }
}
$stmt->close();
return json_encode($arr);
}
}
```

➔ 4.3.2.8 fichero “tomadoresController.php”

Este fichero contiene todas las funciones que realizan las acciones relacionadas con los clientes y tomadores de polizas en el sistema.

```
<?php

// Clase principal.
class TomadoresController{

    private $conexion;

    // Función constructora para conectarnos a la base de datos.
    public function __construct()
    {
        $this->conexion = Database::connect();
        date_default_timezone_set("America/Argentina/Buenos_Aires");
    }

    // Función para mostrar los datos de un tomador/cliente.
    public function mostrarDatosCliente($post)
    {
        $conexion = mysqli_connect('localhost', 'root', '', 'aseguradora');

        $idtomador = $post['idtomador'];

        $arr = array('success'=>false);

        $query = "SELECT T.id, T.nombre, T.documento, T.persona, T.nacionalidad, P.nombre AS provincia, proper(L.nombre) AS localidad, T.cp, T.calle, T.genero, T.telefono, T.correo, T.fecha_nac
                  FROM tomador T
                  INNER JOIN provincia P ON P.id = T.provincia
                  INNER JOIN localidad L ON L.id = T.localidad
                  WHERE T.id = '$idtomador'";

        $resultado = $this->conexion->query($query);

        while ($tomador = mysqli_fetch_row($resultado)) {
            $_SESSION['T.id']      = $tomador[0];
            $arr['nombre']         = $tomador[1];
            $arr['documento']      = $tomador[2];
            $arr['persona']        = $tomador[3];
            $arr['nacionalidad']   = $tomador[4];
            $arr['provincia']      = $tomador[5];
            $arr['localidad']       = $tomador[6];
```

```

        $arr[ 'cp' ]           = $tomador[ 7 ];
        $arr[ 'calle' ]         = $tomador[ 8 ];
        $arr[ 'genero' ]        = $tomador[ 9 ];
        $arr[ 'telefono' ]       = $tomador[ 10 ];
        $arr[ 'correo' ]          = $tomador[ 11 ];
        $arr[ 'fecha_nac' ]       = $tomador[ 12 ];
        $arr[ 'success' ]         = true;
    }

    echo json_encode(array('data' => $arr));
}

// Función para grabar un nuevo cliente/tomador.
public function grabarCliente($post)
{
    try{

        $mysqli = $this->conexion;

        $error = false;

        $arr = array('success'=>false);

        $datos = $post['d'];

        // Validamos los datos ingresados.
        $nom = filter_var($datos[0], FILTER_SANITIZE_FULL_SPECIAL_CHARS);
        if($nom==FALSE || is_null($nom)) {$error = true;}

        $docu = filter_var($datos[1], FILTER_SANITIZE_FULL_SPECIAL_CHARS);
        if($docu==FALSE || is_null($docu)) {$error = true;}

        $per = filter_var($datos[2], FILTER_SANITIZE_FULL_SPECIAL_CHARS);
        if($per==FALSE || is_null($per)) {$error = true;}

        $nac = filter_var($datos[3], FILTER_SANITIZE_FULL_SPECIAL_CHARS);
        if($nac==FALSE || is_null($nac)) {$error = true;}

        $pro = filter_var($datos[4], FILTER_SANITIZE_FULL_SPECIAL_CHARS);
        if($pro==FALSE || is_null($pro)) {$error = true;}

        $loc = filter_var($datos[5], FILTER_SANITIZE_FULL_SPECIAL_CHARS);
        if($loc==FALSE || is_null($loc)) {$error = true;}
    }
}

```

```

$cp = filter_var($datos[6], FILTER_SANITIZE_FULL_SPECIAL_CHARS);
if($cp === FALSE || is_null($cp)) {$error = true;}

$call = filter_var($datos[7], FILTER_SANITIZE_FULL_SPECIAL_CHARS);
if($call === FALSE || is_null($call)) {$error = true;}

$gen = filter_var($datos[8], FILTER_SANITIZE_FULL_SPECIAL_CHARS);
if($gen === FALSE || is_null($gen)) {$error = true;}

$tel = filter_var($datos[9], FILTER_SANITIZE_FULL_SPECIAL_CHARS);
if($tel === FALSE || is_null($tel)) {$error = true;}

if($datos[10] == ''){
    $corr = '';
} else{
    $corr = filter_var($datos[10], FILTER_VALIDATE_EMAIL);

    if($corr === FALSE) {$error = true;}
}

$fechNac = filter_var($datos[11], FILTER_SANITIZE_FULL_SPECIAL_CHARS);
if($fechNac === FALSE || is_null($fechNac)) {$error = true;}

if(!$error){

    // Formateamos los datos con mayúsculas.
    $nom1 = ucwords($nom);
    $per1 = ucwords($per);
    $nac1 = ucwords($nac);
    $pro1 = ucwords($pro);
    $loc1 = ucwords($loc);
    $call1 = ucwords($call);
    $gen1 = ucwords($gen);

    // Generamos un numero aleatorio para el usuario.
    $num_tom = rand(9999999, 99999999);

    $mysqli->set_charset("utf8");
    $sql = "INSERT INTO tomador (num_tom, nombre, documento, persona, nacionalidad, provincia, localidad, cp, calle, genero, telefono, correo, fecha_nac) VALUES (?,?,?,?,?,?,?,?,?,?,?,?,?,?)";
    $stmt = $mysqli->prepare($sql);
    if ($stmt !== FALSE) {

```

```

        $stmt->bind_param('ssssssssssss', $num_tom, $nom1, $docu, $per1, $nac1, $pr
o1, $loc1, $cp, $call1, $gen1, $tel, $corr, $fechNac);
        $stmt->execute();

        // Buscamos el id del tomador registrado
        $sql = "SELECT MAX(id) AS id FROM tomador";
        $stmt = $mysqli->prepare($sql);
        $stmt->execute();
        $rs = $stmt->get_result();
        if ($rs->num_rows > 0) {
            while ($fila = $rs->fetch_array()) {
                $idTomador = $fila[0];
            }
        }

        $stmt->close();
        $arr = array('success'=>true, 'idTomador'=>$idToma
dor);
    }

    return json_encode($arr);

}catch(Exception $e){
    return $e->getMessage();
}
}

// Funcion para mostrar la informacion de un cliente/tomador.
public function mostrarDatosTomador($post)
{
    $id = $post['id'];
    $arr = array();
    $mysqli = $this->conexion;
    $sql = "SELECT T.id, T.num_tom, T.nombre, T.documento, T.perso
na, T.nacionalidad, P.nombre as provinciaNombre, proper(L.nombre) as l
ocalidadNombre, T.cp, T.calle, T.genero, T.telefono, T.correo, T.fecha
_nac, T.localidad as localidadId, T.provincia as provinciaId
    FROM tomador T
    INNER JOIN provincia P ON P.id = T.provincia
    INNER JOIN localidad L ON L.id = T.localidad
    WHERE T.id = ?";
    $stmt = $mysqli->prepare($sql);

    if($stmt!==FALSE){
        $stmt->bind_param('i', $id);
        $stmt->execute();
    }
}

```

```

        $resultado = $stmt->get_result();

        if($resultado->num_rows > 0){

            while($dato = $resultado->fetch_array()){

                $arr['id'] = $dato[0];
                $arr['nroTom'] = $dato[1];
                $arr['nombre'] = $dato[2];
                $arr['documento'] = $dato[3];
                $arr['persona'] = $dato[4];
                $arr['nacionalidad'] = $dato[5];
                $arr['provinciaNombre'] = $dato[6];
                $arr['localidadNombre'] = $dato[7];
                $arr['cp'] = $dato[8];
                $arr['calle'] = $dato[9];
                $arr['genero'] = $dato[10];
                $arr['telefono'] = $dato[11];
                $arr['correo'] = $dato[12];
                $arr['fecha_nac'] = $dato[13];
                $arr['localidadId'] = $dato[14];
                $arr['provinciaId'] = $dato[15];

                $arr['success'] = true;
            }
        }

        $stmt->close();
        echo json_encode($arr);
    }

    // Función para modificar la información de un cliente/tomador.
    public function modificarDatosTomador($post)
    {
        $datos = $post['datos'];

        $arr = array('success'=>false, 'accion'=>0);

        $error = false;

        // Validamos los datos ingresados.
        $nom = filter_var($datos[0], FILTER_SANITIZE_FULL_SPECIAL_CHARS);
        if($nom==FALSE || is_null($nom)) {$error = true;}

        $docu = filter_var($datos[1], FILTER_SANITIZE_FULL_SPECIAL_CHARS);
        if($docu==FALSE || is_null($docu)) {$error = true;}
    }
}

```

```

        $per = filter_var($datos[2], FILTER_SANITIZE_FULL_SPECIAL_CHARS);
        if($per === FALSE || is_null($per)) {$error = true;}

        $nac = filter_var($datos[3], FILTER_SANITIZE_FULL_SPECIAL_CHARS);
        if($nac === FALSE || is_null($nac)) {$error = true;}

        $fechNac = filter_var($datos[4], FILTER_SANITIZE_FULL_SPECIAL_CHARS);
        if($fechNac === FALSE || is_null($fechNac)) {$error = true;}

        $pro = filter_var($datos[5], FILTER_SANITIZE_FULL_SPECIAL_CHARS);
        if($pro === FALSE || is_null($pro)) {$error = true;}

        $cp = filter_var($datos[6], FILTER_SANITIZE_FULL_SPECIAL_CHARS);
        if($cp === FALSE || is_null($cp)) {$error = true;}

        $loc = filter_var($datos[7], FILTER_SANITIZE_FULL_SPECIAL_CHARS);
        if($loc === FALSE || is_null($loc)) {$error = true;}

        $tel = filter_var($datos[8], FILTER_SANITIZE_FULL_SPECIAL_CHARS);
        if($tel === FALSE || is_null($tel)) {$error = true;}

        $call = filter_var($datos[9], FILTER_SANITIZE_FULL_SPECIAL_CHARS);
        if($call === FALSE || is_null($call)) {$error = true;}

        $gen = filter_var($datos[10], FILTER_SANITIZE_FULL_SPECIAL_CHARS);
        if($gen === FALSE || is_null($gen)) {$error = true;}

        if($datos[11] == ''){
            $corr = '';
        }else{
            $corr = filter_var($datos[11], FILTER_VALIDATE_EMAIL);
            if($corr === FALSE) {$error = true;}
        }

        $id = $datos[12];

        if(!$error){

            $nom1 = ucwords($nom);
            $per1 = ucwords($per);

```

```

$nac1 = ucwords($nac);
$pro1 = ucwords($pro);
$loc1 = ucwords($loc);
$call1 = ucwords($call);
$gen1 = ucwords($gen);

$sql =
"SELECT id, nombre, documento, persona, nacionalidad, provincia,
localidad, cp, calle, genero, telefono, correo, fecha_nac
FROM tomador
WHERE nombre      = ? AND
      documento    = ? AND
      persona       = ? AND
      nacionalidad = ? AND
      provincia     = ? AND
      localidad     = ? AND
      cp            = ? AND
      calle          = ? AND
      genero         = ? AND
      telefono       = ? AND
      correo         = ? AND
      fecha_nac     = ? AND
      id             = ?    ";
}

$mysqli = $this->conexion;
$stmt = $mysqli->prepare($sql);
$stmt-
>bind_param('ssssssssssi', $nom1, $docu, $per1, $nac1, $pro1, $loc1,
$cp, $call1, $gen1, $tel, $corr, $fechNac, $id);
$stmt->execute();
$rs = $stmt->get_result();

if($rs->num_rows == 0){

    // Si no existe ningún registro con los mismos datos,
ejecutamos la consulta.
    $sql = "UPDATE tomador
            SET nombre      = ?,
                documento    = ?,
                persona       = ?,
                nacionalidad = ?,
                provincia     = ?,
                localidad     = ?,
                cp            = ?,
                calle          = ?,
                genero         = ?,
                telefono       = ?,
                correo         = ?,
                fecha_nac     = ?"
}

```

```

        WHERE id      = ? ";

$stmt = $mysqli->prepare($sql);

if ($stmt!==FALSE) {
    $stmt-
>bind_param('ssssssssssssi', $nom1, $docu, $per1, $nac1, $pro1, $loc1,
$cp, $call1, $gen1, $tel, $corr, $fechNac, $id);
    $stmt->execute();
    $arr = array('success'=>true, 'accion'=>1);
}

}else{
    $arr = array('success'=>true, 'accion'=>2);
}
}
$stmt->close();
echo json_encode($arr);
}

// Función para eliminar un cliente/tomador del sistema.
public function eliminarTomador($id)
{

    //Acción 1 El tomador fue eliminado correctamente
    //Acción 2 No se ejecuto la consulta, no fue posible eliminar
el tomador

$id = $id['id'];
$arr = array('success'=>false, 'accion'=>0);
$mysqli = $this->conexion;

$esTomador = "SELECT P.clienteid
    FROM poliza P
    WHERE P.clienteid = ?";

$stmt = $mysqli->prepare($esTomador);
$stmt->bind_param('i', $id);
$stmt->execute();
$result = $stmt->get_result();

// Si el cliente no es tomador de ninguna poliza, ejecutamos
la consulta, de lo contrario no se realizara la acción.
if($result->num_rows == 0){
    $sql = "DELETE FROM tomador WHERE id = ?";
    $stmt = $mysqli->prepare($sql);
    $stmt->bind_param('i', $id);
    if($stmt->execute()){
        $arr = array('success'=>true , 'accion'=>1);
    }
}
}

```

```

        }else{
            $arr = array('success'=>true, 'accion'=>2);
        }
    }else{
        $arr = array('success'=>true, 'accion'=>2, 'cantPol' => $result->num_rows);
    }

    $stmt->close();

    echo json_encode($arr);
}

// Función para listar los clientes/tomadores en un select.
public function listarTomadores()
{
    $tomadores = '';
    $mysqli = $this->conexion;
    $tomadores = array();

    $sql = "SELECT id, nombre, documento FROM tomador";

    $stmt = $mysqli->prepare($sql);
    if($stmt!==FALSE){
        $stmt->execute();

        $rs = $stmt->get_result();
        if ($rs->num_rows > 0) {
            foreach($rs as $tom){
                $arr = array();
                $arr['id'] = $tom['id'];
                $arr['nombre'] = $tom['nombre'];
                $arr['doc'] = $tom['documento'];
                $tomadores[] = $arr;
            }
        }

        $stmt->close();
        echo json_encode($tomadores);
    }
}
}

```

➔ 4.3.2.9 fichero “usersController.php”

Este fichero contiene todas las funciones que realizan las acciones relacionadas con los usuarios del sistema.

```
<?php

// Clase principal.
class UsersController{

    private $conexion;

    // Función constructora para conectarnos a la base de datos.
    public function __construct()
    {
        $this->conexion = Database::connect();
        date_default_timezone_set("America/Argentina/Buenos_Aires");
    }

    // Función para grabar un nuevo usuario.
    public function grabarUser($post)
    {
        $formUser = $post['formUser'];
        $arr['success'] = false;
        $error = false;
        $mysqli = $this->conexion;

        // Validamos los datos ingresados.
        $nombreUser = filter_var($formUser[0], FILTER_SANITIZE_FULL_SPECIAL_CHARS);
        if($nombreUser === FALSE || is_null($nombreUser)) {$error = true;}

        $documentoUser = filter_var($formUser[1], FILTER_SANITIZE_FULL_SPECIAL_CHARS);
        if($documentoUser === FALSE || is_null($documentoUser)){ $error = true; }

        $nacionalidadUser = filter_var($formUser[2], FILTER_SANITIZE_FULL_SPECIAL_CHARS);
        if($nacionalidadUser === FALSE || is_null($nacionalidadUser)) { $error = true; }

        $provinciaUser = filter_var($formUser[3], FILTER_VALIDATE_INT);
        if($provinciaUser === FALSE || is_null($provinciaUser)) { $error = true; }
```

```

        $localidadUser = filter_var($formUser[4], FILTER_VALIDATE_INT)
;
        if($localidadUser === FALSE || is_null($localidadUser)) {$error = true; }

        $direccionUser = filter_var($formUser[5], FILTER_SANITIZE_FULL_SPECIAL_CHARS);
        if($direccionUser === FALSE || is_null($direccionUser)) {$error = true; }

        $nacUser = filter_var($formUser[6], FILTER_SANITIZE_FULL_SPECIAL_CHARS);
        if($nacUser === FALSE || is_null($nacUser)) {$error = true; }

        $telefonoUser = filter_var($formUser[7], FILTER_SANITIZE_FULL_SPECIAL_CHARS);
        if($telefonoUser === FALSE || is_null($telefonoUser)) {$error = true; }

        $correoUser = filter_var($formUser[8], FILTER_VALIDATE_EMAIL);
        if($correoUser === FALSE || is_null($correoUser)) {$error = true; }

        $generoUser = filter_var($formUser[9], FILTER_SANITIZE_FULL_SPECIAL_CHARS);
        if($generoUser === FALSE || is_null($generoUser)) {$error = true; }

        $rolUser = filter_var($formUser[10], FILTER_SANITIZE_FULL_SPECIAL_CHARS);
        if($rolUser === FALSE || is_null($rolUser)) {$error = true; }

        if(! $error){

            $numUser = rand(999999999,999999999);
            $usuario = rand(10000,99999). '-' . $documentoUser;
            $pass = rand(999999999, 9999999999);
            $hashPass = password_hash($pass, PASSWORD_DEFAULT, ['cost' => 10]);

            //Guardamos la contraseña en una variable de session para despues enviarla limpia por correo
            session_start();
            $_SESSION['pass'] = $pass;

            if($rolUser == 'usuario'){
                $rol = 'Usuario';
            }else if($rolUser == 'admin'){


```

```

        $rol = 'Administrador';
    }

    // Formateamos los datos con mayúsculas.
    $nombreUser = ucwords($nombreUser);
    $nacionalidadUser = ucwords($nacionalidadUser);
    $direccionUser = ucwords($direccionUser);
    $generoUser = ucwords($generoUser);

    $estado = 'Activo';

    $sql = "INSERT INTO usuarios (nro, nombre, documento, nacionalidad, provincia, localidad, direccion, fecha_nac, telefono, correo, genero, usuario, pass, rol, estado)
            VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)";
    $stmt = $mysqli->prepare($sql);
    if($stmt){
        $stmt-
>bind_param('ssssiissssssss', $numUser, $nombreUser, $documentoUser,
$nacionalidadUser, $provinciaUser, $localidadUser, $direccionUser, $na
cUser, $telefonoUser, $correoUser, $generoUser, $usuario, $hashPass, $
rol, $estado);
        if($stmt->execute()){

            //Obtenemos el id del usuario ingresado
            $conIdUs = "SELECT max(id) AS id FROM usuarios";
            $stmt2 = $mysqli->prepare($conIdUs);
            if($stmt2->execute()){
                $r2 = $stmt2->get_result();
                $infoUs = $r2->fetch_assoc();
                $idUser = $infoUs['id'];

                $arr[ 'success' ] = true;
                $arr[ 'idUser' ] = $idUser;
                $arr[ 'email' ] = $correoUser;
            }
        }else{
            $arr[ 'success' ] = 'error';
        }
    }

}else{
    $arr[ 'success' ] = false;
}

$stmt->close();
echo json_encode($arr);
}

```

```

// Función para dar de baja un usuario.
public function bajaUser($post)
{
    $idUser = $post['idUser'];
    $mysqli = $this->conexion;
    $arr['success'] = false;
    $arr['mismoEst'] = false;

    $sql1 = "SELECT estado FROM usuarios WHERE id = ?";
    $stmt = $mysqli->prepare($sql1);
    if($stmt->bind_param('i', $idUser)) {

        $stmt->execute();
        $r = $stmt->get_result();
        $dat = $r->fetch_assoc();

        // Si el estado es diferente de “Baja” ejecutamos la
        consulta.
        if($dat['estado'] != 'Baja') {

            $sql = "UPDATE usuarios SET estado = 'Baja' WHERE id =
            ?";
            $stmt = $mysqli->prepare($sql);
            if($stmt->bind_param('i', $idUser)){
                $stmt->execute();
                $arr['success'] = true;
                $arr['mismoEst'] = false;
            }
            }else{
                $arr['mismoEst'] = true;
            }
        }

        $stmt->close();
        echo json_encode($arr);
    }

    // Función para reactivar o dar de alta un usuario.
    public function reactivarUser($post)
    {
        $idUser = $post['idUser'];
        $mysqli = $this->conexion;
        $arr['success'] = false;
        $arr['mismoEst'] = false;

        $sql1 = "SELECT estado FROM usuarios WHERE id = ?";
        $stmt = $mysqli->prepare($sql1);
        if($stmt->bind_param('i', $idUser)){

```

```

$stmt->execute();
$r = $stmt->get_result();
$dat = $r->fetch_assoc();

// Ejecutamos la consulta en caso de que el usuario se
encuentre dado de baja.
if($dat['estado'] != 'Activo'){

    $sql = "UPDATE usuarios SET estado = 'Activo' WHERE id
= ?";
    $stmt = $mysqli->prepare($sql);
    if($stmt->bind_param('i', $idUser)){
        $stmt->execute();
        $arr['success'] = true;
        $arr['mismoEst'] = false;
    }
    }else{
        $arr['mismoEst'] = true;
    }
}

$stmt->close();
echo json_encode($arr);
}

// Función para traer la información de un usuario.
public function traerInfoUser($post)
{
    $idUser = $post['idUser'];
    $mysqli = $this->conexion;
    $arr['success'] = false;

    $sql = "SELECT U.nro, U.nombre, U.usuario, U.rol, (SELECT COUNT(rol) FROM usuarios WHERE rol='Administrador') AS cantAdm,
                U.estado, U.documento, U.fecha_nac, U.genero, U.correo, U.telefono, U.nacionalidad, U.direccion,
                proper(LOC.nombre) AS loc, PRO.nombre AS pro
            FROM usuarios U
            INNER JOIN localidad LOC ON LOC.id = U.localidad
            INNER JOIN provincia PRO ON PRO.id = U.provincia
            WHERE U.estado = 'Activo' AND U.id = ?";
    $stmt = $mysqli->prepare($sql);
    $stmt->bind_param('i', $idUser);
    if($stmt->execute()){
        $r = $stmt->get_result();
        if($r->num_rows > 0){
            $info = $r->fetch_assoc();
            $arr['nro'] = $info['nro'];
            $arr['nombre'] = $info['nombre'];
        }
    }
}

```

```

        $arr['usuario'] = $info['usuario'];
        $arr['rol'] = $info['rol'];
        $arr['cantAdm'] = $info['cantAdm'];

        $arr['estado'] = $info['estado'];
        $arr['documento'] = $info['documento'];
        $arr['fecha_nac'] = $info['fecha_nac'];
        $arr['genero'] = $info['genero'];
        $arr['correo'] = $info['correo'];
        $arr['telefono'] = $info['telefono'];
        $arr['nacionalidad'] = $info['nacionalidad'];
        $arr['direccion'] = $info['direccion'];
        $arr['localidad'] = $info['loc'];
        $arr['provincia'] = $info['pro'];

        $arr['success'] = true;
    }
}

$stmt->close();
echo json_encode($arr);
}

// Función para modificar el rol de un usuario.
public function modRolUser($post)
{
    $info = $post['info'];
    $idUser = $info[0];
    $nuevoRol = $info[1];

    $mysqli = $this->conexion;
    $arr['success'] = false;

    $sql = "UPDATE usuarios SET rol = ? WHERE id = ?";
    $stmt = $mysqli->prepare($sql);
    $stmt->bind_param('si', $nuevoRol, $idUser);
    if($stmt->execute()){
        $arr['success'] = true;
    }

    $stmt->close();
    echo json_encode($arr);
}

// Función para traer los datos de un usuario.
public function getMisDatosUser()
{
    // Obtenemos el id del usuario de la sesión actual.
    session_start();
}

```

```

if(isset($_SESSION['user'])){
    $idUser = $_SESSION['user']['id'];
}

$mysqli = $this->conexion;
$arr['success'] = false;

// Realizamos la consulta.
$sql = "SELECT nro, nombre, documento, nacionalidad, provincia
, localidad, direccion, fecha_nac, telefono, correo, genero, usuario
    FROM usuarios
 WHERE estado = 'Activo' AND id = ?";

$stmt = $mysqli->prepare($sql);
$stmt->bind_param('i', $idUser);
if($stmt->execute()){
    $r = $stmt->get_result();
    if($r->num_rows > 0){

        $info = $r->fetch_assoc();

        $arr['nro']           = $info['nro'];
        $arr['nombre']        = $info['nombre'];
        $arr['documento']     = $info['documento'];
        $arr['nacionalidad'] = $info['nacionalidad'];
        $arr['provincia']    = $info['provincia'];
        $arr['localidad']     = $info['localidad'];
        $arr['direccion']    = $info['direccion'];
        $arr['fecha_nac']    = $info['fecha_nac'];
        $arr['telefono']      = $info['telefono'];
        $arr['correo']         = $info['correo'];
        $arr['genero']        = $info['genero'];
        $arr['usuario']       = $info['usuario'];

        $arr['success'] = true;
    }
}
$stmt->close();
echo json_encode($arr);
}

// Función para modificar la información del usuario (Mis datos).
public function setMisDatosUser($post)
{
    $info = $post['info'];
    $arr['segPass'] = null;
    $arr['diferenteInfo'] = null;
    $arr['userDisp'] = null;
}

```

```

$arr[ 'update' ] = null;
$error = false;
$mysqli = $this->conexion;

// Traemos el id del usuario en la sesión actual.
session_start();
if(isset($_SESSION['user'])){
    $idUser = $_SESSION['user']['id'];
}

// Validamos los datos.
$nombreUser = filter_var($info[1], FILTER_SANITIZE_FULL_SPECIAL_CHARS);
if($nombreUser === FALSE || is_null($nombreUser)) {$error = true;}

$documentoUser = filter_var($info[2], FILTER_SANITIZE_FULL_SPECIAL_CHARS);
if($documentoUser === FALSE || is_null($documentoUser)){ $error = true; }

$nacionalidadUser = filter_var($info[3], FILTER_SANITIZE_FULL_SPECIAL_CHARS);
if($nacionalidadUser === FALSE || is_null($nacionalidadUser)) { $error = true; }

$provinciaUser = filter_var($info[4], FILTER_VALIDATE_INT);
if($provinciaUser === FALSE || is_null($provinciaUser)) { $error = true; }

$localidadUser = filter_var($info[5], FILTER_VALIDATE_INT);
if($localidadUser === FALSE || is_null($localidadUser)) { $error = true; }

$direccionUser = filter_var($info[6], FILTER_SANITIZE_FULL_SPECIAL_CHARS);
if($direccionUser === FALSE || is_null($direccionUser)) { $error = true; }

$nacUser = filter_var($info[7], FILTER_SANITIZE_FULL_SPECIAL_CHARS);
if($nacUser === FALSE || is_null($nacUser)) { $error = true; }

$telefonoUser = filter_var($info[8], FILTER_SANITIZE_FULL_SPECIAL_CHARS);
if($telefonoUser === FALSE || is_null($telefonoUser)) { $error = true; }

$correoUser = filter_var($info[9], FILTER_VALIDATE_EMAIL);

```

```

        if($correoUser === FALSE || is_null($correoUser)) {$error = true
;}

        $generoUser = filter_var($info[10], FILTER_SANITIZE_FULL_SPECI
AL_CHARS);
        if($generoUser === FALSE || is_null($generoUser)) {$error = true
;}

        $usuarioUser = filter_var($info[11], FILTER_SANITIZE_FULL_SPEC
IAL_CHARS);
        if($usuarioUser === FALSE || is_null($usuarioUser)) {$error = tr
ue;}

        $pass1 = filter_var($info[12], FILTER_SANITIZE_FULL_SPECIAL_CH
ARS);
        if($pass1 === FALSE || is_null($pass1)) {$error = true;}

        $pass2 = filter_var($info[13], FILTER_SANITIZE_FULL_SPECIAL_CH
ARS);
        if($pass2 === FALSE || is_null($pass2)) {$error = true;}

        if(!$error){

            $seguridadPass = $info[14];

            // Formateamos los datos con mayúsculas.
            $nombreUser = ucwords($nombreUser);
            $nacionalidadUser = ucwords($nacionalidadUser);
            $direccionUser = ucwords($direccionUser);
            $generoUser = ucwords($generoUser);

            // Obtenemos la contraseña actual.
            $stmt1 = $mysqli-
>prepare("SELECT pass FROM usuarios WHERE id = $idUser");
            $stmt1->execute();
            $r1 = $stmt1->get_result();
            $data1 = $r1->fetch_assoc();
            if(password_verify($seguridadPass, $data1['pass'])){
                $arr['segPass'] = true;

                // Comprobamos que no se ingresen los mismos datos.
                $sql2 = "SELECT nombre, documento, nacionalidad, provi
ncia, localidad, direccion, fecha_nac, telefono, correo, genero, usuari
o
                    FROM usuarios
                    WHERE id = $idUser";
                $stmt2 = $mysqli->prepare($sql2);
                $stmt2->execute();
                $r2 = $stmt2->get_result();

```

```

$data2 = $r2->fetch_assoc();
//Si toda la info modificada es igual a la registrada,
no se realizara la modificación.
if( $data2['nombre'] != $nombreUser
|| $data2['documento'] != $documentoUser
|| $data2['nacionalidad'] != $nacionalidadUser
|| $data2['provincia'] != $provinciaUser
|| $data2['localidad'] != $localidadUser
|| $data2['direccion'] != $direccionUser
|| $data2['fecha_nac'] != $nacUser
|| $data2['telefono'] != $telefonoUser
|| $data2['correo'] != $correoUser
|| $data2['genero'] != $generoUser
|| $data2['usuario'] != $usuarioUser
|| $pass1 != ''
&& $pass2 != '' )
{
    $arr['diferenteInfo'] = true;

    //Verificamos que el nuevo nombre de usuario sea único.
$stmt3 = $mysqli-
>prepare("SELECT usuario FROM usuarios WHERE id <> $idUser");
$stmt3->execute();
$contExisteUser = 0;
$contNoExisteUser = 0;
foreach ($stmt3->get_result() as $usuario){

    $userRegistrado = $usuario['usuario'];

    if($userRegistrado == $usuarioUser){
        $contExisteUser++;
    }else{
        $contNoExisteUser++;
    }
}

//Si el usuario no se encuentra ocupado, seguimos con el proceso.
if($contExisteUser == 0){

    $arr['userDisp'] = true;

    //Si la opcion de modificar contraseña fue chequeada (= si), pasamos los valores de pass1 y pass2 al bind_param
    if($info[0]=='si'){
        $sql4 = "UPDATE usuarios
                SET nombre      = ?,
                    documento   = ?,

```

```

        nacionalidad = ?,
        provincia    = ?,
        localidad    = ?,
        direccion   = ?,
        fecha_nac   = ?,
        telefono     = ?,
        correo       = ?,
        genero       = ?,
        usuario      = ?,
        pass         = ?
        WHERE id      = ?";
```

`$hashPass = password_hash($pass1, PASS
WORD_DEFAULT, ['cost' => 10]); //Encriptamos la nueva contraseña
$stmt4 = $mysqli->prepare($sql4);
$stmt4->bind_param('sssiisssssi', $nombreUser, $documentoUser,$nacionalidadUser, $provinciaUser, $localidadUser, $direccionUser, $nacUser, $telefonoUser, $correoUser, $generoUser, $usuarioUser, $hashPass, $idUser);`

`}else if($info[0]=='no') {
 // Ejecutamos la consulta.
 $sql4 = "UPDATE usuarios
 SET nombre = ?,
 documento = ?,
 nacionalidad = ?,
 provincia = ?,
 localidad = ?,
 direccion = ?,
 fecha_nac = ?,
 telefono = ?,
 correo = ?,
 genero = ?,
 usuario = ?
 WHERE id = ?";
 $stmt4 = $mysqli->prepare($sql4);
 $stmt4->bind_param('sssiisssssi', $nombreUser, $documentoUser,$nacionalidadUser, $provinciaUser, $localidadUser, $direccionUser, $nacUser, $telefonoUser, $correoUser, $generoUser, $usuarioUser, $idUser);
}`

`if($stmt4->execute()){
 $arr['update'] = true;
 $info = [$nombreUser, null, null];
 // Actualizamos los datos en la
sesión.
 $this->actualizarSesionUser($info);`

```

        }else{
            $arr[ 'update' ] = false;
        }
        $stmt4->close();

    }else{
        $arr[ 'userDisp' ] = false;
    }

    $stmt3->close();

}else{
    $arr[ 'diferenteInfo' ] = false;
}

$stmt2->close();

}elseif{
    $arr[ 'segPass' ] = false;
}

$stmt1->close();

}elseif{
    $arr[ 'update' ] = false;
}

echo json_encode($arr);
}

// Función para obtener la foto de perfil del usuario.
public function getFotoPerfilUser()
{
    // Traemos el id del usuario de la sesión actual.
    if(!isset($_SESSION)){
        session_start();
        if(isset($_SESSION[ 'user' ][ 'id' ])){
            $idUser = $_SESSION[ 'user' ][ 'id' ];
        }
    }

    $arr[ 'existeImgPerfil' ] = false;

    $mysqli = $this->conexion;
    $sql = "SELECT id, ruta FROM imgperfilusers WHERE idUser = ? ORDER BY id DESC";
    $stmt = $mysqli->prepare($sql);
    if($stmt->bind_param('i', $idUser)){
        $stmt->execute();
    }
}

```

```

    $r = $stmt->get_result();
    $data = $r->fetch_assoc();

    // Comprobamos si el usuario tiene foto de perfil o no.
    if($r->num_rows==0){
        $arr['existeImgPerfil'] = 'no';
    }else{
        $arr['existeImgPerfil'] = 'si';
        $arr['ruta'] = $data['ruta'];
    }

    $stmt->close();
}
echo json_encode($arr);
}

// Función para actualizar la información en tiempo real del
usuario de la sesión actual.
public function actualizarSesionUser($info)
{
    // $info[0] = //nombre
    // $info[1] = //documento
    // $info[2] = //nacionalidad
    // $info[3] = //provincia
    // $info[4] = //localidad
    // $info[5] = //direccion
    // $info[6] = //fecha_nac
    // $info[7] = //telefono
    // $info[8] = //correo
    // $info[9] = //genero
    // $info[10] = //usuario

    $info = $info;

    if(!isset($_SESSION)){
        session_start();
    }

    // Comprobamos que los datos no sean nulos.
    if($info[0]!=NULL){
        $_SESSION['user']['nombre'] = $info[0];
    }
    if($info[1]!=NULL){
        $_SESSION['user']['documento'] = $info[1];
    }
    if($info[2]!=NULL){
        $_SESSION['user']['nacionalidad'] = $info[2];
    }
    if($info[3]!=NULL){

```

```

        $_SESSION['user']['provincia'] = $info[3];
    }
    if($info[4]!=NULL){
        $_SESSION['user']['localidad'] = $info[4];
    }
    if($info[5]!=NULL){
        $_SESSION['user']['direccion'] = $info[5];
    }
    if($info[6]!=NULL){
        $_SESSION['user']['fecha_nac'] = $info[6];
    }
    if($info[7]!=NULL){
        $_SESSION['user']['telefono'] = $info[7];
    }
    if($info[8]!=NULL){
        $_SESSION['user']['correo'] = $info[8];
    }
    if($info[9]!=NULL){
        $_SESSION['user']['genero'] = $info[9];
    }
    if($info[10]!=NULL){
        $_SESSION['user']['usuario'] = $info[10];
    }
}

// Función para eliminar la foto de perfil.
public function eliminarFotoPerfilUser()
{
    $arr['eliminarFotoPerfil'] = false;

    $mysqli = $this->conexion;

    // Traemos el id del usuario de la sesión actual.
    if(!isset($_SESSION)){
        session_start();
        if(isset($_SESSION['user'])){
            $idUser = $_SESSION['user']['id'];
        }
    }

    //Si ya existe una foto, la eliminamos.
    $stmt0 = $mysqli-
>prepare("SELECT ruta FROM imgperfilusers WHERE idUser = ?");
    $existeImg = $stmt0->bind_param('i', $idUser);

    if($existeImg!=FALSE){
        $stmt0->execute();
        $re = $stmt0->get_result();
        $in = $re->fetch_assoc();
    }
}

```

```

        //Si existe un registro, lo borramos de la base de datos y
        //también el archivo de la ubicación de la carpeta.
        if($re->num_rows>=1 && file_exists('../'.$in['ruta'])) ){

            $stmt01 = $mysqli-
>prepare("DELETE FROM imgperfilusers WHERE idUser = $idUser");
            if($stmt01!=FALSE){
                $eliminarRegistro = $stmt01->execute();
                $eliminarRuta = unlink('../'.$in['ruta']);

                if($eliminarRegistro!=FALSE && $eliminarRuta!=FALSE)
E){
                    $arr['eliminarFotoPerfil'] = true;
                }

                if(isset($stmt01)){
                    $stmt01->close();
                }
            }
        }
        if(isset($stmt0)){
            $stmt0->close();
        }

        echo json_encode($arr);
    }

    // Función para reenviar los datos para que otro usuario pueda
    acceder nuevamente al sistema.
    public function reenviarInfoAccesoUser($idUser)
    {
        $idUser = $idUser['idUser'];
        $arr['modPassUser'] = false;
        $mysqli = $this->conexion;

        // Obtenemos el nombre de usuario.
        $sql = "SELECT usuario FROM usuarios WHERE id = ?";
        $stmt = $mysqli->prepare($sql);

        if($stmt!=false){
            $stmt->bind_param('i', $idUser);
            if($stmt->execute()!=false){
                $r = $stmt->get_result();
                foreach($r as $info){
                    $nomUser = $info['usuario'];
                }
            }
        }
    }
}

```

```

        }

    // Generamos una nueva contraseña con numeros
    // aleatorios y reemplazamos la original en la base de datos.
    $pass = rand(999999999, 9999999999);
    $hashPass = password_hash($pass, PASSWORD_DEFAULT, ['cost' =>
10]);

    //Guardamos la contraseña en una variable de session para despues enviarla limpia por correo.
    session_start();
    $_SESSION['pass'] = $pass;

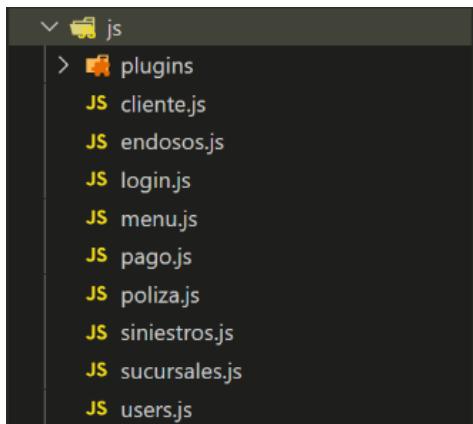
    // Ejecutamos la consulta.
    $sql2 = "UPDATE usuarios SET pass = ? WHERE id = ?";
    $stmt2 = $mysqli->prepare($sql2);
    if($stmt2!=false){
        $stmt2->bind_param('si', $hashPass, $idUser);
        if($stmt2->execute()!=false){
            $arr['modPassUser'] = true;
            $arr['nombreUser'] = $nomUser;
        }
    }

    if($stmt){
        $stmt->close();
    }
    if($stmt2){
        $stmt2->close();
    }

    return json_encode($arr);
}
}

```

4.3.3 Carpeta “js”



Aquí se almacenan todos los ficheros con las funcionalidades del frontend, usando javascript con jquery, como así varias librerías.

En esta carpeta, se encuentran todos los ficheros que contienen las clases principales del sistema, las cuales funcionan como vista en el paradigma MVC, y son las encargadas de conectar con el modelo intermediario y realizar todas las acciones del lado del cliente.

➔ 4.3.3.1 fichero “cliente.js”

Este fichero contiene todas las funciones que realizan las acciones relacionadas al frontend o vista de los clientes/tomadores de pólizas en el sistema.

```
// Hacemos la lectura general del DOM con jQuery.
jQuery(document).ready(function($) {

    // Llamamos a varias funciones al cargar el DOM, o al realizar alguna
    // acción.
    funcionalidadesTomador();
    listadoTomadores();
    $("#provincia").on('change', function(){
        mostrarCpC($("#provincia").val(), 0);
    });

    $("#cp").on('change', function(){
        mostrarLocalidadesC($("#cp").val())
    });

    $('#select_cliente').on('change',function(){
        mostrarDatosCliente($('#select_cliente').val());
    });

    $('#grabarCliente').on('click', function(){
        validarCliente(1);
    });

    $('#grabarClienteSec').on('click', function(){
        validarCliente(2);
    });

    $(document).off('click', '.editarDatosTom');
    $(document).on('click', '.editarDatosTom', function(){
        $('#modificarDatosTomadorModal').modal('open');
        var id = $(this).attr('value');
        mostrarDatosTomador(id);
    });

    $(document).off('click', '.bajaTom');
    $(document).on('click', '.bajaTom', function(){
        var id = $(this).attr('value');
        eliminarTomador(id);
    });

    $(document).off('click', '.verTom');
    $(document).on('click', '.verTom', function(){
        var id = $(this).attr('value');
        consultarTomador(id);
    });
});
```

```

// Función para validar los datos del formulario de cliente/tomador.
function validarCliente(parametro, datos)
{
    var valNombre = /^[A-Záéíóñáéíóñ]{0}?[A-Záéíóñáéíóñ\']+[\'\s]+([A-Záéíóñáéíóñ]{0}?[A-Záéíóñáéíóñ\'])+[\s]?([A-Záéíóñáéíóñ]{0}?[A-Záéíóñáéíóñ\'])?$/;
    var valFecha = /^\d{1,2}/\d{1,2}/\d{2,4}$/;
    var email = /^[a-zA-Z0-9._%+-]{1,64}@[a-zA-Z0-9.-]{1,63}\.{1,125}[a-zA-Z]{2,63}$/i;

    $.validator.addMethod("validarNombre", function(value, element) {
        return this.optional(element) || valNombre.test(value) ;
    });

    $.validator.addMethod("validarTipoPer", function(value, element) {
        return this.optional(element) || value == 'fisica' || value == 'juridica';
    });

    $.validator.addMethod("validarFecha", function(value, element) {
        return this.optional(element) || valFecha.test(value) ;
    });

    $.validator.addMethod("validarGenero", function(value, element) {
        return this.optional(element) || value == 'masculino' || value == 'femenino' || value == 'indeterminado';
    });

    $.validator.addMethod('email', function(value, element){
        return this.optional(element) || email.test(value);
    });

    // Dependiendo el parámetro que llegue, se mostrara el mensaje en un div determinado.

    if(parametro==1){
        claseUl = '.mensaje.mensajeModal';
    }else if(parametro==2){
        claseUl = '.mensaje.ul2';
    }else if(parametro==3){
        claseUl = '.mensaje.mensajeModal';
    }

    // Usamos la librería jquery validate para validar los campos.

    $("#nuevoCliente").validate({
        rules: {
            "nombre": {required:true, validarNombre:true},
            "documento": {required:true, digits:true, rangelength: [6, 8]},
            "persona": {required:true, validarTipoPer:true},
            "nacionalidad": {required:true, lettersonly:true},
            "fech_nac": {required:true, validarFecha:true},
            "provincia": {required:true, rangelength: [1, 24]},
            "cp": {required:true, rangelength: [0, 9421]},
            "localidad": {required:true, rangelength: [1, 22963]},
            "telefono": {required:true, digits:true, min:6},
        }
    });
}

```

```

        "calle": {required:true},
        "genero": {required:true, validarGenero:true},
        "correo": {email:true}
    },
    messages: {
        "nombre": {required:"Debe ingresar el Nombre", validarNombre:"El Nombre ingresado es incorrecto"},
        "documento": {required:"Debe ingresar el Documento", rangelength:["El Documento ingresado está incompleto", digits:"El Documento ingresado es incorrecto"]},
        "persona": {required:"Debe ingresar el tipo de Persona", validarTipoPer:"El tipo de persona seleccionado es incorrecto"},
        "nacionalidad": {required:"Debe ingresar la Nacionalidad", lettersonly:"El Pais ingresado es incorrecto"},
        "fech_nac": {required:"Debe ingresar la fecha de Nacimiento", validarFecha:"La Fecha de Nacimiento/Orig ingresada es incorrecta"},
        "provincia": {required:"Debe seleccionar una Provincia", rangelength:"La Provincia es incorrecta"},
        "cp": {required:"Debe seleccionar un Codigo Postal", rangelength:"El Codigo Postal es incorrecto"},
        "localidad": {required:"Debe seleccionar una Localidad", rangelength:"La Localidad es incorrecta"},
        "telefono": {required:"Debe ingresar un numero de Telefono", digits:"El Telefono ingresado es incorrecto", min:"El Telefono ingresado es incorrecto"},
        "calle": {required:"Debe ingresar una calle"},
        "genero": {required:"Debe seleccionar el Genero", validarGenero:"El Genero seleccionado es incorrecto"},
        "correo": {email:"El E-mail ingresado es incorrecto"}
    },

    highlight: function(element) {
        if($(element).closest('.campo').hasClass('select2')){
            $(element).siblings('span').find('.selection').find('.select2-selection').addClass('claseError');
        }else{
            $(element).closest('.campo').addClass('claseError');
        }
    },

    unhighlight: function(element) {
        if($(element).closest('.campo').hasClass('select2')){
            $(element).siblings('span').find('.selection').find('.select2-selection').removeClass('claseError');
        }else{
            $(element).closest('.campo').removeClass('claseError');
        }
    },
    errorElement : 'li',
    errorLabelContainer: claseUl,
    submitHandler: function(form){

```

```

        switch (parametro)
        {
            case 1:
                //Le pasamos el parametro 'seleccionar' a la función de
                grabar, para que una vez grabado el cliente/tomador, se seleccione en el
                select.
                grabarCliente('seleccionar');
                break;

            case 2:
                grabarCliente();
                break;

            case 3:
                modificarDatosTomador(datos);
                break;
        }
    });
}

// Función para mostrar los datos del tomador.
function mostrarDatosCliente(idtomador)
{
    jQuery.ajax({
        type:"post",
        data:{param:40000, idtomador:idtomador},
        url: "scripts/aseguradora.php",
        dataType:'json',
        success: function(r){
            if(r.data['success']==true){

                $('#modalcliente').modal('close');

                $("#info").css('display', 'block');

                $("#nombreDatos").val(r.data['nombre']);
                $("#docDatos").val(r.data['documento']);
                $("#tipDatos").val(r.data['persona']);
                $("#nacDatos").val(r.data['nacionalidad']);
                $("#proDatos").val(r.data['provincia']);
                $("#locDatos").val(r.data['localidad']);
                $("#cpDatos").val(r.data['cp']);
                $("#calleDatos").val(r.data['calle']);
                $("#genDatos").val(r.data['genero']);
                $("#telDatos").val(r.data['telefono']);
                $("#corrDatos").val(r.data['correo']);
                $("#feNacDatos").val(r.data['fecha_nac']);
            }else{
                alertify.error('Ocurrio un problema al mostrar la informacion');
            };
            //$("#info").css('display', 'none');
            $('#modalcliente').modal('close');
        }
    });
}

```

```

        }
    });
}

// Función para grabar un nuevo cliente.
function grabarCliente(param)
{
    var nom = $('#nombre').val();
    var docu = $('#documento').val();
    var per = $('#persona').val();
    var nac = $('#nacionalidad').val();
    var pro = $('#provincia').val();
    var loc = $('#localidad').val();
    var cp = $('#cp').val();
    var call = $('#calle').val();
    var gen = $('#genero').val();
    var tel = $('#telefono').val();
    var corr = $('#correo').val();
    var fechNac = $('#fech_nac').val();

    var datos = [nom, docu, per, nac, pro, loc, cp, call, gen, tel, corr, fechNac];

    jQuery.ajax({
        url: "scripts/aseguradora.php",
        type: 'post',
        dataType: 'json',
        data: {param:40001, d:datos},
        success: function(r){
            if(r.success){

                // Limpiamos los campos.
                $("#info").css('display', 'block');

                $('#nombre, #documento, #nacionalidad, #calle, #telefono, #correo, #fech_nac').val('');
                $('#persona').val('0');
                $('#provincia').val('nulo').trigger('change');
                $('#cp, #localidad').empty();
                $('#cp').append('<option value="can" disabled selected>Seleccione Provincia</option>');
                $('#localidad').append('<option value="nulo" disabled selected>Seleccione Código Postal</option>');
                $('#genero').val('0');

                $('#mensaje').html("");
                alertify.success('Tomador Registrado con Exito');

            }else{

                $('#nombre, #documento, #nacionalidad, #calle, #telefono, #correo, #fech_nac').val('');
                $('#persona').val('0');
                $('#provincia').val('nulo').trigger('change');
            }
        }
    });
}

```

```

        $("#cp, #localidad").empty();
        $("#cp").append('<option value="can" disabled selected>Seleccione Provincia</option>');
        $("#localidad").append('<option value="nulo" disabled selected>Seleccione Código Postal</option>');
        $("#genero").val('0');

        $("#mensaje").html("");
        alertify.error('Ocurrió un problema al registrar el tomador')
    ;
}

// Si llega el parámetro "Seleccionar", lo hacemos a través de la función.
if(param=='seleccionar'){
    listadoTomadores(r.idTomador);
    mostrarDatosCliente(r.idTomador);
}

$('#modalcliente').modal('close');
}
});

}

// Función para mostrar las localidades en el select, por cada llamada, limpiamos el listado y más abajo recorremos los datos traídos de la BBDD y los agregamos al select.
function mostrarLocalidadesC(cp, loc)
{
    jQuery("#localidad").empty();
    jQuery("#localidad").append('<option value="" disabled selected>Seleccione</option>');
    $.ajax({
        type:"post",
        data:{param:80000, cp:cp},
        url: "scripts/aseguradora.php",
        dataType:'json',
        success: function(r){
            if(r!=null){
                $.each(r.data, function(i,val){
                    if(loc==val.localidadId){
                        $("#localidad").append('<option value="' + val.localidadId + '" selected>' + val.localidadNombre + '</option>');
                    }else{
                        $("#localidad").append('<option value="' + val.localidadId + '">' + val.localidadNombre + '</option>');
                    }
                });
            }else{
                alertify.error('A ocurrido un error al mostrar las localidades');
            }
        }
    });
}

```

```

}

// Función para mostrar los códigos postales en el select, por cada llamada,
limpiamos el listado y más abajo recorremos los datos traídos de la BBDD y
los agregamos al select.
function mostrarCpC(idPro, cp)
{
    jQuery("#cp").empty();
    jQuery("#cp").append('<option value="nulo" disabled selected>Seleccione</option>');
    $.ajax({
        type: 'post',
        dataType: 'json',
        data: {param:80002, id:idPro},
        url: 'scripts/aseguradora.php',
        success: function(r){
            if(r!=null){
                $.each(r.data, function(i, val){
                    if(cp==val.cpNum){
                        $("#cp").append('<option value="' + val.cpNum + '" selected>' + val.cpNum + '</option>');
                    }else{
                        $("#cp").append('<option value="' + val.cpNum + '">' + val.cpNum + '</option>');
                    }
                });
            }else{
                alertify.error('Ha ocurrido un error al mostrar los códigos p
ostales');
            }
        }
    });
}

// Función para mostrar la información de un cliente/tomador, a través de una
petición ajax.
function mostrarDatosTomador(id)
{
    $.ajax({
        url: 'scripts/aseguradora.php',
        dataType: 'json',
        type: 'post',
        data: {param:40002, id:id},
        success: function(r){
            if(r.success){

                $(".tituloFormulario.tomador").html('datos del tomador ' + r.nroTom);

                $("#nroTom").html(r.nroTom);

                persona = r.persona.toLowerCase();
                genero = r.genero.toLowerCase();

                $("#nombre").val(r.nombre);
            }
        }
    });
}

```

```

        $("#documento").val(r.documento);
        $("#persona").val(persona);
        $("#nacionalidad").val(r.nacionalidad);
        $("#provincia").val(r.provinciaId).trigger('change');
        mostrarCpC(r.provincialId, r.cp);
        mostrarLocalidadesC(r.cp, r.localidadId);
        $("#calle").val(r.calle);
        $("#genero").val(genero);
        $("#telefono").val(r.telefono);
        $("#correo").val(r.correo);
        $("#fech_nac").val(r.fecha_nac);

        // Enviamos el parámetro 3 a la función de validar.
        $(document).off('click', '#grabarClienteAdmin');
        $(document).on('click', '#grabarClienteAdmin', function(){
            validarCliente(3, id);
        });

    }else{
        alertify.error('Hubo un error al mostrar la informacion del tomador');
    }
});

// Función para modificar los datos del cliente/tomador.
function modificarDatosTomador(id)
{
    no = $("#nombre").val();
    dd = $("#documento").val();
    pe = $("#persona").val();
    na = $("#nacionalidad").val();
    fe = $("#fech_nac").val();
    pr = $("#provincia").val();
    cp = $("#cp").val();
    lo = $("#localidad").val();
    te = $("#telefono").val();
    ca = $("#calle").val();
    ge = $("#genero").val();
    co = $("#correo").val();

    nuevosDatos = [no, dd, pe, na, fe, pr, cp, lo, te, ca, ge, co, id];

    $.ajax({
        url: 'scripts/aseguradora.php',
        type: 'post',
        dataType: 'json',
        data: {param:40003, datos:nuevosDatos},
        success: function(r){
            if(r.success){

                // Accion 1 = Tomador modificado.
                if(r.accion == 1){


```

```

        alertify.success('Datos del tomador modificados con exito');
    });

    $("#modificarDatosTomadorModal").modal('close');
    tablaModificarDatosTomadores.ajax.reload();

    // Accion 2 = Se ingresaron los mismos datos.
    }else if(r.accion == 2){

        alertify.notify('No se realizaron modificaciones');
        $("#modificarDatosTomadorModal").modal('close');
        tablaModificarDatosTomadores.ajax.reload();

        //Sino = Problema en el servidor.
    }else{

        alertify.error('Ocurrio un problema al modificar los datos del tomador');
        $("#modificarDatosTomadorModal").modal('close');
        tablaModificarDatosTomadores.ajax.reload();

    }
}else{
    alertify.error('Ocurrio un problema al realizar la peticion')
;
    $("#modificarDatosTomadorModal").modal('close');
    tablaModificarDatosTomadores.ajax.reload();
}
}

});

}

// Función para eliminar un cliente/tomador.
function eliminarTomador(id)
{
    // Preguntamos a través de un mensaje de alerta.
    alertify.confirm('Eliminar cliente', '¿Dar de baja el cliente? Se elimina por completo del sistema, y sus datos no podrán recuperarse.',
        function(){

            $.ajax({
                url: 'scripts/aseguradora.php',
                dataType: 'json',
                type: 'post',
                data: {param: 40004, id:id},
                success: function(r){
                    if(r.success){

                        // Acción 1 = Cliente eliminado.
                        if(r.accion == 1)
                        {
                            alertify.success('Cliente dado de baja con exito');
                            tablaModificarDatosTomadores.ajax.reload();
                        }
                    }
                }
            });
        }
    );
}

```

```

        // Acción 2 = El cliente es tomador de una o más pólizas.
    }else if(r.accion == 2)
    {
        if(r.cantPol == 1){
            texto = 'El cliente no puede darse de baja ya que es
tomador de una poliza';
        }else{
            texto = 'El cliente no puede darse de baja ya que es
tomador de '+r.cantPol+' pólizas';
        }
        alertify.alert('Eliminar cliente', texto).set(configura
cionAlert('ok'));
        tablaModificarDatosTomadores.ajax.reload();

    }else
    {
        alertify.error('Ocurrió un problema al dar de baja el
cliente');
        tablaModificarDatosTomadores.ajax.reload();
    }
    }else
    {
        alertify.error('Ocurrió un problema al dar de baja el cli
ente');
        tablaModificarDatosTomadores.ajax.reload();
    }
}
});

},
function(){}
).set(configuracionAlert('okCancel','Si')));

}

// Función para consultar o ver la información de un tomador.
function consultarTomador(id)
{
    $("#consultarTomModal").modal('open');
    $.ajax({
        url: 'scripts/aseguradora.php',
        type: 'post',
        dataType: 'json',
        data: {param: 40002, id:id},
        success: function(r){
            if(r.success){
                $("#nroTom").html('#'+r.nroTom);
                $(".infoTom").html(r.nombre+' '+r.documento);
                $(".nomT").html(r.nombre);
                $(".docT").html(r.documento);
                $(".perT").html(r.persona);
                $(".nacioT").html(r.nacionalidad);
                $(".proT").html(r.provinciaNombre);
                $(".locT").html(r.localidadNombre);
            }
        }
    });
}

```

```

        $(".cpT").html(r.cp);
        $(".callT").html(r.calle);
        $(".genT").html(r.genero);
        $(".telT").html(r.telefono);
        $(".corrT").html(r.correo);
        $(".nacimT").html(r.fecha_nac);

    }else{
        alertify.error('Ocurrio un problema al mostrar la informacion
del tomador');
    }
});
}

// Función para listar todos los tomadores en un select.
function listadoTomadores(tomadorSelect){
$.ajax({
    type: 'post',
    dataType: 'json',
    data: {param:40005},
    url: 'scripts/aseguradora.php',
    success: function(r){
        if(r!=null){
            jQuery("#select_cliente").empty();
            jQuery("#select_cliente").append('<option value="0" disabled
selected>Seleccione</option>');
            $.each(r, function(i, val){
                if(tomadorSelect==val.id){
                    $("#select_cliente").append('<option value="'+val.id+
'" selected>' +val.nombre+ ' - '+val.doc+'</option>');
                }else{
                    $("#select_cliente").append('<option value="'+val.id+
'">' +val.nombre+ ' - '+val.doc+'</option>');
                }
            });
        }else{
            alertify.error('Ha ocurrido un error al mostrar el listado de
tomadores');
        }
    }
});
}

// Función para generar la tabla con el listado de tomadores con el plugin
DataTables.
function tablaModificarDatosTomadores()
{
    tablaModificarDatosTomadores = $('#modificarDatosTomadorTabla').DataTable
({
    "language": {
        "zeroRecords": "No se encontraron tomadores",
        "sSearch": "Buscar",
    },
}

```

```

    "ajax": "scripts/tablas/tablaModificarDatosTomadores.php",
    "scrollY": 370,
    "scrollCollapse": true,
    "dom": 'ft',
    "order": [],
    "lengthMenu": [[5, 10], [5, 10]],
    "aoColumnDefs": [
        { 'bSortable': false, 'aTargets': [4] }
    ],

    "columns": [
        {data:"num_tom", title: "Nro Tom",width:""}, 
        {data:"nombre", title: "Nombre",width:""}, 
        {data:"provincia", title: "Provincia",width:""}, 
        {data:"localidad", title: "Localidad",width:""}, 
        {data:"acciones", title: "Acciones",width:"20%"} 
    ]
});

return tablaModificarDatosTomadores;
}

// Función donde se realizan funcionalidades extras en los formularios y
// plugins utilizados para clientes/tomadores.
function funcionalidadesTomador()
{
    $('input, select').addClass('browser-default');

    //Establecemos los tabs a todos los campos con la clase tabs (requerido por
    //materialize para funcionar)
    $('.tabs').tabs();

    //Declaramos como modal todo campo con la clase .modal
    $('.modal').modal({
        dismissible: false
    });

    //Aplicamos la libreria select2 a los campos requeridos
    $('#provincia, #localidad, #cp').select2({width: "100%"});

    //Agregamos el texto a los campos cada vez que recargamos las funciones
    $("#cp").append('<option value="can" disabled selected>Seleccione Provincia</option>');
    $("#localidad").append('<option value="nulo" disabled selected>Seleccione Código Postal</option>');

    //Desabilitamos los campos donde se muestra la informacion del tomador en
    //emitir poliza.
    $("#nombreDatos,#docDatos,#tipDatos,#nacDatos,#proDatos,#locDatos,#cpDatos,#calleDatos,#genDatos,#telDatos,#corrDatos,#feNacDatos").attr('disabled', 'disabled');

    //Se aplica el complemento Datepicker al input llamado "fech_nac" en registro de tomador.
}

```

➔ 4.3.3.2 fichero “endosos.js”

Este fichero contiene todas las funciones que realizan las acciones relacionadas al frontend o vista de los endosos de pólizas en el sistema.

```
// Leemos el DOM a través de jQuery.
$(document).ready(function() {

    // Llamamos a varias funciones directamente o a través de una acción.
    funcionalidadesEndoso();

    // Mostramos un mensaje antes de cancelar el proceso de emisión de endoso.
    $('#cancelarCambiosEndoso, #cancelarCambiosEndosoV, #cancelarCambiosEndosoT, #cancelarCambiosEndosoC').on('click', function(){
        alertify.confirm('Proceso de Endoso', '¿Cancelar proceso de emision de endoso?',
            function(){
                $('#modalSeleccionarEnd, #modalEndosoVehiculo, #modalEndosoTomador, #modalEndosoCobertura').modal('close');
                $("#formularioEndosoV, #formularioEndosoT, formularioEndosoC").validate().destroy();
                tablaGenerarEndoso.ajax.reload();
            },
            function(){}
        ).set(configuracionAlert('okCancel','Si'));
    });

    // Al clickear, abrimos el modal y llamamos a la función pasándole el parámetro correspondiente dependiendo el tipo de endoso seleccionado.
    $(document).off('click', '.editarPol');
    $(document).on('click', '.editarPol', function(){

        $("#continuarModalTipoEnd").attr('disabled', 'disabled');
        $('#selectTipoEndoso').val(0);
        $('#modalSeleccionarEnd').modal('open');

        var id = $(this).attr('value');

        $("#continuarModalTipoEnd, #cancelarModalTipoEnd").off('click');
        $("#continuarModalTipoEnd").on('click', function(){

            if($('#selectTipoEndoso').val()==1){
                traerDatosPoliza(1, id);
                $('#modalEndosoVehiculo').modal('open');
            }else if($('#selectTipoEndoso').val()==2){
                traerDatosPoliza(2, id);
                $('#modalEndosoVehiculo').modal('open');
            }else if($('#selectTipoEndoso').val()==3){
                traerDatosPoliza(3, id);
                $('#modalEndosoVehiculo').modal('open');
            }else if($('#selectTipoEndoso').val()==4){
                traerDatosPoliza(4, id);
                $('#modalEndosoVehiculo').modal('open');
            }
        });
    });
});
```

```

        traerDatosPoliza(4, id);
        $('#modalEndosoTomador').modal('open');
    }else if($('#selectTipoEndoso').val()==5){
        traerDatosPoliza(5, id);
        $('#modalEndosoCobertura').modal('open');
    }
});

$("#cancelarModalTipoEnd").on('click', function(){
    $('#modalSeleccionarEnd').modal('close');
});

$(document).off('click', '.verEnd');
$(document).on('click', '.verEnd', function(){
    $('#modalListadoDeEndososPorPoliza').modal('open');
    var id = $(this).attr('value');
    mostrarEndososPorPoliza(id);
});

$(document).off('click', '.editarDescripcion');
$(document).on('click', '.editarDescripcion', function(){
    var idEnd = $(this).attr('value');
    modificarDescripcion(idEnd);
});
}

// Función para traer los datos de la poliza al realizar un endoso.
function traerDatosPoliza(paramValid, idPoliza)
{
$.ajax({
    url: 'scripts/aseguradora.php',
    type: 'post',
    dataType: 'json',
    data: {param:10007, id:idPoliza},
    success: function(r){
        if(r.data['success']){
            $(".numeroPol").html(r.data['nro']);

            if(r.data['gps']=='Si'){
                $('#GPS').prop('checked', true);
            }else{
                $('#GPS').prop('checked', false);
            }

            if(r.data['0km']=='Si'){
                $('#ceroKm').prop('checked', true);
            }else{
                $('#ceroKm').prop('checked', false);
            }

            if(r.data['eGas']=='Si'){
                $('#eGas').prop('checked', true);
            }
        }
    }
});
}

```

```

}else{
    $('#eGas').prop('checked', false);
}

if(r.data['pasajeros']==''){
    pasajeros = '';
}else if(r.data['pasajeros']!=''){
    pasajeros = r.data['pasajeros'];
}
if(r.data['asientos']==''){
    asientos = '';
}else if(r.data['asientos']!=''){
    asientos = r.data['asientos'];
}

// Completamos los campos con los datos, y en el caso de los selects, llamamos a las funciones para que se seleccione la opción correcta.
$('#suma_asegurada').val(r.data['suma_asegurada']);
$('#select_marca').val(r.data['marca_id']).trigger('change');
llenarModelo(r.data['marca_id'], r.data['modelo_id']);
$('#select_marca').on('change', function(){
    llenarModelo($('#select_marca').val(), 0);
});
$('#patente').val(r.data['patente']);
$('#motor').val(r.data['motor']);
$('#nroChasis').val(r.data['nroChasis']);
$('#nroMotor').val(r.data['nroMotor']);
$('#select_clase').val(r.data['clase']);
$('#selectTipoUso').val(r.data['uso']);
$('#ceroKm').val();
$('#cant_kilometro').val(r.data['Kilometros']);
$('#codigoPostal').val(r.data['codigoPostal']).trigger('change');
mostrarLocalidades(r.data['codigoPostal'], r.data['localidad']);
$("#codigoPostal").on('change', function(){
    mostrarLocalidades($("#codigoPostal").val(), 0);
});
$('#coberturaAd').val(r.data['coberturaAd']).trigger('change');
$('#combustible').val(r.data['combustible']);
$('#eGas').val();
$('#anio').val(r.data['anio']);
$('#pasajeros').val(pasajeros);
$('#asientos').val(asientos);
$('#color').val(r.data['color']);

$('#select_cliente').val(r.data['clienteid']).trigger('change');
$('#select_cobertura').val(r.data['coberturaid']).trigger('change');

r.data['idPoliza'] = idPoliza;
var datos = r.data;

```

```

        // Enviamos los datos a validar.
        validarEndoso(paramValid, datos);

    }else{
        alertify.error('Ha ocurrido un problema al mostrar la informacion');
    }
}
});

// Funcion para validar los datos ingresados en el formulario de endoso.
function validarEndoso(parametro, info)
{
    // Dependiendo el parámetro que llegue, se validaran los campos correspondientes.
    if(parametro == 1 || parametro == 2 || parametro ==3){

        sumaAsegurada = /^[1-9]{1}[0-9]{4,7}$|[1-9]{1}[0-9]{1}\.[0-9]{3}$|[1-9]{1}[0-9]{2}\.[0-9]{3}$|[1-9]{1}\.[0-9]{3}\.[0-9]{3}$|[1-9]{1}[0-9]{1}\.[0-9]{3}\.[0-9]{3}$/;
        $.validator.addMethod('sumaAsegurada', function(value, element){
            return this.optional(element) || sumaAsegurada.test(value);
        });

        patente = /^[a-zA-Z]{3}[0-9]{3}$|[a-zA-Z]{2}[0-9]{3}[a-zA-Z]{2}$/;
        $.validator.addMethod("patente", function(value, element) {
            value.replace(/ /g, "");
            return this.optional(element) || patente.test(value);
        });

        nroChasis = /^[a-zA-Z0-9]{18}$/;
        $.validator.addMethod("nroChasis", function(value, element) {
            return this.optional(element) || nroChasis.test(value);
        });

        nroMotor = /^[a-zA-Z]{2}[0-9]{3}[a-zA-Z]{2}$/;
        $.validator.addMethod("nroMotor", function(value, element) {
            return this.optional(element) || nroMotor.test(value);
        });

        cantKm = /^[1-9]{1}[0-9]{1,6}$|[1-9]{1}[0-9]{1}\.[0-9]{3}$|[1-9]{1}[0-9]{2}\.[0-9]{3}$|[1-9]{1}\.[0-9]{3}\.[0-9]{3}$|/[^1-9]{1,7}$|[1-9]{1}\.[1-9]{3}$|[1-9]{1}[0-9]{1}\.[1-9]{3}$|[1-9]{1}[0-9]{2}\.[1-9]{3}$|[1-9]{1}\.[0-9]{3}\.[0-9]{3}$/;
        $.validator.addMethod('cantKm', function(value, element){
            return this.optional(element) || cantKm.test(value);
        });

        anio = /^[1]{1}[9]{1}[5-9]{1}[0-9]{1}$|[2]{1}[0]{1}[0]{1}[0]{1}[0-9]{1}$|[2]{1}[0]{1}[1]{1}[0-9]{1}$|[2]{1}[0]{1}[2]{1}[0-1]{1}$/;
        $.validator.addMethod('anio', function(value, element){
            return this.optional(element) || anio.test(value);
        });
    }
}

```

```

        $.validator.addMethod("validarComb", function(value, element) {
            return this.optional(element) || value=="Nafta" || value=="GasOil" || value=="Energia Electrica";
        });

        $("#formularioEndosoV").validate({
            rules:{
                "suma_asegurada": {required: true, sumaAsegurada: true},
                "selec_marca": {required: true, rangelength:[1,79]},
                "select_modelo1": {required: true, rangelength:[1,1005]},
                "patente": {required: true, patente: true},
                "motor": {required: true},
                "nroChasis": {required: true, nroChasis: true},
                "nroMotor": {required: true, nroMotor: true},
                "select_clase": {required: true, rangelength:[1,14], number: true},
                "selectTipoUso": {required: true, rangelength:[1,25], number: true},
                "cant_kilometro": {required: true, cantKm:true},
                "codigoPostal": {required: true, rangelength:[1,22963], number: true},
                "vehiculoLocalidad": {required: true, rangelength:[1,22963], number: true},
                "coberturaAd": {required: true, rangelength:[1,6], number: true},
                "combustible": {required: true, validarComb:true},
                "anio": {required: true, anio: true},
                "pasajeros": {number: true},
                "asientos": {number: true},
                "color": {lettersonly: true}
            },
            messages:{
                "suma_asegurada": {required: "Debe ingresar la Suma Asegurada", sumaAsegurada: "La Suma Asegurada es incorrecta"},
                "selec_marca": {required: "Debe seleccionar una Marca", rangeLength:"La Marca seleccionada es incorrecta"},
                "select_modelo1": {required: "Debe seleccionar un Modelo", rangeLength:"El Modelo seleccionado es incorrecto"},
                "patente": {required: "Debe ingresar la Patente", patente: "La Patente ingresada es incorrecta"},
                "motor": {required: "Debe ingresar el Motor"},
                "nroChasis": {required: "Debe ingresar el Numero de Chasis", nroChasis: "El numero de Chasis es incorrecto"},
                "nroMotor": {required: "Debe ingresar el Numero de Motor", nroMotor: "El Numero de Motor es incorrecto"},
                "select_clase": {required: "Debe seleccionar la Clase", rangeLength:"La Clase seleccionada es incorrecta", number:"La Clase seleccionada es incorrecta 2"},
                "selectTipoUso": {required: "Debe seleccionar el Uso", rangeLength:"El Uso seleccionado es incorrecto", number:"El Uso seleccionado es incorrecto 2"}
            }
        });
    
```

```

        "cant_kilometro": {required: "Debe ingresar la Cantidad de Kilometros", cantKm:"El valor de Cantidad de Kilometros es incorrecto"},  

        "codigoPostal": {required: "Debe seleccionar un Código Postal", rangelength:"El Código Postal seleccionado es incorrecto", number:"El Código Postal seleccionado es incorrecto 2"},  

        "vehiculoLocalidad": {required: "Debe seleccionar la Localidad", rangelength:"La Localidad seleccionada es incorrecta", number:"La Localidad seleccionada es incorrecta 2"},  

        "coberturaAd": {required: "Debe seleccionar una Cobertura Adicional", rangelength:"La Cobertura Adicional seleccionada es incorrecta", number:"La Cobertura Adicional seleccionada es incorrecta 2"},  

        "combustible": {required: "Debe seleccionar un tipo de Combustible", validarComb:"El tipo de Combustible seleccionado es incorrecto"},  

        "anio": {required: "Debe ingresar el Año", anio:"El Año ingresado es incorrecto"},  

        "pasajeros": {number:"El valor ingresado de Pasajeros es incorrecto"},  

        "asientos": {number:"El valor ingresado de Asientos es incorrecto"},  

        "color": {lettersonly:"El valor ingresado de Color es incorrecto"}  

    },  

    highlight: function(element){  

        if($(element).closest('.campo').hasClass('select2')){  

            $(element).siblings('span').find('.selection').find('.select2-selection').addClass('claseError');  

        }else{  

            $(element).closest('.campo').addClass('claseError');  

        }  

    },  

    unhighlight: function(element) {  

        if($(element).closest('.campo').hasClass('select2')){  

            $(element).siblings('span').find('.selection').find('.select2-selection').removeClass('claseError');  

        }else{  

            $(element).closest('.campo').removeClass('claseError');  

        }  

    },  

    errorElement : 'li',  

    errorLabelContainer: '#mensajeEndoso',  

    submitHandler: function(form){  

        procesarEndoso(info)  

    }  

});  

}else if(parametro == 4){  

    $('#select_cliente').change(function(){  

        if ($(this).val()!="")  

        {  

            $(this).valid();  

        }
    });
}

```

```

        });

        $("#formularioEndosoT").validate({
            rules:{
                "select_cliente":{required:true}
            },
            messages:{
                "select_cliente":{required: "Debe seleccionar un Tomador"}
            },
            highlight: function(element){
                $(element).siblings('span').find('.selection').find('.select2-selection').addClass('claseError');
            },
            unhighlight: function(element){
                $(element).siblings('span').find('.selection').find('.select2-selection').removeClass('claseError');
            },
            errorElement: 'li',
            errorLabelContainer: '#mensajeEndoso',

            submitHandler: function(form){
                procesarEndoso(info);
            }
        });

    }else if(parametro == 5){

        $('#select_cobertura').change(function(){
            if ($(this).val()!="")
            {
                $(this).valid();
            }
        });

        $("#formularioEndosoC").validate({
            rules:{'select_cobertura': {required: true} },
            messages:{'select_cobertura': {required: "Debe seleccionar un Tipo de Cobertura"} },
            highlight: function(element){
                $(element).closest('.campo').addClass('claseError');
            },
            unhighlight: function(element){
                $(element).closest('.campo').removeClass('claseError');
            },
            errorElement : 'li',
            errorLabelContainer: '#mensajeEndoso',
            submitHandler: function(form){
                procesarEndoso(info);
            }
        });
    }
}

```

```

// Funcion para recolectar todos los datos del endoso y enviarlos a la
función para grabar.
function procesarEndoso(datos)
{
    /*var fechaEmisEnd      = $('#fechaEm').val();
    var horaEmisEnd       = $('#horaEm').val();*/
    var fechEmPoliza      = datos['P.fecha_emision'];
    var horEmisPol         = datos['P.horaEmis'];
    var iniVigPol          = datos['P.vigencia_inicio'];
    var finVigPol          = datos['P.vigencia_fin'];
    var horaInVig          = datos['P.horaInVig'];

    var sumaAs             = $('#suma_asegurada').val();
    var marca              = $('#select_marca').val();
    var modelo              = $('#select_modelo1').val()
    var patente             = $('#patente').val();
    var motor               = $('#motor').val();
    var nroChasis          = $('#nroChasis').val();
    var nroMotor            = $('#nroMotor').val();
    var clase               = $('#select_clase').val();
    var uso                 = $('#selectTipoUso').val();
    var cantKms             = $('#cant_kilometro').val();
    var cp                  = $('#codigoPostal').val();
    var vehiculoLocalidad   = $('#vehiculoLocalidad').val();
    var cobAd               = $('#coberturaAd').val();
    var combustible          = $('#combustible').val();
    var anio                = $('#anio').val();
    var pasajeros            = $('#pasajeros').val();
    var asientos             = $('#asientos').val();
    var color               = $('#color').val();

    var cliente             = $('#select_cliente').val();
    var cobertura           = $('#select_cobertura').val();

    var idPoliza            = datos['idPoliza'];

    if($("#eGas").is(':checked')){
        var eGas = 'Si';
    }else{
        var eGas = 'No';
    }

    if($("#GPS").is(':checked')){
        var gps = 'Si';
    }else{
        var gps = 'No';
    }

    if($("#ceroKm").is(':checked')){
        var ceroKm = 'Si';
    }else{
        var ceroKm = 'No';
    }
}

```

```

var nuevosDatEnd = [ fechEmPoliza,
                     horEmisPol,
                     iniVigPol,
                     finVigPol,
                     horaInVig,
                     sumaAs,
                     marca,
                     modelo,
                     patente,
                     motor,
                     nroChasis,
                     nroMotor,
                     clase,
                     uso,
                     gps,
                     ceroKm,
                     cantKms,
                     cp,
                     vehiculoLocalidad,
                     cobAd,
                     combustible,
                     eGas,
                     anio,
                     pasajeros,
                     asientos,
                     color,
                     cliente,
                     cobertura,
                     idPoliza ]; // 29 elementos

//var informacionEndoso = [idPoliza, datos['P.nro'], fechaEmisEnd, horaEmisEnd];
modificarDatosPoliza(nuevosDatEnd, idPoliza);
}

// Función para efectuar la modificación de los datos de la poliza.
function modificarDatosPoliza(nuevosDatos, idPoliza)
{
$.ajax({
    url: 'scripts/aseguradora.php',
    type: 'post',
    dataType: 'json',
    data: {param:10008, datos:nuevosDatos},
    success: function(r){
        if(r.success){

            // Si la acción devuelta es = 1: Se efectua el endoso.
            if(r.accion == 1){

                if($('#selectTipoEndoso').val() == 1){
                    paramTipoEnd = 1;
                }else if($('#selectTipoEndoso').val() == 2){
                    paramTipoEnd = 2;
                }
            }
        }
    }
});
}

```

```

        }else if($('#selectTipoEndoso').val()==3){
            paramTipoEnd = 3;
        }else if($('#selectTipoEndoso').val()==4){
            paramTipoEnd = 4;
        }else if($('#selectTipoEndoso').val()==5){
            paramTipoEnd = 5;
        }

        registrarEndoso(idPoliza, paramTipoEnd);

        // Si la acción devuelta es = 2: No se realizan
        modificaciones, ya que los datos son los mismos.
        }else if(r.accion==2){
            alertify.notify('No se realizaron modificaciones en la po
liza');
            $('#modalSeleccionarEnd, #modalEndosoVehiculo, #modalEndo
soTomador, #modalEndosoCobertura').modal('close');

        }else{
            alertify.error('Ocurrio un problema al procesar el endoso
');
            $('#modalSeleccionarEnd, #modalEndosoVehiculo, #modalEndo
soTomador, #modalEndosoCobertura').modal('close');
        }

        tablaGenerarEndoso.ajax.reload();

    }else{
        alertify.error('Ocurrio un problema al procesar el endoso');
        $('#modalSeleccionarEnd, #modalEndosoVehiculo, #modalEndosoTo
mador, #modalEndosoCobertura').modal('close');
    }
}

});

// Función para realizar el registro del endoso.
function registrarEndoso(idPoliza, paramTipoEnd)
{
    var tipoEnd = paramTipoEnd;

    //Usamos la variable tipoEnd para saber el tipo de endoso que vamos a reg
istrar. El valor lo obtenemos por el select o lo pasamos por parametro.

    $.ajax({
        url: 'scripts/aseguradora.php',
        type: 'post',
        dataType: 'json',
        data: {param:20000, tipoEnd:tipoEnd, idPoliza:idPoliza},
        success: function(r){
            if(r.success){

```

```

        $('#modalCambiarEstadoPoliza, #modalSeleccionarEnd, #modalEndosoVehiculo, #modalEndosoTomador, #modalEndosoCobertura').modal('close');

        tablaGenerarEndoso.ajax.reload();

        if(tipoEnd == 1){
            alertify.success('Endoso "Modificacion de Riesgo" registrado con exito');
        }else if(tipoEnd == 2){
            alertify.success('Endoso "Modificacion de Datos Nominales" registrado con exito');
        }else if(tipoEnd == 3){
            alertify.success('Endoso "Modificacion de Datos del Vehiculo" registrado con exito');
        }else if(tipoEnd == 4){
            alertify.success('Endoso "Modificacion de Asegurado" registrado con exito');
        }else if(tipoEnd == 5){
            alertify.success('Endoso "Cambio de Cobertura" registrado con exito');
        }else if(tipoEnd == 6){
            alertify.success('Endoso "Cambio de Estado de Poliza" registrado con exito');
        }

        }else{
            alertify.error('Ha ocurrido un problema al registrar el Endoso');
        }
    });
});

// Función para modificar la descripción de un endoso.
function modificarDescripcion(idEndoso)
{
    $('#modalDescripcionEndoso').modal('open');
    $.ajax({
        url: 'scripts/aseguradora.php',
        type: 'post',
        dataType: 'json',
        data: {param:20002, idEndoso:idEndoso},
        success: function(r){
            if(r.data['success']){
                // Llenamos el campo con la descripción, y enviamos la nueva descripción por parámetro a la función.
                $('#desc').val(r.data['descripcion']);
                $('#numeroEndDesc').html(r.data['endosoNum']);
                actualizarDescripcion(r.data['descripcion'], r.data['idEndoso']);
            }
        }
    });
}

```

```

        alertify.error('Ha ocurrido un problema al mostrar la informa
cion solicitada');
    }
}
});

// Función para grabar la nueva descripción del endoso.
function actualizarDescripcion(desAct, idEndoso)
{
    $('#aceptarDescEnd').off('click');
    $('#aceptarDescEnd').on('click', function(){
        var nuevaDesc = $('#desc').val();
        if(desAct == nuevaDesc){
            alertify.notify('No se realizaron cambios en la descripción del e
ndoso');
            $('#modalDescripcionEndoso').modal('close');
        }else{
            $.ajax({
                url: 'scripts/aseguradora.php',
                type: 'post',
                dataType: 'json',
                data: {param:20003, nuevaDesc:nuevaDesc, idEndoso:idEndoso},
                success: function(r){
                    if(r.success){
                        $('#modalDescripcionEndoso').modal('close');
                        alertify.success('Descripción modificada');
                        tablaListadoEndososPorPoliza.ajax.reload();
                    }else{
                        $('#modalDescripcionEndoso').modal('close');
                        alertify.error('Ha ocurrido un problema al modificar
la descripción del endoso');
                    }
                }
            });
        }
    });
}

// Función utilizada para sumar días a una fecha.
function sumarDias(fecha, dias)
{
    fecha.setDate(fecha.getDate() + dias);

    var di = fecha.getDate();
    var me = fecha.getMonth() + 1;
    var anio = fecha.getFullYear();

    if(di < 10){
        dia = "0"+di;
    }else{
        dia = di;
    }
}

```

```

if(me < 10){
    mes = "0"+me;
}else{
    mes = me;
}

fecha = dia+'/'+mes+'/'+anio;

return fecha;
}

// Función para mostrar el listado de endosos de una poliza.
function mostrarEndososPorPoliza(idPoliza)
{
    $.ajax({
        url: 'scripts/aseguradora.php',
        type: 'post',
        dataType: 'json',
        data: {param:20001, id:idPoliza},
        success: function(r){
            if(r.success){

                // Llenamos los campos.
                $('#numeroPolListado').html(r.numPol);
                $('#nombreTomador').html(r.tom);

                // Usamos el plugin DataTable para generar la tabla con el
                // listado de endosos.
                tablaListadoEndososPorPoliza = $('#mostrarEndososPorPoliza').
                DataTable({
                    "destroy": true,
                    "language": {
                        "zeroRecords": "No se encontraron resultados",
                        "sSearch": "Buscar Endoso",
                    },
                    "scrollY": 370,
                    "bPaginate": false,
                    "dom": 'ft',
                    "order": [[1, "desc"],[2, "desc"]],
                    "lengthMenu": [[5, 10], [5, 10]],
                    "aoColumnDefs": [
                        { 'bSortable': false, 'aTargets': [5] }
                    ],
                    "ajax":{
                        "url": "scripts/tablas/tablaMostrarEndososPorPoliza.p
hp",
                        "method": 'POST',
                        "data":{idPol:idPoliza},
                        "dataSrc": ""
                    },
                    "columns": [
                        {data:"nro", title: "Nro end.", width:"10%"}
                ,

```

```

        {data:"fecha", title: "Registro", width:"10%"}  

        , {data:"hora", title: "Hora", width:"10%"}  

        , {data:"tipo", title: "Tipo", width:"32%"}  

        , {data:"descripcion", title: "Descripcion", width:"35%"}  

        , {data:"btn", title: "", width:"3%"},  

    ]  

});  
  

} else{  

    alertify.error('Ha ocurrido un problema al mostrar la informacion solicitada');  

}  

});  

});  

}  
  

// Función donde generamos una tabla para mostrar el listado de polizas a través del plugin DataTable.  

function tablaGenerarEndoso()  

{
    tablaGenerarEndoso = $('#mostrarPolizasTabla').DataTable({  

        "language": {  

            "zeroRecords": "No se encontraron polizas registradas en la base de datos.",  

            "sSearch": "Buscar poliza",  

        },  

        "ajax": "scripts/tablas/tablaGenerarEndoso.php",  

        "scrollY": 370,  

        "scrollCollapse": true,  

        "dom": 'ft',  

        "order": [[5, "desc"]],  

        "lengthMenu": [[5, 10], [5, 10]],  

        "aoColumnDefs": [  

            { 'bSortable': false, 'aTargets': [6] }  

        ],  

        "columns": [  

            {data:"nro", title: "Poliza Nro",width:""},  

            {data:"tomador", title: "Tomador",width:""},  

            {data:"vehiculo", title: "Vehiculo",width:""},  

            {data:"vigin", title: "Inicio Vig",width:""},  

            {data:"vigfin", title: "Fin Vig",width:""},  

            {data:"estado", title: "Estado",width:""},  

            {data:"boton", title: "",width:""}  

        ]
    });
}  
  

return tablaGenerarEndoso;
}

```

```

// Función donde generamos una tabla para mostrar el listado de endosos a
través del plugin DataTable.
function tablaListadoEndosos()
{
    tablaListadoEndosos = $('#listadoEndososTabla').DataTable({
        "language": {
            "zeroRecords": "No se encontraron endosos registrados",
            "sSearch": "Buscar poliza",
        },
        "ajax": "scripts/tablas/tablaListadoEndosos.php",
        "scrollY": 370,
        "scrollCollapse": true,
        "dom": 'ft',
        "order": [[2, "asc"]],
        "aoColumnDefs": [
            { 'bSortable': false, 'aTargets': [3] }
        ],
        "columns": [
            {data:"nro", title: "Poliza Nro",width:""}, 
            {data:"tomador", title: "Tomador",width:""}, 
            {data:"estado", title: "Estado",width:""}, 
            {data:"boton", title: "",width:""}
        ]
    });
}

return tablaListadoEndosos;
}

// Función donde realizamos funcionalidades extras en todos los formularios y
demás elementos relacionados a los endosos.
function funcionalidadesEndoso()
{
    // Aplicamos la función a todas las clases .tab para que funcione.
    $('.tabs').tabs();
    // Aplicamos la función para que funcione.
    $('#selectTipoEndoso').formSelect();

    //Declaramos como modal todo campo con la clase .modal
    $('.modal').modal({
        dismissible: false
    });

    $('input, select').addClass('browser-default');

    $('#tomadorField .tituloFormulario').remove();
    $('#info input').removeClass('browser-default');

    //Funciones para mostrar botones y formularios correspondientes según el
    tipo de endoso seleccionado

    function actDesCampos(accion){
        todosCampos = `#suma_asegurada, #select_marca, #select_modelo1, #pate
nte, #motor, #nroChasis,

```

```

        #nroMotor, #select_clase, #selectTipoUso, #GPS, #ceroKm, #cant_kilome
tro, #codigoPostal,
        #vehiculoLocalidad, #coberturaAd, #combustible, #eGas, #anio, #pasaje
ros, #asientos, #color`;
    if(accion=='d'){
        $(todosCampos).prop('disabled', true).css('background', '#eee');
    }else if(accion=='a'){
        $(todosCampos).prop('disabled', false).css('background', 'transpa
rent');
    }

}

// Función para limpiar los campos de select.
function limpiarSelects(){
    $("#select_clase, #selectTipoUso, #codigoPostal, #vehiculoLocalidad,
#coberturaAd, #combustible").val(0).trigger('change');
    $("#codigoPostal").val('n').trigger('change');
}

// Mostramos el formulario correcto dependiendo del tipo de endoso que
seleccionemos.
$('#selectTipoEndoso').off('change');
$('#selectTipoEndoso').on('change', function(){

    $("#continuarModalTipoEnd").attr('disabled', false);
    actDesCampos('d'); //Por cada change, desactivamos todos los campos

    if($('#selectTipoEndoso').val()==1{

        $("#tituloDatosVehiculo").html("Modificacion de Riesgo");
        camposActivar = "#select_marca, #select_modelo1";
        $(camposActivar).prop('disabled', false).css('background', 'trans
parent');

        $("#select_modelo1").on('change', function(){
            if(
                $("#suma_asegurada").val()!='' || $("#patente").val()!=''
                || $("#motor").val()!='' || $("#nroChasis").val()!=''
                || $("#nroMotor").val()!='' || $("#cant_kilometro").val()
!='' || $("#anio").val()!=''
            ){
                alertify.confirm('Modificacion de riesgo', '¿Desea conser
var toda la informacion del vehiculo tal cual?', function(){}, function()
{
                camposDesactivar = `#suma_asegurada, #patente, #motor
, #nroChasis, #nroMotor, #select_clase, #selectTipoUso, #GPS, #ceroKm, #cant_k
ilometro, #codigoPostal, #vehiculoLocalidad, #coberturaAd, #combustible, #eGas
, #anio, #pasajeros, #asientos, #color`;
                $(camposDesactivar).prop('disabled', false).css('back
ground', 'transparent').val('');
                limpiarSelects();
                $("#GPS, #ceroKm, #eGas").prop('checked', false);
            }
        })
    }
})
}

```

```

        }).set(configuracionAlert('okCancel', 'Conservar', 'Ingresa
r nueva')));
    }
});

}else if($('#selectTipoEndoso').val()==2){

    $("#tituloDatosVehiculo").html("Modificacion de Datos Nominales")
;

    camposActivar = `#suma_asegurada, #patente, #nroChasis, #nroMotor
`;
    $(camposActivar).prop('disabled', false).css('background', 'trans
parent');

}else if($('#selectTipoEndoso').val()==3){

    $("#tituloDatosVehiculo").html("Modificacion datos del vehiculo")
;

    camposActivar = `#motor, #select_clase, #selectTipoUso, #GPS, #ce
roKm, #cant_kilometro, #codigoPostal,
                    #vehiculoLocalidad, #coberturaAd, #combustible,
#eGas, #anio, #pasajeros, #asientos, #color`;
    $(camposActivar).prop('disabled', false).css('background', 'trans
parent');

}else if($('#selectTipoEndoso').val()==4){

    $("#tituloTomadorEndoso").html("Modificacion de Asegurado");
    $("#nuevoTomador").remove();

}else if($('#selectTipoEndoso').val()==5){

    $("#tituloCoberturaEndoso").html("Cambio de cobertura");
}

}); //Cerramos el modal de listado de endosos al presionar aceptar
$('#cerrarEndososPorPoliza').on('click', function(){
    $('#modalListadoDeEndososPorPoliza').modal('close');
});

$('#cancelarDescEnd').on('click', function(){
    $('#modalDescripcionEndoso').modal('close');
});

//Abilitamos el boton de guardar cambios al realizar una modificacion en
el campo de descripcion de endoso
$('#aceptarDescEnd').prop('disabled', true);

$('#desc').on('change', function(){
    if($('#desc').val() == ''){
        $('#aceptarDescEnd').prop('disabled', true);
    }else{
        $('#aceptarDescEnd').prop('disabled', false);
    }
});
}
}

```

➔ 4.3.3.3 fichero “login.js”

Este fichero contiene todas las funciones que realizan las acciones relacionadas al frontend o vista del inicio o cierre de sesión en el sistema.

```
// Leemos el DOM a través de jQuery.
jQuery(document).ready(function(){

    // Llamamos a la función para listar las sucursales en el select.
    listadoSucursalesSelect();

    $('input, select').addClass('browser-default');

    // Aplicamos estilos dependiendo las acciones que realicemos.
    $("select").on('change', function(){
        $(this).css('color', '#fff');
    });

    $("input, select").on('click', function(){
        $(this).siblings('i').css('color', '#f5f5f5');
    });

    $("input, select").on('focusout', function(){
        $(this).siblings('i').css('color', '#757575');
    });

    // Llamamos a la función para validar el inicio de sesión.
    validarLogin()

    // Salimos del sistema
    $(".logout").off('click');
    $(".logout").on('click', function(){
        logout();
    })
})

// Función para validar los datos ingresados en el formulario de inicio de sesión.
function validarLogin()
{
    $("#formLogin").validate({
        rules: {
            "u": {required: true},
            "p": {required: true},
            "s": {required: true}
        },
        messages: {
            "u": {required: "Debe ingresar un Usuario"},
            "p": {required: "Debe ingresar una Contraseña"},
            "s": {required: "Debe seleccionar una Sucursal"}
        },
    })
}
```

```

        highlight: function (element) {
            $(element).addClass('claseError');
            $(element).siblings('i').addClass('claseErrorIcono');
        },
        unhighlight: function (element){
            $(element).removeClass('claseError');
            $(element).siblings('i').removeClass('claseErrorIcono');
        },
        errorElement: 'li',
        errorLabelContainer: '#errorLogin',
        submitHandler: function(){
            ingresarSistema()
        }
    })
}

// Función para ingresar al sistema.
function ingresarSistema()
{
    // Leemos la información ingresada.
    var info = [$("#u").val(), $("#p").val(), $("#selectSuc").val()];

    // Llenamos el campo con un mensaje.
    $(".mjeLogin").html('La información ingresada no coincide con nuestros registros').slideUp(200);

    $.ajax({
        url: 'scripts/aseguradora.php',
        type: 'POST',
        dataType: 'json',
        data: {param:90000, info:info},
        beforeSend: function(){
            $("#envFormLogin").html('')
        .show().attr('disabled', 'disabled')
            $(".btn-large").css('padding', '3px')
        },
        success: function(r){

            // Función que nos permite realizar el proceso de validación.
            function accion()
            {
                $(".btn-large").css('padding', '9px')
                if(r.existe){ // Si el usuario existe.

                    if(r.passCorr=='si'){ // Si la contraseña es valida.

                        if(r.sucDisp){ // Si la sucursal es valida.

                            if(r.activo=='si'){ // Si el usuario está activo.

                                $("#envFormLogin").html('Logueado').show().at
                                tr('disabled', 'disabled')
                            }
                        }
                    }
                }
            }
        }
    })
}

```

```

        // Redireccionamos a la vista principal.
        location.href = 'indexMaqueta.php'

        // Si el usuario está dado de baja, mostramos un
        // mensaje.
        }else if(r.activo=='no'){

            $(".mjeLogin").html('Usted no tiene permiso p
ara acceder al sistema permanentemente').slideDown(200)
                $("#envFormLogin").html('<span> Reintentar <i
class="fas fa-redo-alt"></i> </span>').show().attr('disabled', false)
                    setTimeout(() => { $(".mjeLogin").slideUp(200
) }, 3000);
            }

            // Si la sucursal no está disponible.
            }else if(r.sucDisp==false){
                $(".mjeLogin").html('La Sucursal no se encuentra
registrada en el sistema').slideDown(200)
                    $("#envFormLogin").html('<span> Reintentar <i cla
ss="fas fa-redo-alt"></i> </span>').show().attr('disabled', false)
                        setTimeout(() => { $(".mjeLogin").slideUp(200) },
3000);
            }

            // Si la contraseña no es correcta.
            }else if(r.passCorr=='no'){
                $(".mjeLogin").html('La contraseña ingresada es incor
recta').slideDown(200)
                    $("#envFormLogin").html('<span> Reintentar <i class="fas
fa-redo-alt"></i> </span>').show().attr('disabled', false)
                        setTimeout(() => { $(".mjeLogin").slideUp(200) }, 300
0);
            }

            // Si el usuario o correo no existe.
            }else if(r.existe==false){
                $(".mjeLogin").html('El Usuario o Correo ingresado no coi
ncide con nuestros registros').slideDown(200)
                    $("#envFormLogin").html('<span> Reintentar <i class="fas
fa-redo-alt"></i> </span>').show().attr('disabled', false)
                        setTimeout(() => { $(".mjeLogin").slideUp(200) }, 3000);
            }

            setTimeout(() => { accion() }, 2000);
        }
    })
}

// Función para cerrar sesión.
function logout(){

    var nombre = $(".nombreUser").html();

    texto = nombre+', ¿Deseas salir del sistema?'

    alertify.confirm('Cerrar sesion', texto,

```

```

function(){

    $.ajax({
        url: 'scripts/aseguradora.php',
        type: 'POST',
        dataType: 'json',
        data: {param:90001},
        success: function(r){
            if(r.success){
                // Redireccionamos a la vista de inicio de sesión.
                location.href = 'index.php';
            }
        }
    });

},
function(){}
.set(configuracionAlert('okCancel','Si')));

}

// Función donde listamos todas las sucursales en un select.
function listadoSucursalesSelect(){
    $("#selectSuc").empty();
    $("#selectSuc").append('<option value="0" selected disabled>Sucursal</option>');
    $.ajax({
        url: 'scripts/aseguradora.php',
        type: 'POST',
        dataType: 'json',
        data: {param:70004},
        success: function(r){
            if(r.success){
                $.each(r.data, function(key, val){
                    $("#selectSuc").append('<option value="'+val.id+'">' +val.nombre+ ' - '+val.iden+ '</option>');
                });
            }
        }
    });
}

```

➔ 4.3.3.4 fichero “menu.js”

Este fichero contiene todas las funciones que realizan las acciones relacionadas al frontend o vista del menú principal del sistema.

```
// Fichero del menu con las funcionalidades de javascript, que la unica funcion que cumple, es al apretar los botones del menú, es poder cargar con el evento load, un fichero de php que hace referencia a las vistas de ese apartado.

jQuery(document).ready(function($){

    $('input, select').addClass('browser-default');

    // Al presionar el boton de configuracion de usuario, desplegamos el menú.
    $(".configuracionUser").on('click', function(){
        $(".menuUser").slideToggle(200);
    })

    $(".contenido, .sidebar").click(function() {
        $(".menuUser").slideUp(200);
    });

    // Funcionamiento del menu
    $('.opMen').off('click');
    $('.opMen').on('click', function(){

        $('.sub-ul-li-a').removeClass('sub-activado'); // Desactivamos el submenú.

        //Cada vez que abrimos y cerramos el menu, agregamos o sacamos una flecha.
        f = $(this).children('a').children('.flecha').children();

        $(this).children('.sub').slideToggle(200, function(){
            if(f.hasClass('fa-angle-right')){
                f.removeClass('fas fa-angle-right');
                f.addClass('fas fa-angle-down');
            }else{
                f.removeClass('fas fa-angle-down');
                f.addClass('fas fa-angle-right');
            }
        });

        $(this).children('a').addClass('active'); //Activamos la opcion del menu

        // Cuando seleccionamos otra opcion de menu, contraemos el resto.
        var Grupo = $(this).parent();
        var posicion = $(this).index();
        Grupo.children().each(function(index){
            if (index != posicion){
                $(this).children(".sub").slideUp(200);
                $(this).children('a').removeClass('active');
            }
        });
    });
});
```

```

        $(this).children('a').children('.flecha').children().removeClass('fas fa-angle-right');
        $(this).children('a').children('.flecha').children().addClass('fas fa-angle-down');
    }
});

})
$('.sub-ul').click(function(e){
    e.stopPropagation();
})

// Si presionamos el boton de inicio
$('#inicio').on('click', function(){
    $("#divTit").css('display', 'block');
    $("#tituloCont").html('Inicio');
    var nombre = $(".nombreUser").html();
    $("#divSubTit").html('Bienvenido '+nombre);
    $('#contenidoC').html('');
})

// Funcionamiento del submenu
$('.sub-ul-li-a').off('click');
$('.sub-ul-li-a').on('click', function(){

    $(this).addClass('sub-activado');

    // Recorremos los elementos para aplicar y remover las clases de activo.
    var grup = $(this).parent().parent();
    var pos = $(this).parent().index();
    grup.children().each(function(index){
        if (index != pos){
            $(this).children().removeClass('sub-activado');
        }
    });

    // Obtenemos el id del botón del menú presionado, y dependiendo del
    // nombre, realizamos una acción.
    var nombreId = $(this).attr('id');
    switch(nombreId)
    {
        case 'nuevaPoliza':
            var titulo = 'emision de poliza';
            var subtitle = '';
            $("#contenidoC").load('vista/poliza/elementosPoliza/formulario_poliza.php', function(){

                //Creamos aca el change de marca y cp para usarlo solo una vez y evitar anidacion de funciones
                $('#select_marca').on('change', function(){
                    llenarModelo($('#select_marca').val(), 0);
                });
                $('#codigoPostal').on('change', function(){

```

```

        mostrarLocalidades($("#codigoPostal").val(), 0);
    });

});

break;

case 'verPoliza':
    var titulo = 'consultar polizas';
    var subtitle = '';
    $('#contenidoC').load('vista/poliza/elementosPoliza/consultar_polizas.php', function(){
        tablaConsultarPoliza();
    });
    break;

case 'renovarPoliza':
    var titulo = 'renovacion de polizas';
    var subtitle = 'Listado de polizas vencidas con su respectiva vigencia finalizada';
    $('#contenidoC').load('vista/poliza/elementosPoliza/renovar_poliza.php', function(){
        tablaRenovarPoliza();
    });
    break;

case 'emitirEndoso':
    var titulo = 'generar endoso';
    var subtitle = '';
    $('#contenidoC').load('vista/endosos/elementosEndoso/endosos.php', function(){
        tablaGenerarEndoso();
    });
    break;

case 'verEndoso':
    var titulo = 'consultar endosos';
    var subtitle = '';
    $('#contenidoC').load('vista/endosos/elementosEndoso/listadoEndosos.php', function(){
        tablaListadoEndosos();
    });
    break;

case 'nuevoSiniestro':
    var titulo = 'emitir nuevo siniestro';
    var subtitle = '';
    $('#contenidoC').load('vista/siniestro/cargarSiniestro.php', function(){
        listarPolizas();
        //Creamos aca el change de provincia con localidades para usarlo solo una vez y evitar anidacion
        $("#provinciaAcc").on('change', function(){
            mostrarLocalidadesSin($("#provinciaAcc").val(), 0);
        });
    });
}

```

```

        });
    });
    break;

    case 'adminSiniestro':
        var titulo = 'administrar siniestros';
        var subtitle = '';
        $('#contenidoC').load('vista/siniestro/elementosSiniestro/administrarSiniestro.php', function(){
            tablaAdministrarSiniestro();
        });
        break;

    case 'verSiniestro':
        var titulo = 'consultar siniestros';
        var subtitle = '';
        $('#contenidoC').load('vista/siniestro/elementosSiniestro/consultarSiniestro.php', function(){
            tablaConsultarSiniestro();
        });
        break;

    case 'nuevoTomador1':
        var titulo = 'registrar nuevo tomador';
        var subtitle = '';
        $('#contenidoC').load('vista/tomador/registrarTomador.php');
        break;

    case 'adminTomador':
        var titulo = 'administrar tomadores';
        var subtitle = '';
        $('#contenidoC').load('vista/tomador/adminTomador.php', function(){
            tablaModificarDatosTomadores();
        });
        break;

    case 'nuevaCuota':
        var titulo = 'establecer cuotas';
        var subtitle = 'Listado de polizas las cuales se han configurado o no los parametros de pago.';
        $('#contenidoC').load('vista/pagos/emisionPago.php', function(){
            tablaCondicionesCuotas();
        });
        break;

    case 'facturarPago':
        var titulo = 'facturas de cuotas';
        var subtitle = 'Listado de pagos registrados';
        $('#contenidoC').load('vista/pagos/facturarPago.php', function(){
            tablaFacturarPago();
        });
        break;

    case 'AdmUsers':

```

```
var titulo = 'Administrar usuarios';
var subtitleo = '';
$('#contenidoC').load('vista/users/administrarUsers.php', function(){
    tablaListadoUsers();
});
break;

case 'AdmSucursales':
    var titulo = 'Administrar sucursales';
    var subtitleo = '';
    $('#contenidoC').load('vista/sucursales/administrarSucursales.php', function(){
        tablaAdminSucursales();
    });
    break;
}

// Agregamos al campo el titulo y subtitleo correspondiente.
$("#divTit").css('display', 'block');
$("#tituloCont").html(titulo);
$("#divSubTit").html(subtitleo);
})

});
```

➔ 4.3.3.5 fichero “ pago.js ”

Este fichero contiene todas las funciones que realizan las acciones relacionadas al frontend o vista de los pagos y cuotas de polizas del sistema.

```
// Leemos el DOM a través de jQuery.
jQuery(document).ready(function($) {

    // Llamamos a varias funciones directamente o a través de acciones.
    funcionalidadesPagos();
    actualizarEstadoCuota();

    $(document).off('click', '.btnAdminPago');
    $(document).on('click', '.btnAdminPago', function(){
        var idPol = $(this).attr('value');
        adminPago(idPol);
    });

    $(document).off('click', '#verCuotas');
    $(document).on('click', '#verCuotas', function(){
        mostrarCuotas();
    });

    $("#cancelarCondPago").off('click');
    $("#cancelarCondPago").on('click', function(){
        $("#primaTotal, #primaMensual, #diaCobrar, #fechaCobrar").val('');
        $("#cantCuotas").val(1);
        condicionesCuotasTabla.ajax.reload();
    });

    $(document).off('click', '.facturarPago');
    $(document).on('click', '.facturarPago', function(){
        var pagoId = $(this).attr('value');
        facturarPago(pagoId);
    });

    $(document).off('click', '.verFactPago');
    $(document).on('click', '.verFactPago', function(){
        var pagoId = $(this).attr('value');
        mostrarListadoRecibos(pagoId);
    });

    $(".btnAtrasListCu").off('click');
    $(".btnAtrasListCu").on('click', function(){
        $('#contenidoC').load('vista/pagos/facturarPago.php', function(){
            tablaFacturarPago();
        });
    });

    $(".btnEnviarListCu").off('click');
    $(".btnEnviarListCu").on('click', function(){
        validarEmail(1, null);
    });
})
```

```

});;

$(".btnDescargarListCu").off('click');
$(".btnDescargarListCu").on('click', function(){
    envDescListRecibos('descargar');
});

$("#actualizarListCuotas").off('click');
$("#actualizarListCuotas").on('click', function(){
    $(this).html(''+'actualiza
r').attr('disabled', 'disabled');
    setTimeout(() => {
        $(this).html('<i class="fas fa-sync-
alt"></i> actualizar').attr('disabled', false);
        actualizarEstadoCuota();
        listadoCuotasFacturaTabla.ajax.reload();
    }, 1000);
});

// Función para generar los datos para el pago.
function generarFormularioPago(param, infoRen)
{
    // Agregamos el botón de grabar poliza.
    $('.tabs').tabs('select','test5');
    boton = '<button type="submit" class="btn btn-
flat green" id="cargar_datos"> grabar poliza <i class="fas fa-
save"></i> </button>';
    $(".btnDivPago").append(boton);

    //Si el parametro es 1, generamos la informacion con los datos pasados po
r parámetro.
    if(param==1){

        var mesesVigPol = $("#tipVig").val();
        var inVigPol = $("#vig_ini").val();
        var sumAs = infoRen[1].replace(/\./g, '');

        $("#divPoliza").css('display', 'none');
        $("#cuotasForm").css('display', 'block');

        $('.atrRenPa').off('click');
        $('.atrRenPa').on('click', function(e){
            e.preventDefault();
            $("#divPoliza").css('display', 'block');
            $("#cuotasForm").css('display', 'none');

            $("#primaTotal, #cantCuotas, #primaMensual").val('');
            $("#diaCobrar, #tipoPag").val(0);

            $("#mensajePago").css('display', 'none');
        });
    }
}

```

```

        $("#" + primaTotal, #cantCuotas, #primaMensual, #diaCobrar, #tipoPag"
).css({
    border: '1px solid #ccc',
    background: 'transparent'
});
})

// Obtenemos el id de la poliza y lo pasamos a la función de
validar.
var idPoliza = infoRen[0];
validarPago(3, idPoliza);

// Si el parametro es = 2, obtenemos los datos de los campos.
}else if(param==2){
    var mesesVigPol = $("#tipVig").val();
    var inVigPol    = $("#vig_ini").val();
    var sumAs       = $("#suma_asegurada").val().replace(/\./g,'');

    validarPago(1);
}

// Operamos con la informacion obtenida.
var cantCuotas   = $("#cantCuotas").val();
var primaMensual = $("#primaMensual").val();
var diaCobrar    = $("#diaCobrar").val();
var tipoPag      = $("#tipoPag").val();

// Calculamos la prima total (3% de la suma asegurada).
var primaTotal = ((2*sumAs)/100);
// Asignamos el costo en relación a la duración de la vigencia.
var primaTotal = (primaTotal/12)*mesesVigPol;
// Asignamos en el campo el valor de prima total calculada por el sistema
$("#primaTotal").val(Math.trunc(primaTotal));
$("#cantCuotas").empty();
$("#cantCuotas").append('<option value="0" selected disabled>Seleccione</option>');

// El maximo de cuotas disponible, será el mismo valor que meses de vig.
for (var i=2; i<=mesesVigPol; i++){
    // Calculamos el resto de cantidad de meses de vigencia dividido la
    cantidad de cuotas.
    var numero = mesesVigPol % i;
    if(numero==0){
        //Asignamos el valor a las opciones si el resto es 0.
        $("#cantCuotas").append('<option value="'+i+'>' + i + '</option>');
    }
}

// Al modificar las cuotas se asigna el valor de las primas mensuales.
$("#cantCuotas").off('change');
$("#cantCuotas").on('change', function(){
    var cantCuotas = $("#cantCuotas").val();
}

```

```

var primaMensual = Math.trunc(primaTotal/cantCuotas);

$("#primaMensual").val(primaMensual);
});

// Si cambiamos el valor de la prima total, y reseleccionamos la cantidad
de cuotas, se asigna el valor de las primas mensuales en el campo.
$("#primaTotal").on('keyup');
$("#primaTotal").on('keyup', function(){

    // Cada vez que se modifique la prima, obtendremos el valor.
    var primaTotal = $("#primaTotal").val();
    primaTotal = primaTotal.replace(/\./g, '');
    $("#cantCuotas").val(1);
    $("#primaMensual").val('');

    $("#cantCuotas").off('change');
    $("#cantCuotas").on('change', function(){
        var cantCuotas = $("#cantCuotas").val();
        var primaMensual = Math.trunc(primaTotal/cantCuotas);
        $("#primaMensual").val(primaMensual);
    });
});

//Llenamos el select de dia a cobrar con 28 opciones.
$("#diaCobrar").empty().append('<option value="0" selected disabled>Selec
ciona</option>');
for (var i=1; i<=28; i++){
    $("#diaCobrar").append('<option value="'+ i +'>' + i + '</option>');
}
}

// Funcion para administrar o modificar el pago de una poliza.
function adminPago(idPol)
{
    $("#condicionesCuotasModal").modal('open');

    $(".datosPol").css('display', 'block');
    $(".tipoPag").remove();

    texto = "Con esta operacion se modificará el pago ya registrado con anter
ioridad, dependiendo la fecha en que realize este tipo de operaciones puede v
erse afectado cualquier transaccion en el sistema, como cuotas en deuda o fec
has obsoletas, se recomienda desde un principio establecer las condiciones de
pago ni bien se registre la poliza y no volver a modificar esta informacion,
y en caso de llevarlo a cabo, verificar detenidamente la informacion de pago
luego.";

    alertify.alert('Modificar datos de pago', texto).set(configuracionAlert('
ok'));

    $("#tituloModalPago").html("modificar condiciones de pago");
    $(".nombreBoton").html('modificar');
    $(".divVerCuotas").css('display', 'block');
}

```

```

        $("#" + idPrima).val('');
        $("#" + idCuotas).val(1);

        $("#" + idCuotas).empty();

        $.ajax({
            url: 'scripts/aseguradora.php',
            type: 'post',
            dataType: 'json',
            data: {param:50000, idpoliza:idPol},
            success: function(r){
                if(r.data['success']){
                    $("#" + idMensual).prop('disabled', true);
                    $("#" + idPago).val(r.data['pagoId']);

                    $("#" + idNumPoliza).html      (r.data['polizaNum']);
                    $("#" + idEmision).html       (r.data['polizaEmis']);
                    $("#" + idHoraEmision).html   (r.data['polizaEmisHora']);
                    $("#" + idInicioVig).html    (r.data['polizaInVig']);
                    $("#" + idInicioHora).html   (r.data['polizaHorInVig']);
                    $("#" + idFinVig).html       (r.data['polizaFinVig']);
                    $("#" + idFinHora).html      (r.data['polizaHorInVig']);
                    $("#" + idMesesVig).html     (r.data['polizaMesesVig']);
                    $("#" + idTipVig).val        (r.data['polizaMesesVig']);
                    $("#" + idMarcaP).html      (r.data['marca']);
                    $("#" + idModeloP).html     (r.data['modelo']);
                    $("#" + idMotorP).html      (r.data['motor']);
                    $("#" + idLocalidadP).html   (r.data['vehiculoLocalidad']);
                    $("#" + idProvinciaP).html   (r.data['vehiculoProvincia']);
                    $("#" + idCpP).html         (r.data['cpVehiculo']);
                    $("#" + idAnio).html        (r.data['anio']);
                    $("#" + idCeroKm).html      (r.data['ceroKm']);
                    $("#" + idKms).html         (r.data['kilometros']);
                    $("#" + idEGas).html        (r.data['eGas']);
                    $("#" + idCoberturaDesc).html (r.data['coberturaDesc']);
                    $("#" + idCoberturaNombre).html (r.data['cobertura']);
                    $("#" + idCoberturaAd).html  (r.data['coberturaAd']);
                    $("#" + idSumaAs).html      (r.data['sumaAsegurada']);

                    $("#" + idPrimaTotal).val(r.data['primaTotal'].replace(/\./g,''));
                    //Asignamos el valor de la prima calculada por el sistema en el campo.

                    //Obtenemos el valor de esa prima
                    var prima = $("#" + idPrimaTotal).val();
                    prima = prima.replace(/\./g,'');

                    //Asignamos por defecto el valor de una cuota.
                    $("#" + idMensual).val(prima);
                    $("#" + idCuotas).append('<option value="0" disabled>Seleccion
e</option>');
                }
            }
        });
    }
}

```

```

//El maximo de cuotas disponibles sera el mismo valor de meses de vig.
    for (var i = 2; i <= r.data['polizaMesesVig']; i++){
        //Calculamos el resto de cantidad de meses de vig dividido cantCuotas.
        var numero = r.data['polizaMesesVig'] % i;

        if(numero==0){

            //Asignamos el valor a las opciones si el resto es 0.
            $("#cantCuotas").append('<option value="'+i+'">' + i
+ '</option>');
        }
    }

    //Al modificar las cuotas se asigna el valor de las primas mensuales.
    $("#cantCuotas").off('change');
    $("#cantCuotas").on('change', function(){
        var cantCuotas = $("#cantCuotas").val();
        var primaMensual = Math.trunc(prima/cantCuotas);
        $("#primaMensual").val(primaMensual);
    });

    // Si cambiamos el valor de la prima total, y reseleccionamos la cantidad de cuotas, se asigna el valor de las primas mensuales en el campo.
    $("#primaTotal").on('keyup');
    $("#primaTotal").on('keyup', function(){

        //Cada vez que se modifique la prima, obtendremos el valor.
        var prima = $("#primaTotal").val();
        prima = prima.replace(/\./g,'');
        $("#cantCuotas").val(1);
        $("#primaMensual").val(prima);

        $("#cantCuotas").off('change');
        $("#cantCuotas").on('change', function(){
            var cantCuotas = $("#cantCuotas").val();
            var primaMensual = Math.trunc(prima/cantCuotas);
            $("#primaMensual").val(primaMensual);

            if(cantCuotas==1){
                $("#diaCob").css("display", "none");
                $("#fechaCob").css("display", "block");
            }else{
                $("#diaCob").css('display', 'block');
                $("#fechaCob").css("display", "none");
            }
        });
    });

    // Completamos el select.
    var diaSelect = r.data['polizaInVig'].substr(0,2);

```

```

        $("#diaCobrar").empty();
        for (var i = 1; i <= 28; i++) {
            $("#diaCobrar").append('<option value="'+ i +'">' + i + '</option>');
        }

        // Hacemos otra petición AJAX para completar los datos.
        mostrarInfoPago(idPol);
        validarPago(2, null);
    }
}
});

// Función para mostrar la información del pago.
function mostrarInfoPago(idPol)
{
$.ajax({
    url: 'scripts/aseguradora.php',
    type: 'POST',
    dataType: 'json',
    data: {param: 50002, idPol:idPol},
    success: function(r){
        if(r.success){
            var fecha = r.diaCobrar.substr(0,2);
            if(fecha.substr(0,1) == 0){
                fecha = fecha.substr(1);
            }
            $("#primaTotal").val(r.primaTotal);
            $("#primaMensual").val(r.primaMensual);
            $("#cantCuotas").val(r.cantCuotas).trigger('change');
            $("#diaCobrar").val(fecha).trigger('change');
        }else{
            alertify.error('Ocurrió un problema al mostrar la información de pago');
        }
    }
})
}

// Función para validar los datos ingresados en el formulario de pago.
function validarPago(param, datos)
{
    var valFecha = /^[^d{1,2}]\d{1,2}\d{2,4}$/;
    $.validator.addMethod("validarFecha", function(value, element) {
        return this.optional(element) || valFecha.test(value) ;
    });

    numeroCorr = /^[1-9]{1}[0-9]{3,5}|[1-9]{1}[0-9]{1}\.[0-9]{3}|[1-6]{1}[0-9]{2}\.[0-9]{3}|[1-9]{1}\.[0-9]{3}\.[0-9]{3}$/;
    $.validator.addMethod('numeroCorr', function(value, element){
        return this.optional(element) || numeroCorr.test(value);
    });
}

```

```

$.validator.addMethod('cantCuotasCorr', function(value, element){
    return this.optional(element) || cantCuotasCorrecta(value)==true;
});

$.validator.addMethod('primaTot', function(value, element){
    value = value.replace(/\./g,'');
    return this.optional(element) || value>3000&&value<600000;
});

$.validator.addMethod('primaMen', function(value, element){
    value = value.replace(/\./g,'');
    return this.optional(element) || value>100&&value<60000;
});

$.validator.addMethod('diaCobrar', function(value, element){
    return this.optional(element) || value>0&&value<29;
});

function cantCuotasCorrecta(valorIngresado){

    //El numero de cuotas seleccionado es valido si es divisible por la cantidad de meses de duracion de la vigencia.
    mesesVig = $("#tipVig").val();

    for (let i=2; i<=mesesVig; i++){
        numero = mesesVig % i;
        if(numero==0){
            if(valorIngresado==i){
                return true;
            }
        }
    }
}

$("#cuotasForm").validate({
    rules: {
        "primaTotal": {required:true, primaTot: true, numeroCorr: true, number: true},
        "cantCuotas": {required:true, rangelength:[1,12], cantCuotasCorr:true},
        "primaMensual": {required:true, primaMen: true, number: true},
        "diaCobrar": {required:true, diaCobrar:[1,28]},
        "tipoPag":{required: true}
    },
    messages: {
        "primaTotal": {required:"Debe ingresar una Prima Total", primaTot:"La Prima Total ingresada es incorrecta (Debe ser mayor a 3.000 y menor a 60.000)", numeroCorr:"Ingrese un numero de Prima Total correcto", number:"Ingrese un numero correcto"},

        "cantCuotas": {required:"Debe seleccionar una cantidad de Cuotas", rangelength:"La cantidad de Cuotas ingresada no es valida", cantCuotasCorr:"El numero de Cantidad de Cuotas no es permitido."},
    }
});

```

```

    "primaMensual": {required:"Debe de existir una Prima Mensual", pri
maMen: "La Prima Mensual ingresada es incorrecta", number:"Ingrese un numero
correcto"},

    "diaCobrar": {required:"Debe seleccionar un Dia a Facturar", diaCo
brar:"El Dia a Facturar ingresado no es permitido."},
        "tipoPag":{required: "Debe seleccionar un metodo de pago"}
    },

    highlight: function(element) {
        $(element).addClass('claseError');
    },
    unhighlight: function(element) {
        $(element).removeClass('claseError');
    },

    errorElement : 'li',
    errorLabelContainer: '#mensajePago',

    submitHandler: function(form){

        // Dependiendo el parametro, realizaremos una determinada acción.
        if(param == 1){
            grabarPoliza();
        }else if(param == 2){
            modificarCuotas(datos);
        }else if(param == 3){
            var idPoliza = datos;
            renovarPoliza(idPoliza);
        }

    }
    });
}

// Función para modificar las cuotas de un pago.
function modificarCuotas(datos)
{
    var primaTotal    = $("#primaTotal").val();
    var cantCuotas   = $("#cantCuotas").val();
    var primaMensual = $("#primaMensual").val();
    var diaCobrar    = $("#diaCobrar").val();

    var mesesVig    = $("#mesesVig").html();
    var inicioVig   = $("#inicioVig").html();
    var finVig      = $("#finVig").html();

    var idPago = $("#idPago").val();
    var lapso = mesesVig/cantCuotas;

    //Asignamos el valor del lapso de meses
    for (let index = 1; index < 12; index++) {
        if(lapso==index){
            var lapso = index;
        }
    }
}

```

```

}

texto = 'Valor de la prima total de $'+primaTotal+' que se realizara en '+
+cantCuotas+' pagos de $'+primaMensual+'.<br/> El pago de la prima mensual se
facturara cada '+lapso+' mes/es, en el dia '+diaCobrar+' del mes que asi cor
responda, durante un plazo de '+mesesVig+' mes/es de vigencia.';

var datos = [idPago, primaTotal, cantCuotas, primaMensual, diaCobrar, ini
cioVig, finVig, lapso];

alertify.confirm('Modificar informacion de pago', texto,
function(){

    $.ajax({
        url: 'scripts/aseguradora.php',
        type: 'POST',
        dataType: 'json',
        data: {param: 50001, datos:datos},
        success: function(r){

            if(r){
                alertify.success('Pago modificado correctamente');
            }else{
                alertify.error('Hubo un error al modificar el pago');
            }

            condicionesCuotasTabla.ajax.reload();
            $('#primaTotal, #primaMensual, #diaCobrar, #fechaCobrar")
.val('');
            $('#cantCuotas').val(1);
            $('#condicionesCuotasModal').modal('close');
        }
    })

    },
    function(){}
).set(configuracionAlert('okCancel','Registrar'));
}

// Función para mostrar las cuotas de un pago.
function mostrarCuotas()
{
    $('#mostrarCuotasModal').modal({
        dismissible: true
    });

    $('#mostrarCuotasModal').modal('open');

var idPago = $("#idPago").val();

$.ajax({
    url: 'scripts/aseguradora.php',
    type: 'POST',
    dataType: 'json',

```

```

        data: {param: 50003, idPago:idPago},
        success: function(r){
            if(r.data!==null){
                $(".numeroPoliza").html(r.data['numPol']);

                // Mostramos en el cuerpo de la tabla el contenido HTML
                devuelto.
                $("#tbodyTabla").html(r.tbody);
            }else{
                alertify.error('Ocurrio un error al mostrar las cuotas de esta poliza');
            }
        }
    }

// Función para realizar el proceso de facturación de una cuota.
function facturarPago(pagoId)
{
    $("#detallesCuotasModal").modal('open');
    $(".mostrarTabla").css('display', 'block');
    $(".mostrarRecibo, .atrasDetallesCuotas, .envRecibo, .impRecibo").css('display', 'none');
    $("#cuerpoModal").css('opacity','1');

    $.ajax({
        url: 'scripts/aseguradora.php',
        type: 'POST',
        dataType: 'json',
        data: {param: 50004, pagoId:pagoId},
        success: function(r){
            if(r.success){

                $("#numPol").html(r.numPol);
                $("#tomador").html(r.tomador);
                $("#primaTotal").html('$'+r.primaTot);

                // Contenido de la tabla (Listado de cuotas), usamos
                DataTables.

                $('#tablaListadoCuotas').DataTable().destroy();
                listadoCuotasFacturaTabla = $('#tablaListadoCuotas').DataTable();

                {
                    "language": {
                        "zeroRecords": "No se encontraron polizas sin cuotas establecidas",
                        "sSearch": "Buscar",
                    },
                    "ajax":{
                        "url": "scripts/tablas/listadoCuotasFacturaTabla.php"
                    ,
                        "method": 'POST',
                        "data":{param: 50004, pagoId:pagoId},
                        "dataSrc": ""
                }
            }
        }
    })
}

```

```

        },
        "scrollY": 370,
        "scrollCollapse": true,
        "bPaginate": false,
        "bInfo": false,
        "order": [[1, "asc"]],


        "aoColumnDefs": [
            { 'bSortable': false, 'aTargets': [0] },
            { 'bSortable': false, 'aTargets': [1] },
            { 'bSortable': false, 'aTargets': [2] },
            { 'bSortable': false, 'aTargets': [3] },
            { 'bSortable': false, 'aTargets': [4] },
            { 'bSortable': false, 'aTargets': [5] },
            { 'bSortable': false, 'aTargets': [6] },
            { 'bSortable': false, 'aTargets': [7] }
        ],


        "columns": [
            {data:"idenCuota", title: "Ident", width:"12%"}, 
            {data:"numCuota", title: "Nº", width:"4%"}, 
            {data:"priMen", title: "Prima", width:"8%"}, 
            {data:"lapso", title: "Lapso", width:"28%"}, 
            {data:"diaCobrar", title: "Factura", width:"7%"}, 
            {data:"vtoPago", title: "Vto", width:"10%"}, 
            {data:"estado", title: "Estado", width:"26%"}, 
            {data:"boton", title: "Factura", width:"5%"}]
    ],


    //Depende el estado del pago mostramos el texto de un color.
    "createdRow": function(row,data,index){
        estado = data['estado'];

        if(estado == 'A facturar'){
            $('td', row).eq(6).css('color','#000000de');
        }else if(estado == 'Facturada (Sin pagar)'){
            $('td', row).eq(6).css({'color':'#d2c300de', 'font-weight':'600'});
        }else if(estado == 'Pago aprobado'){
            $('td', row).eq(6).css({'color':'#4caf50', 'font-weight':'600'});
        }else if(estado == 'Pago rechazado'){
            $('td', row).eq(6).css({'color':'#ec9815', 'font-weight':'600'});
        }else if (estado == 'Deuda'){
            $('td', row).eq(6).css({'color':'#f44336', 'font-weight':'600'});
        }
    }
});

```

```

// Al presionar en las opciones, llamamos a la función
correspondiente.
    $(document).off('click', '.generarFactura');
    $(document).on('click', '.generarFactura', function(){
        var cuotaId = $(this).attr('value');
        traerDatosReciboCuota(cuotaId);
        $(".efectuar").css('display', 'grid');
        $(".efectuado, .envRecibo, .impRecibo, #actualizarListCuotas").css('display','none');
    });

    $(document).off('click', '.verFactura');
    $(document).on('click', '.verFactura', function () {
        cuotaId = $(this).attr('value');
        traerDatosReciboCuota(cuotaId);
        $(".efectuar, #actualizarListCuotas").css('display', 'none');
        $(".efectuado").css('display', 'block');
        $(".envRecibo, .impRecibo").css('display','unset');
    });

}else{
    alertify.error('Ocurrio un problema al mostrar la informacion');
}
}
}
}

// Función para mostrar la informacion de un recibo de pago de una cuota.
function traerDatosReciboCuota(cuotaId)
{
    // Mostramos y ocultamos los elementos correspondientes del formulario.
    $(".atrasDetallesCuotas").css('display', 'block');
    $(".atrasDetallesCuotas").off('click');
    $(".atrasDetallesCuotas").on('click', function(){
        $(".mostrarTabla, #actualizarListCuotas").css('display', 'block');
        $(".mostrarRecibo, .atrasDetallesCuotas").css('display', 'none');
        $("#cuerpoModal").css('opacity','1');
        $(".envRecibo, .impRecibo").css('display','none');
    })

    $("#emitirFacturaModal").modal('open');

    $(".mostrarTabla, .estadoAccionRecibo").css('display', 'none');
    $(".mostrarRecibo").css('display', 'block');

    $.ajax({
        url: 'scripts/aseguradora.php',
        type: 'POST',
        dataType: 'json',
        data: {param:50005, idCuota:cuotaId},
        success: function(r){
            if(r.success){

```

```

        $(".recibo .nombreComp").html('');

        $(".nroPol").html(r.numPol);
        $(".nroCuota").html(r.nroCu);
        $(".aCobrar").html('$'+r.priMen);
        $(".aseg").html(r.tom);
        $(".domic").html(r.dom);
        $(".emision").html(r.fact);
        $(".laps").html(r.lapCu);
        $(".vigencia").html(r.vig);
        $(".vtoPago").html(r.vtoPa);
        $(".importe").html('$'+r.priMen);
        $(".vehiculo").html(r.veh);
        $(".anio").html(r.anioVe);
        $(".patente").html(r.pat);
        $(".nroMot").html(r.nroMot);
        $(".sAseg").html('$'+r.sumAs);
        $(".cob").html(r.cob);
        $(".cobAd").html(r.cobAd);
        $(".numPoliza").html(r.numPol);
        $(".fechaEmision").html(r.fact);

        $(".fechaEmisRec").html('el '+r.fechCob);

        // Efectuamos el cobro en efectivo de la cuota.
        $(document).off('click', '.efectuarPago');
        $(document).on('click', '.efectuarPago', function(){
            efectuarPago(cuotaId);
        });

        // Imprimimos el recibo de la cuota.
        $(document).off('click', '.impRecibo');
        $(document).on('click', '.impRecibo', function(){
            imprimirRecibo(cuotaId);
        });

        // Enviamos el recibo de pago por correo electrónico.
        $(document).off('click', '.envRecibo');
        $(document).on('click', '.envRecibo', function(){
            enviarRecibo(cuotaId);
        });

    }else{
        // Si ocurre un problema al mostrar la informacion, los
        campos se mostraran con asteriscos.
        alertify.error('Ocurrio un problema al mostrar la informacion
        ');
        $("#efectuarPago").css('display', 'none');
        $("#cuerpoModal").css('opacity', '0.5');

        $(".nroPol").html('*****');
        $(".aseg").html('*****');

```

```

        $(".domic").html('*****');
        $(".emision").html('*****');
        $(".laps").html('*****');
        $(".vigencia").html('*****');
        $(".vtoPago").html('*****');
        $(".importe").html('*****');
        $(".vehiculo").html('*****');
        $(".anio").html('****');
        $(".patente").html('*****');
        $(".nroMot").html('*****');
        $(".sAseg").html('*****');
        $(".cob").html('*****');
        $(".cobAd").html('*****');
        $(".numPoliza").html('*****');
        $(".fechaEmision").html('*****');

        $(".nroCuota").html('*****');
        $(".lap").html('*****');
        $(".imp").html('*****');
        $(".vto").html('*****');
        $(".tom").html('*****');
        $(".nroPag").html('*****');
        $(".estCuota").html('*****');
    }
}
}

// Función para efectuar el cobro en efectivo de una cuota.
function efectuarPago(cuotaId)
{
    // Realizamos una pregunta de confirmación.
    alertify.confirm('Efectuar Pago', 'Esta opción no puede deshacerse ¿Seguro?', 
        function(){
            $.ajax({
                url: 'scripts/aseguradora.php',
                type: 'POST',
                dataType: 'json',
                data: {param: 50006, cuotaId:cuotaId},
                beforeSend: function(){
                    $("#efectuarPago").html('');
                },
                success: function(r){
                    if(r.permitido){

                        // Si el pago se realiza correctamente, mostramos el
                        // mensaje y la fecha de cobro en lugar del botón de "Efectuar pago".
                        if(r.success){
                            setTimeout(function(){

                                $(".efectuar").css('display', 'none');

```

```

        $(".efectuado").css('display', 'block');
        $(".fechaEmisRec").html('el '+r.fechaCobro);
        $(".envRecibo, .impRecibo").css('display','unset'
);

        alertify.success('Pago efectuado');
        listadoCuotasFacturaTabla.ajax.reload();
        traerDatosReciboCuota(cuotaId);
    },2000);

}else{
    $(".efectuar").css('display', 'block');
    $(".efectuado").css('display', 'none');
    $(".envRecibo, .impRecibo").css('display','none');

    alertify.error('Ocurrio un error al procesar la petición');
    $(".btnAcc").css('display', 'block');
}
}else{
    alertify.error('No se permite efectuar esta cuota.');
    $("#efectuarPago").html('Efectuado').show().attr('disabled',
'disabled');
}

})
},
function(){}
).set(configuracionAlert('okCancel','Si')));
}

// Función para actualizar el estado de las cuotas.
function actualizarEstadoCuota()
{
$.ajax({
    url: 'scripts/aseguradora.php',
    type: 'POST',
    dataType: 'json',
    data: {param:50007},
    success: function(r){
        if(r){
            //listadoCuotasFacturaTabla.ajax.reload();
        }
    }
})
}

// Función para imprimir el recibo de pago de una cuota.
function imprimirRecibo()
{
var ticket = document.querySelector(".contRecibo");

var ventana = window.open('', 'PRINT', 'height=600,width=1000');

```

```

        //ventana.document.write('<link rel="stylesheet" href="css/estilos.css">');
    );
    ventana.document.write(ticket.innerHTML);
    ventana.document.close();
    ventana.focus();
    ventana.onload = function () {
        ventana.print();
        ventana.close();
    };

    return true;
}

// Función para enviar el recibo de cuota.
function enviarRecibo(cuotaId)
{
    $(".modalEnviarCorreo").modal('open');

    $.ajax({
        url: 'scripts/aseguradora.php',
        type: 'POST',
        dataType: 'json',
        data: {param:50009, cuotaId:cuotaId},
        success: function(r) {

            if(r.success) {

                info = [r.tomador, r.nroPago, r.nroCuota];
                validarEmail(0, info);
                // Si el cliente tiene correo, lo mostramos en el campo.
                if(r.existeCorreo=='si'){
                    $(".estadoAccionRecibo").css('display', 'block');
                    $(".correoLab").html('Correo electronico');
                    $(".correo").val(r.correo);
                    // Si el cliente no tiene correo, debera llenarse el
                    campo.
                }else if (r.existeCorreo == 'no'){
                    $(".correoLab").html('Ingrese un correo electronico');
                }
                }else {
                    alertify.error('Ha ocurrido un problema al realizar la petici
on al servidor');
                }
            }
        })
    }

function validarEmail(param, data)
{
    $("#formCorreo").validate({
        rules: {
            "correo": { required: true, email: true }
        },
        messages: {

```

```

        "correo": { required: "Debe ingresar una dirección de correo electrónico", email: "Debe ingresar una dirección de correo electrónico válida" },
        },
        highlight: function (element) {
            $(element).css({ 'border': '1px solid #ff00005c', 'background': '#ff00000a' });
        },
        unhighlight: function (element) {
            $(element).css({ 'border': '1px solid #ccc', 'background': 'transparent' });
        },
        errorElement: 'label',
        errorLabelContainer: '.mjeCorr',
        submitHandler: function(form){
            if(param==0){
                enviarCorreo(data);
            }else if(param==1){
                envDescListRecibos('enviar');
            }
        }
    })
}

// Función para efectuar el envío del correo electrónico con el recibo.
function enviarCorreo(data)
{
    // Le pasamos por Ajax la información necesaria.
    var correo = $(".correo").val();

    info = [correo, data[0], data[1], data[2]];

    $.ajax({
        url: 'scripts/emails/enviarEmailRecibo.php',
        type: 'POST',
        dataType: 'json',
        data: {info: info},
        beforeSend(){
            $("#btnEnviarRecibo").html('');
        show();
            $("#btnEnviarRecibo, #cerrarModalEnviarRec").attr('disabled', 'disabled');
        },
        success: function(r) {
            if(r.envCorr){
                alertify.success('Comprobante enviado por correo con éxito');
                $("#btnEnviarRecibo").html('Enviar');
                $(".modalEnviarCorreo").modal('close');
                $("#btnEnviarRecibo, #cerrarModalEnviarRec").attr('disabled', false);
            }else{

```

```

        $("#btnEnviarRecibo").html('Reintentar');
        $("#btnEnviarRecibo, #cerrarModalEnviarRec").attr('disabled',
false);
        alertify.error('Error al enviar el comprobante por correo');
    }

}

})

// Función para mostrar el listado de recibos de una poliza caducada o
renovada.
function mostrarListadoRecibos(pagoId)
{
    $("#divTit").css('display', 'block');
    $("#tituloCont").html('Ver recibos de cuotas');
    $("#divSubTit").html('Listado de recibos de todas las cuotas de la poliza
');

$.ajax({
    url: 'scripts/aseguradora.php',
    type: 'POST',
    dataType: 'json',
    data: {param:50010, pagoId:pagoId},
    success: function(r) {
        if(r.resp['success']){
            // Cargamos en el contenedor el fichero con los recibos cuyo
            contenido es obtenido de una variable de sesión de php.
            $('#contenidoC').load('vista/pagos/mostrarListadoRecibos.php
');

        }else{
            alertify.error('Ha ocurrido un problema al realizar la petición
            al servidor');
        }
    }
})
}

// Función para enviar o descargar el listado de recibos de una poliza.
function envDescListRecibos(accion)
{
    // Dependiendo el parametro que llegue, se enviara por correo o se
descargara directamente.
    if(accion=='descargar'){
        location.href = 'scripts/PDFs/listadoRecibosPDF.php';
    }else if(accion=='enviar'){

        var info = [
            $("#email").val(), //E-mail

```

```

        $(".aseg").html(), //tomador
        $(".nroPol").html() //nroPoliza
    ];

    $.ajax({
        url: 'scripts/emails/enviarListadoRecibos.php',
        type: 'POST',
        dataType: 'json',
        data: {info:info},
        beforeSend(){
            $(".btnEnviarListCu").html('</i> Reenviar').attr('disabled', false);
            }else{
                alertify.error('Error al enviar el comprobante por correo');
                $(".btnEnviarListCu").html('<i class="fas fa-share"></i> Reintentar').attr('disabled', false);
            }
        }
    })
}

// Función donde generamos la tabla con el listado de pagos para poder modificar las cuotas, usando el plugin de DataTables.
function tablaCondicionesCuotas()
{
    condicionesCuotasTabla = $('#condicionesCuotasTabla').DataTable({
        "language": {
            "zeroRecords": "No se encontraron polizas sin cuotas establecidas"
        },
        "sSearch": "Buscar",
        },
        "ajax": "scripts/tablas/tablaModificarCuotas.php",
        "scrollY": 370,
        "scrollCollapse": true,
        "bPaginate": false,
        "bInfo": false,
        "order": [[5, "desc"]],
        "aoColumnDefs": [
            { 'bSortable': false, 'aTargets': [6] }

        ],
        "columns": [

```

```

        {data:"nro", title: "Nro",width:""},  

        {data:"tomador", title: "Tomador",width:""},  

        {data:"vehiculo", title: "Vehiculo",width:""},  

        {data:"vigin", title: "Inicio Vig",width:""},  

        {data:"vigfin", title: "Fin Vig",width:""},  

        {data:"reg", title: "Registro",width:""},  

        {data:"boton", title: "Cuotas",width:""}  

    ]  

});  
  

return condicionesCuotasTabla;  

}  
  

// Función donde generamos la tabla con el listado de polizas para facturar  

el pago, usando el plugin DataTables.  

function tablaFacturarPago()  

{
    tablaFacturarPago = $("#verPagoTabla").DataTable({  

        "language": {  

            "zeroRecords": "No se encontraron pagos registrados en la base de  

datos",  

            "sSearch": "Buscar Pagos",  

        },  
  

        "ajax": "scripts/tablas/tablaFacturarPago.php",  

        "scrollY": 370,  

        "scrollCollapse": true,  

        "bPaginate": false,  

        "bInfo": false,  

        "order": [[4, "desc"]],  

        "aoColumnDefs": [  

            { 'bSortable': false, 'aTargets': [5] }  

        ],  
  

        "columns": [  

            {data:"nro" , title: "Pago num",width:""},  

            {data:"polizaNum", title: "Poliza num",width:""},  

            {data:"tomador" , title: "Tomador",width:""},  

            {data:"nroCuotas", title: "Nº Cuotas",width:""},  

            {data:"estado" , title: "Estado",width:""},  

            {data:"boton" , title: "",width:""}  

        ]  

});  

}

```

```
// Función donde agregamos funcionalidades extras para los formularios y
// elementos de pagos y cuotas.
function funcionalidadesPagos()
{
    $('input, select').addClass('browser-default');
    // Ocultamos algunos elementos.
    $(".divVerCuotas").css('display', 'none');
    $("#primaMensual").prop('disabled', true);
    // Evitamos que los modales se cierren con un click fuera.
    $('.modal').modal({
        dismissible: false
    });

    // Configuramos el modal para ingresar el correo electrónico al
    $(".modalEnviarCorreo").modal({
        onCloseEnd: function(){
            $("#formCorreo").validate().destroy();
        },
        dismissible: false,
        endingTop: '20%'
    })
}
```

➔ 4.3.3.6 fichero “poliza.js”

Este fichero contiene todas las funciones que realizan las acciones relacionadas al frontend o vista de las polizas del sistema.

```
// Leemos el DOM a través de jQuery.
jQuery(document).ready(function(){

    // Llamamos a varias funciones directamente o a través de acciones.
    funcionalidadesPoliza();
    actualizarEstadosPoliza();
    listadoCoberturas();

    $("#siguientePoliza").on('click', function(){
        validarPoliza(1);
    });

    $(document).off('click', '.renovarPoliza');
    $(document).on('click', '.renovarPoliza', function(){
        var idPoliza = $(this).attr('value');
        var nroPoliza = $(this).attr("data-id");
        var sumAs = $(this).attr('data-sumAs');

        var info = [idPoliza, nroPoliza, sumAs];
        mostrarInfoRenovacion(info);
    });

    $(document).off('click', '.verConPol')
    $(document).on('click', '.verConPol', function(){
        var idPoliza = $(this).attr('value');

        $(document).off('click', '.btnVolverConPol')
        $(document).on('click', '.btnVolverConPol', function () {
            $(this).css('background', '#e0e0e0');
            $('#contenidoC').load('vista/poliza/elementosPoliza/consultar_polizas.php', function () {
                tablaConsultarPoliza();
            });
        });
        mostrarInformacion(idPoliza);

        $(document).off('click', '.btnActConPol');
        $(document).on('click', '.btnActConPol', function(){
            $(this).html(''+'actualizar').attr('disabled', 'disabled');
            setTimeout(() => {
                mostrarInformacion(idPoliza);
            }, 3000);
        })
    });
});
```

```

        $("#actListPol").off('click');
        $("#actListPol").on('click', function(){
            $(this).html(''+'actualiza-
r').attr('disabled', 'disabled');
            setTimeout(() => {
                $(this).html('<i class="fas fa-sync-
alt"></i> actualizar').attr('disabled', false);
                actualizarEstadosPoliza();
                tablaConsultarPoliza.ajax.reload();
            }, 1000);
        });
    });

// Función para validar los datos ingresados durante todo el proceso de
emisión de poliza.
function validarPoliza(param, info)
{
    var valFecha = /^(\d{1,2})/(\d{1,2})/(\d{2,4})$/;
    $.validator.addMethod("validarFecha", function(value, element) {
        return this.optional(element) || valFecha.test(value) ;
    });

    $.validator.addMethod("validarTF", function(value, element) {
        return this.optional(element) || validarTipoFecha(value)==true;
    });

    $.validator.addMethod("validarTipVig", function(value, element) {
        return this.optional(element) || valTipoVig(value)==true;
    });

    $.validator.addMethod("validarTipFinVig", function(value, element) {
        return this.optional(element) || validarTipoFechaInVig(value)==true;
    });

    // Función interna de validación, para comprobar que las fechas de
vigencia sean ingresadas correctamente en relación al dia y horario actual.
    function validarTipoFecha(fecha){

        // Hora actual.
        horaActual = moment(new Date()).format('HH:mm');
        // 12 del medio dia.
        horaComparar = moment('12:00', 'HH:mm').format('HH:mm');
        // Fecha ingresada por el usuario, la cual se corroborara después.
        fechaIngresada = moment($("#vig_ini").val(), 'DD/MM/YYYY').format('MM/
DD/YYYY');

        // Si la hora actual es mayor a la hora a comparar (12 medio dia),
la fecha a comprarla, será un dia después al actual.
        if(horaActual>horaComparar){
            var fechaComparar = moment(new Date()).add(1, 'days').format('MM/
DD/YYYY'); //Un dia despues de hoy.
        }
    }
}

```

```

        // Si la hora actual es menor a 12 del medio dia, la fecha a comparar
        // será el dia actual.
        }else if(horaActual<horaComparar){
            var fechaComparar = moment(new Date()).format('MM/DD/YYYY'); //Fe
            cha actual.
        }

        // Si la fecha ingresada es mayor a la fecha a comparar.
        if(fechaIngresada>fechaComparar){
            return true;
        // Si la fecha ingresada es menor.
        }else if(fechaIngresada<fechaComparar){
            return false;
        // Si la fecha ingresada es igual.
        }else if(fechaIngresada==fechaComparar){
            return true;
        }
    }

    // Función interna de validación para comprobar que el valor ingresado
    // para la cantidad de meses de vigencia sea correcto.
    function valTipoVig(valor){
        for (let i=4; i<=12; i++) {
            if(valor==i){
                return true;
            }
        }
    }

    // Función interna de validación para comprobar que la fecha de fin de
    // vigencia coincida con la cantidad de meses posterior a la fecha de inicio de
    // vigencia.
    function validarTipoFechaInVig(inVig){

        inicioVigencia = moment(inVig, 'DD/MM/YYYY'); //Inicio de vigencia in
        gresado
        finVigencia = moment($("#vig_fin").val(), 'DD/MM/YYYY'); //Fin de vi
        gencia del input
        cantMeses = $("#tipVig").val();
        diferencia = finVigencia.diff(inicioVigencia, 'months'); //Obtenemos
        la diferencia en meses

        //Si la diferencia en meses entre el inicio de vigencia y el fin de v
        igencia es igual a la cantidad de meses, devuelve true.
        if(cantMeses==diferencia){
            return true;
        }else if(cantMeses!=diferencia){
            return false;
        }
    }

    $("#divPoliza").validate({
        rules: {

```

```

        "vig_ini": {required:true, validarFecha: true, validarTF: true, v
alidarTipFinVig: true},
        "vig_fin": {required:true, validarFecha: true},
        "tipVig": {required:true, validarTipVig: true}
    },
    messages: {
        "vig_ini": {required:"Seleccione una fecha de Inicio de Vigencia"
, validarFecha: "El tipo de fecha ingresado no es valido", validarTF: "La Fec
ha de inicio de vigencia no es permitida", validarTipFinVig: "La Fecha de Fin
de Vigencia ingresada no es permitida"},
        "vig_fin": {required:"Seleccione una fecha de Fin de Vigencia", v
alidarFecha: "El tipo de fecha ingresado no es valido"},
        "tipVig": {required:"Seleccione una opcion par Fin de vigencia",
validarTipVig: "La opcion de Fin de Vigencia ingresada no es valida."}
    },

    highlight: function(element) {
        $(element).closest('.hasDatepicker').addClass('claseError');
        $(element).closest('.tipVig').addClass('claseError');
    },
    unhighlight: function(element) {
        $(element).closest('.hasDatepicker').removeClass('claseError'
);
        $(element).closest('.tipVig').removeClass('claseError');
    },

    errorElement : 'li',
    errorLabelContainer: '#mensajePoliza',

    submitHandler: function(form){
        if(param == 1){
            validarTomador();
        }else if(param == 2){
            generarFormularioPago(1, info);
        }
    }
});
}

// Función para validar el campo de tomador, usando el plugin jquery
validate.
function validarTomador()
{
    $('.tabs').tabs('select','test2');
    boton = '<button type="submit" class="btn btn-
flat green" id="siguienteTomador" value="Enviar"> siguiente <i class="fas fa-
caret-right"></i> </button>';
    $(".btnDivTomador").html(boton);

    $("#siguienteTomador").attr('disabled', false);

    $('#select_cliente').change(function(){
        if ($(this).val()!="")

```

```

        {
            $(this).valid();
        }
    });

    $("#divTomador").validate({
        rules:{
            "select_cliente":{required:true}
        },
        messages:{
            "select_cliente":{required: "Debe seleccionar un Tomador"}
        },
        highlight: function(element){
            $(element).siblings('span').find('.selection').find('.select2-selection').addClass('claseError');
        },
        unhighlight: function(element) {
            $(element).siblings('span').find('.selection').find('.select2-selection').removeClass('claseError');
        },
        errorElement : 'li',
        errorLabelContainer: '#mensajeTomador',
        submitHandler: function(form){
            validarVehiculo();
        }
    });
    //});
}

// Función para validar los campos del formulario de vehiculo al emitir una poliza, usando el plugin jquery validate.
function validarVehiculo()
{
    $('.tabs').tabs('select','test3');
    boton = '<button type="submit" class="btn btn-flat green" id="siguienteVehiculo" value="Enviar"> siguiente <i class="fas fa-caret-right"></i> </button>';
    $(".btnDivVehiculo").html(boton);

    $("#siguienteVehiculo").attr('disabled', false);

    $('#select_marca, #select_modelo1, #codigoPostal, #vehiculoLocalidad').change(function(){
        if ($(this).val()!="")
        {
            $(this).valid();
        }
    });
}

```

```

sumaAsegurada = /^[1-9]{1}[0-9]{4,7}$|[1-9]{1}[0-9]{1}\.[0-9]{3}$|[1-9]{1}[0-9]{2}\.[0-9]{3}$|[1-9]{1}\.[0-9]{3}\.[0-9]{3}$|[1-9]{1}[0-9]{1}\.[0-9]{3}\.[0-9]{3}$/;
$.validator.addMethod('sumaAsegurada', function(value, element){
    return this.optional(element) || sumaAsegurada.test(value);
});

patente = /^[a-zA-Z]{3}[0-9]{3}$|[a-zA-Z]{2}[0-9]{3}[a-zA-Z]{2}$/;
$.validator.addMethod("patente", function(value, element) {
    value.replace(/ /g, "");
    return this.optional(element) || patente.test(value);
});

nroChasis = /^[a-zA-Z0-9]{18}$/;
$.validator.addMethod("nroChasis", function(value, element) {
    return this.optional(element) || nroChasis.test(value);
});

nroMotor = /^[a-zA-Z]{2}[0-9]{3}[a-zA-Z]{2}$/;
$.validator.addMethod("nroMotor", function(value, element) {
    return this.optional(element) || nroMotor.test(value);
});

cantKm = /^[1-9]{1}[0-9]{1,6}$|[1-9]{1}[0-9]{1}\.[0-9]{3}$|[1-9]{1}[0-9]{2}\.[0-9]{3}$|[1-9]{1}\.[0-9]{3}\.[0-9]{3}$/;
$.validator.addMethod('cantKm', function(value, element){
    return this.optional(element) || cantKm.test(value);
});

anio = /^[1]{1}[9]{1}[5-9]{1}[0-9]{1}$|[2]{1}[0]{1}[0]{1}[0]{1}[0-9]{1}$|[2]{1}[0]{1}[1]{1}[0-9]{1}$|[2]{1}[0]{1}[2]{1}[0-1]{1}$/;
$.validator.addMethod('anio', function(value, element){
    return this.optional(element) || anio.test(value);
});

$.validator.addMethod("validarComb", function(value, element) {
    return this.optional(element) || value=="Nafta" || value=="GasOil"
" || value=="Energia Electrica";
});

//$("#siguienteVehiculo").on('click', function(){

    $("#divVehiculo").validate({
        rules:{
            "suma_asegurada": {required: true, sumaAsegurada: true},
            "selec_marca": {required: true, rangelength:[1,79]},
            "select_modelo1": {required: true, rangelength:[1,1005]},
            "patente": {required: true, patente: true},
            "motor": {required: true},
            "nroChasis": {required: true, nroChasis: true},
            "nroMotor": {required: true, nroMotor: true},
            "select_clase": {required: true, rangelength:[1,14], number:
true},
        }
    });
});

```

```

        "selectTipoUso": {required: true, rangelength:[1,25], number: true},
        "cant_kilometro": {required: true, cantKm:true},
        "codigoPostal": {required: true, rangelength:[1,22963], number: true},
        "vehiculoLocalidad": {required: true, rangelength:[1,22963], number: true},
        "coberturaAd": {required: true, rangelength:[1,6], number: true},
        "combustible": {required: true, validarComb:true},
        "anio": {required: true, anio: true},
        "pasajeros": {number: true},
        "asientos": {number: true},
        "color": {lettersonly: true}
    },
    messages:{
        "suma_asegurada": {required: "Debe ingresar la Suma Asegurada", sumaAsegurada: "La Suma Asegurada es incorrecta"},
        "selec_marca": {required: "Debe seleccionar una Marca", rangeLength:"La Marca seleccionada es incorrecta"},
        "select_modelo1": {required: "Debe seleccionar un Modelo", rangeLength:"El Modelo seleccionado es incorrecto"},
        "patente": {required: "Debe ingresar la Patente", patente: "La Patente ingresada es incorrecta"},
        "motor": {required: "Debe ingresar el Motor"},
        "nroChasis": {required: "Debe ingresar el Numero de Chasis", nroChasis: "El numero de Chasis es incorrecto"},
        "nroMotor": {required: "Debe ingresar el Numero de Motor", nroMotor: "El Numero de Motor es incorrecto"},
        "select_clase": {required: "Debe seleccionar la Clase", rangeLength:"La Clase seleccionada es incorrecta", number:"La Clase seleccionada es incorrecta 2"},
        "selectTipoUso": {required: "Debe seleccionar el Uso", rangeLength:"El Uso seleccionado es incorrecto", number:"El Uso seleccionado es incorrecto 2"},
        "cant_kilometro": {required: "Debe ingresar la Cantidad de Kilometros", cantKm:"El valor de Cantidad de Kilometros es incorrecto"},
        "codigoPostal": {required: "Debe seleccionar un Codigo Postal", rangeLength:"El Codigo Postal seleccionado es incorrecto", number:"El Codigo Postal seleccionado es incorrecto 2"},
        "vehiculoLocalidad": {required: "Debe seleccionar la Localidad", rangeLength:"La Localidad seleccionada es incorrecta", number:"La Localidad seleccionada es incorrecta 2"},
        "coberturaAd": {required: "Debe seleccionar una Cobertura Adicional", rangeLength:"La Cobertura Adicional seleccionada es incorrecta", number:"La Cobertura Adicional seleccionada es incorrecta 2"},
        "combustible": {required: "Debe seleccionar un tipo de Combustible", validarComb:"El tipo de Combustible seleccionado es incorrecto"},
        "anio": {required: "Debe ingresar el Año", anio:"El Año ingresado es incorrecto"},
        "pasajeros": {number:"El valor ingresado de Pasajeros es incorrecto"},
    }
}

```

```

        "asientos": {number:"El valor ingresado de Asientos es incorrecto"},
        "color": {lettersonly:"El valor ingresado de Color es incorrecto"}
    },
    highlight: function(element){
        if($(element).closest('.campo').hasClass('select2')){
            $(element).siblings('span').find('.selection').find('.select2-selection').addClass('claseError');
        }else{
            $(element).closest('.campo').addClass('claseError');
        }
    },
    unhighlight: function(element) {
        if($(element).closest('.campo').hasClass('select2')){
            $(element).siblings('span').find('.selection').find('.select2-selection').removeClass('claseError');
        }else{
            $(element).closest('.campo').removeClass('claseError');
        }
    },
    errorElement : 'li',
    errorLabelContainer: '#mensajeVehiculo',

    submitHandler: function(form){
        validarCobertura();
    }
});

//});

}

// Función para validar el campo de cobertura al emitir una poliza, usando el plugin jquery validate.
function validarCobertura()
{
    $('.tabs').tabs('select','test4');
    boton = '<button type="submit" class="btn btn-flat green" id="siguienteCobertura"> siguiente <i class="fas fa-caret-right"></i> </button>';
    $(".btnDivCobertura").html(boton);

    $("#siguienteCobertura").attr('disabled', false);

    $('#select_cobertura').change(function(){
        if ($(this).val()!="")
        {
            $(this).valid();
        }
    });
}

//$("#cargar_datos").on('click', function(){


```

```

        $("#divCobertura").validate({
            rules:{ "select_cobertura": {required: true} },
            messages:{ "select_cobertura": {required: "Debe seleccionar un Tipo de Cobertura"} },
            highlight: function(element){
                $(element).closest('.campo').addClass('claseError');
            },
            unhighlight: function(element){
                $(element).closest('.campo').removeClass('claseError');
            },
            errorElement : 'li',
            errorLabelContainer: '#mensajeCobertura',
            submitHandler: function(form){
                generarFormularioPago(2, null);
            }
        });
    });

// Función para llenar un select con la lista de modelos una vez se haya seleccionado la marca.
function llenarModelo(marca, modeloModificar)
{
    jQuery('#select_modelo1').empty();
    jQuery('#select_modelo1').append('<option value="" disabled selected>Seleccione un modelo</option>');
    jQuery.ajax({
        type:"post",
        data:{param:10000, marca:marca},
        url: "scripts/aseguradora.php",
        dataType:'json',
        success: function(r){
            if(r!=null){
                jQuery.each(r.data, function(k,v){
                    if(modeloModificar==v.id){
                        jQuery('#select_modelo1').append('<option value="' + v.id + '" selected>' + v.nombre + '</option>');
                    }else{
                        jQuery('#select_modelo1').append('<option value="' + v.id + '">' + v.nombre + '</option>');
                    }
                });
            }
        });
    });
}

// Función para mostrar la información de una póliza.
function mostrarInformacion(idpoliza)
{
    // Cargamos el fichero de información en el contenedor general.
    $('#contenidoC').load('vista/poliza/elementosPoliza/informacionPoliza.php');
}

```

```

jQuery.ajax({
    url: "scripts/aseguradora.php",
    type: "post",
    data: {param:10003, idpoliza:idpoliza},
    dataType: 'json',
    success: function (r) {
        if (r.success) {

            $(".btnActConPol").html('<i class="fas fa-sync-alt"></i> Actualizar').attr('disabled', false)

            infPol = '<p>Numero <span>' + r.info[0] + '</span></p>' +
                '<p>Emitida el <span>' + r.info[1] + '</span> (' + r.info[2] +
                ') por <span>' + r.info['ne'] + '</span> (' + r.info['nroe'] + ') en ' + r.info['ns'] +
                ' (' + r.info['is'] +')</p>' +
                '<p>Vigente desde el <span>' + r.info[3] + '</span> hasta el
                <span>' + r.info[4] + '</span> 12:00hs (<span>' + r.info[5] + ' meses</span>)</p>' +
                '<p> Fecha de anulacion <span>' + r.info[6] + '</span></p>';
            $(".infPol").html(infPol);

            infEndSin = '<p>Numero de endosos realizados <span>' + r.info[7] + '</span></p>' +
                '<p> Numero de siniestros involucrados <span>' + r.info[8] + '</span></p>';
            $(".infEndSin").html(infEndSin);
            infCo = '<p><span>' + r.info[9] + '</span></p>';
            $(".infCo").html(infCo);
            infCoA = '<p><span>' + r.info[10] + '</span></p>';
            $(".infCoA").html(infCoA);

            infTom = '<p>Numero <span>' + r.info[12] + '</span> </p>' +
                '<p> Nombre completo <span>' + r.info[13] + '</span> </p>' +
                '<p> Nro documento <span>' + r.info[14] + '</span> </p>' +
                '<p> Tipo de persona <span>' + r.info[15] + '</span> </p>' +
                '<p> Nacionalidad <span>' + r.info[16] + '</span> </p>' +
                '<p> Domicilio <span>' + r.info[11] + '</span> </p>' +
                '<p> Código postal <span>' + r.info[17] + '</span> </p>' +
                '<p> Fecha nacimiento <span>' + r.info[18] + '</span>
                </p>' +
                '<p> Genero <span>' + r.info[19] + '</span> </p>' +
                '<p> Telefono <span>' + r.info[20] + '</span> </p>' +
                '<p> Correo <span>' + r.info[21] + '</span> </p>' +
            $(".infTom").html(infTom);

            if (r.info[25] == 1){
                var metodo = 'Pago en efectivo';
            }else if (r.info[25] == 2){
                var metodo = 'Tarjeta de Credito Asociada';
            }
        }
    }
});

```

```

infPag = '<span class="subtitulo">Metodo de pago</span>' +
          '<p> Numero de pago <span> ' + r.info[22] +'</span> <
/p>' +
          '<p> Fecha de registro <span> ' + r.info[23] +'</span>
</p>' +
          '<p> Prima total <span>' + '$'+r.info[24] +'</span> <
/p>' +
          '<p> Metodo de pago <span>' + metodo +'</span> </p>' +
          '<p> Estado actual <span>' + r.info[26] +'</span> </p
>';
        $(".infPag").html(infPag);

//Mostramos el listado de cuotas
$.each(r.cuotas, function(k,v){
    cuotas = '<tr>' +
              '<td>' +v.numero_cuota+'</td>' +
              '<td>' +v.iden_cuota+'</td>' +
              '<td>' +'$'+v.prima_mensual+'</td>' +
              '<td>' +v.estadoCuota+'</td>' +
              '</tr>';
    $('#tbodyCuo').append(cuotas);
})

$.each(r.estados, function(k, v){
    if(k == 0){
        $(".estAct").append('<span>' + v.estado + '</span>');
    }else{
        $(".estAnt").append('<p> <span>' +v.estado+'</span> </p
>');
    }
})

Iz ='<p> Suma asegurada <span>+' +'$'+ r.info[32] +'</span> </p
>' +
        '<p> Marca <span> ' + r.info[27] +'</span> </p>' +
        '<p> Modelo <span> ' + r.info[28] +'</span> </p>' +
        '<p> Motor <span> ' + r.info[33] +'</span> </p>' +
        '<p> Patente <span> ' + r.info[34] +'</span> </p>' +
        '<p> Nro chasis <span> ' + r.info[35] +'</span> </p>' +
        '<p> Nro motor <span> ' + r.info[36] +'</span> </p>' +
        '<p> Clase <span> ' + r.info[29] +'</span> </p>' +
        '<p> Uso <span> ' + r.info[30] +'</span> </p>' +
        '<p> GPS <span> ' + r.info[37] +'</span> </p>';

De ='<p> Cero Km <span> ' + r.info[38] +'</span> </p>' +
        '<p> Cantidad de Kilometros <span> ' + r.info[39] +' Kms'
+'</span> </p>' +
        '<p> Localidad <span> ' + r.info[31] +' </span> </p>' +
        '<p> Codigo postal <span> ' + r.info[40] +'</span> </p>' +
        '<p> Combustible <span> ' + r.info[41] +'</span> </p>' +
        '<p> Equipo de gas <span> ' + r.info[42] +'</span> </p>' +
        '<p> Año <span> ' + r.info[43] +'</span> </p>' +
        '<p> Color <span> ' + r.info[44] +'</span> </p>'+
```

```

        '<p> Asientos <span>' + r.info[45] +'</span> </p>'+
        '<p> Pasajeros <span>' + r.info[46] +'</span> </p>';

        $(".Iz").html(Iz);
        $(".De").html(De);

    } else {
        alertify.error('Ocurrio un problema al mostrar la informacion
.');
    }
}

});

}

// Función para grabar una nueva poliza.
function grabarPoliza()
{
    if($("#GPS").is(':checked')){
        var gps = 'Si';
    }else{
        var gps = 'No';
    }
    if($("#ceroKm").is(':checked')){
        var ceroKm = 'Si';
    }else{
        var ceroKm = 'No';
    }
    if($("#eGas").is(':checked')){
        var eGas = 'Si';
    }else{
        var eGas = 'No';
    }

    // Datos del vehiculo.
    var sumAs = $('#suma_asegurada').val();
    var mar = $('#select_marca').val();
    var mod = $('#select_modelo1').val();
    var pat = $('#patente').val();
    var mo = $('#motor').val();
    var nroCh = $('#nroChasis').val();
    var nroMot = $('#nroMotor').val();
    var cla = $('#select_clase').val();
    var us = $('#selectTipoUso').val();
    var kms = $('#cant_kilometro').val();
    var cp = $('#codigoPostal').val();
    var loc = $('#vehiculoLocalidad').val();
    var cobAd = $('#coberturaAd').val();
    var com = $('#combustible').val();
    var an = $('#anio').val();
    var pa = $('#pasajeros').val();
    var as = $('#asientos').val();
    var co = $('#color').val();
}

```

```

var datosVehiculo = [sumAs, mar, mod, pat, mo, nroCha, nroMot, cla, us, g
ps, ceroKm, kms, cp, loc, cobAd, com, eGas, an, pa, as, co];

// Datos de la poliza.
var nroPol    = $('#nroPoliza').html();
var vig_ini   = $('#vig_ini').val();
var vig_fin   = $('#vig_fin').val();
var tipVig    = $('#tipVig').val();

var cli = $('#select_cliente').val();

var cob = $('#select_cobertura').val();

var datosPoliza = [nroPol, vig_ini, vig_fin, cli, cob, tipVig];

// Datos del pago.
var primaTotal  = $("#primaTotal").val();
var cantCuotas = $("#cantCuotas").val();
var primaMensual = $("#primaMensual").val();
var diaCobrar   = $("#diaCobrar").val();

var lapso = (tipVig/cantCuotas).toString();

var metodoPago = $("#tipoPag").val();

var texto = 'Valor de la prima total de $'+primaTotal+' que se realizara en '+cantCuotas+' pagos de $'+primaMensual+'.<br/> El pago de la prima mensual se facturara cada '+lapso+' mes/es, en el dia '+diaCobrar+' del mes que asi corresponda, durante un plazo de '+tipVig+' mes/es de vigencia.';

var datosPago = [primaTotal, cantCuotas, primaMensual, diaCobrar, vig_ini, vig_fin, lapso, metodoPago];

alertify.confirm('Registrar Pago', texto,
function(){
$.ajax({
    type:"post",
    data:{param:10002, datosPoliza:datosPoliza, datosVehiculo:datosVehiculo, datosPago:datosPago},
    url: "scripts/aseguradora.php",
    dataType:'json',
    success: function(r){
        if(r.success){

            $('#contenidoC').load('vista/poliza/elementosPoliza/formulario_poliza.php');

            alertify.success('Poliza cargada con exito.');

        }else{
            alertify.error('Error al cargar la poliza.');
        }
    }
});
});

```

```

        },
        function(){}
    ).set(configuracionAlert('okCancel','Registrar'));
}

// Función para listar en un select todas las localidades una vez se haya
seleccionado el código postal.
function mostrarLocalidades(cp, LocalidadSelect)
{
    jQuery("#vehiculoLocalidad").empty();
    jQuery("#vehiculoLocalidad").append('<option value="" disabled selected>S
eleccione Localidad</option>');
    $.ajax({
        type:"post",
        data:{param:80000, cp:cp},
        url: "scripts/aseguradora.php",
        dataType:'json',
        success: function(r){
            if(r!=null){
                jQuery.each(r.data, function(i,val){
                    if(localidadSelect==val.localidadId){
                        jQuery('#vehiculoLocalidad').append('<option value="'
+ val.localidadId + '" selected>' + val.localidadNombre + '</option>');
                    }else{
                        jQuery('#vehiculoLocalidad').append('<option value="'
+ val.localidadId + '">' + val.localidadNombre + '</option>');
                    }
                });
            }
        });
    });
}

// Función para mostrar la información de la poliza en el proceso de
renovación.
function mostrarInfoRenovacion(info)
{
    var idPoliza = parseInt(info[0]);
    var nroPoliza = info[1];
    var sumAs = info[2];
    var idPoliza = [idPoliza];
    //Comprobamos si la poliza se encuentra en deuda, y de ahí permitimos ren
    ovarla o no.
    $.ajax({
        url: "scripts/aseguradora.php",
        type: 'POST',
        dataType: 'json',
        data: {param: 50008, data:idPoliza},
        success: function(r){
            if(r.success){
                var fact = r.fact;
                var apro = r.apro;
                var rech = r.rech;
                var deud = r.deud;

```

```

var caCu = r.cantCuotas;
var nroP = r.nroPago;

//Si la cantidad de cuotas es igual a la cantidad de cuotas con pago aprobado (Todas las cuotas pagadas).
if(caCu==apro){

    $('#contenidoC').load('vista/poliza/formularios/formulariosRenovarPoliza.php', function(){
        funcionalidadesPoliza();

        $('#divSubTit').html(''); //Eliminamos el subtítulo
        $('.titForPol').html('datos de poliza '+nroPoliza);
        $('.infoCabeceraEmisPoliza, .divResumenVigPoliza').remove();

        //Obtenemos los datos por parámetro y los enviamos a la función de validación.
        var infoRen = [idPoliza, sumAs];
        validarPoliza(2, infoRen);

        $('.atrRenPol').off('click');
        $('.atrRenPol').on('click', function(e){
            e.preventDefault();
            $('#contenidoC').load('vista/poliza/elementosPoliza/renovar_poliza.php', function(){
                tablaRenovarPoliza();
            });
        });
    });

}else{
    texto =
        <div class="alertRenPol">
            <div class="ARP1"> ¡ Importante ! </div>
            <div class="ARP2"> Para llevar a cabo una renovación, todas las cuotas de la poliza deben estar con su correspondiente pago aprobado sin excepción. </div>
            <div class="ARP3"> Verifique el pago nro: <span class="ARP3-1"> `+nroP+` </span></div>

            <div class="ARP4">
                <div class="ARP4-1"> Detalles </div>
                <div class="ARP4-2">
                    Cuotas con pagos aprobados: <span class="ARP4-2-1"> `+apro+` </span>
                </div>
                <div class="ARP4-3">
                    Cuotas con pagos sin efectuar: <span class="ARP4-3-1"> `+(fact+rech+deud)+` </span>
                </div>
                <div class="ARP4-3-2">
                    Facturados: <span class="ARP4-3-2-1"> `+fact+` </span>
                </div>
            </div>
}

```

```

                <div class="ARP4-3-
2"> Rechazados: <span class="ARP4-3-2-2"> `+rech+` </span> </div>
                    <div class="ARP4-3-
2"> En deuda: <span class="ARP4-3-2-3"> `+deud+` </span> </div>
                        </div>
                    </div>
                `;

            alertify.alert('Renovar Poliza', texto).set(configuracion
Alert('ok'));

            $('.alertRenPol').parent('.ajs-content').parent('.ajs-
body').parent('.ajs-dialog').css('top', '0px');
        }
    }else{
        alertify.error('Ocurrio un error al procesar la peticion');
    }

}
}

// Función para renovar la poliza.
function renovarPoliza(idPoliza)
{
    var infoRenPol =
    [
        idPoliza,
        $('#vig_ini').val(),
        $('#tipVig').val(),
        $('#vig_fin').val(),
        $('#primaTotal').val(),
        $('#cantCuotas').val(),
        $('#primaMensual').val(),
        $('#diaCobrar').val(),
        $('#tipoPag').val()
    ];

    $.ajax({
        url: "scripts/aseguradora.php",
        type: "post",
        dataType: "json",
        data: {param:10005, data:infoRenPol},
        success: function(r){
            if (r.renPoliza){
                alertify.success('Poliza renovada con exito');
                $('#contenidoC').load('vista/poliza/elementosPoliza/renovar_p
oliza.php', function(){
                    tablaRenovarPoliza();
                });
            }else{
                alertify.error('Ocurrio un problema al renovar la poliza.');
            }
        }
    });
}

```

```

        }
    })
}

// Función para actualizar los estados de las polizas.
function actualizarEstadosPoliza() {
    $.ajax({
        url: 'scripts/aseguradora.php',
        type: 'POST',
        dataType: 'json',
        data: { param: 10006},
        success: function (r) {
            if (r.success) {
                //alertify.success('Estados de poliza actualizado!!!');
            }
        }
    })
}

// Función para mostrar la informacion de una cobertura al momento de seleccionar una.
function listadoCoberturas()
{
    $("#infoCobertura")
    $("#select_cobertura").empty().append('<option value="0" selected disabled>Seleccione</option>');

    $.ajax({
        url: "scripts/aseguradora.php",
        type: "post",
        dataType: "json",
        data: {param:10001},
        success: function(r){
            if(r!=null){

                $.each(r, function(k, val){
                    $("#select_cobertura").append('<option value="'+val.id+'"'+'+val.nombre+': '+val.descripcion+'</option>');
                })

                //Al seleccionar la cobertura se mostrara la informacion de la misma.
                $('#select_cobertura').on('change', function(){

                    var idCob = $('#select_cobertura').val();

                    for (let i=0; i<=5; i++) {
                        if(idCob==r[i]['id']){
                            $('#infoCobertura').html(r[i]['informacion']);
                        }
                    }
                });
            }
        }
    })
}

```

```

        }
    });
}

// Función donde generamos la tabla con el listado de polizas para poder
// consultar su informacion, usando el plugin DataTable.
function tablaConsultarPoliza()
{
    tablaConsultarPoliza = $('#consultarPolizasTabla').DataTable({
        "language": {
            "zeroRecords": "No se encontraron polizas",
            "sSearch": "Buscar",
        },
        "ajax": "scripts/tablas/tablaConsultarPoliza.php",
        "scrollY": 370,
        "bPaginate": false,
        "dom": 'Bft',
        "order": [[5, "desc"]],
        "lengthMenu": [[5, 10], [5, 10]],
        "aoColumnDefs": [
            { 'bSortable': false, 'aTargets': [0] },
            { 'bSortable': false, 'aTargets': [3] },
            { 'bSortable': false, 'aTargets': [6] },
            { 'bSortable': false, 'aTargets': [7] }
        ],
        "columns": [
            {data:"nro", title: "Nro", width:"8%"}, 
            {data:"tomador", title: "Tomador", width:"18%"}, 
            {data:"vehiculo", title: "Vehiculo", width:"12%"}, 
            {data:"vigPol", title: "Vigencia", width:"16%"}, 
            {data:"anulac", title: "Anulacion", width:"4%"}, 
            {data:"emision", title: "Emision", width:"12%"}, 
            {data:"estado", title: "Estado", width:"8%"}, 
            {data:"info", title: "Info", width:"19%"}, 
            {data:"boton", title: "Ver", width:"2%"}],
        "buttons": [
            {
                text: `<select id="selectVistaListPoliza">
                    <option value="0">Todas</option>
                    <option value="1">Espera</option>
                    <option value="2">Vigentes</option>
                    <option value="3">Vencidas</option>
                    <option value="4">Renovadas</option>
                    <option value="5">Anuladas</option>
                </select>`,
                className: 'browser-default'
            }
        ],
        // 0 - Vigente
    });
}

```

```

// 1 - Vencida
// 2 - Anulada
// 3 - Vigente-deuda
// 4 - Renovada
// 5 - Espera

"createdRow": function(row,data,index){
    estado = data['estado'];

    if(estado=='Vigente'){
        $('td', row).eq(6).css({'color':'#333', 'font-
weight':'600'});
    }else if(estado=='Vencida'){
        $('td', row).eq(6).css({'color':'orange', 'font-
weight':'600'});
    }else if(estado=='Anulada'){
        $('td', row).eq(6).css({'color':'red', 'font-weight':'600'});
    }else if(estado=='Vigente-deuda'){
        $('td', row).eq(6).css({'color':'#d7d700', 'font-
weight':'600'});
    }else if(estado=='Renovada'){
        $('td', row).eq(6).css({'color':'#2e4eadb8', 'font-
weight':'600'});
    }else if(estado=='Espera'){
        $('td', row).eq(6).css('color','#333');
    }

    // Mostramos la informacion segun la opcion elegida.
    $('#selectVistaListPoliza").on('change', function(){
        val = $(this).val();

        function mostrar(est){
            $('td', row).css('display', 'revert');
            if(data['estado']!=est){
                $('td', row).css('display', 'none');
            }
        }

        if(val==0){
            $('td', row).css('display', 'revert');
        }else if(val==1){
            mostrar('Espera')
        }else if(val==2){
            mostrar('Vigente')
        }else if(val==3){
            mostrar('Vencida')
        }else if(val==4){
            mostrar('Renovada')
        }else if(val==5){
            mostrar('Anulada')
        }else if(val==6){

        }
    })
}

```

```

        }

    });

    return tablaConsultarPoliza;
}

// Función donde generamos la tabla con el listado de polizas vencidas, para
// poder renovarlas, usando el plugin DataTabes.
function tablaRenovarPoliza()
{
    tablaRenovarPoliza = $('#renovarPolizaTabla').DataTable({
        "language": {
            "zeroRecords": "No hay polizas vencidas registradas",
            "sSearch": "Buscar",
        },
        "ajax": "scripts/tablas/tablaRenovarPoliza.php",
        "scrollY": 370,
        "scrollCollapse": true,
        "dom": 'ft',
        "order": [[3, "desc"]],
        "aoColumnDefs": [
            { 'bSortable': false, 'aTargets': [0] },
            { 'bSortable': false, 'aTargets': [1] },
            { 'bSortable': false, 'aTargets': [2] },
            { 'bSortable': false, 'aTargets': [3] },
            { 'bSortable': false, 'aTargets': [4] }
        ],
        "columns": [
            {data:"nro", title: "Nro", width:"20%"}, 
            {data:"tomador", title: "Tomador", width:"20%"}, 
            {data:"vigPol", title: "Vigencia", width:"20%"}, 
            {data:"emision", title: "Emision", width:"15%"}, 
            {data:"anulac", title: "Anulacion", width:"14%"}, 
            {data:"renovar", title: "Renovar", width:"11%"}]
    });
}

return tablaRenovarPoliza;
}

// Función donde tendremos todas las funcionalidades extras para los
// formularios y elementos relacionados a las polizas.
function funcionalidadesPoliza()
{
    $('input, select').addClass('browser-default');

    //Establecemos los tabs a todos los campos con la clase tabs (requerido p
    or materialize para funcionar).
    $('.tabs').tabs();

    // Declaramos como modal todo campo con la clase .modal.
    $('.modal').modal({

```

```

       -dismissible: false
    });

    //Desabilitamos los campos donde se muestra la informacion del tomador en
    emitir poliza.
    $("#nombreDatos,#docDatos,#tipDatos,#nacDatos,#proDatos,#locDatos,#cpDatos,#calleDatos,#genDatos,#telDatos,#corrDatos,#feNacDatos").attr('disabled', 'disabled').removeClass('browser-default');

    //Si la hora en la que se registra la poliza es superior a las 12 (medio
    dia), el dia de inicio de vigencia MINIMO a seleccionar es el siguiente al
    presente.
    var dt = new Date();
    var hor = dt.getHours();
    if(hor >= 12){
        var diaMinimo = 1;
    }else{
        var diaMinimo = 0;
    }

    //Generamos el numero de poliza.
    var letras = ['A','B','C','D','E','F','G','H','I','J','K','L','M','N','O',
    'P','Q','R','S','T','U','V','W','X','Y','Z'];
    var l = (letras[Math.round(Math.random()*26)]+letras[Math.round(Math.random()*26)]);
    var num = Math.floor(1e9 + (Math.random() * 9e14));
    var numPol = l+num;
    $('#nroPoliza').html(numPol);

    //Usamos Datepicker en los campos "vig_ini" y "vig_fin" estableciendo el
    rango minimo de 4 meses entre fecha y fecha.
    $("#vig_fin").prop({'disabled': true, 'placeholder': 'Seleccione una opcion
    anterior'});
    $("#tipVig").prop('disabled', true);
    $("#vig_ini, #vig_fin").prop('readonly', true).css('background', 'transpa
    rent');

    $("#vig_ini, #tipVig, #vig_fin").on('mouseover', function(){
        $("#vig_ini").prop('readonly', true);
        $("#vig_fin").prop('disabled', true);
    });

    $(function(){
        $("#vig_ini").datepicker({
            dayNames: ["Domingo","Lunes","Martes","Miercoles","Jueves","Viern
            es","Sabado"],
            dayNamesMin: ["Do","Lu","Ma","Mi","Ju","Vi","Sa"],
            monthNames: ["Enero","Febrero","Marzo","Abril","Mayo","Junio","Ju
            lio","Agosto","Septiembre","Octubre","Noviembre","Diciembre"],
            monthNamesShort: ["Ene","Feb","Mar","Abr","May","Jun","Jul","Ago
            ","Sep","Oct","Nov","Dic"],
            dateFormat: 'dd/mm/yy',
            minDate: diaMinimo,
    });
}

```

```

        altField: "#vigenciaInicioInput",
        altFormat: "DD d MM yy",

        onSelect: function (fecha) {

            $("#tipVig").val(0);
            $("#tipVig").prop('disabled', false);

            $("#vig_fin").val('');

            //Segun la cantidad de meses que elijamos se establecen las fechas con el rango automaticamente.
            $("#tipVig").on('change', function(){

                var opcion = $("#tipVig").val();

                fecha = $("#vig_ini").datepicker('getDate');

                //Cada vez que seleccionemos la cant de meses que dura la vigencia, seteamos el input de fin de vigencia con la cantidad de meses.
                for(let i=1; i<=12; i++){
                    if(opcion==i){
                        fecha.setMonth(fecha.getMonth() + i);
                    }
                }
                $("#vig_fin").datepicker("setDate", fecha);

                //Al modificar el input oculto, mostramos el cuadro de informacion de vigencia y asignamos la info a los span.
                $(".divResumenVigPoliza").css('display', 'block');
                var inicioVigInput = $("#vigenciaInicioInput").val();
                var finVigInput = $("#vigenciaFinInput").val();
                $("#vigenciaInicioSpan").html(inicioVigInput);
                $("#vigenciaFinSpan").html(finVigInput);
            });

        });

        $("#vig_fin").datepicker({
            dayNames: ["Domingo", "Lunes", "Martes", "Miercoles", "Jueves", "Viernes", "Sabado"],
            dayNamesMin: ["Do", "Lu", "Ma", "Mi", "Ju", "Vi", "Sa"],
            monthNames: ["Enero", "Febrero", "Marzo", "Abril", "Mayo", "Junio", "Julio", "Agosto", "Septiembre", "Octubre", "Noviembre", "Diciembre"],
            monthNamesShort: ["Ene", "Feb", "Mar", "Abr", "May", "Jun", "Jul", "Ago", "Sep", "Oct", "Nov", "Dic"],
            dateFormat: 'dd/mm/yy',

            altField: "#vigenciaFinInput",
            altFormat: 'DD d MM yy',

            onClose: function (fecha) {
                $("#vig_ini").datepicker("option", "maxDate", fecha);
            }
        });
    
```

```

        });
    });

    //Se aplica la libreria select2 a los inputs requeridos en emision de poliza.
    $('#select_cliente, #select_marca, #select_modelo1, #codigoPostal, #vehiculoLocalidad').select2({width: "100%"});

    // Condicion para ingresar cantidad de kms correctamente.
    $("#ceroKm").on('click', function(){
        if($("#ceroKm").is(':checked')){
            $("#cant_kilometro").prop('disabled', true).css('background', '#e6e6fa');
            $("#cant_kilometro").val('0');
        }else{
            $("#cant_kilometro").prop('disabled', false).css('background', 'transparent');
            $("#cant_kilometro").val('');
            $("#cant_kilometro").focus();
        }
    });

    // Mostrar campos correctamente de cantidad de pasajeros.
    $("#pasajeros, #asientos").prop('min', 1);
    $("#selectTipoUso").on('change', function(){
        if ($("#selectTipoUso").val() == 14){
            $("#pasajeros, #asientos").val('');
            $("#pasajeros, #asientos").prop('max', 15);
        }else{
            $("#pasajeros").prop('max', 60);
        }
    });

    $("#atrasTomador").off('click');
    $("#atrasTomador").on('click', function(){
        $('.tabs').tabs('select','test1');
        $(".btnDivTomador").html('');
    });

    $("#atrasVehiculo").on('click');
    $("#atrasVehiculo").on('click', function(e){
        $('.tabs').tabs('select','test2');
        $(".btnDivVehiculo").html('');
    });

    $("#atrasCobertura").on('click');
    $("#atrasCobertura").on('click', function(e){
        $('.tabs').tabs('select','test3');
        $(".btnDivCobertura").html('');
    });

    $("#atrasPago").on('click');
    $("#atrasPago").on('click', function(e){
        $('.tabs').tabs('select','test4');
    });

```

```

        $(".btnDivPago").html('');
    });

    // Mostramos un cuadro de confirmación antes de cancelar el proceso de
    // emisión de poliza.
    $('#cancelar_datos').on('click', function(e){
        e.preventDefault();
        alertify.confirm('Emision de Poliza', '¿Cancelar proceso de emision d
e poliza?',
            function(){$('#contenidoC').load('vista/poliza/elementosPoliza/formu
lario_poliza.php');},
            function(){}
        ).set(configuracionAlert('okCancel','Si')));
    });

    // Aplicamos estilos al select buscador de consultar polizas.
    estilos = {
        'margin': '7px 0px 0px',
        'height': '26px',
        'padding': '7px 0px 0px',
        'border-radius': '4px',
        'width': '9rem',
        'font-size': '14px',
        'border': '1px solid #ccc'
    }

    $('#selectVistaListPoliza').parent().parent().css(estilos);
}

```

➔ 4.3.3.7 fichero “siniestros.js”

Este fichero contiene todas las funciones que realizan las acciones relacionadas al frontend o vista de los siniestros en el sistema.

```
// Cargamos el DOM a través de jQuery.
jQuery(document).ready(function($) {

    // Llamamos a varias funciones directamente o a través de acciones.
    funcionalidadesSiniestro();
    validarSiniestro();

    $("#cancelarCambiarEstadoSin").on('click', function(){
        $("#modalCambiarEstado").modal('close');
    });

    $("#btnVolverModificarSin").on('click', function(){
        $('#contenidoC').load('vista/siniestro/elementosSiniestro/administrarSiniestro.php', function(){
            tablaAdministrarSiniestro();
        });
    });

    $(document).off('click', '.ModRestoDatos');
    $(document).on('click', '.ModRestoDatos', function(){
        var id = $(this).attr('value');
        traerDatosSin(id);
    });

    $(document).off('click', '#verSiniestro');
    $(document).on('click', '#verSiniestro', function(){
        var id = $(this).attr('value');
        mostrarDatosSiniestro(id);
    });

    $(document).off('click', '.modImagenSin');
    $(document).on('click', '.modImagenSin', function(){
        var idSin = $(this).attr('value');
        $("#modalAdminImgSin").modal('open');
        mostrarImgSiniestro(idSin);
    });

    $(document).off('click', '.eliminarImg');
    $(document).on('click', '.eliminarImg', function(){
        var idImg = $(this).attr('value');
        eliminarImgSin(idImg);
    });

});

});
```

```

// Función para validar los datos ingresados en todos los formularios a lo largo de la emisión de un siniestro, utilizando el plugin jquery validate.
function validarSiniestro()
{
    var valFecha = /^(\d{1,2})\/(\d{1,2})\/(\d{2,4})$/;
    $.validator.addMethod("validarFecha", function(value, element) {
        return this.optional(element) || valFecha.test(value) ;
    });
    $.validator.addMethod("valFechaCorrecta", function(value, element) {
        return this.optional(element) || valFechaCorrecta(value)==true
    });

    var valHora = /^([0-2][0-3])|([0-1][0-9]):[0-5][0-9]$/;
    $.validator.addMethod("validarHora", function(value, element) {
        return this.optional(element) || valHora.test(value);
    });
    $.validator.addMethod("valHoraCorrecta", function(value, element) {
        return this.optional(element) || validarHoraCorrecta(value)==true
    });

    // Función interna de validación para corroborar que la fecha de ocurrencia ingresada sea correcta.
    function valFechaCorrecta(fecha){
        var fechaIngresada = moment($("#fechaOcurrencia").val(), 'DD/MM/YYYY').format('MM/DD/YYYY')
        var fechaActual = moment(new Date()).format("MM/DD/YYYY");
        if(fechaIngresada>fechaActual){
            return false;
        }else if(fechaIngresada<=fechaActual){
            return true;
        }
    }

    // Función interna de validación, que corrobora que la hora de ocurrencia ingresada sea correcta.
    function validarHoraCorrecta(hora){

        var horaUsuario = moment(hora, 'HH:mm').format('HH:mm');
        var horaActualMenosDiez = moment(new Date()).subtract(10, 'minutes').format("HH:mm"); //Horario actual menos 10 minutos

        //Obtenemos la fecha ingresada y la fecha actual.
        var fechaIngresada = moment($("#fechaOcurrencia").val(), 'DD/MM/YYYY').format('MM/DD/YYYY')
        var fechaActual = moment(new Date()).format("MM/DD/YYYY");

        console.log('')

        //Si el dia en que ocurrio el siniestro es el presente, la hora máxima a seleccionar es 10 minutos antes a la actual.
        if(fechaIngresada==fechaActual){
            if(horaUsuario>horaActualMenosDiez){
                return false;
            }else if(horaUsuario<horaActualMenosDiez){

```

```

        return true;
    }else if(horaUsuario==horaActualMenosDiez){
        return true;
    }
}else if(fechaIngresada>fechaActual){
    return false;
}else if(fechaIngresada<fechaActual){
    return true;
}
}

$("#divSiniestro").validate({
    rules:{
        "fechaOcurrencia": {required:true, validarFecha:true, valFechaCorrecta:true},
        "horaOcurrencia": {required:true, validarHora:true, valHoraCorrecta:true},
        "LesMuer": {required:true},
        "selectPolizas": {required:true},
        "TipoSin": {required:true}
    },
    messages: {
        "fechaOcurrencia": {required:"Debe seleccionar la Fecha de Ocurrencia", validarFecha:"El formato de Fecha de Ocurrencia es incorrecto", valFechaCorrecta: "La Fecha de Ocurrencia ingresada no esta permitida"}, 
        "horaOcurrencia": {required:"Debe seleccionar la Hora de Ocurrencia", validarHora:"El formato de Hora de Ocurrencia es incorrecto" , valHoraCorrecta: "La hora de Ocurrencia ingresada no esta permitida"}, 
        "LesMuer": {required:"Debe seleccionar si Hubo Lesion/Muerte"}, 
        "selectPolizas": {required:"Debe seleccionar una Poliza"}, 
        "TipoSin": {required:"Debe seleccionar un Tipo de Siniestro"}
    },
    highlight: function(element) {
        if($(element).hasClass('select2')){
            $(element).siblings('span').find('.selection').find('.select2-selection').addClass('claseError');
        }else{
            $(element).addClass('claseError');
        }
    },
    unhighlight: function(element) {
        if($(element).hasClass('select2')){
            $(element).siblings('span').find('.selection').find('.select2-selection').removeClass('claseError');
        }else{
            $(element).removeClass('claseError');
        }
    },
    errorElement : 'li',
    errorLabelContainer: '#mensajeSiniestro',
})

```

```

        submitHandler: function(form){
            validarDenunciante();
        }
    });

// Función para validar los datos del denunciante al emitir un siniestro,
usando el plugin jquery validate.
function validarDenunciante()
{
    $('.tabs').tabs('select','test2');
    boton = '<button type="submit" class="btn btn-
flat green" id="siguienteDenun"> siguiente <i class="fas fa-caret-
right"></i> </button>';
    $(".btnSigDen").html(boton);

    $.validator.addMethod("validarSelectDenAs", function(value, element) {
        return this.optional(element) || value==1 || value==2;
    });

    var valNombre = /^[A-Záéíóúñáéíóúñ]{0}?[A-Zá-
Záéíóñáéíóñ']+[\s])+([A-Záéíóñáéíóñ]{0}?[A-Zá-
Záéíóñáéíóñ])+[\s]?([A-Záéíóñáéíóñ]{0}?[A-Zá-Záéíóñáéíóñ'\'])?$/;
    $.validator.addMethod("validarNombre", function(value, element) {
        return this.optional(element) || valNombre.test(value) ;
    });

    var valDNI = /^\d{8}(?:[-\s]\d{4})?$/;
    $.validator.addMethod("valDNI", function(value, element) {
        return this.optional(element) || valDNI.test(value) ;
    });

    var email = /^[^\w.%+]{1,64}@(?:[A-Z0-9-]{1,63}\.){1,125}[A-Z]{2,63}$/i;
    $.validator.addMethod('email', function(value, element){
        return this.optional(element) || email.test(value);
    });

    $("#divDenunciante").validate({
        rules:{
            "selectDenunAseg": {required:true, validarSelectDenAs:true},
            "nombreDenunciante": {required:true, validarNombre:true},
            "dniDenunciante": {required:true, valDNI:true},
            "domicDenunciante": {required:true},
            "telefonoDenunciante": {required:true, rangelength:[10,15]},
            "emailDenunciante": {email: true}
        },
        messages: {
            "selectDenunAseg": {required:"Debe seleccionar si el Denuncian-
te es la misma persona que el Asegurado", validarSelectDenAs:"La opcion selec-
cionada no es permitida"},

            "nombreDenunciante": {required:"Debe ingresar el Nombre del Deun-
ciante", validarNombre:"El Nombre del Denunciante es incorrecto"},

        }
    });
}

```

```

        "dniDenunciante":      {required:"Debe ingresar el DNI del Deunciante", valDNI:"El DNI del Denunciante es incorrecto"},  

        "domicDenunciante":    {required:"Debe ingresar el Domicilio del Denunciante"},  

        "telefonoDenunciante": {required:"Debe ingresar un Telefono del Denunciante", rangelength:"El Telefono del Denunciante ingresado no es valido"}  

,  

        "emailDenunciante":   {email: "Debe ingresar un E-Mail correcto"}  

},  

highlight: function(element) {  

    $(element).addClass('claseError');  

},  

unhighlight: function(element) {  

    $(element).removeClass('claseError');  

},  

errorElement : 'li',  

errorLabelContainer: '#mensajeDenunciante',  

submitHandler: function(form){  

    validarAccidente();  

}  

});  

}  

// Función para validar los datos del accidente al emitir un siniestro,  

usando jquery validate.  

function validarAccidente()  

{  

    $('.tabs').tabs('select','test3');  

    valor = 0;  

    if($('#TipoSin').val() == 5){  

        valor = 5;  

        boton = '<button type="submit" class="btn btn-flat green" id="siguienteAcc"> siguiente <i class="fas fa-caret-right"></i> </button>';  

    }else{  

        valor = 10;  

        boton = `<button type="submit" class="btn btn-flat green" id="cargarSiniestro"> grabar siniestro <i class="fas fa-save"></i> </button>  

            <button type="button" class="btn btn-flat red" id="cancelarSin" onclick="cancelarProceso()"> cancelar <i class="fas fa-times"></i> </button>`;  

    }  

    $(".btnAcc").html(boton);  

    $('#provinciaAcc, #localidadAcc').change(function(){  

        if ($this).val() != ""  

        {  

            $(this).valid();  

        }  

    });
}

```

```

$("#divAccidente").validate({
    rules:{
        "descripAcc": {required:true, minlength:[25]},
        "provinciaAcc": {required:true},
        "localidadAcc": {required:true},
        "calleAcc": {minlength:[2]},
        "alturaAcc": {minlength:[2]}
    },
    messages: {
        "descripAcc": {required:"Debe ingresar la Descripcion", minlength:"La Descripcion del Accidente es muy corta"},
        "provinciaAcc": {required:"Debe seleccionar una Provincia"},
        "localidadAcc": {required:"Debe seleccionar una Localidad"},
        "calleAcc": {minlength:"La Calle ingresada es muy corta"},
        "alturaAcc": {minlength:"La Altura ingresada es muy corta"}
    },
    highlight: function(element) {
        if($(element).hasClass('select2')){
            $(element).siblings('span').find('.selection').find('.select2-selection').addClass('claseError');
        }else{
            $(element).addClass('claseError');
        }
    },
    unhighlight: function(element) {
        if($(element).hasClass('select2')){
            $(element).siblings('span').find('.selection').find('.select2-selection').removeClass('claseError');
        }else{
            $(element).removeClass('claseError');
        }
    },
    errorElement : 'li',
    errorLabelContainer: '#mensajeAccidente',

    submitHandler: function(form){
        if(valor==5){
            validarReclamante();
        }else if(valor==10){
            grabarSiniestro();
        }
    }
});
}

// Función para validar los datos del reclamante al emitir un siniestro,
// usando el plugin jquery validate.
function validarReclamante()
{
    $('.tabs').tabs('select','test4');
}

```

```

boton = '<button type="submit" class="btn btn-flat green" id="siguienteRec"> siguiente <i class="fas fa-caret-right"></i> </button>';
$(".btnSigRec").html(boton);

var valNombre = /^[A-ZáéíóúñÁÉÍÓÚÑÁÉÍÓÚÑ][\s]+([A-ZáéíóúñÁÉÍÓÚÑÁÉÍÓÚÑ][\s]+)?[A-ZáéíóúñÁÉÍÓÚÑÁÉÍÓÚÑ][\s]+)?$/;
$.validator.addMethod("validarNombre", function(value, element) {
    return this.optional(element) || valNombre.test(value) ;
});

var valDNI = /^\d{8}(-\d{4})?$/;
$.validator.addMethod("valDNI", function(value, element) {
    return this.optional(element) || valDNI.test(value) ;
});

$("#divReclamante").validate({
    rules:{
        "nombreReclamante": {required:true, validarNombre:true},
        "domicReclamante": {required:true},
        "telefonoReclamante": {required:true},
        "dniReclamante": {required:true, valDNI:true}
    },
    messages: {
        "nombreReclamante": {required:"Debe ingresar el Nombre del Reclamante", validarNombre:"El Nombre del Reclamante es incorrecto"},
        "domicReclamante": {required:"Debe ingresar el Domicilio del Reclamante"},
        "telefonoReclamante": {required:"Debe ingresar el Telefono del Reclamante"},
        "dniReclamante": {required:"Debe ingresar el DNI del Reclamante", valDNI:"El DNI del Reclamante es incorrecto"}
    },
    highlight: function(element) {
        $(element).addClass('claseError');
    },
    unhighlight: function(element) {
        $(element).removeClass('claseError');
    },
    errorElement : 'li',
    errorLabelContainer: '#mensajeReclamante',

    submitHandler: function(form){
        validarConductor();
    }
});
}

// Función para validar los datos del conductor/tercero al emitir un nuevo siniestro, usando el plugin jquery validate.
function validarConductor()
{

```

```

$('.tabs').tabs('select','test5');
boton = `<button type="submit" class="btn btn-flat green" id="cargarSiniestro"> grabar siniestro <i class="fas fa-save"></i> </button>
          <button type="button" class="btn btn-flat red" id="cancelarSin" onclick="cancelarProceso()"> cancelar <i class="fas fa-times"></i> </button>`;
$(".btnSinConductor").html(boton);

$.validator.addMethod("validarSelectCondRec", function(value, element) {
    return this.optional(element) || value==1 || value==2;
});

var valNombre = /^[A-ZáéíóñÁÉÍÓÑáéíóñ]{0}?[A-ZáéíóñÁÉÍÓÑáéíóñ\']+[\\s]+([A-ZáéíóñÁÉÍÓÑáéíóñ]{0}?[A-ZáéíóñÁÉÍÓÑáéíóñ\']+[\\s]?([A-ZáéíóñÁÉÍÓÑáéíóñ]{0}?[A-ZáéíóñÁÉÍÓÑáéíóñ\']))?$/;
$.validator.addMethod("validarNombre", function(value, element) {
    return this.optional(element) || valNombre.test(value) ;
});

var valDNI = /^\d{8}(?:[-\s]\d{4})?$/;
$.validator.addMethod("valDNI", function(value, element) {
    return this.optional(element) || valDNI.test(value) ;
});

$("#divConductor").validate({
    rules:{
        "selectCondRec": {required:true, validarSelectCondRec:true},
        "nombreConductor": {required:true, validarNombre:true},
        "domicConductor": {required:true},
        "telefonoConductor": {required:true},
        "dniConductor": {required:true, valDNI:true}
    },
    messages: {
        "selectCondRec": {required:"Debe seleccionar si el Conductor es la misma persona que el Reclamante", validarSelectCondRec:"La opcion seleccionada no es permitida"},
        "nombreConductor": {required:"Debe ingresar el Nombre del Conductor", validarNombre:"El Nombre del Conductor ingresado es incorrecto"},
        "domicConductor": {required:"Debe ingresar el Domicilio del Conductor"},
        "telefonoConductor": {required:"Debe ingresar el Telefono del Conductor"},
        "dniConductor": {required:"Debe ingresar el DNI del Conductor", valDNI:"El DNI del Conductor ingresado es incorrecto"}
    },
    highlight: function(element) {
        $(element).addClass('claseError');
    },
    unhighlight: function(element) {
        $(element).removeClass('claseError');
    },
    errorElement : 'li',

```

```

        errorLabelContainer: '#mensajeConductor',
        submitHandler: function(form){
            grabarSiniestro();
        }
    });
}

// Función para validar todos los campos de los formularios al momento de
// modificar la información de un siniestro, usando el plugin jquery validate.
function validarModSiniestro(idSiniestro, fechaHoraMaxima)
{
    $.validator.setDefaults({
        ignore: []
    });

    //siniestro
    var valFecha = /^(\d{1,2})/(\d{1,2})/(\d{2,4})$/;
    $.validator.addMethod("validarFecha", function(value, element) {
        return this.optional(element) || valFecha.test(value) ;
    });
    $.validator.addMethod("valFechaCorrecta", function(value, element) {
        return this.optional(element) || valFechaCorrecta(value)==true
    });
    var valHora = /^([0-2][0-3])|([0-1][0-9]):[0-5][0-9]$/;
    $.validator.addMethod("validarHora", function(value, element) {
        return this.optional(element) || valHora.test(value);
    });
    $.validator.addMethod("valHoraCorrecta", function(value, element) {
        return this.optional(element) || validarHoraCorrecta(value)==true
    });
    // Función interna para validar la fecha de ocurrencia.
    function valFechaCorrecta(fecha){
        var fechaIngresada = moment($("#fechaOcurrencia").val(), 'DD/MM/YYYY')
        .format('MM/DD/YYYY')
        var fechaMaxPermitida = moment(fechaHoraMaxima, 'DD/MM/YYYY HH:mm').format('MM/DD/YYYY'); //Fecha de registro
        if(fechaIngresada>fechaMaxPermitida){
            return false;
        }else if(fechaIngresada<=fechaMaxPermitida){
            return true;
        }
    }
    // Función interna para validar la hora.
    function validarHoraCorrecta(hora){

        var fechaMaxPermitida = moment(fechaHoraMaxima, 'DD/MM/YYYY HH:mm').format('MM/DD/YYYY'); //Fecha de registro
        var horaMaxPermitida = moment(fechaHoraMaxima, 'DD/MM/YYYY HH:mm').subtract(10, 'minutes').format("HH:mm");//Hora de registro menos 10 minutos

        var horaIngresada = moment(hora, 'HH:mm').format('HH:mm');
    }
}

```

```

//Obtenemos la fecha ingresada y la fecha actual
var fechaIngresada = moment($("#fechaOcurrencia").val(), 'DD/MM/YYYY')
.format('MM/DD/YYYY')
var fechaActual = moment(new Date()).format("MM/DD/YYYY");

//Si el dia en que ocurrio el siniestro es el presente, la hora maxima a seleccionar es 10 minutos antes a la actual
if(fechaIngresada>=fechaMaxPermitida){
    if(horaIngresada>horaMaxPermitida){
        return false;
    }else if(horaIngresada<horaMaxPermitida){
        return true;
    }else if(horaIngresada==horaMaxPermitida){
        return true;
    }
}else if(fechaIngresada<fechaMaxPermitida){
    return true;
}

//Denunciante
$.validator.addMethod("validarSelectDenAs", function(value, element) {
    return this.optional(element) || value==1 || value==2;
});

var valNombre = /^[A-Záéíóúñáéíóúñ]{0}?[A-Záéíóñáéíóúñ'\']+[\s]+([A-Záéíóúñáéíóúñ]{0}?[A-Záéíóñáéíóúñ'\']+)[\s]?([A-Záéíóúñáéíóúñ]{0}?[A-Záéíóúñáéíóúñ'\'])?$/;
$.validator.addMethod("validarNombre", function(value, element) {
    return this.optional(element) || valNombre.test(value) ;
});

var valDNI = /^\d{8}(?:[-\s]\d{4})?$/;
$.validator.addMethod("valDNI", function(value, element) {
    return this.optional(element) || valDNI.test(value) ;
});

var email = /^[^\w.%+]{1,64}@(?:[A-Z0-9-]{1,63}\.){1,125}[A-Z]{2,63}$/i;
$.validator.addMethod('email', function(value, element){
    return this.optional(element) || email.test(value);
});

$.validator.addMethod("validarSelectCondRec", function(value, element) {
    return this.optional(element) || value==1 || value==2;
});

$("#modificarSiniestro").validate({
    rules:{
        "fechaOcurrencia": {required:true, validarFecha:true, valFechaCorrecta:true},
        "horaOcurrencia": {required:true, validarHora:true, valHoraCorrecta:true},
        "LesMuer": {required:true},
        "selectPolizas": {required:true},
    }
});

```

```

    "TipoSin":           {required:true},

    "selectDenunAseg":   {required:true, validarSelectDenAs:true},
    "nombreDenunciante": {required:true, validarNombre:true},
    "dniDenunciante":   {required:true, valDNI:true},
    "domicDenunciante": {required:true},
    "telefonoDenunciante": {required:true, rangelength:[10,15]},
    "emailDenunciante":  {email: true},

    "decripAcc":         {required:true, minlength:[25]},
    "provinciaAcc":      {required:true},
    "localidadAcc":      {required:true},
    "calleAcc":          {minlength:[2]},
    "alturaAcc":         {minlength:[2]},

    "nombreReclamante":  {required:true, validarNombre:true},
    "domicReclamante":   {required:true},
    "telefonoReclamante": {required:true},
    "dniReclamante":     {required:true, valDNI:true},

    "selectCondRec":     {required:true, validarSelectCondRec:true},
    "nombreConductor":   {required:true, validarNombre:true},
    "domicConductor":    {required:true},
    "telefonoConductor": {required:true},
    "dniConductor":      {required:true, valDNI:true}
  },
  messages: {
    "fechaOcurrencia": {required:"Debe seleccionar la Fecha de Ocurrencia", validarFecha:"El formato de Fecha de Ocurrencia es incorrecto", valFechaCorrecta: "La Fecha de Ocurrencia ingresada no esta permitida"},

    "horaOcurrencia": {required:"Debe seleccionar la Hora de Ocurrencia", validarHora:"El formato de Hora de Ocurrencia es incorrecto", valHoraCorrecta: "La hora de Ocurrencia ingresada no esta permitida"},

    "LesMuer": {required:"Debe seleccionar si Hubo Lesion/Muerte"},

    "selectPolizas": {required:"Debe seleccionar una Poliza"},

    "TipoSin": {required:"Debe seleccionar un Tipo de Siniestro"},

    "selectDenunAseg": {required:"Debe seleccionar si el Denunciante es la misma persona que el Asegurado", validarSelectDenAs:"La opcion seleccionada no es permitida"},

    "nombreDenunciante": {required:"Debe ingresar el Nombre del Denunciante", validarNombre:"El Nombre del Denunciante es incorrecto"},

    "dniDenunciante": {required:"Debe ingresar el DNI del Denunciante", valDNI:"El DNI del Denunciante es incorrecto"},

    "domicDenunciante": {required:"Debe ingresar el Domicilio del Denunciante"},

    "telefonoDenunciante": {required:"Debe ingresar un Telefono del Denunciante", rangelength:"El Telefono del Denunciante ingresado no es valido"},

    "emailDenunciante": {email: "Debe ingresar un Email correcto"},

  }
}

```

```

        "decripAcc":      {required:"Debe ingresar la Descripcion", minlength:"La Descripcion del Accidente es muy corta"},  

        "provinciaAcc":   {required:"Debe seleccionar una Provincia"},  

        "localidadAcc":   {required:"Debe seleccionar una Localidad"},  

        "calleAcc":        {minlength:"La Calle ingresada es muy corta"},  

        "alturaAcc":       {minlength:"La Altura ingresada es muy corta"},  

        "nombreReclamante": {required:"Debe ingresar el Nombre del Reclamante", validarNombre:"El Nombre del Reclamante es incorrecto"},  

        "domicReclamante": {required:"Debe ingresar el Domicilio del Reclamante"},  

        "telefonoReclamante": {required:"Debe ingresar el Telefono del Reclamante"},  

        "dniReclamante": {required:"Debe ingresar el DNI del Reclamante", valDNI:"El DNI del Reclamante es incorrecto"},  

        "selectCondRec": {required:"Debe seleccionar si el Conductor es la misma persona que el Reclamante", validarSelectCondRec:"La opcion seleccionada no es permitida"},  

        "nombreConductor": {required:"Debe ingresar el Nombre del Conductor", validarNombre:"El Nombre del Conductor ingresado es incorrecto"},  

        "domicConductor": {required:"Debe ingresar el Domicilio del Conductor"},  

        "telefonoConductor": {required:"Debe ingresar el Telefono del Conductor"},  

        "dniConductor": {required:"Debe ingresar el DNI del Conductor", valDNI:"El DNI del Conductor ingresado es incorrecto"}  

    },  

    highlight: function(element){  

        if($(element).hasClass('validarSin')){  

            $('.tabs').tabs('select','test1');  

        }else if($(element).hasClass('validarDen')){  

            $('.tabs').tabs('select','test2');  

        }else if($(element).hasClass('validarAcc')){  

            $('.tabs').tabs('select','test3');  

        }else if($(element).hasClass('validarRec')){  

            $('.tabs').tabs('select','test4');  

        }else if($(element).hasClass('validarCond')){  

            $('.tabs').tabs('select','test5');  

        }  

        if($(element).hasClass('select2')){  

            $(element).siblings('span').find('.selection').find('.select2-selection').addClass('claseError');  

        }else{  

            $(element).addClass('claseError');  

        }  

    },  

    unhighlight: function(element) {  

        if($(element).hasClass('select2')){  

            $(element).siblings('span').find('.selection').find('.select2-selection').removeClass('claseError');  

        }else{  


```

```

        $(element).removeClass('claseError');
    }
},
errorElement : 'li',
errorLabelContainer: '#mensajeModSin',
submitHandler: function(form){
    modificarDatosSin(idSiniestro);
}
});
}

// Función para traer y mostrar los datos del denunciante, en caso de querer
mostrarlos al indicar que es la misma persona que el tomador.
function traerDatosDenunciante(id)
{
    var idPol = id;
    $.ajax({
        type:"post",
        data:{param:30000, data:idPol},
        url: "scripts/aseguradora.php",
        dataType:'json',
        success: function(r){
            if(r.data['success']==true)
            {
                $('#nombreDenunciante').val(r.data['nombre']);
                $('#telefonoDenunciante').val(r.data['telefono']);
                $('#dniDenunciante').val(r.data['documento']);
                $('#domicDenunciante').val(r.data['domicilio']);
                $('#emailDenunciante').val(r.data['correo']);
            }
        }
    });
}

// Función para grabar un nuevo siniestro.
function grabarSiniestro()
{
    if($("#selectDenunAseg").val()==1){
        var tomadDenunciante = 'SI';
    }else if($("#selectDenunAseg").val()==2){
        var tomadDenunciante = 'NO';
    }

    if($("#selectCondRec").val()==1){
        var condReclamante = 'SI';
    }else if($("#selectCondRec").val()==2){
        var condReclamante = 'NO';
    }

    if($('#TipoSin').val() != 5){
        var tomadReclamante = 'SI';
    }else{

```

```

var tomadReclamante = 'NO';
}

var datosSiniestro =
[
    $('#nroSin').html(),
    $('#fechaOcurrencia').val(),
    $('#horaOcurrencia').val(),
    $('#selectPolizas').val(),
    $('#LesMuer').val(),
    $('#TipoSin').val(),

    $('#nombreDenunciante').val(),
    $('#telefonoDenunciante').val(),
    $('#dniDenunciante').val(),
    $('#emailDenunciante').val(),
    $('#domicDenunciante').val(),

    $('#decripAcc').val(),
    $('#calleAcc').val(),
    $('#alturaAcc').val(),
    $('#localidadAcc').val(),
    $('#provinciaAcc').val(),

    $('#nombreReclamante').val(),
    $('#domicReclamante').val(),
    $('#telefonoReclamante').val(),
    $('#dniReclamante').val(),

    $('#nombreConductor').val(),
    $('#domicConductor').val(),
    $('#telefonoConductor').val(),
    $('#dniConductor').val(),

    tomadDenunciante,
    condReclamante,
    tomadReclamante
];

$.ajax({
    url: "scripts/aseguradora.php",
    type: 'post',
    dataType: 'json',
    data: {param: 30001, data: datosSiniestro},
    success: function(r)
    {
        if(r.success)
        {
            alertify.success('Siniestro cargado con exito');
            $('#contenidoC').load('vista/siniestro/cargarSiniestro.php');

        }else
        {
    
```

```

        alertify.error('Error al cargar el siniestro');
        $('#contenidoC').load('vista/siniestro/cargarSiniestro.php');
    }
}
});

// Mostramos un mensaje de confirmación antes de cancelar el proceso de
emisión de un siniestro.
function cancelarProceso(){
    alertify.confirm('Emision de siniestro', '¿Cancelar proceso de emision de
siniestro?',
    function(){ $('#contenidoC').load('vista/siniestro/cargarSiniestro.php');
},
    function(){}).set(configuracionAlert('okCancel','Si'));
}

// Función para modificar el estado de un siniestro.
function cambiarEstadoSin(idSiniestro, idEstadoActual)
{
    $('#modalCambiarEstado').modal('open');

    $("#selectEstadoSiniestro").val(idEstadoActual);
    var idSiniestro = idSiniestro;
    var idEstadoActual = idEstadoActual;

    $("#aceptarCambiarEstadoSin").attr('disabled', true);
    $("#selectEstadoSiniestro").on('change', function(){
        $("#aceptarCambiarEstadoSin").attr('disabled', false);
    });

    $("#aceptarCambiarEstadoSin").off('click');
    $("#aceptarCambiarEstadoSin").on('click', function(){

        var idNuevoEstSin = $("#selectEstadoSiniestro").val();

        // Si el nuevo estado es igual al ya establecido, no se realizara la
petición Ajax.
        if(idNuevoEstSin != null && idNuevoEstSin != idEstadoActual){
            var datos = [idSiniestro, idNuevoEstSin];
            $.ajax({
                url: 'scripts/aseguradora.php',
                type: 'post',
                dataType: 'json',
                data: {param: 30002, data: datos},
                success: function(r){
                    if(r.success){

                        $('#modalCambiarEstado').modal('close');
                        tablaAdministrarSiniestro.ajax.reload();
                        alertify.success('Estado modificado correctamente');
                    }else{
                        $('#modalCambiarEstado').modal('close');
                    }
                }
            });
        }
    });
}

```

```

        alertify.error('Ocurrio un error al modificar el esta
do');
    }
}
});

}else{
    $('#modalCambiarEstado').modal('close');
    alertify.notify('No se realizaron cambios');
    tablaAdministrarSiniestro.ajax.reload();
}
});
}

// Función para traer y mostrar los datos del siniestro.
function traerDatosSin(idSiniestro)
{
    // Cargamos el fichero php con la vista en el contenedor principal.
    $('#contenidoC').load('vista/siniestro/elementosSiniestro/modificarSiniest
ro.php', function(){
        funcionalidadesSiniestro();
        $(".textoCabeceraNuevoSin").html('');
        //Quitamos del DOM elementos que no usamos
        elementos = `#siguienteSiniestro, #atrasDenunciante, #atrasAccidente,
#atrasReclamante, #atrasConductor,
            #mensajeSiniestro, #mensajeDenunciante, #mensajeAccidente,
#mensajeReclamante, #mensajeConductor`;
        $(elementos).remove();

        $.ajax({
            url: 'scripts/aseguradora.php',
            type: 'post',
            dataType: 'json',
            data: {param: 30003, data: idSiniestro},
            success: function(r){
                if(r.success){

                    texto = `Poliza <span class="textoCabNuevoSinAzul">`+r.nr
oPol+`</span>.
                        Siniestro <span class="textoCabNuevoSinAzul">`+r
.nroSiniestro+`</span>
                        denunciado el <span class="textoCabNuevoSinAzul"
>`+r.fechaDen+`</span>`;
                    $(".textoCabeceraNuevoSin").html(texto);

                    //Convierto la fecha a formato valido y se lo asigno a da
tepicker para asignar un limite maximos
                    fechaMax = moment(r.fechaDen, 'DD/MM/YYYY HH:mm').format(
'DD/MM/YYYY');
                    $("#fechaOcurrencia").datepicker( "option", "maxDate", f
echaMax);

                    $("#nroSin").val(r.nroSiniestro);
                    $("#fechaDenuncia").val(r.fechaDen);
                }
            }
        });
    }
}

```

```

        $("#" + id).val(r.descripcion);
        $("#" + id).trigger('change');
    }
}

function() {
    var tipo = $("#" + id).val();
    if (tipo == "SINIESTRO") {
        // Dependiendo del tipo de siniestro seleccionado,
        // mostraremos ciertos elementos o no.
        $("#" + id).val(r.tipoSiniestro).trigger('change', function() {
            funcionalidadesSiniestro();
        });

        if (r.tomadDenunciante == 'SI') {
            $("#" + id).val(1).trigger('change');
        } else if (r.tomadDenunciante == 'NO') {
            $("#" + id).val(2).trigger('change');
            $("#" + id).val(r.nomDen);
            $("#" + id).val(r.telDen);
            $("#" + id).val(r.dniDen);
            $("#" + id).val(r.domDen);
            $("#" + id).val(r.emailDen);
        }
    }

    $("#" + id).val(r.descripcionAcc);

    $('#' + id).val(r.provinciaAcc).trigger('change');
    mostrarLocalidadesSin(r.provinciaAcc, r.localidadAcc);
    $('#' + id).on('change', function() {
        mostrarLocalidadesSin($("#" + id).val(), 0);
    });
    $("#" + id).val(r.calleAcc);
    $("#" + id).val(r.alturaAcc);

    $("#" + id).val(r.nombreTercero);
    $("#" + id).val(r.domTercero);
    $("#" + id).val(r.telTercero);
    $("#" + id).val(r.docTercero);

    if (r.condReclamante == 'SI') {
        $("#" + id).val(1).trigger('change');
    } else if (r.condReclamante == 'NO') {
        $("#" + id).val(2).trigger('change');
        $("#" + id).val(r.nombreConductor);
        $("#" + id).val(r.telConductor);
        $("#" + id).val(r.domConductor);
        $("#" + id).val(r.dniConductor);
    } else if (r.condReclamante == '') {
        $("#" + id).val(0);
    }

    //Al modificar la opcion, limpiamos o volvemos a mostrar
    //los datos.
    camposDenunciante = "#nombreDenunciante, #telefonoDenunciante,
    #dniDenunciante, #domicDenunciante, #emailDenunciante";
    $("#" + id).on('change', function() {
        if ($("#this").val() == 1) {

```

```

        $(camposDenunciante).val('').prop('disabled', true);
    });
} else if($(this).val() == 2){
    $('#nombreDenunciante').val(r.nomDen);
    $('#telefonoDenunciante').val(r.telDen);
    $('#dniDenunciante').val(r.dniDen);
    $('#domicDenunciante').val(r.domDen);
    $('#emailDenunciante').val(r.emailDen);
}
})
camposConductor = "#nombreConductor, #telefonoConductor,
#domicConductor, #dniConductor";
$("#selectCondRec").on('change', function(){
    if($(this).val() == 1){
        $(camposConductor).val('').prop('disabled', true)
    };
} else if($(this).val() == 2){
    $('#nombreConductor').val(r.nombreConductor);
    $('#telefonoConductor').val(r.telConductor);
    $('#domicConductor').val(r.domConductor);
    $('#dniConductor').val(r.dniConductor);
}
})
//Pasamos el horario maximo y la fecha maxima permitida
fechaHoraMaxima = r.fechaDen //Formato fecha y hora juntas;
validarModSiniestro(idSiniestro, fechaHoraMaxima);

} else{
    alertify.error('Ha ocurrido un problema al mostrar la informacion del siniestro solicitado');
}
});
});
});

// Función para procesar la modificación de los datos del siniestro.
function modificarDatosSin(idSiniestro)
{
    if($("#TipoSin").val() == 5){
        tomadorReclama = 'NO';
    } else if($("#TipoSin").val() != 5){
        tomadorReclama = 'SI';
    }

    if($("#selectDenunAseg").val() == 1){
        tomadorDenuncia = 'SI';
    } else if($("#selectDenunAseg").val() == 2){
        tomadorDenuncia = 'NO';
    }

    conductorReclamante = '';
}

```

```

if($("#selectCondRec").val() == 1){
    conductorReclamante = 'SI';
} else if($("#selectCondRec").val() == 2){
    conductorReclamante = 'NO';
}

// Almacenamos el valor de todos los campos en un arreglo.
datosSin = [$("#fechaOcurrencia").val(), $("#horaOcurrencia").val(), $("#LesMuer").val(), $("#TipoSin").val(),
            $("#nombreDenunciante").val(), $("#dniDenunciante").val(), $("#domicDenunciante").val(),
            $("#telefonoDenunciante").val(), $("#emailDenuncia").val(),
            $("#decripAcc").val(), $("#provinciaAcc").val(), $("#localida
dAcc").val(), $("#calleAcc").val(), $("#alturaAcc").val(),
            $("#nombreReclamante").val(), $("#domicReclamante").val(),
            $("#telefonoReclamante").val(), $("#dniReclamante").val(),
            $("#nombreConductor").val(), $("#domicConductor").val(),
            $("#telefonoConductor").val(), $("#dniConductor").val(),
            idSiniestro, tomadorReclama, tomadorDenuncia, conductorReclam
ante];

$.ajax({
    url: 'scripts/aseguradora.php',
    type: 'post',
    dataType: 'json',
    data: {param: 30004, datosSin: datosSin},
    success: function(r){
        if(r.success){

            // Si la acción es = 1: Siniestro se modifica.
            if(r.accion == 1){
                alertify.success('Siniestro modificado correctamente');
                $("#modalModRestoDatos").modal('close');
            // Si la acción es = 2: Los datos son iguales a los
            grabados.
            }else if(r.accion == 2){
                alertify.notify('No se realizaron modificaciones');
                $("#modalModRestoDatos").modal('close');

            }else{
                alertify.error('Ocurrió un problema al modificar los dato
s del siniestro');
                $("#modalModRestoDatos").modal('close');
            }

        }else{
            alertify.error('Ocurrió un problema al realizar la petición d
e modificación de datos del siniestro');
            $("#modalModRestoDatos").modal('close');
        }

        $('#contenidoC').load('vista/siniestro/elementosSiniestro/adminis
trarSiniestro.php', function(){
            tablaAdministrarSiniestro();
        }
    }
}

```

```

        tablaAdministrarSiniestro.ajax.reload();
    });
}
});

// Función para listar en un select todas las localidades una vez
seleccionada la provincia.
function mostrarLocalidadesSin(id, locSelect){
    jQuery('#localidadAcc').empty();
    jQuery("#localidadAcc").append('<option value="" disabled selected>Selecc
    ión</option>');
    $.ajax({
        type:"post",
        data:{param:80001, id:id},
        url: "scripts/aseguradora.php",
        dataType:'json',
        success: function(r){
            if(r!=null){
                $.each(r.data, function(i,val){
                    if(locSelect==val.localidadId){
                        $("#localidadAcc").append('<option value="' + val.loc
                        alidadId + '" selected>' + val.localidadNombre + '</option>');
                    }else{
                        $("#localidadAcc").append('<option value="' + val.loc
                        alidadId + '">' + val.localidadNombre + '</option>');
                    }
                });
            }
        });
    });
}

// Función para mostrar los datos del siniestro.
function mostrarDatosSiniestro(idSiniestro)
{
    var idSiniestro = idSiniestro;
    $.ajax({
        type:"post",
        data:{param:30005, id:idSiniestro},
        url: "scripts/aseguradora.php",
        dataType:'json',
        success: function(r){
            if(r.success){
                // Cargamos el fichero php con la vista en el contenedor
                general.
                $('#contenidoC').load('vista/siniestro/elementosSiniestro/ver
                DatosSiniestro.php', function(){

                    $(".nroSin").html('#'+r.nroSiniestro);
                    $(".fechaDen").html(r.fechaOc+' - '+r.horaOc);
                    $(".fechaOcu").html(r.fechaDen);
                    $(".tipoSin").html(r.tipoSiniestro);
                    $(".lesionMuerte").html(r.lesionMuerte);

```

```

        $(".estadoSin").html(r.estadoSiniestro);

        $(".numPol").html('#'+r.nroPol);
        $(".tomador").html(r.tomador);
        $(".vigencia").html(r.vigencia_inicio+' al '+r.vigencia_f
in+' a las 11:59');
        $(".estadoPol").html(r.estadoPoliza);

        $(".denunTom").html(r.tomadDenunciante);
        $(".nombreDen").html(r.nomDen);
        $(".dniDen").html(r.dniDen);
        $(".domiDen").html(r.domDen);
        $(".telDen").html(r.telDen);
        $(".emailDen").html(r.emailDen);

        $(".ubiAcc").html(r.localidadAcc+', '+r.provinciaAcc+', '
+r.calleAcc+' '+r.alturaAcc);
        $(".descAcc").html(r.descripcionAcc);

        $(".tomRec").html(r.tomadReclamante);
        $(".nombreRec").html(r.nombreTercero);
        $(".domRec").html(r.domTercero);
        $(".telRec").html(r.telTercero);
        $(".docRec").html(r.docTercero);

        $(".condRec").html(r.condReclamante);
        $(".nombreCond").html(r.nombreConductor);
        $(".domCond").html(r.domConductor);
        $(".telCond").html(r.telConductor);
        $(".docCond").html(r.dniConductor);

        /*En caso de que sea la misma persona, no mostramos los c
ampos innecesarios (Vacios)*/
        if(r.tomadDenunciante=='SI'){
            $(".nombreDen, .dniDen, .domiDen, .telDen, .emailDen"
).closest('.infoSin-3').css('display', 'none');
        }
        if(r.tomadReclamante=='SI'){
            $(".nombreRec, .domRec, .telRec, .docRec").closest('.'
.infoSin-3').css('display', 'none');
        }
        if(r.condReclamante=='SI'){
            $(".nombreCond, .domCond, .telCond, .docCond").closes
t('.infoSin-3').css('display', 'none');
        }

    });
}else{
    alertify.error('Ocurrio un problema al mostrar la informacion
.');
}
});
```

```

// Función donde traemos y mostramos las imágenes del siniestro.
function mostrarImgSiniestro(idSin)
{
    $("#nroSiniestroPie").html('sin imágenes registradas');
    $("#marcaModeloPie").html('');

    $("#nuevaImagenBtn").off('click');
    $("#nuevaImagenBtn").on('click', function(){
        $("#imagenSin").trigger('click');
    });

    $.ajax({
        type:"post",
        data:{param:30006, idSin:idSin},
        url: "scripts/aseguradora.php",
        dataType:'json',
        success: function(r){
            $(".gallery-container").html(r.data);
            $("#nroSiniestroPie").html(r.info['nroSin']);
            $("#marcaModeloPie").html(r.info['marcaModVe']);
        }
    });

    grabarImgSiniestro(idSin);
}

// Función para grabar una nueva imagen del siniestro.
function grabarImgSiniestro(idSin)
{
    var idSin = idSin;

    $("#cargarImagen").off('click');
    $("#cargarImagen").on('click', function(){

        // Al llamar a la variable, se oculta el mensaje.
        var estadoOut = setTimeout(function(){
            $("#estadoSubida").fadeOut(1000);
        },2000);

        var imagenes = document.getElementsByClassName("gallery-card").length;

        // Solo se pueden cargar 12 imágenes por siniestro.
        if(imagenes<12){

            var imagen = $('#imagenSin');

            var image_data = imagen.prop('files')[0];

            var formData = new FormData();
            formData.append('imagen', image_data);
            formData.append('idSin', idSin);
        }
    });
}

```

```

$.ajax({
    url: "scripts/imagenes/procesarImgSin.php",
    type: "POST",
    data: formData,
    contentType: false,
    cache: false,
    processData: false,
    beforeSend: function(d){
        $("#cargarImagen").prop('disabled', true);
        $("#estadoSubida").html('').show();
    },
    success: function(r){
        $("#cargarImagen").prop('disabled', false);
        if(r == 1){
            //alertify.success('Imagen cargada');
            $("#imagenSin").val('');
            $("#estadoSubida").html('').show();
            estadoOut;
            mostrarImgSinistro(idSin);

        }else if(r == 10){
            $("#estadoSubida").html('');
            alertify.alert('Cargar imagen', 'Debe seleccionar una
imagen').set(configuracionAlert('ok'));
        }else if(r == 20){
            $("#estadoSubida").html('').show();
            estadoOut;
            alertify.alert('Cargar imagen', 'El formato de la ima
gen no coincide con los admitidos (.JPG / .PNG / .JPEG)').set(configuracionAl
ert('ok'));
        }else if(r == 30){
            $("#estadoSubida").html('').show();
            alertify.alert('Cargar imagen', 'El tamaño de la ima
gen es superior al maximo permitido (5 Mb)').set(configuracionAlert('ok'));
            estadoOut;
        }else if(r == 40){
            $("#estadoSubida").html('').show();
            alertify.error('No se ha podido grabar esa imagen');
            estadoOut;
        }else if(r == 50){
            $("#estadoSubida").html('').show();
            alertify.error('Ha ocurrido un problema al realizar l
a peticion al servidor');
            estadoOut;
        }
    });
});
}};


```

```

        alertify.alert('Cargar imagen', 'Solo se permiten cargar hasta 12
imagenes').set(configuracionAlert('ok'));
    }

});

}

// Función para eliminar una imagen del siniestro.
function eliminarImgSin(idImg)
{
    var idImg = idImg;

    // Mostramos un mensaje de confirmación antes de eliminarla.
    alertify.confirm('Eliminar imagen', '¿Eliminar esta imagen?',
        function(){
            $.ajax({
                url: 'scripts/aseguradora.php',
                type: 'POST',
                dataType: 'json',
                data: {param:30007, idImg:idImg},
                success: function(r){
                    if(r.success){
                        mostrarImgSiniestro(r.idSin);
                    }else{
                        alertify.error('Ocurrió un error al eliminar la imagen');
                    }
                }
            });
        },
        function(){}
    ).set(configuracionAlert('okCancel','Si'));
}

// Función para mostrar el listado de polizas al momento de cargar un nuevo
siniestro.
function listarPolizas(polizaSelect, getUnica)
{
    jQuery('#selectPolizas').empty();
    jQuery('#selectPolizas').append('<option value="0" disabled selected>Sele
ccione una Poliza</option>');
    jQuery.ajax({
        type:"post",
        data:{param:10004},
        url: "scripts/aseguradora.php",
        dataType:'json',
        success: function(r){
            if(r!=null){
                if(getUnica==null){
                    // Cuando no pasamos por parámetro la variable "getUnica".
                    jQuery.each(r.data, function(k,v){
                        // Si llega polizaSelect por parámetro, se
seleccionara esa misma.
                        if(polizaSelect==v.id){

```

```

                jQuery('#selectPolizas').append('<option value="'+
+v.id+'" selected>' + v.nro + ' - ' + v.tomador + ' - ' + v.doc + '</option>');
            }else{
                if(v.estado==0){ //Deshabilitamos las polizas que
estan vencidas o con deuda.
                    jQuery('#selectPolizas').append('<option valu
e="'+v.id+'">' + v.nro + ' - ' + v.tomador + ' - ' + v.doc + '</option>');
                }else if(v.estado==1){
                    jQuery('#selectPolizas').append('<option valu
e="'+v.id+'" disabled>' + v.nro + ' - ' + v.tomador + ' - ' + v.doc + ' - '+' (Poliza Ven
cida)' + '</option>');
                }else if(v.estado==3){
                    jQuery('#selectPolizas').append('<option valu
e="'+v.id+'" disabled>' + v.nro + ' - ' + v.tomador + ' - ' + v.doc + ' - '+' (Poliza c
on deudas)' + '</option>');
                }
            }
        });
        // Si el getUnica es = 1, solo se mostrara la poliza
seleccionada que llegue por parametro (polizaSelect).
    }else if(getUnica==1){
        jQuery.each(r.data, function(k,v){
            if(polizaSelect==v.id){
                jQuery('#selectPolizas').append('<option value="'+
+v.id+'" selected>' + v.nro + ' - ' + v.tomador + ' - ' + v.doc + '</option>');
            }
        });
    }
}
});

// Función donde generamos la tabla con el listado de siniestros registrados,
utilizando el plugin jquery validate, para poder modificar sus datos.
function tablaAdministrarSiniestro()
{
    tablaAdministrarSiniestro = $('#administrarSiniestroTabla').DataTable({
        "ajax": "scripts/tablas/tablaAdministrarSiniestro.php",
        "columns": [
            {data:"nroSin", title: "Nro Siniestro", width:""}, 
            {data:"nroPol", title: "Nro Poliza", width:""}, 
            {data:"tomador", title: "Tomador", width:""}, 
            {data:"tipo", title: "Tipo", width:""}, 
            {data:"fechDen", title: "Denuncia", width:""}, 
            {data:"estado", title: "Estado", width:""}, 
            {data:"botones", title: "Administrar", width:""}
        ],
        "language": {
            "zeroRecords": "No se encontraron siniestros registrados en la ba
se de datos.",
            "sSearch": "Buscar",
        },
        "scrollY": 370,
    });
}

```

```

        "scrollCollapse": true,
        "dom": 'ft',
        "order": [[4, "desc"]],
        "lengthMenu": [[5, 10], [5, 10]],
        "aoColumnDefs": [
            { 'bSortable': false, 'aTargets': [5] }
        ]
    });

    return tablaAdministrarSiniestro;
}

// Función donde generamos la tabla con el listado de siniestros registrados,
// utilizando el plugin jquery validate para poder consultar su informacion.
function tablaConsultarSiniestro()
{
    $('#consultarSiniestroTabla').DataTable({
        "ajax": "scripts/tablas/tablaConsultarSiniestro.php",
        "columns": [
            {data:"nroSin", title:"Nro Siniestro"},
            {data:"nroPol", title:"Nro Poliza"},
            {data:"tomador", title:"Tomador"},
            {data:"fechDen", title: "Fecha Denuncia"},
            {data:"estado", title: "Estado"},
            {data:"boton", title: ""}
        ],
        "language": {
            "zeroRecords": "No se encontraron siniestros registrados en la base de datos.",
            "sSearch": "Buscar",
        },
        "dom": 'ft',
        "scrollY": 370,
        "scrollCollapse": true,
        "order": [[4, "desc"]],
        "aoColumnDefs": [
            { 'bSortable': false, 'aTargets': [5] }
        ]
    });

}

// Función donde tenemos funcionalidades extras para los formularios y los elementos relacionados a los siniestros.
function funcionalidadesSiniestro()
{
    // Aplicamos las funciones de materializeCss.
    $('.tabs').tabs();
    $('.collapsible').collapsible();
    $('.modal').modal({
        dismissible: false
    });
    $('input, select').addClass('browser-default');
}

```

```

    //Se aplica la libreria select2 a los inputs requeridos en emision de siniestro.
    $('#selectPolizas, #provinciaAcc, #localidadAcc, #provinciaAccMod, #localidadAccMod').select2({width: "100%"});

    // Generamos el numero de siniestro y lo mostramos en el campo.
    var letrasSin = ['A','B','C','D','E','F','G','H','I','J','K','L','M','N','O','P','Q','R','S','T','U','V','W','X','Y','Z'];
    var lSin = (letrasSin[Math.round(Math.random()*26)]+letrasSin[Math.round(Math.random()*26)]);
    var numSin = Math.floor(1e9 + (Math.random() * 9e14));
    var numSiniestro = numSin+lSin;
    $('#nroSin, #fechaDenuncia, #fechaOcurrencia, #fechaOcurrenciaMod').prop('readonly', true);
    $('#nroSin').html(numSiniestro);

    //Se aplica el complemento Datepicker al input llamado "fechaOcurrencia" en emision de siniestro, pero se aplica una opcion para que solo se pueda seleccionar una fecha anterior a la actual.
    $(function(){
        $("#fechaOcurrencia").datepicker({
            dayNames: ["Domingo","Lunes","Martes","Miercoles","Jueves","Viernes","Sabado"],
            dayNamesMin: ["Do","Lu","Ma","Mi","Ju","Vi","Sa"],
            monthNames: ["Enero","Febrero","Marzo","Abril","Mayo","Junio","Julio","Agosto","Septiembre","Octubre","Noviembre","Diciembre"],
            monthNamesShort: ["Ene","Feb","Mar","Abr","May","Jun","Jul","Ago","Sep","Oct","Nov","Dic"],
            dateFormat: "dd/mm/yy",
            maxDate: 0
        });
    });

    // Mantenemos la integridad de los inputs en caso de ser modificados.
    $('#fechaOcurrencia').on('mouseover', function(){
        $(this).prop('readonly', true);
    });
    $('#horaOcurrencia').on('mouseover', function(){
        $(this).prop('type', 'time');
    });

    // Segun la opcion del select denunciante-asegurado que elijamos se habilitaran los campos o no.
    inputsDenunciante = '#nombreDenunciante, #telefonoDenunciante, #dniDenunciante, #domicDenunciante, #emailDenunciante';
    $(inputsDenunciante).val('').prop('disabled', true);

    $("#selectDenunAseg").on('change', function(){
        if($(this).val()==1){
            $(inputsDenunciante).val('').prop('disabled', true);
            $(inputsDenunciante).on('mouseover', function(){
                $(inputsDenunciante).prop('disabled', true)
            })
        }
    })

```

```

        })
    }else if($(this).val()==2){
        $(inputsDenunciante).val(' ').prop('disabled', false);
        $(inputsDenunciante).on('mouseover', function(){
            $(inputsDenunciante).prop('disabled', false)
        })
    }else{
        $(inputsDenunciante).val(' ').prop('disabled', true);
        $(inputsDenunciante).on('mouseover', function(){
            $(inputsDenunciante).prop('disabled', true)
        })
    }
}

//Si requerimos de agregar la informacion de terceras personas, segun el tipo de siniestro elegido, estaran disponible los campos o no.
inputsReclamante = '#nombreReclamante, #domicReclamante, #telefonoReclamante, #dniReclamante';
inputsConductor = '#selectCondRec, #nombreConductor, #domicConductor, #telefonoConductor, #dniConductor';

$("#TipoSin").on('change', function(){
    if($("#TipoSin").val()==5){

        $("#tab4, #tab5").parent('.tab').removeClass('disabled');
        $("#tab3").children('i').css('display','inline-block');
        $("#test4").css('display', 'none').mouseover(()=> $("#test4").css('display','block'));
        $("#test5").css('display', 'none').mouseover(()=> $("#test5").css('display','block'));
        $("#tab4, #tab5").mouseover(()=> $("#tab4, #tab5").parent('.tab').removeClass('disabled'));

        $("#selectCondRec").val(0);
        $(inputsReclamante).prop('disabled', false);
        $(inputsConductor).prop('disabled', false);

    }else{

        $("#tab4, #tab5").parent('.tab').addClass('disabled');
        $("#tab3").children('i').css('display','none');
        $("#test4, #test5").css('display', 'none').mouseover(()=> $("#test4, #test5").css('display','none'));
        $("#tab4, #tab5").mouseover(()=> $("#tab4, #tab5").parent('.tab').addClass('disabled'));

        $("#selectCondRec").val(0);
        $(inputsReclamante).prop('disabled', true);
        $(inputsConductor).prop('disabled', true);
    }
}

```

```

        //Quitamos el boton para que no quede el actual en caso de modificar
el tipo de siniestro.
        $(".btnAcc").html('');
    });

    camposConductor = '#nombreConductor, #domicConductor, #telefonoConductor,
#dniConductor';

    $("#selectCondRec").on('change', function(){
        if($(this).val() == 1){
            $(camposConductor).val('').prop('disabled', true);
            $(camposConductor).on('mouseover', function(){
                $(camposConductor).prop('disabled', true)
            })
        }else if($(this).val() == 2){
            $(camposConductor).val('').prop('disabled', false);
            $(camposConductor).off('mouseover')
        }else{
            $(camposConductor).val('').prop('disabled', true);
            $(camposConductor).on('mouseover', function(){
                $(camposConductor).prop('disabled', true)
            })
        }
    })

    //Establecemos limites maximos y minimos en los campos de anio.
    $("#anio").prop('min', 1950);
    $("#anio").prop('max', 2021);

    //Al presionar el boton atras en cada seccion, se muestra la pantalla anterior.
    $("#atrasDenunciante").off('click');
    $("#atrasDenunciante").on('click', function(e){
        $('.tabs').tabs('select', 'test1');
        $('.btnSigDen').html('');
    });
    $("#atrasAccidente").off('click');
    $("#atrasAccidente").on('click', function(e){
        $('.tabs').tabs('select', 'test2');
        $('.btnAcc').html('');
    });
    $("#atrasReclamante").off('click');
    $("#atrasReclamante").on('click', function(e){
        $('.tabs').tabs('select', 'test3');
        $('.btnSigRec').html('');
    });
    $("#atrasConductor").off('click');
    $("#atrasConductor").on('click', function(e){
        $('.tabs').tabs('select', 'test4');
        $('.btnSinConductor').html('');
    });

    $("#volverConsultarSin").on('click', function(){


```

```
        $('#contenidoC').load('vista/siniestro/elementosSiniestro/consultarSi  
niestro.php', function(){  
    tablaConsultarSiniestro();  
});  
});  
  
// Parametros para lightbox js (Librería de imágenes).  
lightbox.option({  
    'resizeDuration': 0,  
    'fadeDuration': 0,  
    'wrapAround': true,  
    'albumLabel': "%1/%2",  
    'imageFadeDuration': 0,  
    'wrapAround': true  
});  
}
```

➔ 4.3.3.8 fichero “sucursales.js”

Este fichero contiene todas las funciones que realizan las acciones relacionadas al frontend o vista de las sucursales en el sistema.

```
// Cargamos el DOM con jQuery.
jQuery(document).ready(function($){

    // Cargamos los ficheros php con las vistas al presionar en las opciones.
    $(".btnAgregarSucursal").off('click');
    $(".btnAgregarSucursal").on('click', function(){
        $("#contenidoC").load('vista/sucursales/nuevaSucursal.php', function()
    ){
        validarSucursal();
    });
});

$(".volverSucursal").off('click');
$(".volverSucursal").on('click', function(){
    $('#contenidoC').load('vista/sucursales/administrarSucursales.php', function(){
        tablaAdminSucursales();
    });
});

$("#cpSucursal, #localidadSucursal, #provinciaSucursal").select2();

$("#provinciaSucursal").on('change', function(){
    idProv = $("#provinciaSucursal").val();
    mostrarCp(idProv);
})

$("#cpSucursal").on('change', function(){
    cp = $("#cpSucursal").val();
    mostrarLocalidades(cp);
});

$(document).off('click', '#verSucursal');
$(document).on('click', '#verSucursal', function(){
    idSuc = $(this).attr('value');
    verInfoSucursal(idSuc);
});

$(document).off('click', '#eliSucursal');
$(document).on('click', '#eliSucursal', function(){
    idSuc = $(this).attr('value');
    eliminarSucursal(idSuc);
});
```

```

$(document).off('click', '#modSucursal');
$(document).on('click', '#modSucursal', function(){
    idSuc = $(this).attr('value');

    var info = traerInfoSucursal(idSuc);

    $('#contenidoC').load('vista/sucursales/modificarSucursal.php', function(){
        validarSucursal(idSuc);

        $('.sucursalNro').html(info['nombre'] + ' - ' + info['iden']);

        $('#nombreSucursal').val(info['nombre']);
        $('#cuitSucursal').val(info['cuit']);
        $('#direccionSucursal').val(info['direccion']);
        $('#provinciaSucursal').val(info['provinciaId']);
        mostrarCp(info['provinciaId'], info['cp']);
        mostrarLocalidades(info['cp'], info['localidadId']);
    });
});

});

// Función para mostrar el listado de códigos postales en un select una vez
seleccionada una provincia.

function mostrarCp(idProv, cpSelect){
    $("#cpSucursal").empty();

    $.ajax({
        url: 'scripts/aseguradora.php',
        type: 'post',
        dataType: 'json',
        data: {param: 80002, id: idProv},
        success: function(r){
            if(r!=null){
                $("#cpSucursal").append('<option value="0" selected disabled>
Seleccione</option>');
                $.each(r.data, function(key, val){
                    if(cpSelect==val.cpNum){
                        $("#cpSucursal").append('<option value="' + val.cpNum +
" selected>' + val.cpNum + '</option>')
                    }else{
                        $("#cpSucursal").append('<option value="' + val.cpNum +
">' + val.cpNum + '</option>')
                    }
                })
            }else{
                alertify.error('Ocurrio un error al mostrar las localidades')
            }
        }
    });
}

```

```

        })
    }

// Función para listar todas las localidades en un select una vez
seleccionado un código postal.
function mostrarLocalidades(cp, selectLoc){

    $("#localidadSucursal").empty();

    $.ajax({
        url: 'scripts/aseguradora.php',
        type: 'post',
        dataType: 'json',
        data: {param:80000, cp:cp},
        success: function(r){
            if(r!=null){
                $("#localidadSucursal").append('<option value="0" selected di
sabled>Seleccione</option>');
                $.each(r.data, function(key, val){
                    if(selectLoc==val.localidadId){
                        $("#localidadSucursal").append('<option value="'+val.
localidadId+'" selected>' +val.localidadNombre+ '</option>');
                    }else{
                        $("#localidadSucursal").append('<option value="'+val.
localidadId+'">' +val.localidadNombre+ '</option>');
                    }
                })
            }else{
                alertify.error('Ocurrio un error al mostrar las localidades')
            }
        }
    })
}

// Función para validar los campos del formulario cuando agregamos una nueva
sucursal, usando el plugin jquery validate.
function validarSucursal(data){

    var valNombre = /^[a-zA-Z ]{5,30}$/;

    $.validator.addMethod("validarNombre", function (value, element) {
        return this.optional(element) || valNombre.test(value)
    })

    $("#registroSucursal").validate({
        rules: {
            "nombreSucursal": {required: true, minlength: [5], validarNom
bre: true},
            "cuitSucursal": {required: true, rangelength: [10, 15]},
            "direccionSucursal": {required: true, minlength: [5]},
            "provinciaSucursal": {required: true, rangelength: [1, 24]},
            "cpSucursal": {required: true, rangelength: [1, 22963]},
            "localidadSucursal": {required: true, rangelength: [1, 22963]}
        },
    })
}

```

```

messages: {
    "nombreSucursal": {required: "Debe ingresar el Nombre",
        minlength: "El Nombre no puede ser tan corto", validarNombre: "El Nombre
no puede contener numeros"}, 
    "cuitSucursal": {required: "Debe ingresar el Cuit",
        rangelength: "El Cuit ingresado no es valido"}, 
    "direccionSucursal": {required: "Debe ingresar la Direccion",
        minlength: "La informacion en la Direccion es muy limitada"}, 
    "provinciaSucursal": {required: "Debe seleccionar una Provincia",
        rangelength: "La Provincia es incorrecta"}, 
    "cpSucursal": {required: "Debe seleccionar un Codigo Postal",
        rangelength: "El Codigo Postal es incorrecto"}, 
    "localidadSucursal": {required: "Debe seleccionar una Localidad",
        rangelength: "La Localidad es incorrecta"}
}, 

highlight: function (element) {
    if ($(element).hasClass('select2')) {
        $(element).siblings('span').find('.selection').find('.select2-
selection').addClass('claseError');
    } else {
        $(element).addClass('claseError');
    }
}, 

unhighlight: function (element){
    if ($(element).hasClass('select2')) {
        $(element).siblings('span').find('.selection').find('.select2-
selection').removeClass('claseError');
    } else {
        $(element).removeClass('claseError');
    }
}, 

errorElement: 'li',
errorLabelContainer: '#mensajeSucursal', 

submitHandler: function(){

    // Dependiendo del nombre del botón ingresado, registramos o
    modificamos una sucursal.
    var id = $(".btnSubmitSucursal").attr('id');

    if(id=='submitSucursal'){
        registrarSucursal();
    }else if(id=='submitModificarSucursal'){
        modificarSucursal(data);
    }

}
}
}

```

```

// Función para grabar una nueva sucursal.
function registrarSucursal(){
    info = [
        $("#nombreSucursal").val(),
        $("#cuitSucursal").val(),
        $("#direccionSucursal").val(),
        $("#provinciaSucursal").val(),
        $("#cpSucursal").val(),
        $("#localidadSucursal").val()
    ]

    $.ajax({
        url: 'scripts/aseguradora.php',
        type: 'post',
        dataType: 'json',
        data: {param:70000, data:info},
        success: function(r){
            if(r.success){
                $("#nombreSucursal").val(''),
                $("#cuitSucursal").val(''),
                $("#direccionSucursal").val(''),
                $("#provinciaSucursal").val('\n'),
                mostrarCp(null, 0);
                mostrarLocalidades(null, 0);
                $('#contenidoC').load('vista/sucursales/administrarSucursales.php', function(){
                    tablaAdminSucursales();
                });
                alertify.success('Sucursal registrada con exito');
            }else{
                alertify.error('Ocurrio un error al mostrar las localidades')
            }
        }
    });
}

// Función para eliminar una sucursal.
function eliminarSucursal(idSuc)
{
    alertify.confirm('Eliminar sucursal', '¿Esta seguro que desea eliminar esta sucursal? <br> Ya no estara disponible para el inicio de sesion',
        function(){
            $.ajax({
                url: 'scripts/aseguradora.php',
                type: 'post',
                dataType: 'json',
                data: {param:70001, idSuc:idSuc},
                success: function(r){
                    if(r.success){
                        tablaAdminSucursales.ajax.reload();
                        alertify.success('Sucursal eliminada del sistema');
                    }else{

```

```

        alertify.error('Ocurrio un error al realizar la peticion');
    );
    }
}
},
function(){}
.set(configuracionAlert('okCancel','Dar de baja')));
}

// Función para traer y mostrar la informacion de una sucursal.
function traerInfoSucursal(idSuc){

$.ajax({
    url: 'scripts/aseguradora.php',
    type: 'post',
    dataType: 'json',
    async: false, //Usamos para generar una petición no asíncrona.
    data: {param:70002, idSuc:idSuc},
    success: function(r){
        if(r.success){
            info = r.info;
        }else{
            alertify.error('Ocurrio un error al realizar la peticion');
        }
    }
});

return info;
}

// Función para procesar la modificación de los datos de una sucursal.
function modificarSucursal(idSuc){

info = [
    $('#nombreSucursal').val(),
    $('#cuitSucursal').val(),
    $('#direccionSucursal').val(),
    $('#provinciaSucursal').val(),
    $('#cpSucursal').val(),
    $('#localidadSucursal').val(),
    idSuc
]

$.ajax({
    url: 'scripts/aseguradora.php',
    type: 'post',
    dataType: 'json',
    data: {param:70003, data:info},
    success: function(r){

        // Si los datos son iguales a los ya registrados.
        if(r.repetido){
            alertify.notify('No se realizaron modificaciones');
        }
    }
});
}

```

```

}else{
    if(r.success){
        alertify.success('Informacion actualizada con exito');
    }else{
        alertify.error('Ocurrio un error al realizar la peticion');
    }
}

// Limpio los campos.
$("#nombreSucursal").val(''),
$("#cuitSucursal").val(''),
$("#direccionSucursal").val(''),
$("#provinciaSucursal").val('\n'),
mostrarCp(null, 0);
mostrarLocalidades(null, 0);

$('#contenidoC').load('vista/sucursales/administrarSucursales.php',
', function(){
    tablaAdminSucursales();
});

});

}

// Función para ver los datos de una sucursal.
function verInfoSucursal(idSuc){

info = traerInfoSucursal(idSuc);

$('#contenidoC').load('vista/sucursales/verInfoSucursal.php', function(){

$(".sucursalNro").html(info['nombre']);

$(".nomb").html(info['nombre']);
$(".iden").html(info['iden']);
$(".cuit").html(info['cuit']);
$(".dir").html(info['direccion']);
$(".cp").html(info['cp']);
$(".pro").html(info['provincia']);
$(".loc").html(info['localidad']);

});

}

// Función donde generamos una tabla con el listado de todas las sucursales
// para poder administrarlas, usando el plugin DataTables.
function tablaAdminSucursales(){

tablaAdminSucursales = $('#admSucursalesTabla').DataTable({
    "language": {
        "zeroRecords": "No se encontraron Sucursales registradas",
        "sSearch": "Buscar",
    }
}

```

```

        },
        "ajax": "scripts/tablas/tablaAdministrarSucursales.php",
        "scrollY": 370,
        "bPaginate": false,
        "dom": 'Bft',
        "order": [],

        "columns": [
            {data:"iden", title: "Identificador" ,width:""},
            {data:"nom", title: "Nombre" ,width:""}, 
            {data:"dir", title: "Direccion" ,width:""}, 
            {data:"cp", title: "Codigo Postal" ,width:""}, 
            {data:"ubic", title: "Ubicacion" ,width:""}, 
            {data:"acc", title: "Acciones" ,width:"20%"}],
        ],
        // Agregamos los botones de funcionalidad de usuarios.
        "buttons": [
            [
                {
                    text: '<i class="fas fa-plus-circle"></i> Agregar',
                    className: 'green btnAgregarSucursal'
                }
            ]
        );
    });

    return tablaAdminSucursales;
}

```

➔ 4.3.3.9 fichero “users.js”

Este fichero contiene todas las funciones que realizan las acciones relacionadas al frontend o vista de los usuarios del sistema.

```
// Cargamos el DOM con jQuery.
jQuery(document).ready(function(){

    // Llamamos las funciones directamente o a través de acciones.
    validarUser();
    getFotoPerfilUser();

    $('.modal').modal({
        dismissible: false
    });

    $('input, select').addClass('browser-default');

    // Usamos la librería select2 para algunos campos.
    $("#provinciaUser, #localidadUser").select2({width: "100%"})
    // Usamos la librería datepicker para generar un calendario en los
    campos, configurándolo para que la fecha sea correspondiente a una persona
    mayor de 18 años.
    $("#nacUser").prop('readonly', true).datepicker({
        dayNames: ["Domingo", "Lunes", "Martes", "Miércoles", "Jueves", "Viernes",
        "Sábado"],
        dayNamesMin: ["Do", "Lu", "Ma", "Mi", "Ju", "Vi", "Sa"],
        monthNames: ["Enero", "Febrero", "Marzo", "Abril", "Mayo", "Junio", "Julio",
        "Agosto", "Septiembre", "Octubre", "Noviembre", "Diciembre"],
        monthNamesShort: ["Ene", "Feb", "Mar", "Abr", "May", "Jun", "Jul", "Ago", "Se
        p", "Oct", "Nov", "Dic"],
        dateFormat: 'dd/mm/yy',

        changeMonth: true,
        changeYear: true,
        // yearRange: "c-100:c-16",
        minDate: '-100Y',
        maxDate: '-16Y'
    });

    $('.ui-datepicker-title select').removeAttr('style');

    $("#provinciaUser").on('change', function(){
        idProv = $("#provinciaUser").val();
        mostrarLocalidades(idProv);
    })

    $(document).off('click', '.btnNuevoUser');
    $(document).on('click', '.btnNuevoUser', function(){
        $("#contenidoC").load('vista/users/nuevoUser.php')
    });
})
```

```

$(document).off('click', '.btnListadoBajas');
$(document).on('click', '.btnListadoBajas', function(){
    listadoUserBaja()
})

$(".volverUser").off('click');
$(".volverUser").on('click', function() {
    $('#contenidoC').load('vista/users/administrarUsers.php', function(){
        tablaListadoUsers();
    });
});

$(".volverMisDatos, .volverGrabarFotoUser").off('click');
$(".volverMisDatos, .volverGrabarFotoUser").on('click', function() {
    $("#inicio").trigger('click');
});

$(document).off('click', '.reactivarUser')
$(document).on('click', '.reactivarUser', function(){
   idUser = $(this).attr('value')
    reactivarUser(idUser)
});

//Configurar datos del usuario
$(".misDatos").off('click');
$(".misDatos").on('click', function(){
    getMisDatosUser();
});

$(".modFotoUser").off('click');
$(".modFotoUser").on('click', function(){
    $("#contenidoC").load('vista/users/modFotoUser.php');
    $("#divTit").css('display', 'block');
    $("#tituloCont").html('mi foto de perfil');
    $("#divSubTit").html('');
    //getFotoPerfilUser();
});

$(".grabarNuevaFotoUser").off('click');
$(".grabarNuevaFotoUser").on('click', function(){
    modFotoUser();
});
}

// Función para validar los datos del formulario al registrar un nuevo
usuario.
function validarUser()
{
    var valNombre = /^[A-Záéíóñáéíóñ]{0}?[A-Záéíóñáéíóñ]+[\s]+([A-Záéíóñáéíóñ]{0}?[A-Záéíóñáéíóñ]+[\s]?([A-Záéíóñáéíóñ]{0}?[A-Záéíóñáéíóñ\']?)?$/;
    var valFecha = /^[\d{1,2}]/[\d{1,2}]/[\d{2,4}]/;
    var email = /^[^\w.%+]{1,64}@[?:[A-Z0-9-]{1,63}\.]{1,125}[A-Z]{2,63}$/i
}

```

```

$.validator.addMethod("validarNombre", function (value, element) {
    return this.optional(element) || valNombre.test(value)
})

$.validator.addMethod("validarFecha", function (value, element) {
    return this.optional(element) || valFecha.test(value)
})

$.validator.addMethod("validarGenero", function (value, element) {
    return this.optional(element) || value == 'masculino' || value == 'fe
menino' || value == 'otro'
})

$.validator.addMethod("validarRol", function (value, element) {
    return this.optional(element) || value == 'usuario' || value == 'admi
n'
})

$.validator.addMethod('email', function(value, element){
    return this.optional(element) || email.test(value);
});

$("#registroUser").validate({
    rules: {
        "nombreUser": {required: true, validarNombre: true},
        "documentoUser": {required: true, digits: true, rangelength: [
6, 8]}, 
        "nacionalidadUser": {required: true, lettersonly: true},
        "provinciaUser": {required: true, rangelength: [1, 24]},
        "localidadUser": {required: true, rangelength: [1, 22963]},
        "direccionUser": {required: true},
        "nacUser": {required: true, validarFecha: true},
        "telefonoUser": {required: true, digits: true, min: 6},
        "correoUser": {required: true, email: true},
        "generoUser": {required: true, validarGenero: true},
        "rolUser": {required: true, validarRol: true},
        //Modificar usuario y contraseña
        "usuarioUser": {required: true, minlength: 4, maxlength: 35}
    ,
        "pass1": {required: true, minlength: 5},
        "pass2": {required: true, minlength: 5, equalTo: "#pas
s1" }
    },
    messages: {
        "nombreUser": {required: "Debe ingresar el Nombre",
        validarNombre: "El Nombre ingresado es incorrecto"},
        "documentoUser": {required: "Debe ingresar el Documento",
        digits: "El Documento ingresado es incorrecto", rangelength: "El Do
cumento ingresado esta incompleto"},
        "nacionalidadUser": {required: "Debe ingresar la Nacionalidad",
        lettersonly: "El Pais ingresado es incorrecto"},
        "provinciaUser": {required: "Debe seleccionar una Provincia",
        rangelength: "La Provincia es incorrecta"},

    }
});

```

```

    "localidadUser":      {required: "Debe seleccionar una Localidad",
    rangelength: "La Localidad es incorrecta"},
    "direccionUser":     {required: "Debe ingresar una Direccion"},
    "nacUser":           {required: "Debe seleccionar una Fecha de Nac
imiento", validarFecha: "La Fecha ingresada es incorrecta"},
    "telefonoUser":      {required: "Debe ingresar un Telefono",
    digits: "El Telefono ingresado es incorrecto", min: "El Telefono in
gresado es incorrecto"},
    "correoUser":         {required: "Debe ingresar una Direccion de E-
Mail", email: "El E-Mail ingresado es incorrecto"},
    "generoUser":         {required: "Debe seleccionar el Genero",
    validarGenero: "El genero seleccionado es incorrecto"},
    "rolUser":            {required: "Debe seleccionar el Rol",
    validarRol: "El tipo de rol seleccionado es incorrecto"},

    "usuarioUser":        {required: "Debe ingresar un nombre de usuari
o", minlength: "El usuario debe tener al menos 4 caracteres", maxlength
: "El usuario es demasiado largo",
    "pass1":              {required: "Debe ingresar una contraseña",
    minlength: "La contraseña debe tener al menos 5 caracteres"},
    "pass2":              {required: "Debe ingresar nuevamente la contr
aseña", minlength: "La contraseña debe tener al menos 5 caracteres", equalT
o: "Las contraseñas no son iguales"}
    },

    highlight: function (element) {
        if ($(element).hasClass('select2')) {
            $(element).siblings('span').find('.selection').find('.select2
-selection').addClass('claseError');
        } else {
            $(element).addClass('claseError');
        }
    },
    unhighlight: function (element){
        if ($(element).hasClass('select2')) {
            $(element).siblings('span').find('.selection').find('.select2
-selection').removeClass('claseError');
        } else {
            $(element).removeClass('claseError');
        }
    },
    errorElement: 'li',
    errorLabelContainer: '#mensajeUser',

    submitHandler: function(){
        // Dependiendo de la clase que tenga el botón, llamamos una
determinada función.
        if($("#submitUser").hasClass("registrarUser")){
            grabarUser();
        }else if($("#submitUser").hasClass("modificarUser")){
            setMisDatosUser();
        }
    }
}

```

```

        }
    })
}

// Función para listar todas las localidades en un select una vez
seleccionada una provincia.
function mostrarLocalidades(idProv, selectLoc){

    $("#localidadUser").empty();

    $.ajax({
        url: 'scripts/aseguradora.php',
        type: 'post',
        dataType: 'json',
        data: {param:80001, id:idProv},
        success: function(r){
            if(r!=null){
                $("#localidadUser").append('<option value="0" selected disabled>Seleccione</option>');
                $.each(r.data, function(key, val){
                    if(selectLoc==val.localidadId){
                        $("#localidadUser").append('<option value="'+val.localidadId+'" selected>' +val.localidadNombre+ '</option>');
                    }else{
                        $("#localidadUser").append('<option value="'+val.localidadId+'">' +val.localidadNombre+ '</option>');
                    }
                })
            }else{
                alertify.error('Ocurrio un error al mostrar las localidades')
            }
        }
    })
}

// Función para grabar un nuevo usuario.
function grabarUser(){

    formUser = [
        $("#nombreUser").val(),
        $("#documentoUser").val(),
        $("#nacionalidadUser").val(),
        $("#provinciaUser").val(),
        $("#localidadUser").val(),
        $("#direccionUser").val(),
        $("#nacUser").val(),
        $("#telefonoUser").val(),
        $("#correoUser").val(),
        $("#generoUser").val(),
        $("#rolUser").val()
    ]

    $.ajax({
        url: 'scripts/aseguradora.php',

```

```

        type: 'post',
        dataType: 'json',
        data: {param:60000, formUser:formUser},

        success: function(r){
            if(r.success){
                email = r.email
                idUser = r.idUser
                enviarEmailRegistroUser(email, idUser, 'registro')
            }else{
                alertify.error('Ocurrio un problema al registrar el nuevo usuario')
            }
        }
    }

// Función para enviar por email los datos de acceso al nuevo usuario.
function enviarEmailRegistroUser(email, idUser, parametro)
{
    // Dependiendo del parametro, se mostrara un tipo de mensaje u otro.
    if(parametro=='registro'){
        nombreBoton = '.registrarUser';
        contenidoBoton = '<i class="fas fa-check-circle"></i> registrado';
        mensaje = 'Usuario registrado exitosamente';
    }else if(parametro=='reenviar'){
        nombreBoton = '#btnEnviarInfoAcceso';
        contenidoBoton = '<i class="fas fa-check-circle"></i> enviado';
        mensaje = 'Datos de acceso enviados correctamente.';
    }

    const html =
`La informacion para el acceso al sistema fueron enviados por email al usuario. <br>
En caso de no haber recibido la informacion, debe de ponerse en contacto con el administrador.`;

    var info = [email, idUser]

    $.ajax({
        url: 'scripts/emails/enviarEmailRegistroUser.php',
        type: 'post',
        dataType: 'json',
        data: {info:info},
        beforeSend(){
            $(nombreBoton).html('').show().
            attr('disabled', 'disabled');
            $(".volverUser").attr('disabled', 'disabled');
        },
        success: function(r){
            if(r.envCorr){
                alertify.success(mensaje);
                $(nombreBoton).html(contenidoBoton).show().attr('disabled', 'disabled');
            }
        }
    })
}

```

```

        $(".volverUser").attr('disabled', false);

        if(parametro=='reenviar'){
            $(".modalReenviarInfoAcceso").modal('close');
        }

        alertify.alert('Registro de usuarios', html, function(){
            $('#contenidoC').load('vista/users/administrarUsers.php',
function(){
                tablaListadoUsers();
            })
        }).set(configuracionAlert('ok'))
    }else{
        alertify.error('Ocurrio un problema al enviar la informacion
de acceso al usuario')
    }
}

// Función para dar de baja un usuario.
function bajaUser(idUser)
{
    // Mostramos un mensaje de confirmación antes de dar de baja el usuario.
    alertify.confirm('Anular usuario', '¿Estas seguro? El usuario ya no podra
acceder al sistema hasta que se revierta esta opcion.',
function(){

    $.ajax({
        url: 'scripts/aseguradora.php',
        dataType: 'json',
        type: 'post',
        data: {param: 60001, idUser:idUser},
        success: function(r){
            if(r.success){
                if(r.mismoEst){
                    alertify.alert('El usuario ya se encuentra dado de baja e
n el sistema');
                }else{
                    alertify.success('Usuario dado de baja correctamente');
                }
            }else
            {
                alertify.error('Ocurrio un problema al dar de baja el usuario
');
            }
        }
    });
    tablaListadoUsers.ajax.reload()
    $(".btnBajaUser, .btnModRolUser, .btnInfoUser, .btnReenviarInfoAc
cesoUser").css('display', 'none')
    $(".btnNuevoUser, .btnListadoBajas").css('display', 'inline')

    });
}

```

```

        },
        function(){}
    ).set(configuracionAlert('okCancel','Dar de baja')));
}

// Función para reactivar o dar de alta un usuario.
function reactivarUser(idUser)
{
    // Mostramos un mensaje de confirmación antes de reactivar el usuario.
    alertify.confirm('Reactivar user', '¿Estas seguro? El usuario podra ingresar nuevamente al sistema.',

        function(){

            $.ajax({
                url: 'scripts/aseguradora.php',
                dataType: 'json',
                type: 'post',
                data: {param: 60002, idUser:idUser},
                success: function(r){
                    if(r.success){
                        if(r.mismoEst){
                            alertify.alert('El usuario ya se encuentra activo en el sistema');
                        }else{
                            alertify.success('Usuario reactivado correctamente');
                        }
                    }else
                    {
                        alertify.error('Ocurrio un problema al reactivar el usuario')
                    }
                }
            });
        },
        function(){}
    ).set(configuracionAlert('okCancel','Reactivar')));
}

// Función para traer los datos de un usuario.
function traerInfoUser(idUser)
{
    // Deschequeamos algunos campos.
    $(".checkUs, .checkAdm, .enviarModRol").prop({'checked': false, 'disabled': false})

    $("#modalModificarRolUser").modal('open');

    $.ajax({
        url: 'scripts/aseguradora.php',
        type: 'post',
        dataType: 'json',
        data: {param:60003, idUser:idUser},
    })
}

```

```

success: function(r){
    if(r.success){
        // Asignamos la informacion al los campos.
        $(".num").html('#'+r.nro);
        $(".nombre").html(r.nombre);
        $(".usuario").html('@'+r.usuario);

        //Chequeamos la opcion que corresponde, pero si queremos modificar el administrador y solo existe uno, no podremos.
        if(r.rol=='Usuario'){
            $(".checkUs").prop({'checked': true, 'disabled': true});
            $(".checkAdm").prop('checked', false);
        }else if(r.rol=='Administrador'){
            if(r.cantAdm==1){
                $(".checkAdm").prop({'checked': true, 'disabled': true});

                $(".checkUs").off('click');
                $(".checkUs").on('click', function(){
                    $(".checkAdm").prop('checked', true);
                    alertify.alert('Modificar rol usuario', 'No es posible modificar el rol de '+r.nombre+' en el sistema, ya que es el unico usuario administrador.').set(configuracionAlert('ok'));
                });
            }else if (r.cantAdm>1){
                $(".checkAdm").prop({'checked': true, 'disabled': true});

                $(".checkUs").off('click');
                $(".checkUs").on('click', function(){
                    $(".checkUs").prop('checked', true);
                });
            }
        }

        //Enviamos la peticion ajax para cambiar el rol al presionar en enviar.
        $(".enviarModRol").off('click')
        $(".enviarModRol").on('click', function(){
            modRolUser(idUser, r.nombre, r.rol)
        })

        }else{
            alertify.error('Ocurrio un error al realizar la peticion al servidor')
        }
    }
}

// Función para modificar el rol de un usuario.
function modRolUser(idUser, nombre, rol)
{

```

```

if($(".checkAdm").is(':checked')){
    var nuevoRol = 'Administrador';
    texto = '{Desea convertir a '+nombre+' en administrador del sistema?'
};

}else if($(".checkUs").is(':checked')){
    var nuevoRol = 'Usuario';
    texto = '{Desea convertir a '+nombre+' solo en usuario del sistema?';
}

info = [idUser, nuevoRol];

// Si el rol seleccionado es el mismo que el que ya posee el usuario,
mostramos un mensaje.
if(nuevoRol==rol){
    alertify.alert('Modificar rol usuario', ''+nombre+' ya posee el rol
de '+rol+'.').set(configuracionAlert('ok'));
}else{
    alertify.confirm('Modificar rol usuario', texto,
    function(){
        $.ajax({
            url: 'scripts/aseguradora.php',
            type: 'post',
            dataType: 'json',
            data: {param:60004, info:info},
            success: function(re)
            {
                $(".btnBajaUser, .btnModRolUser, .btnInfoUser, .btnReenvi
arInfoAccesoUser").css('display', 'none')
                $(".btnNuevoUser, .btnListadoBajas").css('display', 'inli
ne')
                tablaListadoUsers.ajax.reload()
                $("#modalModificarRolUser").modal('close');

                if(re.success){
                    texto = ''+nombre+' ahora posee los permisos que asi
le adjudica el rol de '+nuevoRol+'.'
                    alertify.alert('Rol modificado', texto).set(configura
cionAlert('ok'))
                    alertify.success('Rol modificado correctamente')
                }else{
                    alertify.error('Ocurrio un problema al modificar el r
ol del usuario')
                }
            }
        })
    },
    function(){}
).set(configuracionAlert('okCancel','Modificar rol'));
}
}

```

```

// Función para mostrar la informacion del usuario.
function infoUser(idUser)
{
    $("#modalVerInfoUser").modal('open');

    $.ajax({
        url: 'scripts/aseguradora.php',
        type: 'post',
        dataType: 'json',
        data: {param:60003, idUser:idUser},
        success: function(r){
            if(r.success){
                $(".nume").html('#'+r.nro);
                $(".usu").html('@'+r.usuario);
                $(".estado").html(r.estado);
                $(".rol").html(r.rol);
                $(".nomb").html(r.nombre);
                $(".doc").html(r.documento);
                $(".naci").html(r.fecha_nac);
                $(".gen").html(r.genero);
                $(".correo").html(r.correo);
                $(".tel").html(r.telefono);
                $(".domi").html(r.direccion+' - '+r.localidad);
                $(".locPais").html(r.provincia+' - '+r.nacionalidad);
            }else{
                alertify.error('Ocurrio un error al realizar la peticion al s
ervidor');
            }
        }
    })
}

// Función para reenviar los datos de acceso a un usuario.
function generarReenvioDatosUser(idUser){

    texto = `¿Estas seguro? Al realizar esta accion, se generara una nueva co
ntraseña
y sera enviada junto al nombre de usuario actual a la direccion de
correo asignada.`;

    // Mostramos un mensaje de confirmación antes de realizar la petición.
    alertify.confirm('Reenviar datos de acceso', texto,
    function(){

        $(".modalReenviarInfoAcceso").modal('open');

        $.ajax({
            url: 'scripts/aseguradora.php',
            type: 'post',
            dataType: 'json',
            data: {param:60003, idUser:idUser},
            success: function(r){
                if(r.success){
                    validarEmail(0, idUser);
                }
            }
        })
    })
}

```

```

        $('.correoUs').val(r.correo);

    }else{
        alertify.error('Ocurrio un error al realizar la peticion
al servidor');
    }
}
})
},
function(){}
).set(configuracionAlert('okCancel','Aceptar', 'Cancelar'));
}

// Función para validar el campo de email, usando el plugin jquery validate.
function validarEmail(param, idUser)
{
    idUsuario = idUser;

    $("#formCorreo").validate({
        rules: {
            "correo": { required: true, email: true }
        },
        messages: {
            "correo": { required: "Debe ingresar una direccion de correo elec
tronico", email: "Debe ingresar una direccion de correo electronico valida" }
        },
        highlight: function (element) {
            $(element).css({ 'border': '1px solid #ff00005c', 'background': '#ff00000a' });
        },
        unhighlight: function (element) {
            $(element).css({ 'border': '1px solid #ccc', 'background': 'trans
parent' });
        },

        errorElement: 'label',
        errorLabelContainer: '.mjeCorr',

        submitHandler: function(form){
            if(param==0){
                reenviarInfoAccesoUser(idUsuario);
            }
        }
    })
}

// Función para enviar por email los nuevos datos de acceso al usuario.
function reenviarInfoAccesoUseridUser(){
    idUsuario = idUser;
    $.ajax({
        url: 'scripts/aseguradora.php',
        type: 'post',
        dataType: 'json',
        data: {param:60009, idUser:idUsuario},

```

```

success: function(r){
    if(r.modPassUser){

        email = $('.correoUs').val();
        nombreUser = r.nombreUser;
        enviarEmailRegistroUser(email, idUser, 'reenviar');

    }else{
        alertify.error('Ocurrio un error al realizar la peticion al s
ervidor');
    }
}
}

// Función para obtener la informacion del usuario el cual se encuentra en la
sesión actual.
function getMisDatosUser()
{
    $("#divTit").css('display', 'block');
    $("#tituloCont").html('mis datos');
    $("#divSubTit").html('');

    $('#contenidoC').load('vista/users/nuevoUser.php', function(){

        $('.registroUser .tituloFormulario').html('<i class="fas fa-user-
edit"></i> Modificar mi informacion');
        $('.registrarUser').html('<i class="fas fa-save"></i> modificar');
        // Ocultamos el campo de rol.
        $('.col.s4.rol').css('display', 'none');

        // Modificamos la clase del botón volver para poder llamarlo
correctamente
        $('.volverUser').addClass('volverMisDatos').removeClass('volverUser');

        // Modificamos la clase del botón submit a modificarUser.
        $('.registrarUser').addClass('modificarUser').removeClass('registrarUs
er');

        //Agregamos los campos de usuario y contraseña al formulario.
        htmlUserPass =
<div class="col s2">
    <div class="switch">
        <label>Modificar Contraseña</label><br>
        <label>
            <input type="checkbox" class="checkModPass">
            <span class="lever"></span>
        </label>
    </div>
</div>

<div class="col s2">
    <label for="usuarioUser">Usuario</label>

```

```

        <input type="text" class="browser-
default" name="usuarioUser" id="usuarioUser">
    </div>

    <div class="col s2">
        <label for="pass1">Contraseña nueva</label>
        <input type="password" class="browser-
default" name="pass1" id="pass1"">
    </div>

    <div class="col s2">
        <label for="pass2">Repetir</label>
        <input type="password" class="browser-
default" name="pass2" id="pass2"">
    </div> `;

    $('.UP').html(htmlUserPass);

    $('#pass1, #pass2').prop('disabled', 'disabled').css('background', '#eee');

    $(".checkModPass").change(function() {
        if($(this).is(":checked")){
            $('#pass1, #pass2').prop('disabled', false).css('background', 'transparent').val('');
        }
        else{
            $('#pass1, #pass2').prop('disabled', 'disabled').css('background', '#eee').val('');
        }
    })

    $.ajax({
        url: 'scripts/aseguradora.php',
        type: 'post',
        dataType: 'json',
        data: {param:60005},
        success: function(r){
            if(r.success){

                var genero = r.genero.toLowerCase();

                $("#nombreUser").val(r.nombre);
                $("#documentoUser").val(r.documento);
                $("#nacionalidadUser").val(r.nacionalidad);
                $("#provinciaUser").val(r.provincia).trigger('change.select2');

                mostrarLocalidades(r.provincia, r.localidad);
                $("#direccionUser").val(r.direccion);
                $("#nacUser").val(r.fecha_nac);
                $("#telefonoUser").val(r.telefono);
                $("#correoUser").val(r.correo);
                $("#generoUser").val(genero);
                $("#usuarioUser").val(r.usuario);
            }
        }
    })
}

```

```

        }else{
            alertify.error('Ocurrio un error al realizar la peticion
al servidor');
        }
    })
});
}

// Función para modificar los datos del usuario que se encuentra en la actual
sesión.
function setMisDatosUser()
{
    alertify.prompt('Modificar mis datos', 'Ingrese su contraseña', '',
    function(evt, value){

        if($(".checkModPass").is(":checked")){
            var checkeado = 'si';
        }
        else{
            var checkeado = 'no';
        }

        info = [
            checkeado,
            $("#nombreUser").val(),
            $("#documentoUser").val(),
            $("#nacionalidadUser").val(),
            $("#provinciaUser").val(),
            $("#localidadUser").val(),
            $("#direccionUser").val(),
            $("#nacUser").val(),
            $("#telefonoUser").val(),
            $("#correoUser").val(),
            $("#generoUser").val(),
            $("#usuarioUser").val(),
            $("#pass1").val(),
            $("#pass2").val(),
            value
        ]

        $.ajax({
            url: 'scripts/aseguradora.php',
            type: 'post',
            dataType: 'json',
            data: {param:60006, info:info},
            beforeSend: function(){
                $("#submitUser").html('').show().attr('disabled', 'disabled');
                $(".volverUser").attr('disabled', 'disabled');
            },
            success: function(r){
                if(r.segPass){
                    if(r.diferenteInfo){

```

```

        if(r.userDisp){
            if(r.update){
                alertify.success('Informacion modificada correctamente');
                getMisDatosUser();
            }else{
                //Ocurrio un problema al modificar la informacion.
                alertify.error('Ocurrio un problema al modificar la informacion');
                getMisDatosUser();
            }
        }else{
            // El usuario ya se encuentra en uso por otro.
            alertify.alert('Modificar mis datos', 'El usuario ya se encuentra en uso, ingrese uno diferente.').set(configuracionAlert('ok'));
        })
        $("#submitUser").html('<i class="fas fa-save"></i> reintentar').show().attr('disabled', false);
    }
    }else{
        // No se realizaron modificaciones.
        alertify.notify('No se realizaron modificaciones');
        getMisDatosUser();
    }
}else{
    // La contraseña ingresada es incorrecta.
    alertify.alert('Modificar mis datos', 'La contraseña es incorrecta, reintente.').set(configuracionAlert('ok'));
    $("#submitUser").html('<i class="fas fa-save"></i> reintentar').show().attr('disabled', false);
}
})
},
function(){}).set(configuracionAlert('password', 'Aceptar'));
}

// Función para traer la foto de perfil del usuario.
function getFotoPerfilUser()
{
$.ajax({
    url: 'scripts/aseguradora.php',
    type:'post',
    dataType:'json',
    data: {param:60007},
    success: function(r){
        if(r.existeImgPerfil=='si'){

            // Mostramos el contenido en formato HTML.
            contenido = `<div class="divActualFotoHijo">
                
            
```

```

        <a class="eliFotoUser"> <i class="fas fa-trash" title="Eliminar foto de perfil"></i></a>
            </div>
        `;
        $(".divActualFoto").html(contenido);
        $(".configuracionUser").html('<a href="#">  </a> ');
        funcionesImgUser();
    }else if(r.existeImgPerfil=='no'){
        $(".divActualFoto").html('<a href="#"> <i class="fas fa-user-circle"></i> </a>');
        $(".configuracionUser").html('<a href="#"> <i class="fas fa-user"></i> </a>');
    }
}
}

// Función donde tenemos todas las funcionalidades relacionadas a la imagen de perfil.
function funcionesImgUser(){
    // Ocultamos el icono de eliminar.
    $(".eliFotoUser").addClass('oculto');
    // Mostramos el icono de eliminar y opacamos el fondo de la imagen al pasar por encima con el mouse.
    $(".fotoActualUser").on('mouseover', function(){
        $(".eliFotoUser").removeClass('oculto');
    })
    $(".fotoActualUser").on('mouseout', function(){
        $(".eliFotoUser").addClass('oculto');
        $(".fotoActualUser").removeClass('claseOpaco');
    })
    $(".eliFotoUser").on('mouseover', function(){
        $(".fotoActualUser").addClass('claseOpaco');
        $(this).removeClass('oculto');
    })
    // Llamamos a la función de eliminar foto al cliquear sobre el icono.
    $(".fotoActualUser, .eliFotoUser").on('click', function(){
        eliminarFotoPerfilUser();
    })
}

// Función para eliminar la foto de perfil del usuario que se encuentra en la sesión actual.
function eliminarFotoPerfilUser(){
    // Mostramos un mensaje de confirmación antes de eliminar la foto.
    alertify.confirm('Eliminar foto de perfil', '¿Dese eliminar su foto de perfil?', 
        function(){
            $.ajax({
                url: 'scripts/aseguradora.php',
                type: 'post',
                dataType: 'json',

```

```

        data: {param:60008},
        success: function(r){
            if(r.eliminarFotoPerfil!=false){
                alertify.success('Foto de perfil eliminada correctamente');
            }
            // Refrescamos la foto de perfil.
            getFotoPerfilUser();
        }else{
            alertify.error('Ocurrio un problema al eliminar la foto de perfil');
        }
    });
},
function(){}
).set(configuracionAlert('okCancel','Eliminar'));

}

// Función para modificar la foto de perfil del usuario que se encuentra en la actual sesión.
function modFotoUser()
{
    // Obtenemos la informacion necesaria para enviar la imagen en formato "formData".
    var fotoPerfil = $(".nuevaFoto");
    var infoImagen = fotoPerfil.prop('files')[0];
    var formData = new FormData();
    formData.append('imagen', infoImagen);

    $.ajax({
        url: 'scripts/imagenes/procesarImgPerfil.php',
        type: 'post',
        data: formData,
        contentType: false,
        cache: false,
        processData: false,
        success: function(r){
            // Foto de perfil actualizada.
            if(r==1){
                alertify.success('Foto de perfil actualizada');
                getFotoPerfilUser();
                $('.nuevaFoto').val('');
            }
            // No se selecciono ninguna foto.
            }else if(r==10){
                alertify.alert('Cargar imagen', 'Debe seleccionar una imagen');
            }.set(configuracionAlert('ok'));
            // El formato de imagen no es valido.
            }else if(r==20){
                alertify.alert('Cargar imagen', 'El formato de la imagen no coincide con los admitidos (.JPG / .PNG / .JPEG)').set(configuracionAlert('ok'));
            }
        }
    );
}

```

```

        // El tamaño de la imagen es mayor a 5 megas.
    }else if(r==30){
        alertify.alert('Cargar imagen', 'El tamaño de la imagen es superior al maximo permitido (5 Mb)').set(configuracionAlert('ok'));
    }else if(r==40){
        alertify.error('No se ha podido registrar la imagen solicitada');
    }else if(r==50){
        alertify.error('Ha ocurrido un problema al realizar la petición al servidor');
    }else if(r==60){
        alertify.error('Ocurrio un problema interno. Reintente mas tarde.');
    }
    // Redireccionamos al inicio.
    $("#inicio").trigger('click');
}

})
}

// Función donde generamos una tabla con el listado de usuarios del sistema,
// usando el plugin DataTables.
function tablaListadoUsers()
{
    tablaListadoUsers = $('#tablaListadoUsers').DataTable({
        "language": {
            "zeroRecords": "No se encontraron usuarios registrados",
            "sSearch": "Buscar usuario",
        },
        "ajax": "scripts/tablas/tablaListadoUsuarios.php",
        "scrollY": 370,
        "bPaginate": false,
        "order": [[3, "desc"]],
        "columns": [
            { data: "boton", width: "" },
            { data: "nro", title: "Número", width: "" },
            { data: "tomador", title: "Nombre", width: "" },
            { data: "rol", title: "Rol", width: "" },
            { data: "estado", title: "Estado", width: "" }
        ],
        "dom": 'Bft',
        // Iniciamos la función select para las filas.
        "select": true,
        "select": {
            style: 'single',
            selector: 'td:first-child',
            "value": '4'
        },
        // Agregamos el checkbox en el td vacío de la tabla.
        "aoColumnDefs": [{
```

```

        orderable: false,
        className: 'selectUser select-checkbox',
        targets: 0
    ],
    // Agregamos los botones de funcionalidad de usuarios.
    "buttons": [
        {
            text: '<i class="fas fa-user-plus"></i> Nuevo',
            className: 'green btnNuevoUser'
        },
        {
            text: '<i class="fas fa-users-slash"></i> Bajas',
            className: 'blue btnListadoBajas'
        },
        {
            text: '<i class="fas fa-user-slash"></i> Baja',
            className: 'red btnBajaUser'
        },
        {
            text: '<i class="fas fa-user-shield"></i> Rol',
            className: 'yellow btnModRolUser'
        },
        {
            text: '<i class="fas fa-question-circle"></i> Info',
            className: 'blue btnInfoUser'
        },
        {
            text: '<a> Reenviar datos de acceso </a> ',
            className: 'btnReenviarInfoAccesoUser'
        }
    ]
});

// Cada vez que seleccionamos una fila, mostramos los botones.
tablaListadoUsers.on('select', function(){

    $(".btnNuevoUser, .btnListadoBajas").css('display', 'none')

    $(document).off('click', '.selectUser')
    $(document).on('click', '.selectUser', function(){
        idUser = $(this).children().attr('value')
    })

    $(".btnBajaUser, .btnModRolUser, .btnInfoUser, .btnReenviarInfoAccesoUser").css('display', 'initial');

    $(".btnBajaUser").off('click');
    $(".btnBajaUser").on('click', function () {
        bajaUser(idUser)
    });

    $(".btnModRolUser").off('click');
})

```

```

        $(".btnModRolUser").on('click', function () {
            traerInfoUser(idUser)
        });

        $(".btnInfoUser").off('click');
        $(".btnInfoUser").on('click', function () {
            infoUser(idUser)
        });

        $(".btnReenviarInfoAccesoUser").off('click');
        $(".btnReenviarInfoAccesoUser").on('click', function(){
            generarReenvioDatosUser(idUser);
        });

    })

    // Cada vez que deseleccionamos una fila, ocultamos los botones.
    tablaListadoUsers.on('deselect', function () {
        $(".btnBajaUser, .btnModRolUser, .btnInfoUser, .btnReenviarInfoAccesoUser").css('display', 'none')
        $(".btnNuevoUser, .btnListadoBajas").css('display', 'inline')
    })

    return tablaListadoUsers
}

// Función donde generamos una tabla con el listado de usuarios dados de baja, usando el plugin DataTables.
function listadoUserBaja()
{
    $("#modalListadoUserBaja").modal('open')

    tablaListadoUserBaja = $('#listadoBajaUser').DataTable({
        "language": {
            "zeroRecords": "No se encontraron usuarios dados de baja",
            "sSearch": "Buscar usuario",
        },
        "ajax": "scripts/tablas/tablaListadoUserBaja.php",
        "scrollY": 370,
        "scrollCollapse": true,
        "dom": 'ft',
        "order": [[3, "desc"]],
        "columns": [
            { data: "nro", title: "Numero", width: "" },
            { data: "tomador", title: "Nombre", width: "" },
            { data: "rol", title: "Rol", width: "" },
            { data: "estado", title: "Estado", width: "" },
            { data: "boton", title: "Reactivar", width: "" }
        ]
    });
}

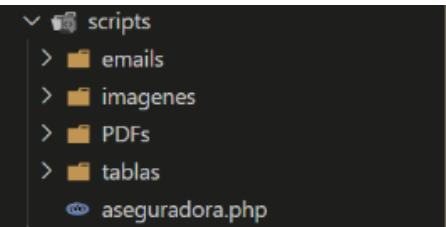
$(".cerrarModalListadoUsBa").off('click');

```

```
$( ".cerrarModalListadoUsBa" ).on('click', function(){
    $("#modalListadoUserBaja").modal('close')
    $('#listadoBajaUser').DataTable().destroy();
})

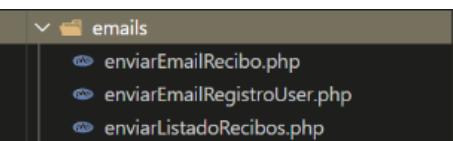
return tablaListadoUserBaja
}
```

✳ 4.3.4 Carpeta “scripts”



En esta carpeta se almacenan todos los ficheros los cuales funcionan como intermediarios entre el frontend y el backend. Funcionarian de forma similar al modelo en la arquitectura MVC.

➡ 4.3.4.1 Carpeta “emails”



Esta carpeta contiene los ficheros cuya funcionalidad se basa en enviar los elementos correspondientes por correo electrónico.

* 4.3.4.1.1 Fichero “enviarEmailRecibo.php”

Este fichero tiene como principal función, enviar por correo electrónico un recibo de pago de una cuota específica.

```
<?php

// Guardamos el fichero con el recibo en un buffer.
ob_start();
require '../vista/pagos/reciboPago.php';
$html = ob_get_clean();

// Usamos la librería MPDF de PHP para generar documentos PDF.
require_once '../Agregados/mpdf/vendor/autoload.php';

// Agregamos los estilos css para el recibo.
$css = file_get_contents('../css/recibos/recibosPDF.css');

// Configuramos la librería.
$mpdf = new \Mpdf\Mpdf([
    "format" => "A4"
]);

// Agregamos el contenido del recibo y sus estilos al documento PDF.
$mpdf->writeHTML($css, \Mpdf\HTMLParserMode::HEADER_CSS);
$mpdf->writeHTML($html, \Mpdf\HTMLParserMode::HTML_BODY);

// Generamos un nombre al archivo.
$nombreComprobante = md5(rand()).'.pdf';

// Procesamos el documento.
$listadoRecibos = $mpdf-
>output($nombreComprobante, \Mpdf\Output\Destination::STRING_RETURN);

$cont = file_put_contents($nombreComprobante, $listadoRecibos);

// Enviamos el correo electrónico
// Utilizamos la librería PHPMAILER de PHP para enviar el email.
include_once('../Agregados/phpmailer/Exception.php');
include_once('../Agregados/phpmailer/PHPMailer.php');
include_once('../Agregados/phpmailer/SMTP.php');

use PHPMailer\PHPMailer\PHPMailer;
use PHPMailer\PHPMailer\SMTP;
use PHPMailer\PHPMailer\Exception;

if (isset($_POST['info'])) {
    $info = $_POST['info'];
    $correo = $info[0];
    $tomador = $info[1];
    $nroPago = $info[2];
    $nroCuota = $info[3];
```

```

}

$arr['envCorr'] = false;
$arr['mje'] = '';
$mensaje = 'Estimado/a '.$tomador.', adjuntamos su factura #'.$nroCuota.'
, correspondiente al pago #'.$nroPago.'. Atte Aseguradora SA.';

$mail = new PHPMailer(true);

try {
    // Configuracion de los datos del emisor del correo.
    $mail->SMTPDebug = 0; //SMTP::DEBUG_SERVER;
    $mail->isSMTP();
    $mail->Host = 'smtp.gmail.com';
    $mail->SMTPAuth = true;
    $mail->Username = 'aseguradora2021@gmail.com';
    $mail->Password = 'aseguradoraSociedadAnonima2021';
    $mail->SMTPSecure = PHPMailer::ENCRYPTION_STARTTLS;
    $mail->Port = 587;

    // Configuracion del email.
    $mail->setFrom('aseguradora2021@gmail.com', 'Aseguradora SA');
    $mail->addAddress($correo);
    $mail->addAttachment($nombreComprobante, 'comprobante.pdf');

    $mail->isHTML(true);
    $mail->Subject = 'Factura';
    $mail->Body = $mensaje;

    if(!$mail->send()){
        $arr['envCorr'] = false;
        $arr['mje'] = 'Error al enviar el mensaje';
    }else{
        $arr['envCorr'] = true;
        $arr['mje'] = 'Correo enviado correctamente';
    }
} catch (Exception $e) {
    $arr['envCorr'] = false;
    $arr['mje'] = "Error al enviar el mensaje. Mailer Error: {$mail->ErrorInfo}";
}
// Una vez lo enviamos, eliminamos el documento generado.
unlink($nombreComprobante);
echo json_encode($arr);
?>

```

* 4.3.4.1.2 Fichero "enviarEmailRegistroUser.php"

Este fichero tiene como principal función, enviar por correo electrónico los datos de acceso al sistema a un usuario.

```
<?php

// Agregamos la librería PHPMAILER de PHP.
include_once('../Agregados/phpmailer/Exception.php');
include_once('../Agregados/phpmailer/PHPMailer.php');
include_once('../Agregados/phpmailer/SMTP.php');

use PHPMailer\PHPMailer\PHPMailer;
use PHPMailer\PHPMailer\SMTP;
use PHPMailer\PHPMailer\Exception;

if (isset($_POST['info'])) {
    $info = $_POST['info'];
    $correo = $info[0];
    $idUser = $info[1];
}

// Obtenemos la contraseña almacenada en la variable de sesión.
session_start();
if(isset($_SESSION['pass'])){
    $pass = $_SESSION['pass'];
}
unset($_SESSION['pass']);

// Validamos los datos.
$error = false;
$correo = filter_var($correo, FILTER_VALIDATE_EMAIL);
if($correo==FALSE || is_null($correo)) {$error = true;}
$idUser = filter_var($idUser, FILTER_VALIDATE_INT);
if($idUser==FALSE || is_null($idUser)) {$error = true;}

// Buscamos en la BBDD los datos que enviaremos (nombre, usuario y pass).
include_once('../config/db.php');
$conexion = Database::connect();
$mysqli = $conexion;
$mysqli->set_charset("utf8");
$sql = "SELECT nombre, usuario, pass FROM usuarios WHERE id = ?";
$stmt = $mysqli->prepare($sql);
if($stmt->bind_param('s', $idUser)){
    $stmt->execute();
    $r = $stmt->get_result();
    $datos = $r->fetch_assoc();
    $nombreUser = $datos['nombre'];
    $user = $datos['usuario'];
}
```

```

$arr['envCorr'] = false;
$arr['mje'] = '';
$mensaje = '<p>Estimado/a '.$nombreUser.', a continuacion se detallara la
informacion necesaria para que pueda acceder al sistema de Aseguradora SA </
p>
<p> <b> USUARIO: '.$user.' <br>CONTRASEÑA: '.$pass.' </b> </p
>
<p>Recomendamos:
<br>1) Bajo ninguna circunstancia comparta con nadie la anterior
información.
<br>2) Una vez complete el inicio de sesión, modifique la mencionada
información para así preservar la seguridad en el sistema.</p>
<p>Atte, Aseguradora SA.</p>';

$mail = new PHPMailer(true);

try {
    // Configuracion del emisor.
    $mail->SMTPDebug = 0; //SMTP::DEBUG_SERVER;
    $mail->isSMTP();
    $mail->Host      = 'smtp.gmail.com';
    $mail->SMTPAuth   = true;
    $mail->Username   = 'aseguradora2021@gmail.com';
    $mail->Password   = 'aseguradoraSociedadAnonima2021';
    $mail->SMTPSecure = PHPMailer::ENCRYPTION_STARTTLS;
    $mail->Port       = 587;

    $mail->setFrom('aseguradora2021@gmail.com', 'Aseguradora SA');
    $mail->addAddress($correo);

    // Creamos el correo.
    $mail->isHTML(true);
    $mail->Subject = 'Informacion de acceso';
    $mail->Body    = $mensaje;

    if(!$mail->send()){
        $arr['envCorr'] = false;
        $arr['mje'] = 'Error al enviar la info por email';
    }else{
        $arr['envCorr'] = true;
        $arr['mje'] = 'Info enviada por email correctamente';
    }
}

} catch (Exception $e) {
    $arr['envCorr'] = false;
    $arr['mje'] = "Error al enviar la info por email. Mailer Error: {$mai
l->ErrorInfo}";
}
echo json_encode($arr);
?>
```

* 4.3.4.1.3 Fichero "enviarListadoRecibos.php"

Este fichero tiene como principal función, enviar por correo electrónico el listado de todos los recibos de las cuotas de una poliza que ya finalizo su ciclo de vida.

```
<?php

if(!isset($_SESSION))
{
    session_start();
}

// Almacenamos en un buffer el fichero con la informacion de los recibos.
ob_start();
require '../vista/pagos/listadoRecibosParaPDF.php';
$html = ob_get_clean();

// Usamos la librería MPDF de PHP para generar un documento pdf.
require_once '../Agregados/mpdf/vendor/autoload.php';

// Llamamos el fichero con los estilos.
$css = file_get_contents('../css/recibos/recibosPDF.css');

// Configuramos la libreira.
$mpdf = new \Mpdf\Mpdf([
    "format" => "A4"
]);

// Generamos el documento pdf.
$mpdf->writeHTML($css, \Mpdf\HTMLParserMode::HEADER_CSS);
$mpdf->writeHTML($html, \Mpdf\HTMLParserMode::HTML_BODY);

// Generamos un nombre aleatorio para el documento.
$nombreListadoRecibos = md5(rand()).'.pdf';

$listadoRecibos = $mpdf-
>output($nombreListadoRecibos, \Mpdf\Output\Destination::STRING_RETURN);

// Guardamos el documento en una variable.
$cont = file_put_contents($nombreListadoRecibos, $listadoRecibos);

// Llamamos a la librería PHPMAILER de PHP para enviar el documento por
correo electrónico.
include_once '../Agregados/phpmailer/Exception.php';
include_once '../Agregados/phpmailer/PHPMailer.php';
include_once '../Agregados/phpmailer/SMTP.php';
use PHPMailer\PHPMailer\PHPMailer;
use PHPMailer\PHPMailer\SMTP;
use PHPMailer\PHPMailer\Exception;
```

```

if (isset($_POST['info'])) {
    $info = $_POST['info'];
    $correo = $info[0];
    $tomador = $info[1];
    $nroPoliza = $info[2];
}

$arr['envCorr'] = false;
$arr['mje'] = '';
$mensaje = 'Estimado/a '.$tomador.', adjuntamos los comprobantes de pago
ya emitidos correspondientes a la poliza #'.$nroPoliza.'. Atte Aseguradora SA
.';

// Realizamos el proceso de envio.
$mail = new PHPMailer(true);

try {
    $mail->SMTPDebug = false;
    $mail->isSMTP();
    $mail->Host = 'smtp.gmail.com';
    $mail->SMTPAuth = true;
    $mail->Username = 'aseguradora2021@gmail.com';
    $mail->Password = 'aseguradoraSociedadAnonima2021';
    $mail->SMTPSecure = PHPMailer::ENCRYPTION_STARTTLS;
    $mail->Port = 587;

    $mail->setFrom('aseguradora2021@gmail.com', 'Aseguradora SA');
    $mail->addAddress($correo);
    $mail->addAttachment($nombreListadoRecibos, 'recibos.pdf');

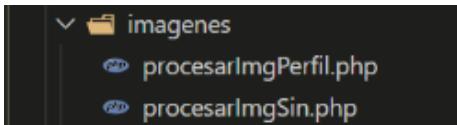
    $mail->isHTML(true);
    $mail->Subject = 'Listado de comprobantes de pago';
    $mail->Body = $mensaje;

    if(!$mail->send()){
        $arr['envCorr'] = false;
        $arr['mje'] = 'Error al enviar el mensaje';
    }else{
        $arr['envCorr'] = true;
        $arr['mje'] = 'Correo enviado correctamente';
    }
} catch (Exception $e) {
    $arr['envCorr'] = false;
    $arr['mje'] = "Error al enviar el mensaje. Mailer Error: {$mail->ErrorInfo}";
}

unlink($nombreListadoRecibos);
echo json_encode($arr);
?>

```

➔ 4.3.4.2 Carpeta “imagenes”



En esta carpeta se almacenan los ficheros cuya finalidad es procesar y grabar las imágenes que se utilicen en el sistema.

* 4.3.4.2.1 Fichero “procesarImgPerfil.php”

Este fichero tiene como principal función, grabar las imágenes de perfil subidas por los usuarios. Al momento de subir una nueva foto, si ya existe una, primero se elimina de la base de datos el registro de esta y también el archivo de la ruta, para luego pasar a grabar la nueva; En caso de que no exista ninguna foto grabada, se realiza el registro directamente.

```
<?php
    include_once('.../config/db.php');
    $conexion = Database::connect();

    $respuesta = 50;

    // Obtenemos el id del usuario de la sesión actual.
    if(!isset($_SESSION)){
        session_start();
        if(isset($_SESSION['user'])){
            $idUser = $_SESSION['user']['id'];
        }
    }

    // Obtenemos los datos de la imagen que llega por parametro.
    if(!empty($_FILES['imagen'])){

        $nombreImagen = $_FILES['imagen']['name'];
        $nomImgTemp = $_FILES['imagen']['tmp_name'];
        $tamañoImagen = $_FILES['imagen']['size'];

        $exp = explode(".", $nombreImagen);
        $ext = end($exp);
        $extPermitida = array('jpg', 'jpeg', 'png');

        // Si ya existe una foto, la eliminamos.
        $stmt0 = $conexion->
prepare("SELECT ruta FROM imgperfilusers WHERE idUser = ?");
        $existeImg = $stmt0->bind_param('i', $idUser);

        if($existeImg !== FALSE){
            $stmt0->execute();
        }
    }
}
```

```

$re = $stmt0->get_result();
$in = $re->fetch_assoc();

// Eliminamos el archivo de la ruta.
if($re->num_rows>=1 && file_exists('../'.$in['ruta'])) {

    // Y el registro de la base de datos.
    $stmt01 = $conexion-
>prepare("DELETE FROM imgperfilusers WHERE idUser = $idUser");
    if($stmt01!==FALSE){
        $stmt01->execute();
        unlink('../'.$in['ruta']);
        $stmt01->close();
    }
}

// Grabamos la nueva imagen.
if(in_array($ext, $extPermitida)){

    // Le asignamos un nombre único a la imagen y alamcenamos la
    ubicación donde será grabada en una variable.
    $imagen = time().'.'.$ext;
    $ubicacion = "../img/imgPerfilUser/".$imagen;

    // Si el archivo es menor a 5mb seguimos.
    if($tamañoImagen < 5242880){

        // Movemos el archivo a la ubicación designada antes.
        $subida = move_uploaded_file($nomImgTemp, $ubicacion);

        // Si el archivo se movio a la ubicación anterior,
        hacemos el registro en la base de datos.
        if($subida!==FALSE){
            $sql = "INSERT INTO imgperfilusers (nombre, idUser, r
uta) VALUES (?,?,?)";
            $ubicacion2 = "img/imgPerfilUser/".$imagen;
            $stmt = $conexion->prepare($sql);

            if ($stmt!==FALSE) {
                $stmt-
>bind_param('sis', $imagen, $idUser, $ubicacion2);
                $stmt->execute();
                $respuesta = true;
                $stmt->close();

            }else{
                $respuesta = 60;
            }
        }else{
            $respuesta = 40;
        }
    }else{
        $respuesta = 30;
    }
}

```

```

}else{
    $respuesta = 20;
}

$stmt0->execute();
}

}else{
$respuesta = 10;
}

echo $respuesta;
?>

```

* 4.3.4.2.2 Fichero “procesarImgSin.php”

Este fichero tiene como principal función, grabar las imágenes de un siniestro subidas por los usuarios. Al momento de subir una nueva foto, se iran registrando con diferentes nombres pero todas relacionadas al mismo id de siniestro.

```

<?php

include_once('.../.../config/db.php');
$conexion = Database::connect();

$respuesta = 50;

// Obtenemos la informacion del archivo que llega por parametro.
if(!empty($_FILES['imagen'])){

    $idSin = $_POST['idSin'];

    $nombreImagen = $_FILES['imagen']['name'];
    $nomImgTemp = $_FILES['imagen']['tmp_name'];
    $tamañoImagen = $_FILES['imagen']['size'];

    // Creamos un arreglo con los elementos del nombre de la imagen a
    // partir de cada punto.
    $exp = explode(".", $nombreImagen);

    // Accedemos al ultimo elemento del array (la extension).
    $ext = end($exp);

    $extPermitida = array('jpg', 'jpeg', 'png');

    // Si la extensión del archivo es permitida, seguimos.
    if(in_array($ext, $extPermitida)){
        $imagen = time().'.'.$ext; // Asignamos el nombre.
        $ubicacion = ".../.../img/ImgSin/".$imagen; // Ubicación.
    }
}

```

```

// Si el tamaño es menor a 5mb, seguimos.
if($tamañoImagen < 5242880){
    // Movemos el archivo a la ubicación anteriormente designada.
    $subida = move_uploaded_file($nomImgTemp, $ubicacion);

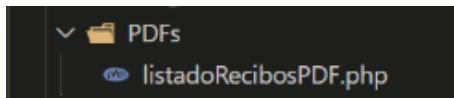
    if($subida){
        $sql = "INSERT INTO img_siniestro (nombre, idSiniestro, u
bicacion) VALUES(?, ?, ?)";
        $ubicacion2 = "img/ImgSin/".$imagen;
        $stmt = $conexion->prepare($sql);
        if ($stmt !== FALSE) {
            $stmt-
            $stmt-
>bind_param('sis', $imagen, $idSin, $ubicacion2);
            $stmt->execute();
            $stmt->close();

            $respuesta = true;
        }
    }else{
        $respuesta = 40;
    }
}else{
    $respuesta = 30;
}
}else{
    $respuesta = 20;
}
else{
    $respuesta = 10;
}

echo $respuesta;
?>

```

➔ 4.3.4.3 Carpeta “PDFs”



Esta carpeta contiene los ficheros cuya función es la de generar documentos pdfs.

* 4.3.4.3.1 Fichero “listadoRecibosPDF.php”

Este fichero tiene como función generar un documento pdf con el listado de todos los recibos de una poliza cuyo ciclo de vida fue finalizado.

```
<?php

// Guardamos en un buffer el fichero con el listado de recibos.
ob_start();
require '../vista/pagos/listadoRecibosParaPDF.php';
$html = ob_get_clean();

// Usamos la librería MPDF de PHP para generar un documento pdf.
require_once '../Agregados/mpdf/vendor/autoload.php';

// Cargamos los estilos del documento.
$css = file_get_contents('../css/recibos/recibosPDF.css');

// Configuramos algunos parámetros de la librería.
$mpdf = new \Mpdf\Mpdf([
    "format" => "A4"
]);

// Generamos el documento.
$mpdf->writeHtml($css, \Mpdf\HTMLParserMode::HEADER_CSS);
$mpdf->writeHtml($html, \Mpdf\HTMLParserMode::HTML_BODY);

// Exportamos el documento.
$mpdf->Output('Recibo-
'.md5(rand()).'.pdf', \Mpdf\Output\Destination::DOWNLOAD);
?>
```

➔ 4.3.4.4 Carpeta “tablas”

```
✓ tablas
  ↳ listadoCuotasFacturaTabla.php
  ↳ tablaAdministrarSiniestro.php
  ↳ tablaAdministrarSucursales.php
  ↳ tablaConsultarPoliza.php
  ↳ tablaConsultarSiniestro.php
  ↳ tablaFacturarPago.php
  ↳ tablaGenerarEndoso.php
  ↳ tablaListadoEndosos.php
  ↳ tablaListadoUserBaja.php
  ↳ tablaListadoUsuarios.php
  ↳ tablaModificarCuotas.php
  ↳ tablaModificarDatosTomadores.php
  ↳ tablaMostrarEndososPorPoliza.php
  ↳ tablaRenovarPoliza.php
```

Esta carpeta es la encargada de contener todos los ficheros cuya función principal, es la de obtener la información de cada una de las tablas del sistema, partiendo de una consulta a la base de datos y exportando los registros obtenidos en un formato legible para poder ser interpretados por las tablas.

* 4.3.4.4.1 Fichero “listadoCuotasFacturaTabla.php”

Este fichero tiene como función traer de la base de datos los registros de todas las cuotas de un pago de una poliza y generar las opciones para las distintas acciones para cada una.

```
<?php
    include_once('.../.../config/db.php');

    if(isset($_POST['pagoId'])){
        $idPago = $_POST['pagoId'];
    }

    $conexion = Database::connect();
    $mysqli = $conexion;
    $mysqli->set_charset("utf8");

    $tbody = array();

    date_default_timezone_set('America/Argentina/Buenos_Aires');

    // Generamos la fecha actual.
    $d = date("d");
    $m = date("m");
    $a = date("Y");
    $fechaActual = $d.'/'.$m.'/'.$a;

    $sql = "SELECT C.iden_cuota, C.numero_cuota, C.prima_mensual, C.lapso, C.
dia_cobrar, C.vto_pago, EC.nombre, C.id, EC.id
```

```

        FROM pagos P
        INNER JOIN cuotas C ON C.pago_id = P.id
        INNER JOIN poliza PO ON PO.id = P.poliza_id
        INNER JOIN estado_cuota EC ON EC.id = C.estado
        INNER JOIN tomador T ON T.id = PO.clienteid
        WHERE P.id = ?";

$stmt = $mysqli->prepare($sql);
if($stmt!==FALSE){

    $stmt->bind_param('i', $idPago);
    $stmt->execute();
    $re = $stmt->get_result();

    // Recorremos los registros.
    if($re->num_rows > 0){
        while ($cuota = $re->fetch_array()){

            $aFacturar = 0;
            $facturada = 1;
            $aprobado = 2;
            $rechazado = 3;
            $deuda = 4;

            $estadoCuota = $cuota[8];

            //if( $cuota[4] != $fechaActual )

            // Generamos los opciones dependiendo del estado de la cuota.
            if($estadoCuota == $facturada){

                $boton = '<a class="btn btn-
flat generarFactura" value="'. $cuota[7].'">
                            <span>Emitir</span>
                        </a>';
            }else if($estadoCuota == $aprobado){
                $boton = '<a class="btn btn-
flat verFactura" value="'. $cuota[7].'">
                            <i class="fas fa-search"></i>
                        </a>';
            }else if($estadoCuota == $rechazado){
                $boton = '<a class="btn btn-
flat generarFactura" value="'. $cuota[7].'">
                            <span>Aprobar</span>
                        </a>';
            }else if($estadoCuota == $deuda){
                $boton = '<a class="btn btn-
flat generarFactura" value="'. $cuota[7].'">
                            <span>Saldar</span>
                        </a>';
            }else{
                $boton = '';
            }
        }
    }
}

```

```
$primaMensual = '$'.$cuota[2] . '';  
  
$arr = array();  
$arr['idenCuota'] = $cuota[0];  
$arr['numCuota'] = $cuota[1];  
$arr['priMen'] = $primaMensual;  
$arr['lapso'] = $cuota[3];  
$arr['diaCobrar'] = $cuota[4];  
$arr['vtoPago'] = $cuota[5];  
$arr['estado'] = $cuota[6];  
$arr['boton'] = $boton;  
  
$tbody[] = $arr;  
}  
}  
$stmt->close();  
}  
  
echo json_encode($tbody)  
  
?>
```

* 4.3.4.4.2 Fichero “tablaAdministrarSiniestro.php”

Este fichero tiene como función traer de la base de datos los registros de todos los siniestros y generar las opciones para realizar las acciones correspondiente para cada uno.

```
<?php
    include_once '../config/db.php';

$conexion = Database::connect();
$mysqli = $conexion;
$tabla = array();

$sql = "SELECT
        S.id,
        S.estado,
        S.nroSiniestro,
        P.nro AS NroPol,
        T.nombre AS Tomador,
        TS.nombre AS Tipo,
        S.fechaDen AS fechDen,
        ES.nombre AS Estado
    FROM siniestro S
    INNER JOIN poliza P ON P.id = S.idPoliza
    INNER JOIN tomador T ON T.id = P.clienteid
    INNER JOIN tiposiniestro TS ON TS.id = S.tipoSiniestro
    INNER JOIN estadosiniestro ES ON ES.id = S.estado";
$stmt = $mysqli->prepare($sql);
if ($stmt!==FALSE) {
    $stmt->execute();
    $rs = $stmt->get_result();
    if ($rs->num_rows > 0){

        // Recorremos los registros.
        while ($mostrar = mysqli_fetch_row($rs)){

            // Generamos los botones.
            $btnModificarEstado = '<a class="btn btn-
flat modificarEstado" value="'.$mostrar[0].'" onclick="cambiarEstadoSin('.$mostrar[0].','.$mostrar[1].')"> <i class="fas fa-pencil-alt"></i></a>';
            $btnAdmin = '<a class="btn btn-
flat ModRestoDatos" value="'.$mostrar[0].'"> <i class="fas fa-pen"></i></a>';
            $btnImgs = '<a class="btn btn-
flat modImagenSin" value="'.$mostrar[0].'"> <i class="fas fa-
images"></i></a>';

            $arr = array();
            $arr['nroSin'] = $mostrar[2];
            $arr['nroPol'] = $mostrar[3];
            $arr['tomador'] = $mostrar[4];
            $arr['tipo'] = $mostrar[5];
            $arr['fechDen'] = $mostrar[6];
        }
    }
}
```

```
        $arr['estado'] = $mostrar[7] . '$btnModificarEstado;
        $arr['botones'] = $btnAdmin . '$btnImgS;

        $tabla[] = $arr;
    }
}

$stmt->close();
echo json_encode(array('data' => $tabla));

?>
```

* 4.3.4.4.3 Fichero "tablaAdministrarSucursales.php"

Este fichero tiene como función traer de la base de datos los registros de todas las sucursales y generar las opciones para realizar las acciones correspondiente para cada uno.

```
<?php
    include_once '../config/db.php';

$conexion = Database::connect();
$mysqli = $conexion;
$tabla = array();

$sql = "SELECT S.id,
            S.iden_sucursal,
            S.nombre,
            S.direccion,
            L.codigopostal,
            CONCAT(proper(L.nombre), ' - ', P.nombre)
        FROM sucursales S
        INNER JOIN provincia P ON P.id = S.provincia
        INNER JOIN localidad L ON L.id = S.localidad";

$stmt = $mysqli->prepare($sql);

if ($stmt!==FALSE) {
    $stmt->execute();
    $rs = $stmt->get_result();

    if ($rs->num_rows > 0)
    {
        // Recorremos los registros.
        while ($mostrar = mysqli_fetch_row($rs)){

            // Generamos los botones.
            $verSuc = '<a class="btn btn-flat" value="'.$mostrar[0].'" id="verSucursal"> <i class="fas fa-search"></i></a>';
            $modSuc = '<a class="btn btn-flat" value="'.$mostrar[0].'" id="modSucursal"> <i class="fas fa-pen"></i></a>';
            $eliSuc = '<a class="btn btn-flat" value="'.$mostrar[0].'" id="eliSucursal"> <i class="fas fa-trash"></i></a>';

            $arr = array();
            $arr['iden'] = $mostrar[1];
            $arr['nom'] = $mostrar[2];
            $arr['dir'] = $mostrar[3];
            $arr['cp'] = $mostrar[4];
            $arr['ubic'] = $mostrar[5];
            $arr['acc'] = $verSuc.' '. $modSuc.' '. $eliSuc;
        }
    }
}
```

```
        $tabla[ ]      = $arr;
    }
}

$stmt->close();
echo json_encode(array('data' => $tabla));
```

* 4.3.4.4.4 Fichero "tablaConsultarPoliza.php"

Este fichero tiene como función traer de la base de datos los registros de todas las polizas y generar las opciones para realizar las acciones correspondiente para cada uno.

```
<?php
    include_once '../config/db.php';

$conexion = Database::connect();
$mysqli = $conexion;
$tabla = array();

$sql = "SELECT
P.id,
P.nro,
CONCAT (T.nombre, ' - ', T.num_tom),
CONCAT(MA.nombre, ' ', M.nombre),
P.fecha_emision,
P.horaEmis,
CONCAT (P.vigencia_inicio, ' - ', P.vigencia_fin),
P.anulacion,
EP.nombre,
EP.descripcion
FROM poliza P
INNER JOIN tomador T ON T.id = P.clienteid
INNER JOIN vehiculo V ON V.id = P.vehiculoid
INNER JOIN marca MA ON MA.id = V.marca_id
INNER JOIN modelo M ON M.id = V.modelo_id
INNER JOIN estadopoliza EP ON EP.id = P.estado";

$stmt = $mysqli->prepare($sql);

if ($stmt!==FALSE) {
    $stmt->execute();
    $rs = $stmt->get_result();

    if ($rs->num_rows > 0)
    {
        // Recorremos los registros.
        while ($mostrar = mysqli_fetch_row($rs)){

            //Agregamos un span con la fecha en formato yyyy/mm/dd para poder ordenarlo en datatables
            $f = str_replace('/', '-', $mostrar[4]);
            $fechaOrdenar = date("Y/m/d", strtotime($f));

            $arr = array();
            $arr['nro'] = $mostrar[1];
            $arr['tomador'] = $mostrar[2];
            $arr['vehiculo'] = $mostrar[3];
        }
    }
}
```

```

        $arr['emision'] = '<span style="display: none">' . $fechaOrden
ar.'</span>' . $mostrar[4] . ' - ' . $mostrar[5];
        $arr['vigPol'] = $mostrar[6];
        $arr['anulac'] = $mostrar[7];
        $arr['estado'] = $mostrar[8];
        $arr['info'] = $mostrar[9];
        $arr['boton'] = '<a class="btn btn-
flat verConPol" value="" . $mostrar[0]."' id="verConPol"> <i class="fas fa-
search"></i></a>';

        $tabla[] = $arr;
    }
}

$stmt->close();
echo json_encode(array('data' => $tabla));

```

* 4.3.4.4.5 Fichero "tablaConsultarSiniestro.php"

Este fichero tiene como función traer de la base de datos los registros de todos los siniestros y generar el botón para visualizar su informacion.

```
<?php
    include_once '../config/db.php';

    $conexion = Database::connect();
    $mysqli = $conexion;
    $tabla = array();

    $sql = "SELECT
        S.id,
        S.nroSiniestro AS nroSin,
        P.nro AS nroPol,
        T.nombre AS tomador,
        S.fechaDen AS fechDen,
        ES.nombre AS estado
    FROM siniestro S
    INNER JOIN poliza P ON P.id = S.idPoliza
    INNER JOIN tomador T ON T.id = P.clienteid
    INNER JOIN tiposiniestro TS ON TS.id = S.tipoSiniestro
    INNER JOIN estadosiniestro ES ON ES.id = S.estado";

    $stmt = $mysqli->prepare($sql);

    if ($stmt!==FALSE) {
        $stmt->execute();
        $rs = $stmt->get_result();

        if ($rs->num_rows > 0)
        {
            // Recorremos los registros.
            while ($mostrar = mysqli_fetch_row($rs)) {

                //Agregamos un span con la fecha en formato yyyy/mm/dd para poder ordenarlo en datatables
                $f = str_replace('/', '-', $mostrar[4]);
                $fechaOrdenar = date("Y/m/d", strtotime($f));

                $arr = array();
                $arr['nroSin'] = $mostrar[1];
                $arr['nroPol'] = $mostrar[2];
                $arr['tomador'] = $mostrar[3];
                $arr['fechDen'] = '<span style="display: none">' . $fechaOrdenar . '</span>' . $mostrar[4];
                $arr['estado'] = $mostrar[5];
                $arr['boton'] = '<a class="btn btn-flat verSiniestro" id="verSiniestro" value="' . $mostrar[0] . '"><i class="fas fa-search"></i></a>';
            }
        }
    }
}
```

```
        $tabla[] = $arr;
    }
}

$stmt->close();
echo json_encode(array('data' => $tabla));
```

* 4.3.4.4.6 Fichero "tablaFacturarPago.php"

Este fichero tiene como función traer de la base de datos los registros de todos los pagos y generar las opciones para realizar las acciones correspondiente para cada uno.

```
<?php
    include_once '../config/db.php';

$conexion = Database::connect();
$mysqli = $conexion;
$tabla = array();

$sql = "SELECT PAGO.id, PAGO.num, POL.nro, CONCAT(TOM.nombre, ' - ', TOM.nombre_tom), (SELECT COUNT(*) FROM cuotas C WHERE C.pago_id = PAGO.id), EP.nombre
        FROM pagos PAGO
        INNER JOIN poliza POL ON POL.id = PAGO.poliza_id
        INNER JOIN tomador TOM ON TOM.id = POL.clienteid
        INNER JOIN estado_pago EP ON EP.id = PAGO.estado";

$stmt = $mysqli->prepare($sql);

if ($stmt!==FALSE) {
    $stmt->execute();
    $rs = $stmt->get_result();
    if ($rs->num_rows > 0)
    {
        // Recorremos los registros.
        while ($mostrar = mysqli_fetch_row($rs)){
            // Asignamos un botón dependiendo el estado del pago.
            if($mostrar[5] == 'Caducado'){
                $boton = '<a class="btn-flat verFactPago" value="'.$mostrar[0].'"><i class="fas fa-search"></i></a>';
            }else{
                $boton = '<a class="btn-flat facturarPago" value="'.$mostrar[0].'"><i class="fas fa-pen"></i></a>';
            }

            $arr = array();
            $arr['nro'] = $mostrar[1];
            $arr['polizaNum'] = $mostrar[2];
            $arr['tomador'] = $mostrar[3];
            $arr['nroCuotas'] = $mostrar[4];
            $arr['estado'] = $mostrar[5];
            $arr['boton'] = $boton;
            $tabla[] = $arr;
        }
    }
    $stmt->close();
}

echo json_encode(array('data' => $tabla));
```

* 4.3.4.4.7 Fichero "tablaGenerarEndoso.php"

Este fichero tiene como función traer de la base de datos los registros de todas las polizas con endosos realizados y generar las opciones para realizar las acciones correspondiente para cada uno.

```
<?php
    include_once '../config/db.php';

$conexion = Database::connect();
$mysqli = $conexion;
$tabla = array();

$sql = "SELECT
        P.id,
        P.nro,
        T.nombre AS Tomador,
        CONCAT(MA.nombre, ' ', M.nombre) AS Vehiculo,
        C.nombre AS Cobertura,
        P.fecha_emision,
        P.emisor,
        P.vigencia_inicio,
        P.vigencia_fin,
        EP.nombre AS estado,
        P.estado
        FROM poliza P
        INNER JOIN tomador T ON T.id = P.clienteid
        INNER JOIN vehiculo V ON V.id = P.vehiculoid
        INNER JOIN marca MA ON MA.id = V.marca_id
        INNER JOIN modelo M ON M.id = V.modelo_id
        INNER JOIN cobertura C ON C.id = P.coberturaid
        INNER JOIN estadopoliza EP ON EP.id = P.estado
        WHERE P.estado = 0 OR P.estado = 3";

$stmt = $mysqli->prepare($sql);
if ($stmt!==FALSE) {
    $stmt->execute();
    $rs = $stmt->get_result();
    if ($rs->num_rows > 0)
    {
        // Recorremos los registros.
        while ($mostrar = mysqli_fetch_row($rs)){

            $arr           = array();
            $arr['nro']     = $mostrar[1];
            $arr['tomador'] = $mostrar[2];
            $arr['vehiculo'] = $mostrar[3];
            $arr['vigin']   = $mostrar[7];
            $arr['vigfin']  = $mostrar[8];
            $arr['estado']   = $mostrar[9];
            $arr['boton']    = ' <a class="btn btn-
flat editarPol" value="'. $mostrar[0] .' "> <i class="fas fa-plus"></i></a> ';
```

```
        $tabla[]      = $arr;
    }
}
$stmt->close();
echo json_encode(array('data' => $tabla));
?>
```

* 4.3.4.4.8 Fichero "tablaListadoEndosos.php"

Este fichero tiene como función traer de la base de datos los registros de todos los endosos de una poliza y generar el botón para consultar su información.

```
<?php
    include_once '../config/db.php';

    $conexion = Database::connect();
    $mysqli = $conexion;
    $tabla = array();
    $sql = "SELECT
        P.id AS idPoliza,
        E.id,
        E.polizaNum,
        E.descripcion,
        E.polizaId,
        E.fechaRegistroEnd,
        E.horaRegistroEnd,
        TE.descripcion,
        T.nombre AS Tomador,
        EP.nombre AS Estado
    FROM endoso E
    INNER JOIN poliza P ON E.polizaId = P.id
    INNER JOIN tomador T ON P.clienteId = T.id
    INNER JOIN tipo_endoso TE ON TE.id = E.tipo
    INNER JOIN estadopoliza EP ON EP.id = P.estado
    GROUP BY P.id";

    $stmt = $mysqli->prepare($sql);
    if ($stmt!==FALSE) {
        $stmt->execute();
        $rs = $stmt->get_result();
        if ($rs->num_rows > 0)
        {
            // Recorremos los registros.
            while ($mostrar = mysqli_fetch_row($rs)){
                $arr = array();
                $arr['nro'] = $mostrar[2];
                $arr['tomador'] = $mostrar[8];
                $arr['estado'] = $mostrar[9];
                $arr['boton'] = '<a class="btn btn-
flat verEnd" value="'.$mostrar[0].'"><i class="fas fa-search"></i></a>';
                $tabla[] = $arr;
            }
        }
        $stmt->close();
        echo json_encode(array('data' => $tabla));
    }
?>
```

* 4.3.4.4.9 Fichero "tablaListadoUserBaja.php"

Este fichero tiene como función traer de la base de datos los registros de todos los usuarios dados de baja y generar el botón para poder darlo de alta.

```
<?php
    include_once '../config/db.php';

    $conexion = Database::connect();
    $mysqli = $conexion;
    $tabla = array();

    $sql = "SELECT id, nro, nombre, rol, estado FROM usuarios WHERE estado =
'Baja'";

    $stmt = $mysqli->prepare($sql);

    if ($stmt!==FALSE) {
        $stmt->execute();
        $rs = $stmt->get_result();

        // Recorremos los registros.
        if ($rs->num_rows > 0)
        {
            while ($mostrar = mysqli_fetch_row($rs)){

                $arr = array();
                $arr['nro'] = $mostrar[1];
                $arr['tomador'] = $mostrar[2];
                $arr['rol'] = $mostrar[3];
                $arr['estado'] = $mostrar[4];
                $arr['boton'] = '<a class="btn-
flat reactivarUser" value="'.$mostrar[0].'"> <i class="fas fa-user-
check"></i></a>';

                $tabla[] = $arr;
            }
        }

        $stmt->close();
        echo json_encode(array('data' => $tabla));
    }
}
```

* 4.3.4.4.10 Fichero "tablaListadoUsuarios.php"

Este fichero tiene como función traer de la base de datos los registros de todos los usuarios.

```
<?php
    include_once '../config/db.php';

    // Obtenemos el id del usuario activo en la actual sesión.
    session_start();
    if(isset($_SESSION['user'])){
        $idSesion = $_SESSION['user']['id'];
    }

    $conexion = Database::connect();
    $mysqli = $conexion;
    $tabla = array();

    $sql = "SELECT id, nro, nombre, rol, estado FROM usuarios WHERE estado =
'Activo' AND id <> ?";

    $stmt = $mysqli->prepare($sql);

    if ($stmt!==FALSE) {
        $stmt->bind_param('i', $idSesion);
        $stmt->execute();
        $rs = $stmt->get_result();

        // Recorremos los registros.
        if ($rs->num_rows > 0)
        {
            while ($mostrar = mysqli_fetch_row($rs)){

                $arr = array();
                $arr['nro'] = $mostrar[1];
                $arr['tomador'] = $mostrar[2];
                $arr['rol'] = $mostrar[3];
                $arr['estado'] = $mostrar[4];
                $arr['boton'] = '<a value="'.$mostrar[0].'"></a>';

                $tabla[] = $arr;
            }
        }
        $stmt->close();
    }

    echo json_encode(array('data' => $tabla));
```

* 4.3.4.4.11 Fichero "tablaModificarCuotas.php"

Este fichero tiene como función traer de la base de datos los registros de todos los pagos de polizas y generar el botón para modificar los parámetros.

```
<?php
    include_once '../config/db.php';

    $conexion = Database::connect();
    $mysqli = $conexion;
    $tabla = array();

    $sql = "SELECT
        P.id,
        P.nro,
        CONCAT(T.nombre, ' - ', T.num_tom) AS Tomador,
        CONCAT(MA.nombre, ' ', M.nombre) AS Vehiculo,
        P.vigencia_inicio,
        P.vigencia_fin,
        P.fecha_emision,
        EP.nombre AS estadoPa
    FROM poliza P
    INNER JOIN tomador T      ON T.id = P.clienteid
    INNER JOIN vehiculo V     ON V.id = P.vehiculoid
    INNER JOIN marca MA       ON MA.id = V.marca_id
    INNER JOIN modelo M        ON M.id = V.modelo_id
    INNER JOIN pagos PA       ON PA.poliza_id = P.id
    INNER JOIN estado_pago EP ON EP.id = PA.estado
    WHERE P.estado = 0 || P.estado = 5";

    $stmt = $mysqli->prepare($sql);
    if ($stmt !== FALSE) {
        $stmt->execute();
        $rs = $stmt->get_result();
        if ($rs->num_rows > 0)
        {
            // Recorremos los registros.
            while ($mostrar = mysqli_fetch_row($rs)){

                $arr           = array();
                $arr['nro']    = $mostrar[1];
                $arr['tomador'] = $mostrar[2];
                $arr['vehiculo'] = $mostrar[3];
                $arr['vigin']   = $mostrar[4];
                $arr['vigfin'] = $mostrar[5];
                $arr['reg']     = $mostrar[6];
                // Generamos el botón en formato HTML.
                $arr['boton']   = '<a class="btn-
flat btnAdminPago" value="'.$mostrar[0].'"><i class="fas fa-pen"></i></a>';
                $tabla[] = $arr;
            }
        }
    }
}
```

```
    }
    $stmt->close();
    echo json_encode(array('data' => $tabla));
?>
```

* 4.3.4.4.12 Fichero “tablaModificarDatosTomadores.php”

Este fichero tiene como función traer de la base de datos los registros de todos los clientes/tomadores y generar las opciones para realizar las distintas acciones para cada uno.

```
<?php
    include_once '../config/db.php';

$conexion = Database::connect();
$mysqli = $conexion;
$tabla = array();

$sql = "SELECT T.id, T.nombre, T.num_tom, P.nombre AS provincia, proper(
L.nombre) AS localidad
        FROM tomador T
        INNER JOIN provincia P ON P.id = T.provincia
        INNER JOIN localidad L ON L.id = T.localidad ";

$stmt = $mysqli->prepare($sql);
if ($stmt!==FALSE) {
    $stmt->execute();
    $rs = $stmt->get_result();
    if ($rs->num_rows > 0)
    {
        // Recorremos todos los registros.
        while ($mostrar = mysqli_fetch_row($rs)) {

            // Generamos los botones en formato HTML.
            $btnEditar = '<a class="btn btn-
flat editarDatosTom" value="'. $mostrar[0].'"><i class="fas fa-pen"></i></a>';
            $btnVer = '<a class="btn btn-
flat verTom" value="'. $mostrar[0].'"><i class="fas fa-search"></i></a>';
            $btnEliminar = '<a class="btn btn-
flat bajaTom" value="'. $mostrar[0].'"><i class="fas fa-trash-alt"></i></a>';

            $arr = array();
            $arr['nombre'] = $mostrar[1];
            $arr['num_tom'] = $mostrar[2];
            $arr['provincia'] = $mostrar[3];
            $arr['localidad'] = $mostrar[4];
            $arr['acciones'] = $btnEditar.' '. $btnVer.' '. $btnEliminar;

            $tabla[] = $arr;
        }
    }
    $stmt->close();
    echo json_encode(array('data' => $tabla));
}

?>
```

* 4.3.4.4.13 Fichero "tablaMostrarEndososPorPoliza.php"

Este fichero tiene como función traer de la base de datos los registros de todos los endosos de una poliza y generar el botón para modificar su descripción.

```
<?php
    include_once('../config/db.php');

    if(isset($_POST['idPol'])){
        $id = $_POST['idPol'];
    }

    $conexion = Database::connect();
    $mysqli = $conexion;
    $mysqli->set_charset("utf8");

    $sql = "SELECT E.polizaNum AS polizaNum, T.nombre AS tomador, E.fechaRegistroEnd, E.horaRegistroEnd,
              TE.descripcion AS tipo, SUBSTRING(E.descripcion, 1, 37) AS descripcion, E.id, E.endosoNum
        FROM endoso E
        INNER JOIN tipo_endoso TE ON TE.id = E.tipo
        INNER JOIN poliza P ON E.polizaId = P.id
        INNER JOIN tomador T ON T.id = P.clienteid
        WHERE E.polizaId = ?
        ORDER BY E.fechaRegistroEnd DESC";

    $stmt = $mysqli->prepare($sql);

    if ($stmt!==FALSE) {
        $stmt->bind_param('i', $id);
        $stmt->execute();
        $rs = $stmt->get_result();
        if ($rs->num_rows > 0)
        {
            // Recorremos los registros.
            while ($mostrar = mysqli_fetch_row($rs)){

                $arr = array();
                $arr['nro'] = $mostrar[7];
                $arr['fecha'] = $mostrar[2];
                $arr['hora'] = $mostrar[3];
                $arr['tipo'] = $mostrar[4];
                $arr['descripcion'] = $mostrar[5].'...';

                // Generamos el botón en formato HTML.
                $arr['btn'] = '<a class="btn btn-flat editarDescripcion" value="'.$mostrar[6].'"><i class="fas fa-pencil-alt"></i></a>';

                $arr1[] = $arr;
            }
        }
    }
}
```

```
        }  
    }  
  
    $stmt->close();  
    echo json_encode($arr1);  
  
?>
```

* 4.3.4.4.14 Fichero "tablaRenovarPoliza.php"

Este fichero tiene como función traer de la base de datos los registros de todas las polizas cuyo estado se encuentra vencido y generar el botón para realizar la renovacion.

```
<?php
    include_once '../config/db.php';

$conexion = Database::connect();
$mysqli = $conexion;
$tabla = array();

$sql = "SELECT
P.id,
P.nro,
CONCAT (T.nombre, ' - ', T.num_tom),
CONCAT (P.vigencia_inicio, ' - ', P.vigencia_fin),
P.fecha_emision,
P.horaEmis,
P.anulacion,
EP.descripcion,
V.suma_asegurada
FROM poliza P
INNER JOIN tomador T ON T.id = P.clienteid
INNER JOIN vehiculo V ON V.id = P.vehiculoid
INNER JOIN marca MA ON MA.id = V.marca_id
INNER JOIN modelo M ON M.id = V.modelo_id
INNER JOIN estadopoliza EP ON EP.id = P.estado
WHERE P.estado = 1";

$stmt = $mysqli->prepare($sql);

if ($stmt!==FALSE) {
    $stmt->execute();
    $rs = $stmt->get_result();

    if ($rs->num_rows > 0)
    {
        // Recorremos los registros.
        while ($mostrar = mysqli_fetch_row($rs)){

            // Agregamos un span con la fecha en formato yyyy/mm/dd para
            // poder ordenarlo en datatables.
            $f = str_replace('/', '-', $mostrar[4]);
            $fechaOrdenar = date("Y/m/d", strtotime($f));

            $arr = array();
            $arr['nro'] = $mostrar[1];
            $arr['tomador'] = $mostrar[2];
            $arr['vigPol'] = $mostrar[3];
        }
    }
}
```

```
$arr['emision'] = '<span style="display: none">' . $fechaOrden  
ar.'</span>' . $mostrar[4] . ' - ' . $mostrar[5];  
$arr['anulac'] = $mostrar[6];  
// Generamos un botón en formato HTML.  
$arr['renovar'] = '<a class="btn btn-  
flat renovarPoliza" value="'. $mostrar[0] .'" data-id="'. $mostrar[1] .'" data-  
sumAs="'. $mostrar[8] .'"> <i class="fas fa-arrow-right"></i></a>';  
  
$tabla[] = $arr;  
}  
}  
}  
  
$stmt->close();  
echo json_encode(array('data' => $tabla));
```

➔ 4.3.4.5 Fichero “aseguradora.php”

Este fichero tiene como función intermediar la vista con el servidor, el 95% de los datos del sistema pasan a través de este modelo, ya que se encarga de organizar el flujo de datos y enviarlo hacia la clase y método correspondiente, para que una vez reciba una respuesta, la devuelva directamente a la función del frontend que organiza la vista.

```
<?php

$error = true;

// Almacenamos en la variable “opcion” el valor del parametro, e incluimos el
fichero que contiene el código para realizar la conexión a la base de datos.
if (isset($_POST['param'])) {
    $opcion = $_POST['param'];

    $error = false;
    include_once('../config/db.php');
}

// Si el parametro existe (No existe error), dependiendo de su valor, a
través de una función switch, pasaremos a instanciar una clase específica y a
enviar los datos recibidos al método correspondiente, guardando los datos
devueltos por el mismo en el objeto.
if(!$error){

    switch ($opcion) {

        // Metodos de la clase “Poliza”.
        case 1000:
            include_once '../controlador/polizasController.php';
            $polizac = new PolizasController();
            $retorno = $polizac->buscarmodelos($_POST);
            break;

        case 10001:
            include_once '../controlador/polizasController.php';
            $polizac = new PolizasController();
            $retorno = $polizac->listadoCoberturas();
            break;

        case 10002:
            include_once '../controlador/polizasController.php';
            $polizac = new PolizasController();
            $retorno = $polizac->grabarVehiculo($_POST);
            break;

        case 10003:
            include_once '../controlador/polizasController.php';
            $polizac = new PolizasController();
            $retorno = $polizac->mostrarInformacion($_POST);
    }
}
```

```

        break;

    case 1004:
        include_once '../controlador/polizasController.php';
        $polizac = new PolizasController();
        $retorno = $polizac->listarPolizas();
        break;

    case 1005:
        include_once '../controlador/polizasController.php';
        $polizac = new PolizasController();
        $retorno = $polizac->renovarPoliza($_POST);
        break;

    case 1006:
        include_once '../controlador/polizasController.php';
        $polizac = new PolizasController();
        $retorno = $polizac->actualizarEstadosPoliza();
        break;

    case 1007:
        include_once '../controlador/polizasController.php';
        $polizac = new PolizasController();
        $retorno = $polizac->traerDatosPoliza($_POST);
        break;

    case 1008:
        include_once '../controlador/polizasController.php';
        $polizac = new PolizasController();
        $retorno = $polizac->modificarDatosPoliza($_POST);
        break;

    // Metodos de la clase "Endosos".
    case 2000:
        include_once '../controlador/endososController.php';
        $endosoc = new EndososController();
        $retorno = $endosoc->grabarEndoso($_POST);
        break;

    case 2001:
        include_once '../controlador/endososController.php';
        $endosoc = new EndososController();
        $retorno = $endosoc->mostrarEndososPorPoliza($_POST);
        break;

    case 2002:
        include_once '../controlador/endososController.php';
        $endosoc = new EndososController();
        $retorno = $endosoc->traerDescripcionEndoso($_POST);
        break;

    case 2003:
        include_once '../controlador/endososController.php';
        $endosoc = new EndososController();

```

```

$retorno = $endosoc->actualizarDescripcion($_POST);
break;

// Metodos de la clase “Siniestros”.
case 3000:
    include_once '../controlador/siniestrosController.php';
    $siniestrosc = new SiniestrosController();
    $retorno = $siniestrosc->traerDatosDenunciante($_POST);
    break;

case 3001:
    include_once '../controlador/siniestrosController.php';
    $siniestrosc = new SiniestrosController();
    $retorno = $siniestrosc->grabarSiniestro($_POST);
    break;

case 3002:
    include_once '../controlador/siniestrosController.php';
    $siniestrosc = new SiniestrosController();
    $retorno = $siniestrosc->modificarEstadoSiniestro($_POST);
    break;

case 3003:
    include_once '../controlador/siniestrosController.php';
    $siniestrosc = new SiniestrosController();
    $retorno = $siniestrosc->traerDatosSiniestro($_POST);
    break;

case 3004:
    include_once '../controlador/siniestrosController.php';
    $siniestrosc = new SiniestrosController();
    $retorno = $siniestrosc->actualizarDatosSin($_POST);
    break;

case 3005:
    include_once '../controlador/siniestrosController.php';
    $siniestrosc = new SiniestrosController();
    $retorno = $siniestrosc->verSiniestro($_POST);
    break;

case 3006:
    include_once '../controlador/siniestrosController.php';
    $siniestrosc = new SiniestrosController();
    $retorno = $siniestrosc->mostrarImgSin($_POST);
    break;

case 3007:
    include_once '../controlador/siniestrosController.php';
    $siniestrosc = new SiniestrosController();
    $retorno = $siniestrosc->eliminarImgSin($_POST);
    break;

// Metodos de la clase “Tomador”.
case 4000:

```

```

include_once '../controlador/tomadoresController.php';
$tomadorc = new TomadoresController();
$retorno = $tomadorc->mostrarDatosCliente($_POST);
break;

case 40001:
    include_once '../controlador/tomadoresController.php';
    $tomadorc = new TomadoresController();
    $retorno = $tomadorc->grabarCliente($_POST);
    break;

case 40002:
    include_once '../controlador/tomadoresController.php';
    $tomadorc = new TomadoresController();
    $retorno = $tomadorc->mostrarDatosTomador($_POST);
    break;

case 40003:
    include_once '../controlador/tomadoresController.php';
    $tomadorc = new TomadoresController();
    $retorno = $tomadorc->modificarDatosTomador($_POST);
    break;

case 40004:
    include_once '../controlador/tomadoresController.php';
    $tomadorc = new TomadoresController();
    $retorno = $tomadorc->eliminarTomador($_POST);
    break;

case 40005:
    include_once '../controlador/tomadoresController.php';
    $tomadorc = new TomadoresController();
    $retorno = $tomadorc->listarTomadores();
    break;

// Metodos de la clase “Pagos”.
case 50000:
    include_once '../controlador/pagosController.php';
    $pagoc = new PagosController();
    $retorno = $pagoc->generarFormularioPago($_POST);
    break;

case 50001:
    include_once '../controlador/pagosController.php';
    $pagoc = new PagosController();
    $retorno = $pagoc->modificarCuotas($_POST);
    break;

case 50002:
    include_once '../controlador/pagosController.php';
    $pagoc = new PagosController();
    $retorno = $pagoc->mostrarInfoPago($_POST);
    break;

```

```

case 50003:
    include_once '../controlador/pagosController.php';
    $pagoc = new PagosController();
    $retorno = $pagoc->mostrarCuotas($_POST);
    break;

case 50004:
    include_once '../controlador/pagosController.php';
    $pagoc = new PagosController();
    $retorno = $pagoc->facturarPago($_POST);
    break;

case 50005:
    include_once '../controlador/pagosController.php';
    $pagoc = new PagosController();
    $retorno = $pagoc->traerDatosReciboCuota($_POST);
    break;

case 50006:
    include_once '../controlador/pagosController.php';
    $pagoc = new PagosController();
    $retorno = $pagoc->efectuarPago($_POST);
    break;

case 50007:
    include_once '../controlador/pagosController.php';
    $pagoc = new PagosController();
    $retorno = $pagoc->actualizarEstadoCuota();
    break;

case 50008:
    include_once '../controlador/pagosController.php';
    $pagoc = new PagosController();
    $retorno = $pagoc->comprobarDeuda($_POST);
    break;

case 50009:
    include_once '../controlador/pagosController.php';
    $pagoc = new PagosController();
    $retorno = $pagoc->enviarRecibo($_POST);
    break;

case 50010:
    include_once '../controlador/pagosController.php';
    $pagoc = new PagosController();
    $retorno = $pagoc->mostrarListadoRecibos($_POST);
    break;

// Metodos de la clase "Usuarios".
case 6000:
    include_once '../controlador/usersController.php';
    $user = new UsersController();
    $retorno = $user->grabarUser($_POST);
    break;

```

```

case 60001:
    include_once '../controlador/usersController.php';
    $user = new UsersController();
    $retorno = $user->bajaUser($_POST);
    break;

case 60002:
    include_once '../controlador/usersController.php';
    $user = new UsersController();
    $retorno = $user->reactivarUser($_POST);
    break;

case 60003:
    include_once '../controlador/usersController.php';
    $user = new UsersController();
    $retorno = $user->traerInfoUser($_POST);
    break;

case 60004:
    include_once '../controlador/usersController.php';
    $user = new UsersController();
    $retorno = $user->modRolUser($_POST);
    break;

case 60005:
    include_once '../controlador/usersController.php';
    $user = new UsersController();
    $retorno = $user->getMisDatosUser();
    break;

case 60006:
    include_once '../controlador/usersController.php';
    $user = new UsersController();
    $retorno = $user->setMisDatosUser($_POST);
    break;

case 60007:
    include_once '../controlador/usersController.php';
    $user = new UsersController();
    $retorno = $user->getFotoPerfilUser($_POST);
    break;

case 60008:
    include_once '../controlador/usersController.php';
    $user = new UsersController();
    $retorno = $user->eliminarFotoPerfilUser($_POST);
    break;

case 60009:
    include_once '../controlador/usersController.php';
    $user = new UsersController();
    $retorno = $user->reenviarInfoAccesoUser($_POST);
    break;

```

```

// Metodos de la clase "Sucursales".
case 7000:
    include_once '../controlador/sucursalesController.php';
    $suc = new SucursalesController();
    $retorno = $suc->registrarSucursal($_POST);
    break;

case 70001:
    include_once '../controlador/sucursalesController.php';
    $suc = new SucursalesController();
    $retorno = $suc->eliminarSucursal($_POST);
    break;

case 70002:
    include_once '../controlador/sucursalesController.php';
    $suc = new SucursalesController();
    $retorno = $suc->traerInfoSucursal($_POST);
    break;

case 70003:
    include_once '../controlador/sucursalesController.php';
    $suc = new SucursalesController();
    $retorno = $suc->modificarSucursal($_POST);
    break;

case 70004:
    include_once '../controlador/sucursalesController.php';
    $suc = new SucursalesController();
    $retorno = $suc->listadoSucursalesSelect($_POST);
    break;

// Metodos de la clase "Funciones globales".
case 8000:
    include_once '../controlador/funcionesGlobalesController.php';
    $globalc = new FuncionesGlobalesController();
    $retorno = $globalc->listadoLocalidadesPorCp($_POST);
    break;

case 80001:
    include_once '../controlador/funcionesGlobalesController.php';
    $globalc = new FuncionesGlobalesController();
    $retorno = $globalc->listadoLocalidadesPorProvincia($_POST);
    break;

case 80002:
    include_once '../controlador/funcionesGlobalesController.php';
    $globalc = new FuncionesGlobalesController();
    $retorno = $globalc->listadoCpPorProvincia($_POST);
    break;

//-----SISTEMA-----
case 9000:
    include_once '../controlador/sistemaController.php';

```

```
$login = new SistemaController();
$retorno = $login->loginSistema($_POST);
break;

case 90001:
    include_once '../controlador/sistemaController.php';
    $login = new SistemaController();
    $retorno = $login->logoutSistema();
    break;

}

// Devolvemos los datos desde la variable retorno.
echo $retorno;

// Si no existe el parametro, se devolvera el error.
}else{
    echo 'Error aseguradora.php';
}
```

✿ 4.3.5 Carpeta “vista”

```
✓ vista
  > endosos
  > maqueta
  > pagos
  > poliza
  > siniestro
  > sucursales
  > tomador
  > users
```

En esta carpeta se almacenan todos los ficheros que corresponden a las vistas del sistema, es decir, todos los formularios y elementos con los que el usuario estará interaccionando directamente.

➔ 4.3.5.1 Carpeta “endosos”

```
✓ endosos
  > divsEndoso
  > elementosEndoso
```

Ficheros relacionados a la vista de los endosos.

* 4.3.5.1.1 Carpeta “divsEndoso”

```
✓ divsEndoso
  > coberturaModificar.php
  > polizaModificar.php
  > tomadorModificar.php
  > vehiculoModificar.php
```

➤ 4.3.5.1.1.1 Fichero “coberturaModificar.php”

```
// Estructura del formulario donde mostramos los tipos de coberturas.
<fieldset id="coberturaField">
    <div class="subtituloFormulario" id="tituloCoberturaEndoso"></div>

    <select class="campo select_cobertura" id="select_cobertura" name="sele
ct_cobertura">
        <option value="0" selected disabled>Seleccione</option>
        // Incluimos el fichero con el listado de coberturas.
        <?php include_once "../poliza/elementosPoliza/mostrarCoberturas.p
hp"; ?>
    </select>

    <div id="infoCobertura" class="col s12"></div>
</fieldset>
```

> 4.3.5.1.1.2 Fichero “polizaModificar.php”

```
// Formulario para ingresar los datos de la sección poliza.
<fieldset class="border p-2" id="polizaField">
    <legend class="w-auto">Endosos Datos de Poliza</legend>

    <form>
        <div class="form-row">

            <div class="form-group col-md-3">
                <fieldset class="border p-2">
                    <label for="fechaPol">Emision de poliza</label>
                    <input type="text" readonly class="form-control-
plaintext" name="fechaPol" id="fechaPol" value="">
                </fieldset>
            </div>

            <div class="form-group col-md-3">
                <fieldset class="border p-2">
                    <label for="horaEmPo">Hora</label>
                    <input type="text" readonly class="form-control-
plaintext" name="horaEmPo" id="horaEmPo" value="">
                </fieldset>
            </div>

            <div class="form-group col-md-3">
                <fieldset class="border p-2">
                    <label for="fechaEm">Emision de endoso</label>
                    <?php date_default_timezone_set("America/Argentina/Buenos_Aires")?>
                    <input type="text" readonly class="form-control-
plaintext" name="fechaEm" id="fechaEm" value="<?=date("d/m/Y")?>">
                </fieldset>
            </div>

            <div class="form-group col-md-3">
                <fieldset class="border p-2">
                    <label for="horaEm">Hora</label>
                    <?php date_default_timezone_set("America/Argentina/Buenos_Aires")?>
                    <input type="text" readonly class="form-control-
plaintext" name="horaEm" id="horaEm" value="<?= date("H:i a")?>">
                </fieldset>
            </div>
        </div>

        <div class="form-row">

            <div class="form-group col-md-2">
                <label for="vig_ini">Inicio vigencia</label>
                <input type="text" class="form-control form-control-
sm" id="vig_ini" name="vig_ini" placeholder="Seleccione">
                <div id="fechInVigR"></div>
            </div>
        </div>
    </form>
</fieldset>
```

```

<div class="form-group col-md-2">
    <label for="vig_finEndoso">Fin de vigencia</label>
    <input type="text" class="form-control form-control-sm" id="vig_finEndoso" name="vig_finEndoso" placeholder="Seleccione">
        <div id="fechFinVigR"></div>
    </div>

    <div class="form-group col-md-2">
        <label for="horaInVig">Hora</label>
        <input type="text" class="form-control form-control-sm" id="horaInVig" name="horaInVig" placeholder="Seleccione">
            <div id="horaInVigR"></div>
        </div>
    </div>
</form>
</fieldset>

```

► 4.3.5.1.1.3 Fichero “tomadorModificar.php”

```

// Formulario de la sección tomador.
<fieldset id="tomadorField">
    <div class="subtituloFormulario" id="tituloTomadorEndoso"></div>
    // Incluimos el fichero con el formulario del tomador.
    <?php require_once '../poliza/formularios/formularioTomador.php' ?>
</fieldset>

<script type="text/javascript" src="js/cliente.js?v=13"></script>

```

> 4.3.5.1.1.4 Fichero “vehiculoModificar.php”

```
// Instaciamos las clases para utilizar la informacion de los métodos.  
<?php  
    include_once('../.../controlador/polizasController.php');  
    include_once('../.../controlador/funcionesGlobalesController.php');  
    include_once('../.../config/db.php');  
    $objSelect = new PolizasController();  
    $objFg = new FuncionesGlobalesController();  
?>  
  
<!--<form id="divVehiculoEndoso" class="formulario">-->  
    <fieldset id="vehiculoField">  
  
        <div class="subtituloFormulario" id="tituloDatosVehiculo"></div>  
  
        <div class="row">  
            <div class="col s2">  
                <label for="suma_asegurada">Suma Asegurada</label>  
                <input type="text" id="suma_asegurada" class="campo" placeholder="560  
.000" name="suma_asegurada">  
            </div>  
  
            <div class="col s3">  
                <label for="select_marca">Marca</label>  
                <select class="campo select2" id="select_marca" name="select_marca">  
                    // Listamos las marcas desde el objeto.  
                    <?= $objSelect->listadoMarcas() ?>  
                </select>  
            </div>  
  
            <div class="col s3">  
                <label for="select_modelo1">Modelo</label>  
                <select class="campo select2" id="select_modelo1" name="select_modelo  
1">  
                    <option value="0" disabled selected>Seleccionar Marca</option>  
                </select>  
            </div>  
  
            <div class="col s2">  
                <label for="patente">Patente</label>  
                <input type="text" id="patente" class="campo" placeholder="ABC 123" m  
axlength="9" name="patente">  
            </div>  
  
            <div class="col s2">  
                <label for="motor">Motor</label>  
                <input type="text" id="motor" class="campo" placeholder="2.0 TSI 20V  
230CV" name="motor">  
            </div>  
        </div>
```

```

<div class="row">
    <div class="col s3">
        <label for="nroChasis">Nro Chasis</label>
        <input type="text" id="nroChasis" class="campo" placeholder="B2N5R6S9
E9R6VJ5T8" name="nroChasis">
    </div>

    <div class="col s2">
        <label for="nroMotor">Nro Motor</label>
        <input type="text" id="nroMotor" class="campo" placeholder="GH323JK
" name="nroMotor">
    </div>

    <div class="col s3">
        <label for="select_clase">Clase</label>
        <select class="campo" id="select_clase" name="select_clase">
            // Listamos las clases de vehiculo desde el objeto.
            <?= $objSelect->listadoClases() ?>
        </select>
    </div>

    <div class="col s4">
        <label for="selectTipoUso">Uso</label>
        <select class="campo" id="selectTipoUso" name="selectTipoUso">
            // Listamos los tipos de uso desde el objeto.
            <?= $objSelect->listadoUsos() ?>
        </select>
    </div>
</div>

<div class="row">
    <div class="col s1">
        <label for="GPS">GPS</label>
        <div class="switch">
            <label> <input type="checkbox" id="GPS" class="filled-
in"> <span class="filled-in"></span> </label>
        </div>
    </div>

    <div class="col s1">
        <label for="ceroKm">0 Km</label>
        <div class="switch">
            <label> <input type="checkbox" id="ceroKm" class="filled-
in"> <span class="filled-in"></span> </label>
        </div>
    </div>

    <div class="col s2">
        <label for="cant_kilometro">Kilometros</label>
        <input type="text" id="cant_kilometro" class="campo" placeholder="124
.000" name="cant_kilometro">
    </div>

```

```

<div class="col s2">
    <label for="codigoPostal">Codigo Postal</label>
    <select class="campo select2" id="codigoPostal" name="codigoPostal">
        // Listamos los códigos postales desde el objeto.
        <?= $objFg->listadoCp() ?>
    </select>
</div>

<div class="col s4">
    <label for="vehiculoLocalidad">Localidad</label>
    <select class="campo select2" id="vehiculoLocalidad" name="vehiculoLo
calidad">
        <option value="0" selected disabled>Seleccionar Codigo Postal</opti
on>
    </select>
</div>
</div>

<div class="row">

    <div class="col s3">
        <label for="coberturaAd">Coberturas Adicionales</label>
        <select class="campo" id="coberturaAd" name="coberturaAd">
            // Listamos las coberturas adicionales desde el objeto.
            <?= $objSelect->listadoCoberturasAd() ?>
        </select>
    </div>

    <div class="col s2">
        <label for="combustible">Combustible</label>
        <select class="campo" id="combustible" name="combustible">
            <option value="0" selected disabled>Seleccione</option>
            <option value="Nafta">Nafta</option>
            <option value="GasOil">Gas Oil</option>
            <option value="Energia Electrica">Energia Electrica</option>
        </select>
    </div>

    <div class="col s1">
        <label for="eGas">Gas</label>
        <div class="switch">
            <label> <input type="checkbox" id="eGas" class="filled-
in"> <span class="filled-in"></span> </label>
        </div>
    </div>

    <div class="col s2">
        <label for="anio">Año</label>
        <input type="number" id="anio" class="campo" placeholder="2017" name=
"anio">
    </div>

    <div class="col s1">
        <label for="pasajeros">Pasajeros</label>

```

```
<input type="number" id="pasajeros" class="campo" placeholder="5" name="pasajeros">
</div>

<div class="col s1">
    <label for="asientos">Asientos</label>
    <input type="number" id="asientos" class="campo" placeholder="5" name="asientos">
</div>

<div class="col s2">
    <label for="color">Color</label>
    <input type="text" id="color" class="campo" placeholder="Blanco" name="color">
</div>

</div>

</fieldset>
<!--</form>-->

<script type="text/javascript" src="js/poliza.js?v=3"></script>
```

* 4.3.5.1.2 Carpeta "elementosEndoso"

```
✓ elementosEndoso
  - endosos.php
  - listadoEndosos.php
```

> 4.3.5.1.2.1 Fichero "endosos.php"

```
// Instanciamos la clase para usar los datos de los métodos.
<?php
    include_once('../.....controlador/endososController.php');
    include_once('../.....config/db.php');
    $objEnd = new EndososController();
?>

// Estructura de la tabla donde muestro la lista de polizas registradas.
<div class="row" id="consultarPolRow">
    <div class="col s12">
        <table id="mostrarPolizasTabla" class="centered striped">
            <thead>
                <tr>
                    <th scope="col">Nro</th>
                    <th scope="col">Tomador</th>
                    <th scope="col">Riesgo</th>
                    <th scope="col">Inicio Vig</th>
                    <th scope="col">Fin Vig</th>
                    <th scope="col">Estado</th>
                    <th scope="col">Mod</th>
                </tr>
            </thead>
            <tbody id=""></tbody>
        </table>
    </div>
</div>

// Modal donde seleccionamos el tipo de endoso.
<div id="modalSeleccionarEnd" class="modal modal-fixed-footer">
    <div class="modal-content formulario">
        <h5 class="tituloFormulario">Tipo de Endoso a registrar</h5>
        <select id="selectTipoEndoso" > <?= $objEnd->
listadoTipoEndosos() ?> </select>
    </div>
    <div class="modal-footer">
        <button type="button" class="btn btn-flat blue modal-
close" id="cancelarModalTipoEnd"> <i class="fas fa-
times"></i> cancelar </button>
        <button type="button" class="btn btn-
flat green" id="continuarModalTipoEnd"> continuar <i class="fas fa-caret-
right"></i> </button>
    </div>
</div>
```

```

        </div>

// Modal donde mostramos el formulario para modificar los datos del vehiculo.
<div id="modalEndosoVehiculo" class="modal modal-fixed-footer">
    <form id="formularioEndosoV" class="formulario">
        <div class="modal-content">
            <div class="tituloFormulario">Endosos poliza: <span class="numeroPol"></span></div>
            <div id="cuerpoModalEndoso" class="modalEndoso">

                // Incluimos el fichero donde tenemos el formulario de vehiculo.
                <?php include_once '../endosos/divsEndoso/vehiculoModificar.php'?
            </div>
            <ul id="mensajeEndoso" class="mensajeModal"></ul>
        </div>
        <div class="row modal-footer">
            <button type="button" class="btn btn-flat blue" id="cancelarCambiosEndosoV"> <i class="fas fa-times"></i> cancelar </button>
            <button type="submit" class="btn btn-flat green" id="guardarCambiosEndosoV"> <i class="fas fa-save"></i> guardar </button>
        </div>
    </form>
</div>

// Modal donde mostramos la sección para modificar el tomador.
<div id="modalEndosoTomador" class="modal modal-fixed-footer">
    <form id="formularioEndosoT" class="formulario">
        <div class="modal-content">
            <div class="tituloFormulario">Endosos poliza: <span class="numeroPol"></span></div>
            <div id="cuerpoModalEndoso" class="modalEndoso">

                // Incluimos el fichero donde tenemos el formulario de tomador.
                <?php include_once '../endosos/divsEndoso/tomadorModificar.php'?
            </div>
            <ul id="mensajeEndoso" class="mensajeModal"></ul>
        </div>
        <div class="row modal-footer">
            <button type="button" class="btn btn-flat blue" id="cancelarCambiosEndosoT"> <i class="fas fa-times"></i> cancelar </button>
            <button type="submit" class="btn btn-flat green" id="guardarCambiosEndosoT"> <i class="fas fa-save"></i> guardar </button>
        </div>
    </form>
</div>

// Modal donde mostramos la sección para modificar el tipo de cobertura.

```

```
<div id="modalEndosoCobertura" class="modal modal-fixed-footer">
    <form id="formularioEndosoC" class="formulario">
        <div class="modal-content">
            <div class="tituloFormulario">Endosos poliza: <span class="numeroPol">
        ></span></div>
        <div id="cuerpoModalEndoso" class="modalEndoso">

            // Incluimos el fichero donde tenemos el formulario de cobertura.
            <?php include_once '../endosos/divsEndoso/coberturaModificar.php' ?>
        </div>
    </div>
    <div class="row modal-footer">
        <button type="button" class="btn btn-flat blue" id="cancelarCambiosEndosoC"> <i class="fas fa-times"></i> cancelar </button>
        <button type="submit" class="btn btn-flat green" id="guardarCambiosEndosoC"> <i class="fas fa-save"></i> guardar </button>
    </div>
</form>
</div>

<script type="text/javascript" src="js/cliente.js"></script>
<script type="text/javascript" src="js/endosos.js"></script>
```

> 4.3.5.1.2.2 Fichero “listadoEndosos.php”

```
// Estructura de la tabla donde mostramos las polizas con endosos realizados.
<div class="row" id="consultarPolRow">
    <div class="col s12">
        <table id="listadoEndososTabla" class="centered striped">
            <thead>
                <tr>
                    <th scope="col">Nro de poliza</th>
                    <th scope="col">Tomador</th>
                    <th scope="col">Estado</th>
                    <th scope="col">Ver</th>
                </tr>
            </thead>
            <tbody></tbody>
        </table>
    </div>
</div>

// Modal donde se muestra el listado de endosos por poliza.
<div id="modalListadoDeEndososPorPoliza" class="modal modal-fixed-footer">
    <div class="modal-content">
        <div class="subtituloFormulario">Endosos <span id="numeroPolizaEn">Poliza: <span id="numeroPolListado"></span> </span></div>
        <div id="cuerpoModalListadoEndosoPorPoliza">
            // Tabla donde mostramos el listado de endosos de una poliza.
            <table id="mostrarEndososPorPoliza" class="mostrarEndososPorPoliza centered striped">
                <thead>
                    <tr>
                        <th scope="col">Num</th>
                        <th scope="col">Fecha</th>
                        <th scope="col">Hora</th>
                        <th scope="col">Tipo</th>
                        <th scope="col">Descripcion</th>
                        <th scope="col"></th>
                    </tr>
                </thead>
                <tbody id="contenidoTablaEn"></tbody>
            </table>
        </div>
    </div>
    <div class="modal-footer">
        <button type="button" class="btn btn-flat blue" id="cerrarEndososPorPoliza"> <i class="fas fa-times"></i> cerrar </button>
    </div>
</div>
```

```
//Modal donde mostramos la descripcion a modificar del endoso seleccionado.
<div id="modalDescripcionEndoso" class="modal modal-fixed-footer">
    <div class="modal-content">
        <div class="subtituloFormulario" id="tituloModalDescripcion">Descripcio
n <span id="numeroPolizaEn">Endoso: <span id="numeroEndDesc"></span> </span>
</div>
        <label for="desc"></label>
        <textarea class="form-control" id="desc" rows="5"></textarea>
    </div>
    <div class="modal-footer">
        <button type="button" class="btn btn-flat blue modal-
close" id="cancelarDescEnd"> <i class="fas fa-times"></i> cancelar </button>
        <button type="button" class="btn btn-
flat green" id="aceptarDescEnd"> <i class="fas fa-
save"></i> guardar </button>
    </div>
</div>

<script type="text/javascript" src="js/endosos.js?v=3"></script>
```

➔ 4.3.5.2 Carpeta “maqueta”

```
maqueta
└── footerScripts.php
    └── linksHead.php
        └── mainContainer.php
            └── sidebarAdmin.php
                └── sidebarUser.php
                    └── topNavbar.php
```

En esta carpeta se encuentran todos los ficheros relacionados a la confección de la vista general o maqueta del sistema.

* 4.3.5.2.1 Fichero “*footerScripts.php*”

```
// Incluimos la librería jQuery.
<script type="text/javascript" src="js/plugins/jQuery/jquery-
3.4.1.min.js"></script>

// Incluimos el script de menú.
<script type="text/javascript" src="js/menu.js?v=4"></script>
// Incluimos el script Login.
<script type="text/javascript" src="js/login.js?v=1"></script>
// Incluimos el script User.
<script type="text/javascript" src="js/users.js?v=1"></script>

// Incluimos el framework materialize css.
<script type="text/javascript" src="js/plugins/Materialize/materialize.min.js
"></script>

// Incluimos el script de Alertify js.
<script type="text/javascript" src="js/plugins/Alertify JS/alertify.min.js"><
/script>

// Incluimos el script de Font awesome.
<script type="text/javascript" src="js/plugins/FontAwesome/fontAwesome.js"></
script>

// Incluimos la librería select2 de jQuery.
<script src="js/plugins/select2/js/select2.min.js"></script>

// Incluimos la librería DatePicker.
<script type="text/javascript" src="js/plugins/DatePicker/js/jquery-
ui.js"></script>

// Incluimos la librería DataTables.
<script type="text/javascript" src="js/plugins/DataTables/datatables.min.js">
</script>
```

```
// Incluimos el plugin jQuery Validation.  
<script type="text/javascript" src="js/plugins/jQuery Validation Plugin/dist/  
jquery.validate.min.js"></script>  
<script type="text/javascript" src="js/plugins/jQuery Validation Plugin/dist/  
additional-methods.min.js"></script>  
  
// Incluimos la librería Lightbox js para imagenes.  
<script type="text/javascript" src="js/plugins/LightBox JS/js/lightbox.min.js  
></script>  
<script type="text/javascript" src="js/plugins/LightBox JS/js/app.js"></scrip  
t>  
  
// Incluimos la librería Moment js para trabajar con fechas.  
<script type="text/javascript" src="js/plugins/Moment/moment.min.js"></script  
>
```

* 4.3.5.2.2 Fichero "linksHead.php"

```
// Estilos CSS de Materialize.  
<link rel="stylesheet" href="js/plugins/Materialize/styles.css">  
  
// Estilos para Select2 (Libreria JQuery).  
<link rel="stylesheet" href="js/plugins/select2/css/select2.min.css"/>  
  
// Estilos para Alertify.  
<link rel="stylesheet" href="js/plugins/Alertify JS/styles.css"/>  
  
// Estilos para Date Picker.  
<link rel="stylesheet" href="js/plugins/DatePicker/css/jquery-ui.min.css">  
  
// Estilos para DataTables.  
<link rel="stylesheet" type="text/css" href="js/plugins/DataTables/datatables.min.css">  
  
// Estilos CSS propios.  
<link rel="stylesheet" href="css/estilos.css">  
  
// Estilos para la libreria Lightbox de imágenes.  
<link rel="stylesheet" href="js/plugins/LightBox JS/css/lightbox.min.css">
```

* 4.3.5.2.3 Fichero "mainContainer.php"

```
// Estructura de la vista del contenedor del sistema (Main container).
<div class="main_container">
    <div class="contenido" id="contenido">
        <div class="menuUser">
            <ul>
                <div class="nombreUser">
                    // Mostramos el nombre del usuario de la actual sesión.
                    <?= $_SESSION['user']['nombre'] ?>
                </div>
                <hr>
                <li class="modFotoUser">
                    Foto de perfil
                    <span class="icoMenUs">
                        <i class="fas fa-user-circle"></i>
                    </span>
                </li>
                <li class="misDatos">
                    Mis datos
                    <span class="icoMenUs">
                        <i class="fas fa-user-edit"></i>
                    </span>
                </li>
                <li class="logout">
                    Salir del sistema
                    <span class="icoMenUs">
                        <i class="fas fa-sign-out-alt"></i>
                    </span>
                </li>
            </ul>
        </div>

        <div id="divTit">
            <h2 class="linea">
                <span id="tituloCont"></span>
            </h2>
        </div>

        <div id="divSubTit">
            <span id="subTituloCont"></span>
        </div>

        // Div donde se cargara todo el contenido central del sistema.
        <div id="contenidoC"></div>
    </div>
</div>
```

* 4.3.5.2.4 Fichero "sidebarAdmin.php"

```
// Estructura del menú principal o sidebar correspondiente al administrador.  
<ul>  
    <!--Inicio-->  
    <li class="opMen">  
        <a href="#" id="inicio" class="active">  
            <span class="icon"><i class="fas fa-home"></i></span>  
            <span class="title">Inicio</span>  
        </a>  
    </li>  
  
    <!--Poliza-->  
    <li class="opMen">  
        <a href="#" id="polizas">  
            <span class="icon"><i class="fas fa-file-contract"></i></span>  
            <span class="title">Polizas</span>  
            <span class="flecha"><i class="fas fa-angle-down"></i></span>  
        </a>  
        <div class="sub">  
            <ul class="sub-ul">  
                <li class="sub-ul-li">  
                    <a id="nuevaPoliza" class="sub-ul-li-a">  
                        <span class="submenuIco"><i class="fas fa-plus"></i></span>  
                        <span class="submenuTit">Emitir</span>  
                    </a>  
                </li>  
                <li class="sub-ul-li">  
                    <a id="verPoliza" class="sub-ul-li-a">  
                        <span class="submenuIco"><i class="fas fa-eye"></i></span>  
                        <span class="submenuTit">Ver</span>  
                    </a>  
                </li>  
                <li class="sub-ul-li">  
                    <a id="renovarPoliza" class="sub-ul-li-a">  
                        <span class="submenuIco"><i class="fas fa-sync-alt"></i></span>  
                        <span class="submenuTit">Renovaciones</span>  
                    </a>  
                </li>  
            </ul>  
        </div>  
    </li>  
  
    <!--Endoso-->  
    <li class="opMen">  
        <a href="#" id="endosos">  
            <span class="icon"><i class="fas fa-file-signature"></i></span>  
            <span class="title">Endosos</span>  
            <span class="flecha"><i class="fas fa-angle-down"></i></span>  
        </a>  
    </li>
```

```

        </a>
    <div class="sub">
        <ul class="sub-ul">
            <li class="sub-ul-li">
                <a id="emitirEndoso" class="sub-ul-li-a">
                    <span class="submenuIco"><i class="fas fa-plus"></i></span>
                    <span class="submenuTit">Emitir</span>
                </a>
            </li>
            <li class="sub-ul-li">
                <a id="verEndoso" class="sub-ul-li-a">
                    <span class="submenuIco"><i class="fas fa-eye"></i></span>
                    <span class="submenuTit">Ver</span>
                </a>
            </li>
        </ul>
    </div>
</li>

<!--Siniestro--&gt;
&lt;li class="opMen"&gt;
    &lt;a href="#" id="siniestros"&gt;
        &lt;span class="icon"&gt;&lt;i class="fas fa-car-crash"&gt;&lt;/i&gt;&lt;/span&gt;
        &lt;span class="title"&gt;Siniestros&lt;/span&gt;
        &lt;span class="flecha"&gt;&lt;i class="fas fa-angle-down"&gt;&lt;/i&gt;&lt;/span&gt;
    &lt;/a&gt;
    &lt;div class="sub"&gt;
        &lt;ul class="sub-ul"&gt;
            &lt;li class="sub-ul-li"&gt;
                &lt;a id="nuevoSiniestro" class="sub-ul-li-a"&gt;
                    &lt;span class="submenuIco"&gt;&lt;i class="fas fa-plus"&gt;&lt;/i&gt;&lt;/span&gt;
                    &lt;span class="submenuTit"&gt;Emitir&lt;/span&gt;
                &lt;/a&gt;
            &lt;/li&gt;
            &lt;li class="sub-ul-li"&gt;
                &lt;a id="adminSiniestro" class="sub-ul-li-a"&gt;
                    &lt;span class="submenuIco"&gt;&lt;i class="fas fa-pencil-alt"&gt;&lt;/i&gt;&lt;/span&gt;
                    &lt;span class="submenuTit"&gt;Administrar&lt;/span&gt;
                &lt;/a&gt;
            &lt;/li&gt;
            &lt;li class="sub-ul-li"&gt;
                &lt;a id="verSiniestro" class="sub-ul-li-a"&gt;
                    &lt;span class="submenuIco"&gt;&lt;i class="fas fa-eye"&gt;&lt;/i&gt;&lt;/span&gt;
                    &lt;span class="submenuTit"&gt;Ver&lt;/span&gt;
                &lt;/a&gt;
            &lt;/li&gt;
        &lt;/ul&gt;
    &lt;/div&gt;
&lt;/li&gt;
</pre>

```

```

<!--Tomador-->
<li class="opMen">
    <a href="#" id="tomadores">
        <span class="icon"><i class="fas fa-user-cog"></i></span>
        <span class="title">Tomadores</span>
        <span class="flecha"><i class="fas fa-angle-down"></i></span>
    </a>
    <div class="sub">
        <ul class="sub-ul">
            <li class="sub-ul-li">
                <a id="nuevoTomador1" class="sub-ul-li-a">
                    <span class="submenuIco"><i class="fas fa-plus"></i></span>
                    <span class="submenuTit">Nuevo</span>
                </a>
            </li>
            <li class="sub-ul-li">
                <a id="adminTomador" class="sub-ul-li-a">
                    <span class="submenuIco"><i class="fas fa-pencil-alt"></i></span>
                    <span class="submenuTit">Administrar</span>
                </a>
            </li>
        </ul>
    </div>
</li>

<!--Pago-->
<li class="opMen">
    <a href="#" id="pagos">
        <span class="icon"><i class="fas fa-money-check-alt"></i></span>
        <span class="title">Pagos</span>
        <span class="flecha"><i class="fas fa-angle-down"></i></span>
    </a>
    <div class="sub">
        <ul class="sub-ul">
            <li class="sub-ul-li">
                <a id="facturarPago" class="sub-ul-li-a">
                    <span class="submenuIco"><i class="fas fa-money-check"></i></span>
                    <span class="submenuTit">Facturar</span>
                </a>
            </li>
            <li class="sub-ul-li">
                <a id="nuevaCuota" class="sub-ul-li-a">
                    <span class="submenuIco"><i class="fas fa-pencil-alt"></i></span>
                    <span class="submenuTit">Modificar cuotas</span>
                </a>
            </li>
        </ul>
    </div>
</li>

```

```

<!--Administracion-->
<li class="opMen">
    <a href="#" id="administrador">
        <span class="icon"><i class="fas fa-cog"></i></span>
        <span class="title">Administrador</span>
        <span class="flecha"><i class="fas fa-angle-down"></i></span>
    </a>
    <div class="sub">
        <ul class="sub-ul">
            <li class="sub-ul-li">
                <a id="AdmUsers" class="sub-ul-li-a">
                    <span class="submenuIco"><i class="fas fa-user-cog"></i></span>
                    <span class="submenuTit">Usuarios</span>
                </a>
            </li>

            <li class="sub-ul-li">
                <a id="AdmSucursales" class="sub-ul-li-a">
                    <span class="submenuIco"><i class="fas fa-building"></i></span>
                    <span class="submenuTit">Sucursales</span>
                </a>
            </li>
        </ul>
    </div>
</li>

```

* 4.3.5.2.5 Fichero "sidebarUser.php"

```
// Estructura del menú principal o sidebar correspondiente al usuario.

<ul>
    <!--Inicio-->
    <li class="opMen">
        <a href="#" id="inicio" class="active">
            <span class="icon"><i class="fas fa-home"></i></span>
            <span class="title">Inicio</span>
        </a>
    </li>

    <!--Poliza-->
    <li class="opMen">
        <a href="#" id="polizas">
            <span class="icon"><i class="fas fa-file-contract"></i></span>
            <span class="title">Polizas</span>
            <span class="flecha"><i class="fas fa-angle-down"></i></span>
        </a>
        <div class="sub">
            <ul class="sub-ul">
                <li class="sub-ul-li">
                    <a id="nuevaPoliza" class="sub-ul-li-a">
                        <span class="submenuIco"><i class="fas fa-plus"></i></span>
                        <span class="submenuTit">Emitir</span>
                    </a>
                </li>
                <li class="sub-ul-li">
                    <a id="verPoliza" class="sub-ul-li-a">
                        <span class="submenuIco"><i class="fas fa-eye"></i></span>
                        <span class="submenuTit">Ver</span>
                    </a>
                </li>
                <li class="sub-ul-li">
                    <a id="renovarPoliza" class="sub-ul-li-a">
                        <span class="submenuIco"><i class="fas fa-sync-alt"></i></span>
                        <span class="submenuTit">Renovaciones</span>
                    </a>
                </li>
            </ul>
        </div>
    </li>

    <!--Endoso-->
    <li class="opMen">
        <a href="#" id="endosos">
            <span class="icon"><i class="fas fa-file-signature"></i></span>
            <span class="title">Endosos</span>
            <span class="flecha"><i class="fas fa-angle-down"></i></span>
        </a>
        <div class="sub">
```

```

<ul class="sub-ul">
    <li class="sub-ul-li">
        <a id="emitirEndoso" class="sub-ul-li-a">
            <span class="submenuIco"><i class="fas fa-plus"></i></span>
            <span class="submenuTit">Emitir</span>
        </a>
    </li>
    <li class="sub-ul-li">
        <a id="verEndoso" class="sub-ul-li-a">
            <span class="submenuIco"><i class="fas fa-eye"></i></span>
            <span class="submenuTit">Ver</span>
        </a>
    </li>
</ul>
</div>
</li>

<!--Siniestro-->
<li class="opMen">
    <a href="#" id="siniestros">
        <span class="icon"><i class="fas fa-car-crash"></i></span>
        <span class="title">Siniestros</span>
        <span class="flecha"><i class="fas fa-angle-down"></i></span>
    </a>
    <div class="sub">
        <ul class="sub-ul">
            <li class="sub-ul-li">
                <a id="nuevoSiniestro" class="sub-ul-li-a">
                    <span class="submenuIco"><i class="fas fa-plus"></i></span>
                    <span class="submenuTit">Emitir</span>
                </a>
            </li>
            <li class="sub-ul-li">
                <a id="adminSiniestro" class="sub-ul-li-a">
                    <span class="submenuIco"><i class="fas fa-pencil-alt"></i></span>
                    <span class="submenuTit">Administrar</span>
                </a>
            </li>
            <li class="sub-ul-li">
                <a id="verSiniestro" class="sub-ul-li-a">
                    <span class="submenuIco"><i class="fas fa-eye"></i></span>
                    <span class="submenuTit">Ver</span>
                </a>
            </li>
        </ul>
    </div>
</li>

<!--Tomador-->
<li class="opMen">
    <a href="#" id="tomadores">
        <span class="icon"><i class="fas fa-user-cog"></i></span>
        <span class="title">Tomadores</span>
    </a>

```

```

        <span class="flecha"><i class="fas fa-angle-down"></i></span>
    </a>
    <div class="sub">
        <ul class="sub-ul">
            <li class="sub-ul-li">
                <a id="nuevoTomador1" class="sub-ul-li-a">
                    <span class="submenuIco"><i class="fas fa-plus"></i></span>
                    <span class="submenuTit">Nuevo</span>
                </a>
            </li>
            <li class="sub-ul-li">
                <a id="adminTomador" class="sub-ul-li-a">
                    <span class="submenuIco"><i class="fas fa-pencil-alt"></i></span>
                    <span class="submenuTit">Administrar</span>
                </a>
            </li>
        </ul>
    </div>
</li>

<!--Pago-->
<li class="opMen">
    <a href="#" id="pagos">
        <span class="icon"><i class="fas fa-money-check-alt"></i></span>
        <span class="title">Pagos</span>
        <span class="flecha"><i class="fas fa-angle-down"></i></span>
    </a>
    <div class="sub">
        <ul class="sub-ul">
            <li class="sub-ul-li">
                <a id="facturarPago" class="sub-ul-li-a">
                    <span class="submenuIco"><i class="fas fa-money-check"></i></span>
                    <span class="submenuTit">Facturar</span>
                </a>
            </li>
            <li class="sub-ul-li">
                <a id="nuevaCuota" class="sub-ul-li-a">
                    <span class="submenuIco"><i class="fas fa-pencil-alt"></i></span>
                    <span class="submenuTit">Modificar cuotas</span>
                </a>
            </li>
        </ul>
    </div>
</li>
</ul>

```

* 4.3.5.2.6 Fichero “topNavbar.php”

```
// Estructura de la vista de la barra superior o top navbar.
<div class="top_navbar">
    <div class="hamburger">
        <div class="one"></div>
        <div class="two"></div>
        <div class="three"></div>
    </div>

    <div class="top_menu">
        
        <ul>
            <li class="configuracionUser">
                <a href="#"></a>
            </li>
        </ul>
    </div>
</div>
```

➔ 4.3.5.3 Carpeta “pagos”

pagos
emisionPago.php
facturarPago.php
listadoRecibosParaPDF.php
mostrarListadoRecibos.php
nuevaCuota.php
reciboPago.php

En esta carpeta se encuentran todos los ficheros relacionados a las vistas y formularios de los pagos y cuotas de polizas del sistema.

* 4.3.5.3.1 Fichero “emisionPago.php”

```
// Estructura de la tabla donde se muestra la lista de polizas o pagos.
<div class="row" id="consultarPolRow">
    <div class="col s12">
        <table id="condicionesCuotasTabla" class="centered striped">
            <thead>
                <tr>
                    <th scope="col">Poliza Nro</th>
                    <th scope="col">Tomador</th>
                    <th scope="col">Vehiculo</th>
                    <th scope="col">Inicio Vig</th>
                    <th scope="col">Fin Vig</th>
                    <th scope="col">Registro</th>
                    <th scope="col"></th>
                </tr>
            </thead>
            <tbody></tbody>
        </table>
    </div>
</div>

// Modal para modificar las condiciones de pago.
<div id="condicionesCuotasModal" class="modal modal-fixed-footer">
    <form id="cuotasForm" class="formulario">
        <div class="modal-content">

            <div id="cuerpoModal">
                // Incluimos el fichero con el recibo de pago.
                <?php include_once 'nuevaCuota.php'; ?>
            </div>

            <ul id="mensajePago" class="mensaje mensajeModal padding"></ul>
        </div>

        <div class="modal-footer">
```

```

        <button type="button" class="btn btn-flat blue modal-
close" id="cancelarCondPago"> <i class="fas fa-times"></i> cerrar </button>
        <button type="submit" class="btn btn-
flat green" id="grabarCondPago"> <i class="fas fa-
save"></i> <span class="nombreBoton"> </span> </button>
    </div>

    </form>
</div>

// Modal con la tabla donde mostramos el listado de cuotas del pago.
<div id="mostrarCuotasModal" class="modal modal-fixed-footer">
    <div class="modal-content">
        <div class="tituloFormulario">Listado de cuotas</div>
        <div class="subtituloFormulario">
            Poliza: <span class="numeroPoliza"></span>
        </div>
        <table class="tablaListadoCuotas centered">
            <thead>
                <tr>
                    <th>Cuota N°</th>
                    <th>Iden Cuota</th>
                    <th>Prima</th>
                    <th>Lapso</th>
                    <th>Factura</th>
                    <th>Vto Pago</th>
                    <th>Estado</th>
                </tr>
            </thead>
            <tbody id="tbodyTabla"></tbody>
        </table>
    </div>
    <div class="modal-footer">
        <button type="button" class="btn btn-flat blue modal-
close" id="cancelarCondPago">
            <i class="fas fa-times"></i> cerrar
        </button>
    </div>
</div>

```

* 4.3.5.3.2 Fichero "facturarPago.php"

```
// Tabla donde se muestra la lista de pagos.  
<div class="row" id="consultarPolRow">  
    <div class="col s12">  
        <table id="verPagoTabla" class="centered striped">  
            <thead>  
                <tr>  
                    <th scope="col">Número</th>  
                    <th scope="col">Poliza Num</th>  
                    <th scope="col">Tomador</th>  
                    <th scope="col">Nº Cuotas</th>  
                    <th scope="col">Estado</th>  
                    <th scope="col">Detalles</th>  
                </tr>  
            </thead>  
            <tbody></tbody>  
        </table>  
    </div>  
</div>  
  
// Modal para ver detalles de las cuotas y para realizar las facturas.  
<div id="detallesCuotasModal" class="modal modal-fixed-footer">  
    <div class="modal-content">  
        <div id="cuerpoModal">  
  
            <div class="mostrarTabla">  
                <div class="tituloFormulario">Consultar Facturas de Pago</div>  
                <div class="textoInformacionCabecera">  
                    Poliza <span id="numPol"></span> - <span id="tomador"></span>  
                    <span class="divPrTo">Prima Total <span id="primaTotal"></span></span>  
                </div>  
            </div>  
  
            // Tabla donde se muestra el listado de cuotas del pago.  
            <table class="tablaListadoCuotas centered" id="tablaListadoCuotas">  
                <thead>  
                    <tr>  
                        <th>Iden cuota</th>  
                        <th>Nº cuota</th>  
                        <th>Prima</th>  
                        <th>Lapso</th>  
                        <th>Factura</th>  
                        <th>Vto Pago</th>  
                        <th>Estado</th>  
                        <th></th>  
                    </tr>  
                </thead>  
                <tbody id="detallesCuotasTbody"></tbody>  
            </table>  
  
</div>
```

```

<div class="mostrarRecibo">
    <div class="tituloFormulario">
        Recibos - Emitir factura #
        <span class="nroCuota">.....</span>
    </div>

    <div class="contRecibo">
        // Incluimos el fichero con el recibo de pago.
        <?php include 'reciboPago.php'?>
    </div>

    <div class="btms-estados">
        <div class="efectuar">
            <a type="button" class="btn-flat efectuarPago" id="efectuarPago">Cobrar</a>
            <span> <i class="fas fa-long-arrow-alt-right"></i> <span class="aCobrar"></span> </span>
        </div>
        <div class="efectuado">
            <i class="fas fa-check-circle icCheckPago"></i><div class="conInfAcc">Pago efectuado <span class="fechaEmisRec"></span> </div>
        </div>
    </div>
</div>
<div class="modal-footer">

    <a class="btn-flat envRecibo"><i class="fas fa-share"></i> enviar</a>
    <a class="btn-flat impRecibo"><i class="fas fa-print"></i> imprimir</a>

    <button type="button" class="btn btn-flat grey" id="actualizarListCuotas"> <i class="fas fa-sync-alt"></i> actualizar </button>
    <button type="button" class="btn btn-flat blue atrasDetallesCuotas" id="atrasDetallesCuotas"> <i class="fas fa-caret-left"></i> volver </button>
    <button type="button" class="btn btn-flat blue modal-close" id="cerrarDetallesCuotas"> <i class="fas fa-times"></i> cerrar </button>
</div>
</div>

// Modal para enviar el recibo por email.
<div class="modalEnviarCorreo modal">
    <form class="formEnviarCorreo formulario" id="formCorreo">

        <div class="modal-content">
            <div class="tituloFormulario">Enviar recibo</div>

```

```
<label for="correo" class="correoLab">Correo electronico</label>
<input type="email" class="correo" name="correo">
<div class="mjeCorr"></div>
</div>

<div class="modal-footer">
    <button type="button" class="btn btn-flat blue modal-
close" id="cerrarModalEnviarRec"> <i class="fas fa-
times"></i> cerrar </button>
    <button type="submit" class="btn btn-
flat green" id="btnEnviarRecibo"> enviar <i class="fas fa-paper-
plane"> </i> </button>
</div>

</form>
</div>

<script type="text/javascript" src="js/pago.js?v=3"></script>
```

* 4.3.5.3.3 Fichero "listadoRecibosParaPDF.php"

```
// Traemos desde la sesión el contenido de todos los recibos de pago.  
<?php  
    if(!isset($_SESSION)) {  
        session_start();  
    }  
    if(isset($_SESSION['infoListadoRecibos'])){  
        $infoRecibos = $_SESSION['infoListadoRecibos'];  
        $srcImg = $infoRecibos['urlImg'].'img/ImgAlt/asegLogo.png';  
    }  
?>  
<body>  
    // Incluimos una hoja de estilos propia para el listado de recibos.  
    <link rel="stylesheet" href="../../css/recibos/listadoRecibosPDF.css">  
  
    <div class="LR2">  
        <div class="IH">  
            <div class="IH-titulo">Recibos</div>  
            <div class="IH-subTitulo">  
                Poliza #<?= $infoRecibos[0]['numPol']?>  
            </div>  
        </div>  
  
        // Por cada ciclo vamos cargando e insertando los datos de un recibo  
        // especifico. Se ejecutaran tantos ciclos como cantidad de cuotas tenga la  
        // poliza. Utilizamos la variable "$i" del for para ir accediendo a los  
        // distintos índices del arreglo de la variable de sesión.  
        <?php for($i=0; $i < $infoRecibos['filas']; $i++): ?>  
        <div class="LR3">  
            <div class="LR4">  
                <div class="recibo">  
  
                    <div class="nombreComp"><img src=<?=$srcImg?> class  
                    ="logoAs" alt="Logo Aseguradora"></div>  
                    <div class="infoComp">  
                        <div> <span class="neg">Cupon oficial Asegurado</span> </div>  
                        <div> <span class="domicEmpr"></span>Av. España 7  
89 Santa Rosa La Pampa</div>  
                        <div> <span class="neg">C.U.I.T:</span> <span class  
                        ="cuitEmpr"><?=$infoRecibos[$i]['cuitEmpr']?>30-70003691-1</div>  
                    </div>  
                    <div class="datosRecibo">  
                        <div> <span class="neg">Poliza/Cuota:</span>  
                            <span class="nroPol"> <?=$infoRecibos[$i]['nroPol']?> </span> / <span class="nroCuota"> <?=$infoRecibos[$i]['nroCuota']?> </span>  
                        </div>  
                        <div> <span class="neg">Tom/Num:</span> <span class  
                        ="aseg"> <?=$infoRecibos[$i]['tom']?> </span> </div>
```

```

        <div> <span class="neg">Domicilio:</span> <span class="domic"> <?= $infoRecibos[$i]['dom'] ?> </span> </div>
        <div>
            <span class="neg">Fecha factuacion:</span> <span class="emision"> <?= $infoRecibos[$i]['fact'] ?> </span>
            <span class="neg">Lapso:</span> <span class="laps"> <?= $infoRecibos[$i]['lapCu'] ?> </span>
        </div>
        <div> <span class="neg">Vigencia de poliza:</span> <span class="vigencia"> <?= $infoRecibos[$i]['vig'] ?> </span> </div>
        <div>
            <div class="lineaVenImp">
                <div class="cont">
                    <span class="neg">Vencimiento del pago:</span> <span class="vtoPago"> <?= $infoRecibos[$i]['vtoPa'] ?> </span>
                    <span class="neg">Importe:</span> <span class="importe"> <?= '$'. $infoRecibos[$i]['priMen'] ?> </span>
                </div>
            </div>
        </div>
        <div class="infoVehic">
            <div> <span class="neg">Vehiculo:</span> <span class="vehiculo"> <?= $infoRecibos[$i]['veh'] ?> </span> </div>
            <div>
                <span class="neg">Año:</span> <span class="anio"> <?= $infoRecibos[$i]['anioVe'] ?> </span>
                <span class="neg">Patente:</span> <span class="patente"> <?= $infoRecibos[$i]['pat'] ?> </span>
                <span class="neg">Motor:</span> <span class="nroMot"> <?= $infoRecibos[$i]['nroMot'] ?> </span>
                <span class="neg">S.Aseg:</span> <span class="sAseg"> <?= '$'. $infoRecibos[$i]['sumAs'] ?> </span>
            </div>
            <div>
                <span class="neg">Cob:</span> <span class="cob"> <?= $infoRecibos[$i]['cob'] ?> </span>
                <span class="neg">Ad:</span> <span class="cobAd"> <?= $infoRecibos[$i]['cobAd'] ?> </span>
            </div>
            <div class="pieRecibo">
                <div class="textoPieRec">El presente plan de pago para la poliza <span class="numPoliza"> <?= $infoRecibos[$i]['numPol'] ?> </span> reemplaza cualquier documento emitido con anterioridad a la fecha <span class="fechaEmision"> <?= $infoRecibos[$i]['fact'] ?> </span>. </div>
            </div>
        </div>
    <?php endfor ?>
</div>
</body>

```

* 4.3.5.3.4 Fichero “mostrarListadoRecibos.php”

```
// Formulario donde se imprime el listado de recibos de una poliza.  
<?php  
    if(!isset($_SESSION)) {  
        session_start();  
    }  
  
    // Obtenemos la informacion almacenada en la variable de sesion.  
    if(isset($_SESSION['infoListadoRecibos'])){  
        $infoRecibos = $_SESSION['infoListadoRecibos'];  
    }  
?>  
<html>  
    <head>  
        <link rel="stylesheet" href="css/estilos.css">  
    </head>  
    <body>  
        // Listado de recibos de una poliza.  
        <div class="LR1">  
            <div class="LR2">  
  
                // Por cada ciclo listamos y generamos un recibo.  
                <?php for($i=0; $i < $infoRecibos['filas']; $i++): ?>  
                    <div class="LR3">  
                        <div class="LR4">  
                            <div class="recibo">  
                                <div class="nombreComp"></div>  
                                <div class="infoComp">  
                                    <div> <span class="neg">Cupon oficial Ase  
gurado</span> </div>  
                                    <div> <span class="domicEmpr"></span>Av.  
España 789 Santa Rosa La Pampa</div>  
                                    <div> <span class="neg">C.U.I.T:</span> <  
span class="cuitEmpr"></span>30-70003691-1</div>  
                                </div>  
                                <div class="datosRecibo">  
                                    <div> <span class="neg">Poliza/Cuota:</sp  
an>  
                                        <span class="nroPol"> <?=$infoRecibos  
[$i]['numPol']?> </span> / <span class="nroCuota"> <?=$infoRecibos[$i]['nroCu  
'?> </span>  
                                            </div>  
                                            <div> <span class="neg">Tom/Num:</span> <  
span class="aseg"> <?= $infoRecibos[$i]['tom'] ?> </span> </div>  
                                            <div> <span class="neg">Domicilio:</span>  
                                            <span class="domic"> <?= $infoRecibos[$i]['dom'] ?> </span> </div>  
                                            <div>  
                                                <span class="neg">Fecha facturacion:</  
span> <span class="emision"> <?= $infoRecibos[$i]['fact'] ?> </span>
```

```

        <span class="neg">Lapso:</span> <span
class="laps"> <?= $infoRecibos[$i]['lapCu'] ?> </span>
                </div>
                <div> <span class="neg">Vigencia de poliz
a:</span> <span class="vigencia"> <?= $infoRecibos[$i]['vig'] ?> </span> </di
v>
                <div>
                    <div class="lineaVenImp">
                        <div class="cont">
                            <span class="neg">Vencimiento
del pago:</span> <span class="vtoPago"> <?= $infoRecibos[$i]['vtoPa'] ?> </s
pan>
                                <span class="neg">Importe:</s
pan> <span class="importe"> <?='$'.\$infoRecibos[$i]['priMen'] ?> </span>
                                </div>
                            </div>
                        </div>
                    <div class="infoVehic">
                        <div> <span class="neg">Vehiculo:</span>
<span class="vehiculo"> <?= $infoRecibos[$i]['veh'] ?> </span> </div>
                        <div>
                            <span class="neg">Año:</span> <span c
lass="anio"> <?= $infoRecibos[$i]['anioVe'] ?> </span>
                            <span class="neg">Patente:</span> <sp
an class="patente"> <?= $infoRecibos[$i]['pat'] ?> </span>
                            <span class="neg">Motor:</span> <span
class="nroMot"> <?= $infoRecibos[$i]['nroMot'] ?> </span>
                            <span class="neg">S.Aseg:</span> <spa
n class="sAseg"> <?='$'.\$infoRecibos[$i]['sumAs'] ?> </span>
                            </div>
                        <div>
                            <span class="neg">Cob:</span> <span c
lass="cob"> <?= $infoRecibos[$i]['cob'] ?> </span>
                            <span class="neg">Ad:</span> <span cl
ass="cobAd"> <?= $infoRecibos[$i]['cobAd'] ?> </span>
                            </div>
                        </div>
                        <div class="pieRecibo">
                            <div class="textoPieRec">El presente plan de
 pago para la poliza <span class="numPoliza"> <?= $infoRecibos[$i]['numPol'] ?>
 </span> reemplaza cualquier documento emitido con anterioridad a la fecha <
span class="fechaEmision"> <?= $infoRecibos[$i]['fact'] ?> </span>. </div>
                            </div>
                        </div>
                    </div>
                <?php endfor ?>
            </div>
        </div>
        <div class="btnsVerListCu">
            <button type="button" class="btn btn-
flat blue btnAtrasListCu"> <i class="fas fa-caret-left"></i> Volver</button>

```

```
        <a type="button" class="btn btn-
flat btnDescargarListCu"> <i class="fas fa-file-download"></i> PDF</a>
        <form id="formCorreo" class="formEnvCorrListadoCuo">
            <button type="submit" class="btn btn-
flat grey btnEnviarListCu"> Enviar <i class="fas fa-share"></i></button>
            <input type="email" class="envEmail browser-
default" id="email" placeholder="E-
mail" name="correo" value=" <?= $infoRecibos[0]['email'] ?> ">
        </form>
    </div>
    <script type="text/javascript" src="js/pago.js?v=3"></script>
</body>
</html>
```

* 4.3.5.3.5 Fichero "nuevaCuota.php"

```
// Formulario con los campos para ingresar los parámetros de pago.
<div class="tituloFormulario" id="tituloModalPago"></div>

<!--<fieldset class="fieldsetCuotas">-->

<div class="datosPol">
    <div class="infoPag-1">
        <div class="infoPag-2">
            <div class="infoPag-3">
                <div class="infoPag-4-titulo">Poliza</div>
                <div class="infoPag-4-contenido">
                    <div> Numero <span id="numPoliza"></span> </div>
                    <div> Emitida el <span id="emision"></span> a las <span id="horaEmision"></span> </div>
                </div>
            <div class="infoPag-3">
                <div class="infoPag-4-titulo">Vigencia</div>
                <div class="infoPag-4-contenido">
                    <div> Desde <span id="inicioVig"></span> <span id="inicioHora"></span> hasta <span id="finVig"></span> <span id="finHora"></span> (<span id="mesesVig"></span> mes/es) </div>
                    <input type="text" class="mesesVigMuestra" id="tipVig">
                </div>
            </div>
        <div class="infoPag-3">
            <div class="infoPag-4-titulo">Vehiculo</div>
            <div class="infoPag-4-contenido">
                <div> <span id="marcaP"></span> <span id="modeloP"></span> <span id="motorP"></span> - <span id="localidadP"></span> <span id="provinciaP"></span> - <span id="cpP"></span> </div>
                <div> Año: <span id="anio"></span> </div>
                <div> Es cero km: <span id="ceroKm"></span> </div>
                <div> Kilometros: <span id="kms"></span> </div>
                <div> Equipo de gas <span id="eGas"></span> </div>
            </div>
        </div>
    <div class="infoPag-3">
        <div class="infoPag-4-titulo">Coberturas</div>
        <div class="infoPag-4-contenido">
            <div> Principal <span id="coberturaDesc"></span> <span id="coberturaNombre"></span> </div> Principal
            <div> Adicional <span id="coberturaAd"></span> </div>
            <div> Suma Asegurada $<span id="sumaAs"></span> </div>
        </div>
    </div>
</div>
```

```

        </div>
    </div>

<div class="row">

    <div class="col s2">
        <label for="primaTotal">Prima total $</label>
        <input type="number" id="primaTotal" name="primaTotal">
    </div>

    <div class="col s2">
        <label for="cantCuotas">Cant cuotas</label>
        <select id="cantCuotas" name="cantCuotas"></select>
    </div>

    <div class="col s2">
        <label for="primaMensual">Prima mensual $</label>
        <input type="number" id="primaMensual" name="primaMensual">
    </div>

    <div class="col s2" id="diaCob">
        <label for="diaCobrar">Dia a facturar</label>
        <select id="diaCobrar" name="diaCobrar"></select>
    </div>

    <div class="col s3 tipoPag">
        <label for="tipoPag">Metodo de pago</label>
        <select id="tipoPag" name="tipoPag">
            <option value="0" selected disabled>Seleccione</option>
            <option value="1">Efectivo</option>
            <option value="2" disabled>Asociar Tarjeta de Credito/Debito</option>
        </select>
    </div>

    <div class="col s2 divVerCuotas">
        <label for="verCuotas">Ver cuotas</label>
        <a type="button" id="verCuotas" class="btn btn-flat"> <i class="fas fa-eye"></i> </a>
    </div>

    <input type="text" id="idPago">
</div>

<script type="text/javascript" src="js/pago.js?v=3"></script>

// Comentamos la sección para ingresar los datos de una tarjeta de crédito.

<!--
<div class="row datosTarjeta">
    <hr class="hrPago">
-->
```

```
<div class="col s4">
    <label for="tarjetaNum">Número de tarjeta</label>
    <input type="text" class="browser-
default" placeholder="0000 0000 0000 0000" id="tarjetaNum" name="tarjetaNum">
</div>

<div class="col s4">
    <label for="tarjetaNombre">Nombre</label>
    <input type="text" class="browser-
default" placeholder="Tal cual figura en la tarjeta" id="tarjetaNombre" name=
"tarjetaNombre">
</div>

<div class="col s1">
    <label for="tarjetaMesCad">Mes caduc</label>
    <input type="text" class="browser-
default" placeholder="MM" id="tarjetaMesCad" name="tarjetaMesCad">
</div>

<div class="col s1">
    <label for="tarjetaAnCad">Año caduc</label>
    <input type="text" class="browser-
default" placeholder="AAAA" id="tarjetaAnCad" name="tarjetaAnCad">
</div>

<button type="button" id="comprobarTarjeta" class="waves-light btn-
small blue">Comprobar</button>-->

<!--</fieldset>-->
```

* 4.3.5.3.6 Fichero "reciboPago.php"

```
// Formulario donde mostramos los datos de un recibo de pago específico.  
<?php  
  
    $srcImg = 'img/ImgAlt/asegLogo.png'; //Definimos la url de la imagen.  
  
    if(!isset($_SESSION))  
    {  
        session_start();  
    }  
  
    // Obtenemos la información del recibo desde la variable de sesión.  
    if(isset($_SESSION['recibo']))  
    {  
        $info = $_SESSION['recibo'];  
  
        if(isset($_SESSION['urlImg'])){  
            //Como llamamos al fichero desde diferentes lugares, la url de la  
            //imagen/loho debe de modificarse para asesorar que se muestre correctamente.  
            $srcImg = $_SESSION['urlImg'].$srcImg;  
        }  
  
        // Si no existe ningún recibo en la sesión, se muestran * en los campos.  
    }else{  
        $info['numPol'] = '*****';  
        $info['nroCu'] = '*****';  
        $info['tom'] = '***** / *****';  
        $info['dom'] = '***** ***** *****';  
        $info['fact'] = '*****';  
        $info['lapCu'] = '***** *****';  
        $info['vig'] = '***** *****';  
        $info['vtoPa'] = '*****';  
        $info['priMen'] = '*****';  
        $info['veh'] = '***** ***** ***** ***';  
        $info['anioVe'] = '****';  
        $info['pat'] = '*****';  
        $info['nroMot'] = '*****';  
        $info['sumAs'] = '*****';  
        $info['cob'] = '**** ** *****';  
        $info['cobAd'] = '***** ***** *****';  
        $info['numPol'] = '*****';  
        $info['fact'] = '*****';  
    }  
  
?>  
  
// Asignamos las variables con los datos en los campos.  
<html class="htmlRecibo">  
  
    <link rel="stylesheet" href="css/recibos/recibosPDF.css">  
  
    <div class="recibo">
```

```

        <div class="nombreComp"><img src=<?= $srcImg ?>" class="logoAs" alt="Logo Aseguradora"></div>
        <div class="infoComp">
            <div> <span class="neg">Cupon oficial Asegurado</span> </div>
            <div> <span class="domicEmpr"></span>Av. España 789 Santa Rosa La Pampa</div>
            <div> <span class="neg">C.U.I.T:</span> <span class="cuitEmpr"></span>30-70003691-1</div>
        </div>

        <div class="datosRecibo">
            <div> <span class="neg">Poliza/Cuota:</span>
                <span class="nroPol"> <?= $info['numPol'] ?> </span> / <span class="nroCuota"> <?= $info['nroCu'] ?> </span>
            </div>
            <div> <span class="neg">Tom/Num:</span> <span class="aseg"> <?= $info['tom'] ?> </span> </div>
            <div> <span class="neg">Domicilio:</span> <span class="domic"> <?= $info['dom'] ?> </span> </div>
            <div>
                <span class="neg">Fecha factuacion:</span> <span class="emision"> <?= $info['fact'] ?> </span>
                <span class="neg">Lapso:</span> <span class="laps"> <?= $info['lapCu'] ?> </span>
            </div>
            <div> <span class="neg">Vigencia de poliza:</span> <span class="vigenzia"> <?= $info['vig'] ?> </span> </div>
            <div>
                <div class="lineaVenImp">
                    <div class="cont">
                        <span class="neg">Vencimiento del pago:</span> <span class="vtoPago"> <?= $info['vtoPa'] ?> </span>
                        <span class="neg">Importe:</span> <span class="importe"> <?=$info['priMen'] ?> </span>
                    </div>
                    </div>
                </div>
            </div>
        </div>

        <div class="infoVehic">
            <div> <span class="neg">Vehiculo:</span> <span class="vehiculo"> <?= $info['veh'] ?> </span> </div>
            <div>
                <span class="neg">Año:</span> <span class="anio"> <?= $info['anioVe'] ?> </span>
                <span class="neg">Patente:</span> <span class="patente"> <?= $info['pat'] ?> </span>
                <span class="neg">Motor:</span> <span class="nroMot"> <?= $info['nroMot'] ?> </span>
                <span class="neg">S.Aseg:</span> <span class="sAseg"> <?=$info['sumAs'] ?> </span>
            </div>
        </div>
    
```

```
        <span class="neg">Cob:</span> <span class="cob"> <?= $info['cob'] ?> </span>
        <span class="neg">Ad:</span> <span class="cobAd"> <?= $info['cobA
d'] ?> </span>
</div>
</div>

<div class="pieRecibo">
<div class="textoPieRec">El presente plan de pago para la poliza <spa
n class="numPoliza"> <?= $info['numPol'] ?> </span> reemplaza cualquier docum
ento emitido con anterioridad a la fecha <span class="fechaEmision"> <?= $inf
o['fact'] ?> </span>. </div>
</div>
</div>
</html>
```

⇒ 4.3.5.4 Carpeta “poliza”

poliza
elementosPoliza
formularios
div_cobertura.php
div_pago.php
div_poliza.php
div_tomador.php
div_vehiculo.php

En esta carpeta se almacenan todos los ficheros que contienen las vistas y formularios relacionados a las polizas del sistema.

* 4.3.5.4.1 Carpeta “elementosPoliza”

elementosPoliza
consultar_polizas.php
formulario_poliza.php
informacionPoliza.php
renovar_poliza.php

> 4.3.5.4.1.1 Fichero “consultar_polizas.php”

```
// Tabla donde se muestran todas las polizas registradas y su informacion.
<div class="row" id="consultarPolRow">
    <div class="col s12">
        <table id="consultarPolizasTabla" class="centered">
            <thead>
                <tr>
                    <th scope="col">Nro</th>
                    <th scope="col">Tomador</th>
                    <th scope="col">Vehiculo</th>
                    <th scope="col">Emision</th>
                    <th scope="col">Vigencia</th>
                    <th scope="col">Anulacion</th>
                    <th scope="col">Estado</th>
                    <th scope="col">Info</th>
                    <th scope="col">Ver</th>
                </tr>
            </thead>
            <tbody id="info"></tbody>
        </table>
        <div class="btnActPolPadre">
            <button type="button" class="btn btn-flat blue btnActualizarListPolizas" id="actListPol"> <i class="fas fa-sync-alt"></i> actualizar </button>
        </div>
    </div>
```

```
    </div>
</div>

<script type="text/javascript" src="js/poliza.js?v=3"></script>
```

› 4.3.5.4.1.2 Fichero “formulario_poliza.php”

```
// Estructura del formulario general para la emisión de una nueva poliza.
<ul id="tabFormularioPoliza" class="tabs tabs-fixed-width tab-demo z-depth-1">
    <li class="tab"><a id="tab1" href="#test1" class="active">Poliza <i class="fas fa-chevron-circle-right"></i> </a> </li>
    <li class="tab"><a id="tab2" href="#test2">Tomador <i class="fas fa-chevron-circle-right"></i> </a></li>
    <li class="tab tabVehiculo"><a id="tab3" href="#test3">Vehiculo <i class="fas fa-chevron-circle-right"></i> </a></li>
    <li class="tab"><a id="tab4" href="#test4">Cobertura <i class="fas fa-chevron-circle-right"></i> </a></li>
    <li class="tab"><a id="tab5" href="#test5">Pago</a></li>
</ul>
// Incluimos en cada tab los diferentes formularios de las secciones para el
registro.
<div id="test1" class="col s12"> <?php include_once "../div_poliza.php"?>
</div>
<div id="test2" class="col s12"> <?php include_once "../div_tomador.php"?>
</div>
<div id="test3" class="col s12"> <?php include_once "../div_vehiculo.php"?>
</div>
<div id="test4" class="col s12"> <?php include_once "../div_cobertura.php"?>
</div>
<div id="test5" class="col s12"> <?php include_once "../div_pago.php"?> </div>
>

<script type="text/javascript" src="js/poliza.js?v=3"></script>
```

> 4.3.5.4.1.3 Fichero “informacionPoliza.php”

```
// Vista de la sección para mostrar los datos de una poliza.
<div class="row">
    // Sección donde mostramos los datos generales de la poliza en si.
    <div class="infoGeneral">
        <div class="datosPoliza">
            <span class="titulo">Poliza</span>
            <div class="infPol"> </div>

            <div class="infAdPol">
                <span class="subtitulo">Informacion</span>
                <div class="infEndSin"></div>

                <span class="subtitulo">Cobertura</span>
                <div class="infCo"> </div>

                <span class="subtitulo">Adicional</span>
                <div class="infCoA"> </div>
            </div>
        </div>
        // Sección donde mostramos la información del tomador de la poliza.
        <div class="datosTomador">
            <span class="titulo">Tomador</span>
            <div class="infTom"></div>
        </div>

        <div class="datosPago">
            <span class="titulo">Pagos</span>
            <div class="infPag"></div>
            <div class="infCuo">
                <span class="subtitulo">Cuotas</span>
                // Tabla donde mostramos las cuotas de la poliza.
                <table>
                    <thead>
                        <tr>
                            <th>Numero</th>
                            <th>Identificador</th>
                            <th>Prima</th>
                            <th>Estado</th>
                        </tr>
                    </thead>
                    <tbody id="tbodyCuo"></tbody>
                </table>
            </div>
        </div>
    </div>

    // Sección donde mostramos la línea del tiempo de estados de la poliza.
    <div class="datosEstados">
        <span class="titulo">Estados</span>
        <p><span class="subtitulo">Actual</span></p>
        <p class="estAct"></p>
    </div>
</div>
```

```
<p><span class="subtitulo">Anteriores</span></p>
<div class="estAnt"></div>
</div>
// Seccion donde mostramos la informacion del vehiculo de esa poliza.
<div class="datosVehiculo">
    <span class="titulo">Vehiculo / Riesgo</span>
    <div class="infVe">
        <div class="Iz"></div>
        <div class="De"></div>
    </div>
</div>
</div>

</div>

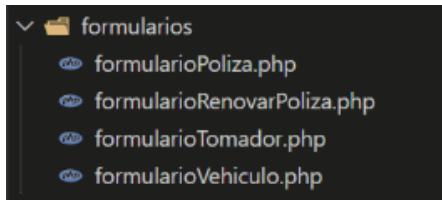
<div class="row botonesVerPol">
    <button type="button" class="btn btn-
flat blue btnVolverConPol"> <i class="fas fa-caret-
left"></i> volver </button>
    <button type="button" class="btn btn-
flat blue btnActConPol"> <i class="fas fa-sync-alt"></i> actualizar </button>
</div>
```

> 4.3.5.4.1.4 Fichero “renovar_poliza.php”

```
// Tabla donde se muestra el listado de polizas vencidas para renovar.
<div class="row" id="consultarPolRow">
    <div class="col s12">
        <table id="renovarPolizaTabla" class="centered renovarPolizaTabla">
            <thead>
                <tr>
                    <th scope="col">Nro</th>
                    <th scope="col">Tomador</th>
                    <th scope="col">Vigencia</th>
                    <th scope="col">Emision</th>
                    <th scope="col">Anulacion</th>
                    <th scope="col">Renovar</th>
                </tr>
            </thead>
            <tbody id="tbodyRenovarPoliza"></tbody>
        </table>
    </div>
</div>

<script type="text/javascript" src="js/poliza.js?v=3"></script>
```

* 4.3.5.4.2 Carpeta "formularios"



> 4.3.5.4.2.1 Fichero "formularioPoliza.php"

```
// Formulario con los campos para ingresar los datos de la poliza.
<h2 class="tituloFormulario titForPol">datos de poliza</h2>

<div class="infoCabeceraEmisPoliza">
    <span class="textoCabecer">Se asignara el numero <span id="nroPoliza"></span> a la poliza a registrar</span>
</div>

<div class="row">

    <div class="col s5">
        <label for="vig_ini" class="vig_ini">Inicio de vigencia</label>
        <input type="text" id="vig_ini" placeholder="Seleccione" name="vig_ini">
    </div>

    <div class="col s2">
        <label for="tipVig" class="tipVig">Fin de vigencia</label>
        <select name="tipVig" id="tipVig" class="tipVig">
            <option value="0" selected disabled>Seleccione vigencia</option>
            <option value="4">4 Meses</option>
            <option value="5">5 Meses</option>
            <option value="6">6 Meses</option>
            <option value="7">7 Meses</option>
            <option value="8">8 Meses</option>
            <option value="9">9 Meses</option>
            <option value="10">10 Meses</option>
            <option value="11">11 Meses</option>
            <option value="12">12 Meses</option>
        </select>
    </div>

    <div class="col s5">
        <label for="vig_fin" class="vig_fin">Fecha de finalizacion</label>
        <input type="text" id="vig_fin" placeholder="Seleccione" name="vig_fin">
    </div>
</div>

// Recuadro donde mostramos la informacion de vigencia.
```

```
<div class="divResumenVigPoliza">
    <!-- <div class="resumen1">Complete la informacion</div> -->
    <div class="resumen">
        <input type="text" id="vigenciaInicioInput">
        <input type="text" id="vigenciaFinInput">
        La vigencia establecida sera desde el <span id="vigenciaInicioSpan"><
        /span> hasta el <span id="vigenciaFinSpan"></span> a las 11:59 AM
    </div>
</div>
```

> 4.3.5.4.2.2 Fichero “formularioRenovarPoliza.php”

```
// Vista para renovar una poliza.
<div id="formRenPol">
    // Formulario donde se muestra datos de la poliza.
    <div>
        <form class="formulario formRenovacionPol" id="divPoliza">
            <fieldset>

                <div class="row" id="formPolRen">
                    <?php require_once '../formularios/formularioPoliza.php' ?>
                </div>

                <div class="botRenPol">
                    <button type="button" class="btn btn-
flat blue atrRenPol"> <i class="fas fa-caret-left"></i> atras </button>
                    <button type="submit" class="btn btn-
flat green sigRenPol"> siguiente <i class="fas fa-caret-right"></i> </button>
                </div>

            </fieldset>
        </form>

        <ul id="mensajePoliza" class="mensaje"></ul>
    </div>

    // Formulario donde se muestra datos del pago.
    <div>
        <form class="formulario cuotasForm" id="cuotasForm">
            <fieldset>
                <div class="tituloFormulario titRenPago">Datos del pago</div>

                <div>
                    // Incluimos el formulario de condiciones de pago.
                    <?php require_once '../pagos/nuevaCuota.php' ?>
                </div>

                <div class="botRenPa">
                    <button type="submit" class="btn btn-
flat blue atrRenPa"> <i class="fas fa-caret-left"></i> atras </button>
                    <button type="submit" class="btn btn-
flat green sigRenPa"> <i class="fas fa-save"></i> renovar </button>
                </div>
            </fieldset>
        </form>
        <ul id="mensajePago" class="mensaje"></ul>
    </div>
</div>
```

> 4.3.5.4.2.3 Fichero “formularioTomador.php”

```
// Formulario con los campos para ingresar los datos del tomador.  
<h2 class="tituloFormulario">datos del tomador</h2>  
  
<div class="elementosSupNuevoTom">  
    <div class="elSupSelectTom">  
        // Incluimos el listado de todos los clientes registrados.  
        <select name="select_cliente" id="select_cliente"></select>  
    </div>  
    <div class="elSupBtnNuevoTom">  
        <button href="#modalcliente" type="button" class="btn btn-flat green modal-trigger" id="nuevoTomador"> <i class="fas fa-plus"></i> nuevo </button>  
    </div>  
</div>  
  
// Vista de los datos de tomador.  
  
<div id="preInfo">  
    <div id="info" class="row">  
        <div class="col s3">  
            <label for="nombreDatos">Nombre</label>  
            <input type="text" id="nombreDatos" class="transparente">  
            <label for="docDatos">Documento</label>  
            <input type="text" id="docDatos" class="transparente">  
            <label for="tipDatos">Tipo de Persona</label>  
            <input type="text" id="tipDatos" class="transparente">  
        </div>  
  
        <div class="col s3">  
            <label for="nacDatos">Nacionalidad</label>  
            <input type="text" id="nacDatos" class="transparente">  
            <label for="proDatos">Provincia</label>  
            <input type="text" id="proDatos" class="transparente">  
            <label for="locDatos">Localidad</label>  
            <input type="text" id="locDatos" class="transparente">  
        </div>  
  
        <div class="col s3">  
            <label for="cpDatos">Codigo Postal</label>  
            <input type="text" id="cpDatos" class="transparente">  
            <label for="calleDatos">Calle</label>  
            <input type="text" id="calleDatos" class="transparente">  
            <label for="genDatos">Genero</label>  
            <input type="text" id="genDatos" class="transparente">  
        </div>  
  
        <div class="col s3">  
            <label for="telDatos">Telefono</label>  
            <input type="text" id="telDatos" class="transparente">  
            <label for="corrDatos">Correo</label>  
        </div>  
    </div>
```

```
<input type="text" id="corrDatos" class="transparente">
<label for="feNacDatos">Nacimiento</label>
<input type="text" id="feNacDatos" class="transparente">
</div>
</div>
</div>
```

> 4.3.5.4.2.4 Fichero “formularioVehiculo.php”

```
// Formulario con los campos para ingresar los datos del vehiculo.  
// Instanciamos las clases para mostrar los datos de sus métodos.  
<?php  
    include_once('../.../controlador/polizasController.php');  
    include_once('../.../controlador/funcionesGlobalesController.php');  
    include_once('../.../config/db.php');  
    $objSelect = new PolizasController();  
    $objFg = new FuncionesGlobalesController();  
?>  
  
<h2 class="tituloFormulario">datos del vehiculo</h2>  
  
<div class="row">  
    <div class="col s2">  
        <label for="suma_asegurada">Suma Asegurada</label>  
        <input type="text" id="suma_asegurada" class="campo" placeholder="560.000"  
        " name="suma_asegurada">  
    </div>  
  
    <div class="col s3">  
        <label for="select_marca">Marca</label>  
        <select class="campo select2" id="select_marca" name="selec_marca">  
            // Listamos todas las marcas desde el objeto.  
            <?= $objSelect->listadoMarcas() ?>  
        </select>  
    </div>  
  
    <div class="col s3">  
        <label for="select_modelo1">Modelo</label>  
        <select class="campo select2" id="select_modelo1" name="select_modelo1">  
            <option value="0" disabled selected>Seleccionar Marca</option>  
        </select>  
    </div>  
  
    <div class="col s2">  
        <label for="patente">Patente</label>  
        <input type="text" id="patente" class="campo" placeholder="ABC123 (Sin espacios)"  
        maxlength="7" name="patente">  
    </div>  
  
    <div class="col s2">  
        <label for="motor">Motor</label>  
        <input type="text" id="motor" class="campo" placeholder="2.0 TSI 20V 230C  
        V" name="motor">  
    </div>  
</div>  
  
<div class="row">  
    <div class="col s3">  
        <label for="nroChasis">Nro Chasis</label>
```

```

<input type="text" id="nroChasis" class="campo" placeholder="B2N5R6S9E9R6
VJ5T8" name="nroChasis">
</div>

<div class="col s2">
    <label for="nroMotor">Nro Motor</label>
    <input type="text" id="nroMotor" class="campo" placeholder="GH323JK" na
me="nroMotor">
</div>

<div class="col s3">
    <label for="select_clase">Clase</label>
    <select class="campo" id="select_clase" name="select_clase">
        // Listamos las clases de vehiculo desde el objeto.
        <?= $objSelect->listadoClases() ?>
    </select>
</div>

<div class="col s4">
    <label for="select_uso">Uso</label>
    <select class="campo" id="selectTipoUso" name="selectTipoUso">
        // Listamos los tipos de usos de vehiculo desde el objeto.
        <?= $objSelect->listadoUso() ?>
    </select>
</div>
</div>

<div class="row">

    <div class="col s1">
        <label for="GPS">GPS</label>
        <div class="switch">
            <label> <input type="checkbox" id="GPS" class="filled-
in" name="GPS"> <span class="filled-in"></span> </label>
        </div>
    </div>

    <div class="col s1">
        <label for="cerokm">0 Km</label>
        <div class="switch">
            <label> <input type="checkbox" id="cerokm" class="filled-
in" name="cerokm"> <span class="filled-in"></span> </label>
        </div>
    </div>

    <div class="col s2">
        <label for="cant_kilometro">Kilometros</label>
        <input type="text" id="cant_kilometro" class="campo" placeholder="124.000
" name="cant_kilometro">
    </div>

    <div class="col s2">
        <label for="codigoPostal">Codigo Postal</label>
        <select class="campo select2" id="codigoPostal" name="codigoPostal">

```

```

        // Listamos todos los códigos postales desde el objeto.
        <?= $objFg->listadoCp() ?>
    </select>
</div>

<div class="col s4">
    <label for="vehiculoLocalidad">Localidad</label>
    <select class="campo select2" id="vehiculoLocalidad" name="vehiculoLocalidad">
        <option value="0" selected disabled>Seleccionar Codigo Postal</option>
    </select>
</div>
</div>

<div class="row">
<div class="col s3">
    <label for="coberturaAd">Coberturas Adicionales</label>
    <select class="campo" id="coberturaAd" name="coberturaAd">
        // Listamos las coberturas adicionales desde el objeto.
        <?= $objSelect->listadoCoberturasAd() ?>
    </select>
</div>

<div class="col s2">
    <label for="combustible">Combustible</label>
    <select class="campo" id="combustible" name="combustible">
        <option value="0" selected disabled>Seleccione</option>
        <option value="Nafta">Nafta</option>
        <option value="GasOil">Gas Oil</option>
        <option value="Energia Electrica">Energia Electrica</option>
    </select>
</div>

<div class="col s1">
    <label for="eGas">Gas</label>
    <div class="switch">
        <label> <input type="checkbox" id="eGas" class="filled-in" name="eGas"> <span class="filled-in"></span> </label>
    </div>
</div>

<div class="col s1">
    <label for="anio">Año</label>
    <input type="text" id="anio" class="campo" placeholder="2017" name="anio">
</div>

<div class="col s1">
    <label for="pasajeros">Pasajeros</label>
    <input type="number" id="pasajeros" class="campo" placeholder="5" name="pasajeros">
</div>

<div class="col s1">

```

```
<label for="asientos">Asientos</label>
<input type="number" id="asientos" class="campo" placeholder="5" name="as
ientos">
</div>

<div class="col s2">
    <label for="color">Color</label>
    <input type="text" id="color" class="campo" placeholder="Blanco" name="co
lor">
</div>
</div>
```

* 4.3.5.4.3 Fichero "div_cobertura.php"

```
// Formulario que representa la sección de cobertura al emitir una poliza.
<form id="divCobertura" class="formulario">
    <fieldset id="fieldsetCobertura">
        <h2 class="tituloFormulario">tipo de cobertura</h2>
        <div class="col s12">
            <select class="campo select_cobertura" id="select_cobertura" name="select_cobertura"></select>
            </div>

            // Sección donde mostramos la información de la cobertura.
            <div id="infoCobertura" class="col s12"></div>

            <div id="botonesCobertura">
                <button type="button" class="btn btn-flat blue" id="atrasCobertura"><i class="fas fa-caret-left"></i> atras </button>
                <span class="btnDivCobertura"></span>
            </div>

        </fieldset>

        <ul id="mensajeCobertura" class="mensaje"></ul>
    </form>
```

* 4.3.5.4.4 Fichero "div_pago.php"

```
// Formulario que representa la sección de pago al emitir una poliza.
<form id="cuotasForm" class="formulario">
  <fieldset id="fieldsetPago">

    <h2 class="tituloFormulario">Datos de pago</h2>
    // Incluimos el formulario con los campos para ingresar los parámetros de
    pago.
    <?php require_once '../pagos/nuevaCuota.php' ?>

    <div id="botonesPago">
      <button type="button" class="btn btn-
flat blue" id="atrasPago"> <i class="fas fa-caret-left"></i> atras </button>
      <span class="btnDivPago"></span>
      <button type="button" class="btn btn-
flat red" id="cancelar_datos" value="Enviar"> cancelar <i class="fas fa-
times"></i> </button>

    </div>

  </fieldset>
</form>

<ul id="mensajePago" class="mensaje"></ul>
```

* 4.3.5.4.5 Fichero "div_poliza.php"

```
// Formulario que representa la sección de poliza al emitir una poliza.
<form id="divPoliza" class="formulario">
  <fieldset id="fieldsetPoliza">
    // Incluimos el formulario de poliza.
    <?php require_once 'formularios/formularioPoliza.php' ?>
    <button type="submit" class="btn btn-
flat green" id="siguientePoliza" value="Enviar"> siguiente <i class="fas fa-
caret-right"></i> </button>
  </fieldset>
</form>

<ul id="mensajePoliza" class="mensaje"></ul>
```

* 4.3.5.4.6 Fichero "div_tomador.php"

```
// Formulario que representa la sección de tomador al emitir una poliza.
<form id="divTomador" class="formulario">
    <fieldset id="fieldsetTomador">
        // Incluimos el formulario del tomador.
        <?php require_once 'formularios/formularioTomador.php' ?>
        <div id="cliR"></div>
        <div id="botonesTomador">
            <button type="button" class="btn btn-
flat blue" id="atrasTomador"> <i class="fas fa-caret-
left"></i> atras </button>
            <span class="btnDivTomador"></span>
        </div>
    </fieldset>
    <ul id="mensajeTomador" class="mensaje ul2"></ul>
</form>

// Ventana modal para registrar un nuevo cliente.
<div id="modalcliente" class="modal modal-fixed-footer" >
    <form id="nuevoCliente" class="formulario">
        <div class="modal-content">
            <div id="cuerpoModal">
                // Incluimos el formulario de nuevo tomador.
                <?php include_once '../tomador/nuevoTomador.php'; ?>
            </div>
            <ul id="mensajeTomador" class="mensaje mensajeModal"></ul>
        </div>
        <div class="modal-footer">
            <button type="button" class="btn btn-flat blue modal-
close" id="cancelarCliente"> <i class="fas fa-times"></i> cerrar </button>
            <button type="submit" class="btn btn-
flat green" id="grabarCliente"> <i class="fas fa-save"></i> grabar </button>
        </div>
    </form>
</div>

<script type="text/javascript" src="js/cliente.js?v=3"></script>
```

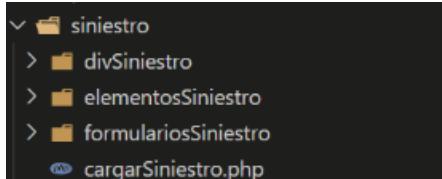
* 4.3.5.4.7 Fichero “div_vehiculo.php”

```
// Formulario que representa la sección de vehiculo al emitir una poliza.
<form id="divVehiculo" class="formulario">
    <fieldset id="fieldsetVehiculo">
        // Incluimos el formulario con los campos para ingresar los datos del
        // vehiculo.
        <?php require_once 'formularios/formularioVehiculo.php' ?>

        <div id="botonesVehiculo">
            <button type="button" class="btn btn-
flat blue" id="atrasVehiculo"> <i class="fas fa-caret-
left"></i> atras </button>
            <span class="btnDivVehiculo"></span>
        </div>

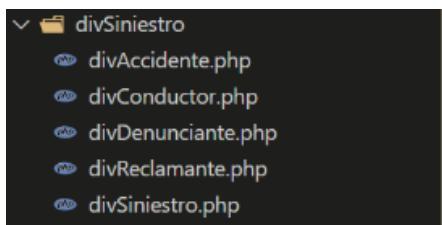
    </fieldset>
</form>
<ul id="mensajeVehiculo" class="mensaje"></ul>
```

➔ 4.3.5.5 Carpeta “siniestro”



En esta carpeta se almacenan todos los ficheros cuyo contenido representa a las vistas y formularios de los siniestros.

* 4.3.5.5.1 Carpeta “divSiniestro”



› [4.3.5.5.1.1 Fichero “divAccidente.php”](#)

```
// Formulario que representa la sección de accidente al registrar un nuevo  
siniestro.  
<form id="divAccidente" class="formulario divAccidente">  
    // Incluimos el formulario con los campos a completar de accidente.  
    <?php require_once 'formulariosSiniestro/formAccidente.php' ?>  
</form>  
  
<ul class="mensaje" id="mensajeAccidente"></ul>
```

› [4.3.5.5.1.2 Fichero “divConductor.php”](#)

```
// Formulario que representa la sección de conductor al registrar un nuevo  
siniestro.  
<form id="divConductor" class="formulario divConductor">  
    // Incluimos el formulario con los campos a completar de conductor.  
    <?php require_once 'formulariosSiniestro/formConductor.php' ?>  
</form>  
  
<ul id="mensajeConductor" class="mensaje"></ul>
```

› [4.3.5.5.1.3 Fichero “divDenunciante.php”](#)

```
// Formulario que representa la sección de denunciante al registrar un nuevo  
siniestro.  
<form id="divDenunciante" class="formulario divDenunciante">  
    // Incluimos el formulario con los campos a completar de denunciante.  
    <?php require_once 'formulariosSiniestro/formDenunciante.php' ?>  
</form>  
  
<ul class="mensaje" id="mensajeDenunciante"></ul>
```

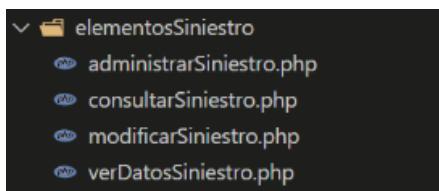
› [4.3.5.5.1.4 Fichero “divReclamante.php”](#)

```
// Formulario que representa la sección de reclamante al registrar un nuevo  
siniestro.  
<form id="divReclamante" class="formulario divReclamante">  
    // Incluimos el formulario con los campos a completar de reclamante.  
    <?php require_once 'formulariosSiniestro/formReclamante.php' ?>  
</form>  
  
<ul id="mensajeReclamante" class="mensaje"></ul>
```

› [4.3.5.5.1.5 Fichero “divSiniestro.php”](#)

```
// Formulario que representa la sección de siniestro al registrar un nuevo  
siniestro.  
<form id="divSiniestro" class="formulario divSiniestro">  
    // Incluimos el formulario con los campos a completar de siniestro.  
    <?php require_once 'formulariosSiniestro/formSiniestro.php' ?>  
</form>  
  
<ul class="mensaje" id="mensajeSiniestro"></ul>
```

* [4.3.5.5.2 Carpeta “elementosSiniestro”](#)



> 4.3.5.5.2.1 Fichero “administrarSiniestro.php”

```
// Instanciamos la clase para los datos de los métodos.  
<?php  
    include_once('../.../controlador/siniestrosController.php');  
    include_once('../.../config/db.php');  
    $objSin = new SiniestrosController();  
?>  
  
// Tabla donde se muestra la lista de siniestros registrados.  
<div class="row" id="consultarPolRow">  
    <div class="col s12">  
        <table id="administrarSiniestroTabla" class="centered striped">  
            <thead>  
                <tr>  
                    <th scope="col">Nro Siniestro</th>  
                    <th scope="col">Nro Poliza</th>  
                    <th scope="col">Tomador</th>  
                    <th scope="col">Tipo</th>  
                    <th scope="col">Denuncia</th>  
                    <th scope="col">Estado</th>  
                    <th scope="col">Administrar</th>  
                </tr>  
            </thead>  
            <tbody></tbody>  
        </table>  
    </div>  
</div>  
  
// Modal para cambiar el estado del siniestro.  
<div id="modalCambiarEstado" class="modal modal-fixed-footer">  
    <div class="modal-content formulario">  
        <div class="tituloFormulario">Cambiar Estado de Siniestro</div>  
        // Listamos los estados de siniestro desde el objeto.  
        <select id="selectEstadoSiniestro" ><?= $objSin->  
listadoEstadosSiniestro() ?></select>  
    </div>  
    <div class="modal-footer">  
        <button type="button" class="btn btn-flat blue modal-close" id="cancelarCambiarEstadoSin"><i class="fas fa-times"></i> cerrar</button>  
        <button type="button" class="btn btn-flat green" id="aceptarCambiarEstadoSin" > <i class="fas fa-save"></i> guardar </button>  
    </div>  
</div>  
  
// Modal para administrar las imágenes del siniestro.  
<div id="modalAdminImgSin" class="modal modal-fixed-footer">  
    <div class="modal-content formulario">  
        <div class="tituloFormulario">Administrar imágenes del siniestro</div>
```

```

// Formulario para seleccionar y cargar imágenes.
<form enctype="multipart/form-data">
    <div class="divBtnsImagenSiniestro row">

        <button type="button" class="btn btn-flat btn-floating blue" id="nuevaImagenBtn"> <i class="fas fa-plus"></i> </button>
        <label for="imagenSin">Nueva imagen</label>
        <input type="file" name="imagenSin" id="imagenSin">

        <div class="btnImg">
            <div id="estadoSubida"> </div>
            <div id="divBot"> <button type="button" class="btn btn-flat green" id="cargarImagen"> <i class="fas fa-upload"></i> cargar </button> </div>
            </div>
        </div>
    </form>
    // Galeria donde se visualizan las imágenes.
    <div class="galeriaImg">
        <div class="gallery-container"></div>
    </div>
</div>
<div class="modal-footer">
    <span class="subtituloFormulario" id="infoPieImgSin">
        Siniestro <span id="nroSiniestroPie"> </span>
        <span id="marcaModeloPie"></span>
    </span>
    <button type="button" class="btn btn-flat blue modal-close" id="cerrarModalAdminImgSin"> <i class="fas fa-times"></i> cerrar </button>
</div>
</div>

<script type="text/javascript" src="js/siniestros.js"></script>

```

» 4.3.5.5.2.2 Fichero “consultarSiniestro.php”

```
// Tabla donde se listan todos los siniestros para consultar sus datos.
<div class="row" id="consultarPolRow">
    <div class="col s12">
        <table id="consultarSiniestroTabla" class="centered striped">
            <thead>
                <tr>
                    <th scope="col">Nro Siniestro</th>
                    <th scope="col">Nro Poliza</th>
                    <th scope="col">Tomador</th>
                    <th scope="col">Fecha Denuncia</th>
                    <th scope="col">Estado</th>
                    <th scope="col">Ver</th>
                </tr>
            </thead>
            <tbody></tbody>
        </table>
    </div>
</div>

<script type="text/javascript" src="js/siniestros.js"></script>
```

» 4.3.5.5.2.3 Fichero “modificarSiniestro.php”

```
// Vista del formulario general para emitir un nuevo siniestro.
<form id="modificarSiniestro" class="formulario">

    <ul id="tabFormularioSiniestro" class="tabs tabs-fixed-width tab-demo z-depth-1 ">
        <li class="tab"><a id="tab1" href="#test1" class="active">Informacion General <i class="fas fa-chevron-circle-right"></i> </a> </li>
        <li class="tab"><a id="tab2" href="#test2">Denunciante <i class="fas fa-chevron-circle-right"></i> </a></li>
        <li class="tab"><a id="tab3" href="#test3">Accidente <i class="fas fa-chevron-circle-right"></i> </a></li>
        <li class="tab"><a id="tab4" href="#test4">Reclamante <i class="fas fa-chevron-circle-right"></i> </a></li>
        <li class="tab"><a id="tab5" href="#test5">Conductor</a></li>
    </ul>
    // Incluimos en cada tab los formularios en su correspondiente sección.
    <div id="test1" class="col s12"> <?php include_once "../formulariosSiniestro/formSiniestro.php"; ?> </div>
    <div id="test2" class="col s12"> <?php include_once "../formulariosSiniestro/formDenunciante.php"; ?> </div>
    <div id="test3" class="col s12"> <?php include_once "../formulariosSiniestro/formAccidente.php"; ?> </div>
    <div id="test4" class="col s12"> <?php include_once "../formulariosSiniestro/formReclamante.php"; ?> </div>
```

```

<div id="test5" class="col s12"> <?php include_once "../formulariosSiniestro/formConductor.php"; ?> </div>

    <div class="btntsModificarSiniestro">
        <button type="button" class="btn btn-flat blue" id="btnVolverModificarSin"> <i class="fas fa-caret-left"></i> volver </button>
        <div class="btntsModSinHijo">
            <button type="submit" class="btn btn-flat green"> confirmar <i class="fas fa-save"></i> </button>
        </div>
    </div>

</form>

<ul id="mensajeModSin" class="mensaje"></ul>

<script type="text/javascript" src="js/siniestros.js"></script>

```

› 4.3.5.5.2.4 Fichero “verDatosSiniestro.php”

```

// Vista donde se mostraran todos los datos del siniestro.
<div class="infoSin-1">

    // Seccion accidente.
    <div class="infoSin-2">
        <div class="infoSin-3-titGeneral">
            <div class="tituloSin">Accidente</div>
        </div>

        <div class="infoSin-3">
            <div class="infoSin-4-titulo">Ubicacion</div>
            <div class="infoSin-4-contenido ubiAcc"></div>
        </div>
        <div class="infoSin-3">
            <div class="infoSin-4-titulo">Descripcion</div>
            <div class="infoSin-4-contenido descAcc"></div>
        </div>
    </div>

    // Seccion poliza.
    <div class="infoSin-2">
        <div class="infoSin-3-titGeneral">
            <div class="tituloSin">Poliza</div>
        </div>

        <div class="infoSin-3">
            <div class="infoSin-4-titulo">Numero de poliza</div>
            <div class="infoSin-4-contenido numPol"></div>
        </div>
        <div class="infoSin-3">

```

```

        <div class="infoSin-4-titulo">Tomador</div>
        <div class="infoSin-4-contenido tomador"></div>
    </div>
    <div class="infoSin-3">
        <div class="infoSin-4-titulo">Vigencia</div>
        <div class="infoSin-4-contenido vigencia"></div>
    </div>
    <div class="infoSin-3">
        <div class="infoSin-4-titulo">Estado actual</div>
        <div class="infoSin-4-contenido estadoPol"></div>
    </div>
</div>

// Seccion siniestro.
<div class="infoSin-2">
    <div class="infoSin-3-titGeneral">
        <div class="tituloSin">Siniestro</div>
    </div>

    <div class="infoSin-3">
        <div class="infoSin-4-titulo">Numero</div>
        <div class="infoSin-4-contenido nroSin"></div>
    </div>
    <div class="infoSin-3">
        <div class="infoSin-4-titulo">Denuncia</div>
        <div class="infoSin-4-contenido fechaDen"></div>
    </div>
    <div class="infoSin-3">
        <div class="infoSin-4-titulo">Ocurrencia</div>
        <div class="infoSin-4-contenido fechaOcu"></div>
    </div>
    <div class="infoSin-3">
        <div class="infoSin-4-titulo">Tipo</div>
        <div class="infoSin-4-contenido tipoSin"></div>
    </div>
    <div class="infoSin-3">
        <div class="infoSin-4-titulo">Lesion/muerte</div>
        <div class="infoSin-4-contenido lesionMuerte"></div>
    </div>
    <div class="infoSin-3">
        <div class="infoSin-4-titulo">Estado</div>
        <div class="infoSin-4-contenido estadoSin"></div>
    </div>
</div>

// Seccion conductor tercero.
<div class="infoSin-2">
    <div class="infoSin-3-titGeneral">
        <div class="tituloSin">Conductor (Tercero)</div>
    </div>

    <div class="infoSin-3">

```

```

        <div class="infoSin-4-
titulo">Misma persona que el reclamante</div>
            <div class="infoSin-4-contenido condRec"></div>
        </div>
        <div class="infoSin-3">
            <div class="infoSin-4-titulo">Nombre</div>
            <div class="infoSin-4-contenido nombreCond"></div>
        </div>
        <div class="infoSin-3">
            <div class="infoSin-4-titulo">Domicilio</div>
            <div class="infoSin-4-contenido domCond"></div>
        </div>
        <div class="infoSin-3">
            <div class="infoSin-4-titulo">Telefono</div>
            <div class="infoSin-4-contenido telCond"></div>
        </div>
        <div class="infoSin-3">
            <div class="infoSin-4-titulo">Documento</div>
            <div class="infoSin-4-contenido docCond"></div>
        </div>
    </div>

// Seccion reclamante tercero.
<div class="infoSin-2">
    <div class="infoSin-3-titGeneral">
        <div class="tituloSin">Reclamante (Tercero)</div>
    </div>

    <div class="infoSin-3">
        <div class="infoSin-4-titulo">Misma persona que el tomador</div>
        <div class="infoSin-4-contenido tomRec"></div>
    </div>
    <div class="infoSin-3">
        <div class="infoSin-4-titulo">Nombre</div>
        <div class="infoSin-4-contenido nombreRec"></div>
    </div>
    <div class="infoSin-3">
        <div class="infoSin-4-titulo">Domicilio</div>
        <div class="infoSin-4-contenido domRec"></div>
    </div>
    <div class="infoSin-3">
        <div class="infoSin-4-titulo">Telefono</div>
        <div class="infoSin-4-contenido telRec"></div>
    </div>
    <div class="infoSin-3">
        <div class="infoSin-4-titulo">Documento</div>
        <div class="infoSin-4-contenido docRec"></div>
    </div>
</div>

// Seccion denunciante.
<div class="infoSin-2">
    <div class="infoSin-3-titGeneral">
        <div class="tituloSin">Denunciante</div>

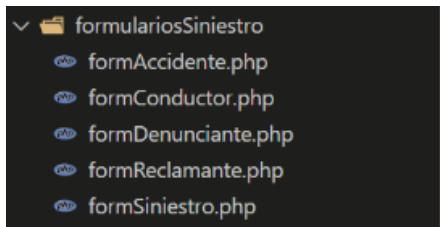
```

```
</div>

<div class="infoSin-3">
    <div class="infoSin-4-titulo">Misma persona que el Tomador</div>
        <div class="infoSin-4-contenido denunTom"></div>
</div>
<div class="infoSin-3">
    <div class="infoSin-4-titulo">Nombre</div>
        <div class="infoSin-4-contenido nombreDen"></div>
</div>
<div class="infoSin-3">
    <div class="infoSin-4-titulo">Documento</div>
        <div class="infoSin-4-contenido dniDen"></div>
</div>
<div class="infoSin-3">
    <div class="infoSin-4-titulo">Domicilio</div>
        <div class="infoSin-4-contenido domiDen"></div>
</div>
<div class="infoSin-3">
    <div class="infoSin-4-titulo">Telefono</div>
        <div class="infoSin-4-contenido telDen"></div>
</div>
<div class="infoSin-3">
    <div class="infoSin-4-titulo">Email</div>
        <div class="infoSin-4-contenido emailDen"></div>
</div>
</div>
</div>

<div class="btnVolverConsultarSinPadre">
    <button type="button" class="btn btn-flat blue" id="volverConsultarSin"> <i class="fas fa-caret-left"></i> volver </button>
</div>
<script type="text/javascript" src="js/siniestros.js"></script>
```

* 4.3.5.5.3 Carpeta "formulariosSiniestro"



> 4.3.5.5.3.1 Fichero "formAccidente.php"

```
// Formulario que contiene los campos a completar de la sección accidente.  
<?php  
    // Dependiendo de la ubicación de donde se llame el presente fichero,  
    // instanciaremos las clases necesarias a través de una ruta compatible.  
    if(file_exists('../controlador/funcionesGlobalesController.php') && file  
    _exists('../config/db.php')){  
        include_once('../controlador/funcionesGlobalesController.php');  
        include_once('../config/db.php');  
    }else if(file_exists('../controlador/funcionesGlobalesController.php'  
) && file_exists('../.....config/db.php')){  
        include_once('../.....controlador/funcionesGlobalesController.php');  
        include_once('../.....config/db.php');  
    }  
    // Instanciamos la clase para utilizar los datos devueltos por los métodos.  
    $objFG = new FuncionesGlobalesController();  
?  
  
// Formulario general con los campos.  
<fieldset id="fieldsetAccidente">  
  
    <h2 class="tituloFormulario">accidente</h2>  
  
    <div class="row">  
        <div class="col s12">  
            <label for="decripcionAcc">Descripción</label>  
            <textarea id="decripcionAcc" name="decripcionAcc" class="validarAcc"></textarea>  
        </div>  
    </div>  
  
    <div class="row">  
        <div class="col s6">  
            <label for="provinciaAcc">Provincia</label>  
            <select id="provinciaAcc" class="select2 validarAcc" name="provinciaAcc">  
                // Llenamos el select con los datos devueltos por el método.  
                <?= $objFG->listadoProvincias() ?>  
            </select>  
        </div>  
        <div class="col s6">
```

```

        <label for="localidadAcc">Localidad</label>
        <select id="localidadAcc" class="select2 validarAcc" name="localidadAcc">
    ">
        <option value="0">Seleccione una Provincia</option>
    </select>
</div>
</div>

<div class="row">
    <div class="col s6">
        <label for="calleAcc">Calle</label>
        <input type="text" id="calleAcc" name="calleAcc" class="validarAcc">
    </div>
    <div class="col s6">
        <label for="alturaAcc">Altura</label>
        <input type="text" id="alturaAcc" name="alturaAcc" class="validarAcc">
    >
    </div>
</div>

<button type="button" class="btn btn-flat blue" id="atrasAccidente"><i class="fas fa-caret-left"></i> atras</button>
<span class="btnAcc"></span>

</fieldset>

```

› 4.3.5.5.3.2 Fichero “formConductor.php”

```

// Formulario que contiene los campos a completar de la sección conductor.
<fieldset id="fieldsetConductor">

    <h2 class="tituloFormulario">conductor (tercero)</h2>

    <div class="selectCondRecPadre">
        <label for="selectCondRec">El Conductor es la misma persona que el Reclamante</label>
        <div>
            <select name="selectCondRec" id="selectCondRec" class="validarCond">
                <option value="0" selected disabled>Seleccione</option>
                <option value="1">Si</option>
                <option value="2">No</option>
            </select>
        </div>
    </div>

    <div class="row">
        <div class="col s6">
            <label for="nombreConductor">Nombre Completo</label>
            <input type="text" id="nombreConductor" name="nombreConductor" class="validarCond">
        </div>
    </div>

```

```

        </div>

        <div class="col s6">
            <label for="domicConductor">Domicilio</label>
            <input type="text" id="domicConductor" name="domicConductor" class="validarCond">
        </div>
    </div>

    <div class="row">
        <div class="col s6">
            <label for="telefonoConductor">Telefono</label>
            <input type="text" id="telefonoConductor" name="telefonoConductor" class="validarCond">
        </div>

        <div class="col s6">
            <label for="dniConductor">DNI</label>
            <input type="text" id="dniConductor" name="dniConductor" class="validarCond">
        </div>
    </div>

    <button type="button" class="btn btn-flat blue" id="atrasConductor"><i class="fas fa-caret-left"></i> atras</button>
    <span class="btnSinConductor"></span>
</fieldset>

```

› 4.3.5.5.3.3 Fichero “formDenunciante.php”

```

// Formulario que contiene los campos a completar de la sección denunciante.
<fieldset id="fielsetDenunciante">

    <h2 class="tituloFormulario">denunciante</h2>

    <div class="selectDenunAsegPadre">
        <label for="selectDenunAseg">El Denunciante es la misma persona que el Asurado</label>
        <select name="selectDenunAseg" id="selectDenunAseg" class="validarDen">
            <option value="0" selected disabled>Seleccione</option>
            <option value="1">Si</option>
            <option value="2">No</option>
        </select>
    </div>

    <div class="row">
        <div class="col s4">
            <label for="nombreDenunciante">Nombre Completo</label>

```

```

<input type="text" id="nombreDenunciante" name="nombreDenunciante" class="form-control" placeholder="Nombre" required="required" data-validation-required-message="Este campo es requerido." data-bbox="137 83 860 130">
</div>

<div class="col s4">
    <label for="dniDenunciante">DNI</label>
    <input type="text" id="dniDenunciante" name="dniDenunciante" class="form-control" placeholder="DNI" required="required" data-validation-required-message="Este campo es requerido." data-bbox="137 150 860 215">
</div>

<div class="col s4">
    <label for="domicilioDenunciante">Domicilio</label>
    <input type="text" id="domicilioDenunciante" name="domicilioDenunciante" class="form-control" placeholder="Domicilio" required="required" data-validation-required-message="Este campo es requerido." data-bbox="137 235 860 315">
</div>

<div class="row">
    <div class="col s4">
        <label for="telefonoDenunciante">Teléfono</label>
        <input type="text" id="telefonoDenunciante" name="telefonoDenunciante" class="form-control" placeholder="Teléfono" required="required" data-validation-required-message="Este campo es requerido." data-bbox="137 355 860 435">
    </div>

    <div class="col s4">
        <label for="emailDenunciante">E-mail</label>
        <input type="text" id="emailDenunciante" name="emailDenunciante" class="form-control" placeholder="E-mail" required="required" data-validation-required-message="Este campo es requerido." data-bbox="137 455 860 535">
    </div>

```

<button type="button" class="btn btn-flat blue" id="atrasDenunciante"><i class="fas fa-caret-left"></i> atras</button>

> 4.3.5.5.3.4 Fichero “formReclamante.php”

```
// Formulario que contiene los campos a completar de la sección reclamante.
<fieldset id="fieldsetReclamante">

    <h2 class="tituloFormulario">reclamante (tercero)</h2>

    <div class="row">
        <div class="col s6">
            <label for="nombreReclamante">Nombre Completo</label>
            <input type="text" id="nombreReclamante" name="nombreReclamante" class="validarRec">
        </div>
        <div class="col s6">
            <label for="domicReclamante">Domicilio</label>
            <input type="text" id="domicReclamante" name="domicReclamante" class="validarRec">
        </div>
    </div>

    <div class="row">
        <div class="col s6">
            <label for="telefonoReclamante">Telefono</label>
            <input type="text" id="telefonoReclamante" name="telefonoReclamante" class="validarRec">
        </div>
        <div class="col s6">
            <label for="dniReclamante">DNI</label>
            <input type="text" id="dniReclamante" name="dniReclamante" class="validarRec">
        </div>
    </div>

    <button type="button" class="btn btn-flat blue" id="atrasReclamante"><i class="fas fa-caret-left"></i> atras</button>
    <span class="btnSigRec"></span>

</fieldset>
```

> 4.3.5.5.3.5 Fichero “formSiniestro.php”

```
// Formulario que contiene los campos a completar de la sección siniestro.  
<?php  
    // Dependiendo de la ubicación de donde se llame el presente fichero,  
    // instanciaremos las clases necesarias a través de una ruta compatible.  
    if(file_exists('../controlador/siniestrosController.php') && file_exists  
('../../../config/db.php')){  
        include_once('../controlador/siniestrosController.php');  
        include_once('../../../config/db.php');  
    }else if(file_exists('../controlador/siniestrosController.php') && fi  
le_exists('../../../config/db.php')){  
        include_once('../controlador/siniestrosController.php');  
        include_once('../../../config/db.php');  
    }  
    // Instanciamos la clase para utilizar los datos devueltos por los métodos.  
    $objSin = new SiniestrosController();  
?  
  
// Formulario general con los campos.  
<fieldset id="fieldsetSiniestro">  
  
    <h2 class="tituloFormulario">siniestro</h2>  
  
    <div class="cabeceraInfoSin">  
        <div class="textoCabeceraNuevoSin"> Se asignara el numero <span id="nroSi  
n" class="textoCabNuevoSinAzul"></span> al siniestro a registrar </div>  
    </div>  
  
    <div class="row">  
        <div class="col s6">  
            <label for="fechaOcurrencia">Fecha de Ocurrencia</label>  
            <input type="text" id="fechaOcurrencia" name="fechaOcurrencia" placeholder="Seleccione" class="validarSin">  
        </div>  
  
        <div class="col s3">  
            <label for="horaOcurrencia">Hora de Ocurrencia</label>  
            <input type="time" id="horaOcurrencia" name="horaOcurrencia" placeholder="Ingrese" class="validarSin">  
        </div>  
  
        <div class="col s3">  
            <label for="LesMuer">Hubo Lesion/Muerte</label>  
            <select id="LesMuer" name="LesMuer" class="validarSin">  
                <option value="0" selected disabled>Seleccione</option>  
                <option value="NO">No</option>  
                <option value="SI">Si</option>  
            </select>  
        </div>  
  
    <div class="col s8">
```

```

        <label for="selectPolizas">Poliza</label>
        <select class="select2 validarSin" id="selectPolizas" name="selectPolizas"></select>
    </div>

    <div class="col s4">
        <label for="TipoSin">Tipo de Siniestro</label>
        <select id="TipoSin" name="TipoSin" class="validarSin">
            // Llenamos el select con los datos devueltos por el método.
            <?= $objSin->listadoTipoSiniestro() ?>
        </select>
    </div>
</div>

<span class="btnSigSin">
    <button type="submit" class="btn btn-flat green" id="siguienteSiniestro"> siguiente <i class="fas fa-caret-right"></i> </button>
</span>
</fieldset>

```

* 4.3.5.5.4 Fichero “cargarSiniestro.php”

```

// Estructura general del formulario para emitir un nuevo siniestro.
<ul id="tabFormularioSiniestro" class="tabs tabs-fixed-width tab-demo z-depth-1 ">
    <li class="tab"><a id="tab1" href="#test1" class="active">Informacion General <i class="fas fa-chevron-circle-right"></i> </a> </li>
    <li class="tab"><a id="tab2" href="#test2">Denunciante <i class="fas fa-chevron-circle-right"></i> </a></li>
    <li class="tab"><a id="tab3" href="#test3">Accidente <i class="fas fa-chevron-circle-right"></i> </a></li>
    <li class="tab"><a id="tab4" href="#test4">Reclamante <i class="fas fa-chevron-circle-right"></i> </a></li>
    <li class="tab"><a id="tab5" href="#test5">Conductor</a></li>
</ul>

// Insertamos en los tabs los formularios con las sección correspondiente.
<div id="test1" class="col s12"> <?php include_once "divSiniestro/divSiniestro.php"; ?> </div>
<div id="test2" class="col s12"> <?php include_once "divSiniestro/divDenunciante.php"; ?> </div>
<div id="test3" class="col s12"> <?php include_once "divSiniestro/divAccidente.php"; ?> </div>
<div id="test4" class="col s12"> <?php include_once "divSiniestro/divReclamante.php"; ?> </div>
<div id="test5" class="col s12"> <?php include_once "divSiniestro/divConductor.php"; ?> </div>

<script type="text/javascript" src="js/siniestros.js"></script>

```

➔ 4.3.5.6 Carpeta “sucursales”

```
✓ └── sucursales
    ├── administrarSucursales.php
    ├── modificarSucursal.php
    ├── nuevaSucursal.php
    └── verInfoSucursal.php
```

Esta carpeta contiene todos los ficheros relacionados a las vistas y formularios correspondientes a las sucursales.

* 4.3.5.6.1 Fichero “administrarSucursales.php”

```
// Tabla donde se muestra la lista de sucursales registradas.
<div class="row">
  <div class="col s12">
    <table id="admSucursalesTabla" class="centered striped">
      <thead>
        <tr>
          <th scope="col">Identificador</th>
          <th scope="col">Nombre</th>
          <th scope="col">Domicilio</th>
          <th scope="col">Cp</th>
          <th scope="col">Ubicacion</th>
          <th scope="col">Acciones</th>
        </tr>
      </thead>
      <tbody></tbody>
    </table>
  </div>
</div>

<script type="text/javascript" src="js/sucursales.js?v=13"></script>
```

* 4.3.5.6.2 Fichero "modificarSucursal.php"

```
// Vista relacionada al formulario donde con los campos para modificar la
sucursal.
<?php
    // Instanciamos las clases para utilizar los datos devueltos por sus
métodos.
    include_once('../controlador/funcionesGlobalesController.php');
    include_once('../config/db.php');
    $objFG = new FuncionesGlobalesController();
?>

// Formulario con los campos.

<form id="registroSucursal" class="formulario registroSucursal">
    <fieldset>
        <div class="tituloFormulario"> <i class="fas fa-
building"></i> modificar datos</div>
        <div class="subtituloFormulario"> <span class="sucursalNro"></span></
div>

        <div class="row">
            <div class="col s4">
                <label for="nombreSucursal">Nombre</label>
                <input type="text" name="nombreSucursal" id="nombreSucursal"
placeholder="Sucursal de X ciudad">
            </div>

            <div class="col s4">
                <label for="cuitSucursal">Cuit</label>
                <input type="text" name="cuitSucursal" id="cuitSucursal" plac
holder="00-00000000-0">
            </div>

            <div class="col s4">
                <label for="direccionSucursal">Direccion</label>
                <input type="text" name="direccionSucursal" id="direccionSucu
rsal" placeholder="Calle x Numero x">
            </div>
        </div>

        <div class="row">
            <div class="col s4">
                <label for="provinciaSucursal">Provincia</label>
                <select class="select2" name="provinciaSucursal" id="provinci
aSucursal">
                    // Listamos las provincias a través del objeto.
                    <?= $objFG->listadoProvincias() ?>
                </select>
            </div>
        </div>
    </fieldset>
</form>
```

```

        <div class="col s4">
            <label for="cpSucursal" >Codigo Postal</label>
            <select class="select2" name="cpSucursal" id="cpSucursal"></s
elect>
        </div>

        <div class="col s4">
            <label for="localidadSucursal">Localidad</label>
            <select class="select2" name="localidadSucursal" id="localida
dSucursal"></select>
        </div>

    </div>

    <div class="col s12 btnsNavSucursal">
        <button type="button" class="volverSucursal btn-
flat MAI blue"> <i class="fas fa-angle-left"></i> volver</button>
        <button type="submit" class="btnSubmitSucursal btn-
flat MAI green" id="submitModificarSucursal"> <i class="fas fa-
save"></i> modificar</button>
    </div>
</fieldset>

<ul id="mensajeSucursal" class="col s12 mensaje"></ul>
</form>

<script type="text/javascript" src="js/sucursales.js?v=14"></script>

```

* 4.3.5.6.3 Fichero “nuevaSucursal.php”

```

// Vista relacionada al formulario donde con los campos para agregar una
nueva sucursal.
<?php
    // Instanciamos las clases para utilizar los datos devueltos por sus
métodos.
    include_once('.../controlador/funcionesGlobalesController.php');
    include_once('.../config/db.php');
    $objFG = new FuncionesGlobalesController();
?>

// Formulario general con los campos.

<form id="registroSucursal" class="formulario registroSucursal">
    <fieldset>
        <div class="tituloFormulario" > <i class="fas fa-
building"></i> agregar nueva sucursal</div>

        <div class="row">
            <div class="col s4">

```

```

        <label for="nombreSucursal">Nombre</label>
        <input type="text" name="nombreSucursal" id="nombreSucursal"
placeholder="Sucursal de X ciudad">
    </div>

    <div class="col s4">
        <label for="cuitSucursal">Cuit</label>
        <input type="text" name="cuitSucursal" id="cuitSucursal" plac-
holder="00-00000000-0">
    </div>

    <div class="col s4">
        <label for="direccionSucursal">Direccion</label>
        <input type="text" name="direccionSucursal" id="direccionSucu-
rsal" placeholder="Calle x Numero x">
    </div>
    </div>

    <div class="row">

        <div class="col s4">
            <label for="provinciaSucursal">Provincia</label>
            <select class="select2" name="provinciaSucursal" id="provinci-
aSucursal">
                // Listamos las provincias a través del objeto.
                <?= $objFG->listadoProvincias() ?>
            </select>
        </div>

        <div class="col s4">
            <label for="cpSucursal" >Codigo Postal</label>
            <select class="select2" name="cpSucursal" id="cpSucursal">
                <option selected disabled>Seleccione Provincia</option>
            </select>
        </div>

        <div class="col s4">
            <label for="localidadSucursal">Localidad</label>
            <select class="select2" name="localidadSucursal" id="localida-
dSucursal">
                <option selected disabled>Seleccione Codigo Postal</optio-
n>
            </select>
        </div>

    </div>

    <div class="col s12 btnsNavSucursal">
        <button type="button" class="volverSucursal btn-
flat MAI blue"> <i class="fas fa-angle-left"></i> volver</button>
        <button type="submit" class="btnSubmitSucursal btn-
flat MAI green" id="submitSucursal"> <i class="fas fa-
save"></i> registrar</button>
    </div>

```

```

</fieldset>

<ul id="mensajeSucursal" class="col s12 mensaje"></ul>
</form>

<script type="text/javascript" src="js/sucursales.js?v=14"></script>

```

* 4.3.5.6.4 Fichero “verInfoSucursal.php”

```

// Vista donde se muestran los datos de la sucursal.
<div class="formulario divVerInfoSucursal">
    <fieldset>
        <div class="tituloFormulario"> <i class="fas fa-building"></i> consultar informacion</div>
        <div class="subtituloFormulario"> <span class="sucursalNro"></span></div>

        <div class="infoUser-1">

            // Datos de la empresa.
            <div class="tituloInfo">Empresa</div>
            <div class="infoUser-2">
                <div class="infoUser-3 row">
                    <div class="infoUser-4 col s4">
                        <div class="infoUser-5-subt">Nombre</div>
                        <div class="infoUser-5-inf nomb"></div>
                    </div>
                    <div class="infoUser-4 col s4">
                        <div class="infoUser-5-subt">Identificador</div>
                        <div class="infoUser-5-inf iden"></div>
                    </div>
                    <div class="infoUser-4 col s4">
                        <div class="infoUser-5-subt">Cuit</div>
                        <div class="infoUser-5-inf cuit"></div>
                    </div>
                </div>
            </div>
        </div>

        // Datos de ubicación,
        <div class="tituloInfo">Ubicacion</div>
        <div class="infoUser-2">
            <div class="infoUser-3 row">
                <div class="infoUser-4 col s3">
                    <div class="infoUser-5-subt">Direccion</div>
                    <div class="infoUser-5-inf dir"></div>
                </div>
                <div class="infoUser-4 col s3">
                    <div class="infoUser-5-subt">Codigo Postal</div>
                    <div class="infoUser-5-inf cp"></div>
                </div>
            </div>
        </div>
    </fieldset>
</div>

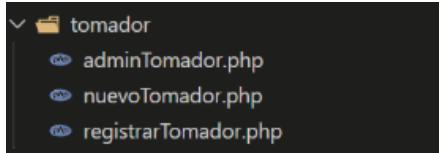
```

```
<div class="infoUser-4 col s3">
    <div class="infoUser-5-subt">Provincia</div>
    <div class="infoUser-5-inf pro"></div>
</div>
<div class="infoUser-4 col s3">
    <div class="infoUser-5-subt">Localidad</div>
    <div class="infoUser-5-inf loc"></div>
</div>
</div>

<div class="col s12 btnsNavSucursal">
    <button type="button" class="volverSucursal btn-flat MAI blue"> <i class="fas fa-angle-left"></i> volver</button>
</div>
</fieldset>
</div>

<script type="text/javascript" src="js/sucursales.js?v=14"></script>
```

➔ 4.3.5.7 Carpeta “tomador”



Carpeta que contiene todos los ficheros relacionados a las vistas y formularios que corresponden a los clientes/tomadores.

* 4.3.5.7.1 Fichero “adminTomador.php”

```
// Tabla donde se muestra la lista de tomadores registrados.
<div class="row" id="consultarPolRow">
    <div class="col s12">
        <table id="modificarDatosTomadorTabla" class="centered striped">
            <thead>
                <tr>
                    <th scope="col">Nombre</th>
                    <th scope="col">Nro Tom</th>
                    <th scope="col">Provincia</th>
                    <th scope="col">Localidad</th>
                    <th scope="col">Acciones</th>
                </tr>
            </thead>
            <tbody></tbody>
        </table>
    </div>
</div>

// Modal para modificar los datos del tomador.
<div id="modificarDatosTomadorModal" class="modal modal-fixed-footer">
    <form id="nuevoCliente" class="formulario">
        <div class="modal-content">

            <div id="cuerpoModal">
                // Incluimos el fichero con el formulario de nuevo tomador.
                <?php include_once 'nuevoTomador.php'; ?>
            </div>

            <ul class="mensaje mensajeModal"></ul>

        </div>

        <div class="modal-footer">
            <button type="button" class="btn btn-flat blue modal-close" id="cancelarClienteAdmin"> <i class="fas fa-times"></i> cerrar </button>
            <button type="submit" class="btn btn-flat green" id="grabarClienteAdmin"> <i class="fas fa-save"></i> grabar </button>
        </div>
    </form>
</div>
```

```

        </div>

        </form>
    </div>

// Modal para consultar los datos del tomador.
<div id="consultarTomModal" class="modal modal-fixed-footer">
    <div class="modal-content">
        <div class="tituloFormulario">Consultar datos tomador <span id="nroTom"></span></div>

        // Vista con los campos.
        <div class="infoTom-1">
            <div class="infoTom-2">
                <div class="infoTom-3">
                    <div class="infoTom-4-titulo">Nombre Completo</div>
                    <div class="infoTom-4-contenido nomT"></div>
                </div>
                <div class="infoTom-3">
                    <div class="infoTom-4-titulo">Documento</div>
                    <div class="infoTom-4-contenido docT"></div>
                </div>
                <div class="infoTom-3">
                    <div class="infoTom-4-titulo">Tipo de Persona</div>
                    <div class="infoTom-4-contenido perT"></div>
                </div>
                <div class="infoTom-3">
                    <div class="infoTom-4-titulo">Nacionalidad</div>
                    <div class="infoTom-4-contenido nacioT"></div>
                </div>
                <div class="infoTom-3">
                    <div class="infoTom-4-titulo">Provincia</div>
                    <div class="infoTom-4-contenido proT"></div>
                </div>
                <div class="infoTom-3">
                    <div class="infoTom-4-titulo">Localidad</div>
                    <div class="infoTom-4-contenido locT"></div>
                </div>
                <div class="infoTom-3">
                    <div class="infoTom-4-titulo">Codigo Postal</div>
                    <div class="infoTom-4-contenido cpT"></div>
                </div>
                <div class="infoTom-3">
                    <div class="infoTom-4-titulo">Calle</div>
                    <div class="infoTom-4-contenido callT"></div>
                </div>
                <div class="infoTom-3">
                    <div class="infoTom-4-titulo">Genero</div>
                    <div class="infoTom-4-contenido genT"></div>
                </div>
                <div class="infoTom-3">
                    <div class="infoTom-4-titulo">Telefono</div>
                    <div class="infoTom-4-contenido telT"></div>
                </div>
            </div>
        </div>
    </div>

```

```

        </div>
        <div class="infoTom-3">
            <div class="infoTom-4-titulo">Correo</div>
            <div class="infoTom-4-contenido corrT"></div>
        </div>
        <div class="infoTom-3">
            <div class="infoTom-4-titulo">Fecha de Nacimiento</div>
            <div class="infoTom-4-contenido nacimT"></div>
        </div>

        </div>
    </div>
</div>

<div class="modal-footer">
    <button type="button" class="btn btn-flat blue modal-
close" id="grabarClienteAdmin"> <i class="fas fa-times"></i> cerrar </button>
</div>
</div>

<script type="text/javascript" src="js/cliente.js?v=3"></script>

```

* 4.3.5.7.2 Fichero “nuevoTomador.php”

```

// Dependiendo de la ubicación de donde se llame el presente fichero,
instanciaremos las clases necesarias a través de la ruta que sea compatible.
<?php
    if(file_exists('../controlador/funcionesGlobalesController.php') && fi
le_exists('../config/db.php')){
        include_once('../controlador/funcionesGlobalesController.php');
        include_once('../config/db.php');
    }else if(file_exists('../../controlador/funcionesGlobalesController.ph
p') && file_exists('../../config/db.php')){
        include_once('../../controlador/funcionesGlobalesController.php');
        include_once('../../config/db.php');
    }
    // Instanciamos la clase.
    $objFG = new FuncionesGlobalesController();
?>

<!!--<form id="nuevoCliente" class="formulario">-->
<fieldset>

    <div class="tituloFormulario tomador">datos del nuevo tomador</div>

    <div class="row">
        <div class="col s4">
            <label for="nombre">Nombre/Razon Social</label>

```

```

        <input type="text" class="campo" name="nombre" id="nombre" placeholder="Nombre Apellido">
    </div>

    <div class="col s4">
        <label for="documento">Documento/CIUT</label>
        <input type="text" class="campo" name="documento" id="documento" maxlength="8" placeholder="0000000000 (Sin espacios)">
    </div>

    <div class="col s4">
        <label for="persona">Persona</label>
        <select class="campo" name="persona" id="persona">
            <option value="0" selected disabled>Seleccione</option>
            <option value="fisica">Física</option>
            <option value="juridica">Jurídica</option>
        </select>
    </div>
</div>

<div class="row">
    <div class="col s4">
        <label for="nacionalidad">Nacionalidad/País</label>
        <input type="text" class="campo" name="nacionalidad" id="nacionalidad" placeholder="Nombre País">
    </div>

    <div class="col s4">
        <label for="fech_nac">Nacimiento/Orig</label>
        <input type="text" class="campo" name="fech_nac" id="fech_nac" placeholder="00/00/0000 (Seleccione)">
    </div>

    <div class="col s4">
        <label for="provincia" >Provincia/Distrito</label>
        <select class="campo select2" name="provincia" id="provincia">
            // Llenamos el select con los datos del objeto.
            <?= $objFG->listadoProvincias() ?>
        </select>
    </div>
</div>

<div class="row">

    <div class="col s4">
        <label for="cp">Código Postal</label>
        <select class="campo select2" name="cp" id="cp"></select>
    </div>

    <div class="col s4">
        <label for="localidad">Localidad</label>
        <select class="campo select2" name="localidad" id="localidad"></select>
    </div>

```

```

        </div>

        <div class="col s4">
            <label for="telefono">Teléfono</label>
            <input type="text" class="campo" name="telefono" id="telefono" placeholder="0000000000 (Sin espacios)">
        </div>
    </div>

    <div class="row">
        <div class="col s4">
            <label for="calle">Calle</label>
            <input type="text" class="campo" name="calle" id="calle" placeholder="Nombre/Número Calle">
        </div>

        <div class="col s4">
            <label for="genero">Género</label>
            <select class="campo" name="genero" id="genero">
                <option value="0" selected disabled>Seleccione</option>
                <option value="masculino">Masculino</option>
                <option value="femenino">Femenino</option>
                <option value="indeterminado">Indeterminado</option>
            </select>
        </div>

        <div class="col s4">
            <label for="correo">Correo</label>
            <input type="email" class="campo" name="correo" id="correo" placeholder="correo@ejemplo.com (Opcional)">
        </div>
    </div>

</fieldset>
<!--</form>-->

```

* 4.3.5.7.3 Fichero “*registrarTomador.php*”

```
// Formulario con los campos para grabar un nuevo tomador.
<form id="nuevoCliente" class="formulario nuevoCliente">
    // Incluimos el formulario de nuevo tomador.
    <?php include_once 'nuevoTomador.php'; ?>

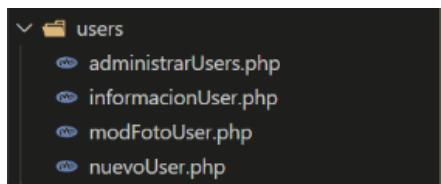
    <div class="bttnsNuevoTomador">
        <div class="bttnsNuevoTomadorHijo">
            <button type="button" class="btn btn-
flat red cancelarClienteSec" id="cancelarClienteSec"> <i class="fas fa-
times"></i> cancelar </button>
            <button type="submit" class="btn btn-
flat green grabarCliente" id="grabarClienteSec"> <i class="fas fa-
save"></i> grabar </button>
        </div>
    </div>

    <ul id="mensajeTomador" class="mensaje ul2 mensajeTomador"></ul>

</form>

<script type="text/javascript" src="js/cliente.js?v=3"></script>
```

➔ 4.3.5.8 Carpeta “users”



En esta carpeta se guardan todos los ficheros correspondientes a las vistas y formularios de los usuarios del sistema.

* 4.3.5.8.1 Fichero “administrarUsers.php”

```
// Tabla donde se muestra la lista de usuarios del sistema.

<div class="row rowListadoUsers">
    <table id="tablaListadoUsers" class="centered">
        <thead>
            <tr>
                <th scope="col"><i class="fas fa-check-circle"></i></th>
                <th scope="col">Número</th>
                <th scope="col">Nombre</th>
                <th scope="col">Rol</th>
                <th scope="col">Estado</th>
            </tr>
        </thead>
        <tbody></tbody>
    </table>
</div>

// Modal donde se muestra el listado de usuarios dados de baja.
<div id="modalListadoUserBaja" class="modal modal-fixed-footer">
    <div class="modal-content">
        <div class="tituloFormulario">Listado de usuarios dados de baja</div>
        <div class="tablaUsrBaja">
            // Tabla con el listado de usuarios dados de baja.
            <table id="listadoBajaUser" class="centered">
                <thead>
                    <tr>
                        <th scope="col">Número</th>
                        <th scope="col">Nombre</th>
                        <th scope="col">Rol</th>
                        <th scope="col">Estado</th>
                        <th scope="col">Activar</th>
                    </tr>
                </thead>
                <tbody></tbody>
            </table>
        </div>
    </div>
    <div class="row modal-footer">
        <button type="button" class="btn btn-flat blue cerrarModalListadoUsBa"> <i class="fas fa-times"></i> cerrar </button>
    </div>
</div>

// Modal para modificar el rol de un usuario.
<div id="modalModificarRolUser" class="modal modal-fixed-footer">
    <div class="modal-content">
        <div class="tituloFormulario">Modificar rol del usuario</div>
        <div class="contentMR">
```

```

// Informacion del usuario.
<div class="div1MR">
    <div class="tituloMR">Informacion del usuario</div>
    <div class="div2MR">
        <div class="subtituloMR">Numero</div>
        <div class="infoMR num"></div>
        <div class="subtituloMR">Nombre</div>
        <div class="infoMR nombre">Erasmo Jesus Jordan</div>
        <div class="subtituloMR">Usuario</div>
        <div class="infoMR usuario">Jordanerasamo7</div>
    </div>
</div>
// Seccion para seleccionar rol.
<div class="div1MR">
    <div class="tituloMR">Rol</div>
    <div class="div2MR">
        <div class="div3MR">
            <p class="div3MR-p1">
                <label>
                    <input class="with-
gap checkUs" name="grupoCheck" type="radio"/>
                    <span>Usuario</span>
                </label>
                <a class="btn-flat"><i class="fas fa-user"></i></a>
            </p>
            <p>
                <label>
                    <input class="with-
gap checkAdm" name="grupoCheck" type="radio"/>
                    <span>Administrador</span>
                </label>
                <a class="btn-flat"><i class="fas fa-user-shield"></i></a>
            </p>
        </div>
    </div>
</div>
<div class="row modal-footer">
    <button type="button" class="btn btn-flat blue modal-
close"> <i class="fas fa-times"></i> cerrar </button>
    <button type="button" class="btn btn-
flat green enviarModRol"> <i class="fas fa-save"></i> confirmar </button>
</div>
</div>

// Modal para visualizar los datos del usuario.
<div id="modalVerInfoUser" class="modal modal-fixed-footer">
    <div class="modal-content">
        <div class="tituloFormulario">Informacion del usuario</div>
        // Incluimos el fichero con los datos del usuario.
        <div> <?php require_once 'informacionUser.php' ?> </div>
    </div>
    <div class="row modal-footer">

```

```

        <button type="button" class="btn btn-flat blue modal-close"> <i class="fas fa-times"></i> cerrar </button>
    </div>
</div>

// Modal para reenviar los datos de acceso.
<div class="modalReenviarInfoAcceso modal">
    <form class="formReenviarInfoAcceso formulario" id="formCorreo">
        <div class="modal-content">
            <div class="tituloFormulario">Reenviar informacion de acceso</div>
            <label for="correo" class="correoLab">Correo electronico</label>
            <input type="email" class="correoUs" name="correo">
            <div class="mjeCorr"></div>
        </div>
        <div class="modal-footer">
            <button type="button" class="btn btn-flat blue modal-close" id="cerrarModalReenviarInfoAcceso"> <i class="fas fa-times"></i> cerrar </button>
            <button type="submit" class="btn btn-flat green" id="btnEnviarInfoAcceso"> enviar <i class="fas fa-paper-plane"> </i> </button>
        </div>
    </form>
</div>

<script type="text/javascript" src="js/users.js?v=3"></script>

```

* 4.3.5.8.1 Fichero “informacionUser.php”

```

// Vista donde se mostraran los datos del usuario.
<div class="infoUser-1">

    // Datos relacionados al sistema.
    <div class="tituloInfo">Sistema</div>
    <div class="infoUser-2">
        <div class="infoUser-3 row">

            <div class="infoUser-4 col s3">
                <div class="infoUser-5-subt">Numero</div>
                <div class="infoUser-5-inf nume"></div>
            </div>
            <div class="infoUser-4 col s3">
                <div class="infoUser-5-subt">Usuario</div>
                <div class="infoUser-5-inf usu"></div>
            </div>
            <div class="infoUser-4 col s3">
                <div class="infoUser-5-subt">Estado actual</div>
            </div>
        </div>
    </div>

```

```

        <div class="infoUser-5-inf estado"></div>
    </div>
    <div class="infoUser-4 col s3">
        <div class="infoUser-5-subt">Rol</div>
        <div class="infoUser-5-inf rol"></div>
    </div>

    </div>

</div>

// Datos personales.
<div class="tituloInfo">Personales</div>
<div class="infoUser-2">

    <div class="infoUser-3 row">

        <div class="infoUser-4 col s3">
            <div class="infoUser-5-subt">Nombre</div>
            <div class="infoUser-5-inf nomb"></div>
        </div>
        <div class="infoUser-4 col s3">
            <div class="infoUser-5-subt">Documento</div>
            <div class="infoUser-5-inf doc"></div>
        </div>
        <div class="infoUser-4 col s3">
            <div class="infoUser-5-subt">Fecha nacimiento</div>
            <div class="infoUser-5-inf naci"></div>
        </div>
        <div class="infoUser-4 col s3">
            <div class="infoUser-5-subt">Genero</div>
            <div class="infoUser-5-inf gen"></div>
        </div>

    </div>
    </div>

// Datos de contacto.
<div class="tituloInfo">Contacto</div>
<div class="infoUser-2">
    <div class="infoUser-3 row">

        <div class="infoUser-4 col s3">
            <div class="infoUser-5-subt">Correo</div>
            <div class="infoUser-5-inf correo"></div>
        </div>
        <div class="infoUser-4 col s3">
            <div class="infoUser-5-subt">Telefono</div>
            <div class="infoUser-5-inf tel"></div>
        </div>
        <div class="infoUser-4 col s3">
            <div class="infoUser-5-subt">Domicilio</div>
            <div class="infoUser-5-inf domi"></div>
        </div>

    </div>

```

```
<div class="infoUser-4 col s3">
    <div class="infoUser-5-subt">Localidad / País</div>
    <div class="infoUser-5-inf locPais"></div>
</div>

</div>
</div>
</div>
```

* 4.3.5.8.1 Fichero "modFotoUser.php"

```
// Vista con el formulario donde se podrá administrar la foto de perfil.
<form id="formNuevaFotoUser" class="formulario formNuevaFotoUser" enctype="multipart/form-data">
    <fieldset>
        <div class="tituloFormulario"> <i class="fas fa-user"></i> modificar foto de perfil</div>

        <label for="actualFoto">Foto actual</label>
        <div class="divActualFoto"></div>

        <label for="nuevaFoto">Cargar foto nueva</label>
        <input type="file" class="nuevaFoto">

        <button type="button" class="btn btn-flat green grabarNuevaFotoUser" > <i class="fas fa-save"></i> aceptar </button>
        <button type="button" class="btn btn-flat blue volverGrabarFotoUser" > <i class="fas fa-caret-left"></i> volver </button>
    </fieldset>
</form>

<script type="text/javascript" src="js/users.js?v=5"></script>
```

* 4.3.5.8.1 Fichero "nuevoUser.php"

```
// Formulario con los campos para registrar un nuevo usuario.
<?php
    include_once('../controlador/funcionesGlobalesController.php');
    include_once('../config/db.php');
    // Instanciamos las clases para utilizar los datos de sus métodos.
    $objFG = new FuncionesGlobalesController();
?>

<form id="registroUser" class="formulario registroUser">
    <fieldset>
        <div class="tituloFormulario" > <i class="fas fa-user"></i> datos del nuevo usuario</div>

        <div class="row">
            <div class="col s4">
                <label for="nombreUser">Nombre completo</label>
                <input type="text" name="nombreUser" id="nombreUser" placeholder="Nombre Apellido">
            </div>
        </div>
    </fieldset>
</form>
```

```

        </div>

        <div class="col s4">
            <label for="documentoUser">Documento</label>
            <input type="text" name="documentoUser" id="documentoUser" maxlength="8" placeholder="0000000000 (Sin espacios)">
        </div>

        <div class="col s4">
            <label for="nacionalidadUser">Nacionalidad</label>
            <input type="text" name="nacionalidadUser" id="nacionalidadUser" placeholder="Argentina">
        </div>

        <div class="row">
            <div class="col s4">
                <label for="provinciaUser" >Provincia/Distrito</label>
                <select class="select2" name="provinciaUser" id="provinciaUser">
                    // Llenamos el select con los datos del objeto.
                    <?= $objFG->listadoProvincias() ?>
                </select>
            </div>

            <div class="col s4">
                <label for="localidadUser">Localidad</label>
                <select class="select2" name="localidadUser" id="localidadUser"></select>
            </div>

            <div class="col s4">
                <label for="direccionUser">Direccion</label>
                <input type="text" name="direccionUser" id="direccionUser" placeholder="Calle - Numero">
            </div>
        </div>

        <div class="row">
            <div class="col s4">
                <label for="nacUser">Fecha nacimiento</label>
                <input type="text" class="datepicker" name="nacUser" id="nacUser" placeholder="00/00/0000">
            </div>

            <div class="col s4">
                <label for="telefonoUser">Teléfono</label>
                <input type="text" name="telefonoUser" id="telefonoUser" placeholder="0000000000 (Sin espacios)">
            </div>

            <div class="col s4">
                <label for="correoUser">E-Mail</label>

```

```

        <input type="email" name="correoUser" id="correoUser" placeholder="correo@ejemplo.com">
            </div>
        </div>

        <div class="row">
            <div class="col s4">
                <label for="generoUser">Género</label>
                <select name="generoUser" id="generoUser">
                    <option value="0" selected disabled>Seleccione</option>
                    <option value="masculino">Masculino</option>
                    <option value="femenino">Femenino</option>
                    <option value="otro">Otro</option>
                </select>
            </div>

            <div class="UP"></div>

            <div class="col s4 rol">
                <label for="rolUser">Rol</label>
                <select name="rolUser" id="rolUser">
                    <option value="0" selected disabled>Seleccione</option>
                    <option value="usuario">Usuario</option>
                    <option value="admin">Administrador</option>
                </select>
            </div>
        </div>

        <div class="col s12 btnsRegistroUser">
            <button type="button" class="volverUser btn-flat MAI blue"> <i class="fas fa-angle-left"></i> volver</button>
            <button type="submit" class="registrarUser btn-flat MAI green" id="submitUser"> <i class="fas fa-save"></i> registrar</button>
        </div>
    </fieldset>

    <ul id="mensajeUser" class="mensaje"></ul>
</form>

<script type="text/javascript" src="js/users.js?v=3"></script>

```

4.3.6 Fichero "index.php"

Fichero principal donde se muestra la vista de inicio de sesión al sistema.

```
// Redireccionamos a la vista correspondiente dependiendo del tipo de usuario
que se encuentre en la actual sesión.
<?php
    session_start();
    if(isset($_SESSION['user'])){
        if($_SESSION['user']['rol']=='Administrador' || $_SESSION['user']['rol']=='Usuario'){
            header('Location: indexMaqueta.php');
        }
    }
?>

// Formulario para iniciar sesión.
<html>
    <head>
        <title>Aseguradora - Login</title>
        <meta http-equiv="Content-type" content="text/html; charset=utf-8" />
        <link rel="stylesheet" href="css/estilosLogin.css">
    </head>
    <body>
        <div class="mjeLogin"></div>

        <div class="login">
            <div class="logoImgLogin">
                
            </div>
            <h1>Acceso al sistema</h1>
            <form method="post" id="formLogin">
                <div class="divInput">
                    <i class="fas fa-user"></i>
                    <input type="text" name="u" id="u" placeholder="Correo o
Userio"/>
                </div>
                <div class="divInput">
                    <i class="fas fa-lock"></i>
                    <input type="password" name="p" id="p" placeholder="Contr
aseña"/>
                </div>
                <div class="divInput">
                    <i class="fas fa-building"></i>
                    <select class="suc" name="s" id="selectSuc"></select>
                </div>
                <button type="submit" id="envFormLogin" class="btn">
                    <span>Entrar <i class="fas fa-sign-in-
alt"></i></span>
                </button>
                <ul id="errorLogin"></ul>
            </form>
```

```

        </div>

        <script type="text/javascript" src="js/plugins/jQuery/jquery-
3.4.1.min.js"></script>
        <script type="text/javascript" src="js/login.js?v=1"></script>
        // Agregamos el plugin jQuery Validation.
        <script type="text/javascript" src="js/plugins/jQuery Validation Plug-
in/dist/jquery.validate.min.js"></script>
        <script type="text/javascript" src="js/plugins/jQuery Validation Plug-
in/dist/additional-methods.min.js"></script>

        // Agregamos la librería Font awesome.
        <script type="text/javascript" src="js/plugins/FontAwesome/fontAwesom-
e.js"></script>

    </body>
</html>
```

✳️ 4.3.7 Fichero “indexMaqueta.php”

Fichero donde se muestra la vista del sistema una vez el usuario haya iniciado sesión.

```

// Mostraremos el tipo de menú principal correspondiente dependiendo del tipo
de usuario que haya iniciado sesión.
<?php
    session_start();
    if(isset($_SESSION['user'])){
        if($_SESSION['user']['rol']=='Administrador'){
            ob_start();
            require_once 'vista/maqueta/sidebarAdmin.php';
            $sidebar = ob_get_clean();
        }else if($_SESSION['user']['rol']=='Usuario'){
            ob_start();
            require_once 'vista/maqueta/sidebarUser.php';
            $sidebar = ob_get_clean();
        }
    }else{
        header('Location: index.php');
    }
?>

// Incluimos las vistas correspondientes de la maqueta.
<!DOCTYPE html>
<html lang="es">
    <head>
        <meta http-equiv="Content-type" content="text/html; charset=utf-8" />
        <title>Aseguradora</title>
```

```
<?php require_once 'vista/maqueta/linksHead.php' ?>
</head>

<body>
    <div class="wrapper">
        <?php require_once 'vista/maqueta/topNavbar.php' ?>
        <div class="sidebar"> <?= $sidebar ?> </div>
        <?php require_once 'vista/maqueta/mainContainer.php' ?>
    </div>

    <?php require_once 'vista/maqueta/footerScripts.php' ?>
</body>
</html>
```


El programa de **Aseguradora** es una herramienta informática concebida para registrar, modificar y analizar información relativa en cuanto a la situación y las necesidades de las empresas de rubro asegurador. El ideal funcionamiento del programa se basa en la previa complementación de conocimientos básicos de los términos y conceptos utilizados en el rubro por parte de los usuarios. Este manual del software comienza con una presentación general de los términos llevados a cabo, y a continuación explica paso a paso las instrucciones necesarias para poder operar bajo cada una de sus funcionalidades.