

Facultad de Ciencias y Sistemas

Sistema de información web para la gestión de fidelización de clientes.

Trabajo Monográfico para optar al título de
Ingeniero de Sistemas

Elaborado por:

Br. Cristopher Josué
Zepeda Tellez
Carnet: 2011-37038

Br. Gloria Esther
Mayorga Mena
Carnet: 2011-36978

Br. Magaly
Magdalena Mayorga
Maradiaga
Carnet: 2010-33172

Tutor:

MSc. Ing. Danilo
Humberto Noguera
Rivera

Contenido

I.	Introducción.....	1
II.	Antecedentes.....	2
III.	Planteamiento del problema	6
IV.	Justificación	7
V.	Objetivos	8
VI.	Marco teórico.....	9
6.1.	Sistema	9
6.1.1.	Sistema de información.....	9
6.1.1.1.	Sistema de información Web.....	10
6.2.	Terminología del negocio	11
6.2.1.	Distribuidor.....	11
6.2.2.	Programa de referidos.....	11
6.2.2.1.	Modelo Piramidal	12
6.2.3.	Recompensa	12
6.2.4.	Fidelización	13
6.3.	Requerimientos de Software	13
6.3.1.	Tipos de requerimientos	13
6.3.1.1.	Requerimientos de negocio.....	14
6.3.1.2.	Requerimientos funcionales	14
6.3.1.3.	Requerimientos no funcionales.....	14
6.3.1.4.	Requerimientos de implementación	16
6.3.1.5.	Requerimientos de usuario.....	16
6.3.2.	Levantamiento de requerimientos	16
6.3.3.	Análisis de requerimiento	18
6.3.4.	Modelado	19
6.3.4.1.	Modelo basado en escenarios	20
6.3.5.	Tabulación (Plantillas Coleman)	21
6.4.	Diseño de Software	21
6.4.1.	UML	22
6.4.2.	UWE	23
6.4.3.	Tipos de modelo.....	24
6.4.3.1.	Modelo de Requerimientos.....	24

6.4.3.2.	Modelo de Contenido	25
6.4.3.3.	Modelo de Navegación.....	25
6.4.3.4.	Modelo de presentación.....	26
6.4.3.5.	Modelo de procesos	27
6.4.4.	Patrones de diseño.....	27
6.4.4.1.	Patrón CQRS.....	28
6.4.5.	Herramientas de Modelado	29
6.4.5.1.	MagicDraw	29
6.4.5.2.	MagicUWE	30
6.4.5.3.	BPMN	30
6.5.	Construcción de Software	30
6.5.1.	Aplicación de página única (SPA)	31
6.5.2.	Alojamiento del sistema Web.....	31
6.5.2.1.	Computación en la nube	32
6.5.2.2.	Plataforma Azure	32
6.5.2.3.	Azure App Service	33
6.5.2.4.	Azure WebJob.....	33
6.5.3.	Protocolos de red.....	34
6.5.3.1.	HTTP	34
6.5.3.2.	HTTPS.....	35
6.5.3.3.	SSL/ TLS.....	35
6.5.3.4.	SFTP.....	36
6.5.4.	REST API.....	36
6.5.4.1.	Verbos HTTP/REST	37
6.5.4.2.	JSON	38
6.5.5.	Tecnologías Front-End.....	39
6.5.5.1.	HTML5	39
6.5.5.2.	Hojas de estilo en cascada (CSS).....	39
6.5.5.3.	JavaScript	40
6.5.5.4.	Frameworks de Front-End	40
6.5.5.4.1.	Angular.....	40
6.5.6.	Tecnologías Back-End	41
6.5.6.1.	.NET	41

6.5.6.2.	Principios SOLID.....	42
6.5.6.3.	JSON Web Token	43
6.5.6.4.	Inyecciones de Dependencia	44
6.5.6.4.1.	Tipos de DI.....	44
6.5.6.4.2.	Tiempos de vida de las DI.....	45
6.5.6.5.	Sistema Gestor de Base de datos	45
6.5.6.5.1.	Servidor SQL Lógico	46
6.5.6.5.2.	Azure SQL Database	46
6.5.6.6.	Entity Framework	47
6.5.6.6.1.	Enfoque DB First	47
Capítulo I: Análisis de requerimientos	48
1.1.	Perfil del negocio	48
1.2.	Descripción de los procesos de negocio	50
1.3.	Objetivos del sistema	53
1.4.	Alcance del sistema.....	53
1.5.	Beneficios del sistema	54
1.6.	Personal o actores involucrados.....	54
1.7.	Definición de usuarios.....	55
1.8.	Requerimientos de negocio.....	56
1.9.	Requerimientos funcionales	57
1.10.	Requerimientos no funcionales.....	57
1.11.	Requerimientos de implementación	58
1.12.	Requerimientos de usuario.....	58
1.13.	Plantillas REM	59
1.14.	Matriz de trazabilidad	61
Capítulo II: Diseño del sistema	62
2.1.	Diseño de la arquitectura del sistema.....	62
2.1.1.	Estilo de arquitectura en capas.....	62
2.2.	Patrón de diseño de la arquitectura	64
2.2.1.	Clean Architecture	64
2.3.	Modelos UWE	66
2.3.1.	Modelo de requerimientos	66
2.3.2.	Modelo de contenido	73

2.3.3.	Modelo de navegación	75
2.3.4.	Modelo de presentación.....	77
2.3.5.	Modelo de procesos	80
2.4.	Diagrama de estados	82
2.5.	Diagrama de componentes	84
2.6.	Diagrama de despliegue.....	85
2.7.	Plantillas de Coleman	86
2.8.	Diseño lógico y físico de la base de datos (3FN)	95
Capítulo III: Construcción del sistema	96
3.1.	Hardware utilizado para la construcción.....	96
3.2.	Software utilizado para la construcción	97
3.3.	Especificación de las clases del sistema.....	97
3.4.	Diseño y uso de APIs.....	100
3.5.	Implementación de patrones de diseño.....	101
3.6.	Organización de los archivos del proyecto.....	104
3.7.	Diseños de las interfaces del sistema.....	108
VII.	Conclusiones	111
VIII.	Recomendaciones	112
IX.	Bibliografía	113
X.	Anexos	117
	Anexo1: Árbol de problemas.....	117
	Anexo2: Árbol de objetivos.....	118
	Anexo3: Modelo de entrevista	119

Anexos de tablas	120
AnexoTabla1: Plantillas REM	120
Tabla 17: Plantilla REM RN-04	120
Tabla 18: Plantilla REM RN-05	120
Tabla 19: Plantilla REM RN-06	120
Tabla 20: Plantilla REM RN-07	121
Tabla 21: Plantilla REM RN-08	121
Tabla 22: Plantilla REM RN-09	121
Tabla 23: Plantilla REM RN-10	122
Tabla 24: Plantilla REM RF-01	122
Tabla 25: Plantilla REM RF-02	123
Tabla 26: Plantilla REM RF-03	123
Tabla 27: Plantilla REM RF-04	123
Tabla 28: Plantilla REM RF-05	124
Tabla 29: Plantilla REM RF-06	124
Tabla 30: Plantilla REM RF-07	124
Tabla 31: Plantilla REM RF-08	124
Tabla 32: Plantilla REM RF-09	125
Tabla 33: Plantilla REM RF-10	125
Tabla 34: Plantilla REM RF-11	125
Tabla 35: Plantilla REM RF-12	126
Tabla 36: Plantilla REM RF-13	126
Tabla 37: Plantilla REM RNF-01	126
Tabla 38: Plantilla REM RNF-02	127
Tabla 39: Plantilla REM RNF-03	127
Tabla 40: Plantilla REM RNF-04	127
Tabla 41: Plantilla REM RNF-05	128
Tabla 42: Plantilla REM RNF-06	128
Tabla 43: Plantilla REM RNF-07	128
Tabla 44: Plantilla REM RNF-08	129
Tabla 45: Plantilla REM RNF-09	129
Tabla 46: Plantilla REM RI-01	129
Tabla 47: Plantilla REM RI-02	130

Tabla 48: Plantilla REM RI-03.....	130
Tabla 49: Plantilla REM RI-04.....	130
Tabla 50: Plantilla REM RI-05.....	131
Tabla 51: Plantilla REM RI-06.....	131
Tabla 52: Plantilla REM RU-01	131
Tabla 53: Plantilla REM RU-02	132
AnexoTabla2: Plantillas de Coleman.....	133
Tabla 54: Plantilla Coleman RN-04.....	133
Tabla 55: Plantilla Coleman RN-05.....	135
Tabla 56: Plantilla Coleman RN-06.....	136
Tabla 57: Plantilla RN-07.....	137
Tabla 58: Plantilla RN-08.....	138
Tabla 59: Plantilla RN-09.....	139
Tabla 60: Plantilla RN-10.....	140
Tabla 61: Plantilla RF-01	142
Tabla 62: Plantilla RF-02	143
Tabla 63: Plantilla RF-03	144
Tabla 64: Plantilla RF-04	144
Tabla 65: Plantilla RF-05	145
Tabla 66: Plantilla RF-06	146
Tabla 67: Plantilla RF-07	146
Tabla 68: Plantilla RF-08	147
Tabla 69: Plantilla RF-09	148
Tabla 70: Plantilla RF-10	148
Tabla 71: Plantilla RF-11	149
Tabla 72: Plantilla RF-12	149
Tabla 73: Plantilla RF-13	150
Tabla 74: Plantilla RNF-01	151
Tabla 75: Plantilla RNF-02	152
Tabla 76: Plantilla RNF-03	152
Tabla 77: Plantilla RNF-04	152
Tabla 78: Plantilla RNF-05	152
Tabla 79: Plantilla RNF-06	153

Tabla 80: Plantilla RNF-07	153
Tabla 81: Plantilla RNF-08	153
Tabla 82: Plantilla Coleman RNF-09	154
Tabla 83: Plantilla Coleman RI-01.....	154
Tabla 84: Plantilla Coleman RI-02.....	154
Tabla 85: Plantilla Coleman RI-03.....	154
Tabla 86: Plantilla Coleman RI-04.....	155
Tabla 87: Plantilla Coleman RI-05.....	155
Tabla 88: Plantilla Coleman RI-06.....	155
Tabla 89: Plantilla Coleman RU-01.....	156
Tabla 90: Plantilla Coleman RU-02.....	156
AnexoTabla3: Software utilizado para la construcción.....	159
Tabla 91: Software – NodeJS	159
Tabla 92: Software – Package Manager	159
Tabla 93: Software – Angular	159
Tabla 94: Software – SQL Server Management Studio	159
Tabla 95: Software – WinSCP	159
Tabla 96: Software – Postman	159

Anexos de ilustraciones	160
Anexo I Ilustración 1: Modelo de requerimientos – Diagramas de casos de uso	160
Ilustración 43: RN-04	160
Ilustración 44: RN-05	161
Ilustración 45: RN-06	162
Ilustración 46: RN-07	163
Ilustración 47: RN-08	164
Ilustración 48: RN-09	165
Ilustración 49: RN-10	166
Ilustración 50: RF-01	167
Ilustración 51: RF-02	168
Ilustración 52: RF-03	169
Ilustración 53: RF-09	170
Ilustración 54: RF-12	171
Anexo II Ilustración 2: Modelo de requerimientos – Diagramas de actividades	172
Ilustración 55: RN-01 Actualizar referidos	172
Ilustración 56: RN-01 Eliminar referido	173
Ilustración 57: RN-02 Mostrar y filtrar contactos	174
Ilustración 58: RN-02 Agregar contacto	175
Ilustración 59: RN-02 Actualizar contacto	176
Ilustración 60: RN-02 Eliminar contacto	177
Ilustración 61: RN-03 Mostrar y filtrar motivos de declinación	178
Ilustración 62: RN-03 Agregar motivo de declinación	179
Ilustración 63: RN-03 Actualizar motivos de declinación	180
Ilustración 64: RN-03 Eliminar motivo de declinación	181
Ilustración 65: RN-04 Mostrar y filtrar recompensas	182
Ilustración 66: RN-04 Agregar recompensa	183
Ilustración 67: RN-04 Actualizar recompensa	184
Ilustración 68: RN-04 Eliminar recompensa	185
Ilustración 69: RF-01 Iniciar sesión	186
Ilustración 70: RF-02 Recuperar contraseña	187
Ilustración 71: RF-03 Cargar menu de navegación	188
Ilustración 72: RF-09 Cerrar sesión	189

Ilustración 73: RN-07 Actualizar estado de referidos	190
Ilustración 74: RN-06 Cambiar visibilidad de referidos.....	191
Ilustración 75: RN-08 Sincronizar referidos	192
Ilustración 76: RN-10 Procesar recompensas.....	193
Anexo lllustración3: Modelo de contenido	194
Ilustración 77: Gestión de referidos	194
Ilustración 78: Gestión de recompensas	195
Anexo lllustración4: Modelo de navegación	196
Ilustración 79: Diagrama de navegación - Inicio de sesión.....	196
Ilustración 80: Diagrama de navegación - Distribuidor.....	197
Ilustración 81: Diagrama de navegación - Administrador	198
Anexo lllustración5: Modelo de presentación	199
Ilustración 82: Presentación - Reportes	199
Ilustración 83: Presentación - Contactos.....	200
Ilustración 84: Presentación - Motivos de declinación.....	201
Ilustración 85: Presentación - Recompensas	202
Anexo lllustración6: Modelo de procesos	203
Ilustración 86: Diagrama de procesos - Inicio de sesión.....	203
Ilustración 87: Diagrama de procesos - Motivos de declinación	204
Ilustración 88: Diagrama de procesos - Recompensas.....	205
Anexo lllustración7: Especificación de las clases del sistema	206
Ilustración 89: Clases - Datafone	206
Ilustración 90: Clases - Datafone.Models.....	207
Ilustración 91: Clases - Datafone.Infrastructure	208
Ilustración 92: Clases - Datafone.Application	209
Anexo lllustración8: Diseño y uso de APIs	210
Ilustración 93: Swagger - Contactos	210
Ilustración 94: Swagger - Motivos de declinacion	210
Ilustración 95: Swagger - Trabajos calendarizados	210
Ilustración 96: Swagger - Navegacion	210
Ilustración 97: Swagger - Tipos de modelo de pagos y recargas	210
Ilustración 98: Swagger - Provincias.....	211
Ilustración 99: Swagger - Referidos	211

Ilustración 100: Swagger - Reportes.....	211
Ilustración 101: Swagger - Recompensas	211
Ilustración 102: Swagger - Usuarios	211
Anexo Ilustración9: Implementación patrones de diseño – Clean Architecture.....	212
Ilustración 103: Dependencia de proyectos - Datafone.Application	212
Ilustración 104: Dependencia de proyectos - Datafone.Domain.....	212
Ilustración 105: Dependencia de proyectos - Datafone.Infrastructure	212
Ilustración 106: Dependencia de proyectos - Datafone.Models.....	212
Ilustración 107: Dependencia de proyectos - Datafone.Persistance.....	213
Anexo Ilustración10: Organización de los archivos del proyecto – Back-End	213
Ilustración 108: Estructura de carpetas y archivos - Datafone.Domain	213
Ilustración 109: Estructura de carpetas y archivos - Datafone.Infrastructure.....	214
Ilustración 110: Estructura de carpetas y archivos - Datafone.Models I	215
Ilustración 111: Estructura de carpetas y archivos - Datafone.Models II.....	215
Ilustración 112: Estructura de carpetas y archivos - Datafone.Persistance I	216
Ilustración 113: Estructura de carpetas y archivos - Datafone.Persistance II.....	216
Anexo Ilustración11: Organización de los archivos del proyecto – Front-End.....	217
Ilustración 114: Estructura de carpetas y archivos - Aplicacion.layout	217
Ilustración 115: Estructura de carpetas y archivos – Aplicación.Modules&Shared....	218
Anexo Ilustración12: Diseños de las interfaces del sistema.....	219
Ilustración 117: Comparación modelo - Menu principal	220
Ilustración 118: Sistema web - Referidos.....	221
Ilustración 119: Sistema web - Agregar referidos	222
Ilustración 120: Sistema web - Actualizar referido	222
Ilustración 121: Sistema web - Eliminar referido	223
Ilustración 122: Comparación modelo - Referidos	223
Ilustración 123: Comparación modelo - Mostrar referidos	224
Ilustración 124: Comparación modelo - Agregar referidos	224
Ilustración 125: Comparación modelo - Actualizar referido	224
Ilustración 126: Comparación modelo - Error en API	225
Ilustración 127: Comparación modelo - Eliminar referido	225
Ilustración 128: Sistema web - Mostrar contactos	226
Ilustración 129: Sistema web - Agregar contactos.....	226

Ilustración 130: Sistema web - Actualizar contactos	227
Ilustración 131: Sistema web - Eliminar contacto.....	227
Ilustración 132: Comparación modelo - Contactos	228
Ilustración 133: Comparación modelo - Mostrar contactos.....	228
Ilustración 134: Comparación modelo - Agregar contacto y Error en campo.....	228
Ilustración 135: Comparación modelo - Eliminar contacto.....	229
Ilustración 136: Comparación modelo - Error en API	229
Ilustración 137: Sistema web - Mostrar motivos de declinación	230
Ilustración 138: Sistema web - Agregar motivo de declinación	230
Ilustración 139: Sistema web - Actualizar motivo de declinación	231
Ilustración 140: Sistema web - Eliminar motivo de declinación.....	231
Ilustración 141: Comparación modelo - Motivos de declinación	232
Ilustración 142: Comparación modelo - Mostrar motivos de declinación.....	232
Ilustración 143: Comparación modelo - Agregar motivo de declinación y Error en campo.....	232
Ilustración 144: Comparación modelo - Actualizar motivo de declinación.....	233
Ilustración 145: Comparación modelo - Eliminar motivo de declinación.....	233
Ilustración 146: Comparación modelo - Error en API	234
Ilustración 147: Sistema web - Mostrar recompensas.....	235
Ilustración 148: Sistema web - Agregar recompensa.....	235
Ilustración 149: Sistema web - Actualizar recompensa.....	236
Ilustración 150: Sistema web - Eliminar recompensa.....	236
Ilustración 151: Comparación modelo - Recompensas.....	237
Ilustración 152: Comparación modelo - Mostrar y filtrar recompensas	237
Ilustración 153: Comparación modelo - Agregar recompensas prepago.....	238
Ilustración 154: Comparación modelo - Agregar recompensa pospago	238
Ilustración 155: Comparación modelo - Actualizar recompensa.....	239
Ilustración 156: Comparación modelo - Eliminar recompensa.....	239
Ilustración 157: Comparación modelo - Error en API	239
Ilustración 158: Sistema web - Reportes I	240
Ilustración 159: Sistema web - Reportes II	241
Ilustración 160: Comparación modelo - Reporte.....	241
Ilustración 161: Comparación modelo - Reporte I.....	242

Ilustración 162: Comparación modelo - Reporte II	242
Ilustración 163: Comparación modelo - Reporte III	242
Ilustración 164: Comparación modelo - Reporte IV	243

Managua, 01 de septiembre de 2023

Msc. Claudia Lucía Benavidez Rugama

Decana

Facultad de Ciencias y Sistemas

Su despacho.

Estimada Ingeniera Benavidez:

Reciba cordiales saludos, nos dirigimos a usted en calidad de estudiantes de la carrera Ingeniería de Sistemas de la Facultad de Ciencias y Sistemas, con el objetivo de solicitar la programación de defensa de nuestro trabajo monográfico titulado **“SISTEMA DE INFORMACION WEB PARA LA GESTION DE FIDELIZACION DE CLIENTES”**.

Esperando una respuesta positiva a nuestra solicitud, agradecemos de antemano su tiempo y consideración.

Atentamente,

Cristopher Zepeda
2011-37038

Gloria Mayorga
2011-36978

Magaly Mayorga
2010-33172

Cc. Archivo personal.

Managua, Nicaragua
Viernes 01 de septiembre 2023

MSc. Ing. Claudia Lucía Benavidez Rugama
Decana
Facultad de Ciencias y Sistemas
Su despacho.

Estimada Ingeniera Benavidez:

Reciba cordiales saludos, por la presente me dirijo a usted con la finalidad de informarle que he revisado trabajo del taller monográfico titulado: “**SISTEMA DE INFORMACION WEB PARA LA GESTION DE FIDELIZACION DE CLIENTES**”, de los bachilleres:

- | | |
|--|-------------------|
| 1. Christopher Josue Zepeda Tellez | 2011-37038 |
| 2. Gloria Esther Mayorga Mena | 2011-36978 |
| 3. Magaly Magdalena Mayorga Maradiaga | 2010-33172 |

Por lo que solicito de su gestión para que el proceso de defensa de dicho trabajo monográfico sea programado y llevado a cabo.

Sin más a qué hacer referencia, me despido deseándole éxito en sus actividades administrativas.

Atentamente,

MSc. Danilo Humberto Noguera Rivera
Tutor
Docente Titular
Facultad de ciencias y sistemas

Cc. Archivo personal

Resumen

El objetivo principal de este proyecto consiste en el desarrollo de un sistema de información web orientado a optimizar la fidelización de clientes en la empresa Datafone Panamá. Para lograrlo, se llevó a cabo un exhaustivo análisis y diseño del sistema, empleando una variedad de herramientas y metodologías.

En la fase de análisis, se identificaron y definieron de manera precisa los requerimientos funcionales y no funcionales del sistema, así como aquellos relacionados con aspectos empresariales y de usuario. Estos requerimientos se documentaron siguiendo las plantillas REM, garantizando así su adecuada especificación.

En el proceso de diseño, se adoptó la metodología UWE, compuesta por cinco modelos esenciales: requerimientos, contenido, navegación, presentación y procesos. Para representar visualmente estos modelos, se empleó la herramienta MagicDraw junto con el complemento MagicUWE.

De manera complementaria, se eligieron y aplicaron los patrones de diseño Clean Architecture y CQRS, incorporando una arquitectura en capas que otorga cohesión y estructura al sistema. Este enfoque estratégico permitió una separación clara de responsabilidades y una organización eficiente de los componentes del sistema.

En la implementación del sistema web, se optó por las tecnologías más actuales, incluyendo .NET 7, Angular V15 y SQL. Estas elecciones tecnológicas proporcionaron una base sólida y moderna para el desarrollo del sistema, asegurando su rendimiento, escalabilidad y mantenibilidad.

La minuciosa definición de requerimientos en la etapa de análisis, los modelos detallados creados mediante UWE en la fase de diseño, así como la selección estratégica de patrones de diseño y arquitectura, fueron factores clave para agilizar el proceso de desarrollo y garantizar su éxito. En conjunto, estos elementos contribuyeron a optimizar los tiempos de ejecución y a obtener un sistema integral y eficiente para la gestión de la fidelización de clientes en Datafone Panamá.

I. Introducción

La presente investigación se refiere a un novedoso sistema de información web para la gestión de fidelización de clientes en la empresa Datafone Panamá, el cual se puede concretar como una estrategia a implementar por dicha organización con el propósito de premiar a los clientes que recomiendan personas a la empresa, esto estimulará que las compras se realicen de forma periódica y permitirá que los clientes existentes elijan Datafone como su compañía telefónica en lugar de un competidor que ofrezca beneficios o servicios similares.

Datafone es el competidor más pequeño del sector de las telecomunicaciones en la república de Panamá, el bajo volumen de clientes nuevos es debido al reducido seguimiento hacia los potenciales clientes esto resulta en un bajo nivel de facturación, adicionalmente la poca capacitación hacia los colaboradores de los distribuidores asociados ha impactado de forma negativa la imagen de la empresa en el mercado.

Por estas razones el sistema de gestión de fidelización de clientes está diseñado a la medida, lo cual permite tener un excelente control y adecuado seguimiento a los clientes nuevos o prospectos, este está basado en un programa de referidos que consiste en un esquema de recompensa; La fidelización de sus clientes fortalecerá la competitividad frente a la competencia e incrementará las utilidades, así como también mejorará la experiencia de cada uno de sus prospectos.

La implementación del sistema de información contribuirá a brindar un servicio de calidad, permitirá a los colaboradores tener un amplio conocimiento del estado de cada prospecto, mejorará la atención al cliente y reducirá las debilidades que se tienen actualmente.

Durante la construcción del sistema de información web, se realizó el análisis de los requerimientos funcionales y no funcionales, con los cuales se diseñó el sistema con la metodología UWE, utilizando las siguientes tecnologías: SPA con .NET 7, Angular v15 y SQL.

II. Antecedentes

Las telecomunicaciones en la república de Panamá forman parte de los servicios públicos, y como tal, se encuentran reguladas por la ley No. 31 (8 febrero de 1996), la cual tiene como objetivo el acelerar la modernización y desarrollo del sector, promover la inversión privada en el mercado, extender su acceso, mejorar la calidad de servicios provistos, promover tarifas bajas al usuario y la competencia leal.

El mercado de las telecomunicaciones se encuentra cubierto por las empresas Milicom, Cable & Wireless, Claro y Datafone; cada una de estas utiliza tecnologías y espectros diferentes Milicom (LTE 700 MHz GSM - UMS/HSPA 850 MHz 1900 MHz), Cable & Wireless (LTE 700 MHz GSM - UMS/HSPA 850 MHz 1900 MHz), Claro (LTE 700 MHz GSM - UMS/HSPA - LTE 1900 MHz) y Datafone (GSM - UMS/HSPA 1900 MHz). En el año 2019 se generó en ventas pospago 929,864 dólares estadounidenses y en ventas prepago 4,699,141 dólares estadounidenses

Datafone realizó su lanzamiento en el año 1998, desde entonces se ha expandido rápidamente para abarcar mercados en todo el Caribe, Centroamérica y el Pacífico. El grupo Datafone, ofrece servicios a consumidores finales y clientes empresariales por igual, ha invertido más de 3,000 millones de dólares en las operaciones a nivel global y continuarán invirtiendo para llevar las últimas tecnologías a los mercados en que se encuentra.

Datafone entró al mercado panameño en el año 2007, adquiriendo la licencia de operación por un monto superior a 75 millones de dólares estadounidenses, desde entonces ha invertido más de 500 millones de dólares estadounidenses en el sector. Actualmente, la empresa posee aproximadamente el 10% del mercado y emplea a 210 colaboradores.

Para el año 2015, el mercado de las telecomunicaciones había cambiado, no solo por la competencia en el mercado, sino por los consumidores, estos cambiaron y su prioridad fue el acceso móvil a servicios de internet. La compañía no solo optó por concentrarse en ese nicho de mercado específico sino también en centrarse en un consumidor que aspira a un servicio cada vez más personalizado.

Para obtener una mayor participación en el mercado, la compañía tomó la decisión de diversificar las líneas de ingreso, inicialmente la compañía era 100% móviles orientada a tecnologías para clientes móviles, al año 2015 se redujo esto al 87% y el objetivo es llevarlo al 70%, dado que empezaron a desarrollar la línea de negocio que tiene que ver con soluciones integradas de conectividad para clientes empresariales.

Adicional a la mejora tecnológica que la compañía ha realizado, Datafone Panamá ha implementado estrategias de marketing tradicionales para la obtención de nuevos clientes, estas estrategias incluyen promociones sobre las recargas realizadas por los clientes, realizar paquetes de internet móvil, mensajería y redes sociales a bajo costo, así como también ha implementado estrategias de fidelización basadas en puntos ganados por las compras realizadas. Las diversas estrategias mencionadas anteriormente se encuentran orientadas al mercado de clientes prepago.

Para el mercado de clientes pospago y clientes empresariales, la empresa ha optado por reducir el costo de los diversos planes, renovar los dispositivos móviles después de un determinado tiempo, ofrecer los diversos equipos móviles en cuotas sin intereses.

Durante la investigación de los antecedentes para este proyecto se procedió a la búsqueda en el repositorio en línea del CNU, en el cual el tema principal está basado en la implementación de un sistema de información para la fidelización de clientes, obteniendo los siguientes resultados:

No se ha identificado un estudio similar a la temática de este proyecto, sin embargo, se encontraron investigaciones relacionadas en tesis de grado, que se enfocan en fidelización de clientes, aumento de las ventas e implementación de sistemas de información orientados a la gestión de los clientes.

Del 2015, tesis de seminario de graduación para optar al título de Ingeniero en Sistemas de Información, elaborado por Katherine Massiel González Guzmán, Alfonso Rubén Mendoza González y Yaritza de los Ángeles Barberena Martínez,

UNAN – FAREM – Chontales, en el que realizaron el proyecto de “Implementación de un prototipo de Sistemas en gestión de relaciones con los clientes para Pyme Nacir Móvil, en la ciudad de Juigalpa Chontales durante el primer semestre del 2014”, se tiene como objetivo Implementar un prototipo de sistemas en gestión de relaciones con los clientes para la Pyme Nacir Móvil, en la ciudad de Juigalpa Chontales durante el primer semestre del 2014, este tipo de proyecto es cualitativo porque realizan diferentes estudios para brindar un mejor servicio al cliente. (González Guzmán, Mendoza González, & Barberena Martínez, 2015).

Este proyecto está relacionado con el tema de investigación, ya que ambos tienen como objetivo incrementar el volumen de clientes y obtener la fidelidad de estos, con ello se pretende brindar solución a las necesidades primordiales, ofreciéndoles un servicio de calidad y una atención altamente calificada. Este proyecto servirá como guía en la etapa de diseño para la realización del sistema web de fidelización de clientes.

Del 2020, tesis de seminario de graduación para optar al título de Licenciado en Mercadotecnia, elaborado por Ana Yanci Martínez Briones, Fanny Janeth Centeno Medina y Gabriela Tatiana Martínez Úbeda, UNAN – FAREM – Estelí, se realizó un estudio de “Análisis de la fidelización de los clientes para el incremento de ventas de la empresa Agua Roca en el II semestre del año 2020 en la ciudad de Estelí”, donde se tiene como objetivo Analizar la fidelización de clientes para el incremento de las ventas de la empresa Agua Roca en el II semestre del año 2020 en la ciudad de Estelí, este tipo de estudio es cualitativo porque proveen estrategias para la atracción y fidelización de clientes. (Martínez Briones, Centeno Medina, & Martínez Úbeda, 2020).

Este estudio está relacionado con el proyecto, ya que ambas investigaciones están enfocadas en fidelizar clientes y aumentar el volumen de ventas. Por medio de estrategias de marketing, se realiza un estudio para tener una mayor captación de clientes, con una mejor experiencia, tanto para colaboradores como clientes, dando mejores resultados a la empresa y aumentar el volumen de clientes. El estudio

aportará el concepto de distribuidor y servirá como base para el modelo de entrevistas en la etapa de análisis.

Del 2016, tesis de seminario de graduación para optar al título de Licenciado en Mercadotecnia, elaborado por Maykeling Gabriela Martínez Gutiérrez y María Félix Pérez Sánchez, UNAN – Managua, se realizó un estudio de “Instrumentos de mercadeo de servicio para lograr la captación de nuevos clientes proporcionarles valor, satisfacción, retención y fidelización”, donde se tiene como objetivo Explicar las generalidades del marketing de servicios para lograr comprender la importancia de generar el valor y satisfacción necesaria de los clientes. (Martínez Gutiérrez & Pérez Sánchez, 2016).

Este tema de investigación aporta en el área de Marketing, ya que a través de este se da a conocer a los clientes el nuevo mecanismo de recompensas que Datafone ofrece, ayuda a que se mantengan informados sobre los beneficios y ayuda a la fidelización hacia la compañía. Esencialmente esta investigación brindará los conceptos de recompensa y fidelización.

III. Planteamiento del problema

En la república de Panamá, existen cuatro competidores en el sector de telecomunicaciones Milicom, Cable & Wireless, Claro y Datafone; de los cuales Datafone es el competidor más pequeño del mercado (bnamericas, 2022).

La empresa Datafone posee el porcentaje de clientes activos en el mercado más pequeño, esto se puede ver reflejado en el bajo volumen de ventas y adquisiciones de servicios por parte de clientes nuevos, han intentado implementar estrategias para mejorar la situación y tener una mejor posición en el mercado, pero sin resultado alguno, por lo que se han identificado factores o debilidades a mejorar como lo son: poco seguimiento los clientes potenciales, poca capacitación hacia colaboradores que pertenecen a distribuidores asociados y reducida motivación que tiene los empleados para ofrecer las líneas de la compañía.

Cada uno de los factores antes mencionados generan un efecto adverso dentro de la compañía como lo son: el bajo rendimiento en la facturación mensual de la empresa, y el porcentaje de mercado que posee la empresa; los factores relacionados a la capacitación y motivación de los colaboradores han estado impactando de forma negativa a la imagen y presencia de la compañía frente a la competencia en el sector de las telecomunicaciones; así mismo el reducido seguimiento por parte de los colaboradores ha generado inconformidad en los potenciales clientes.

Por lo anterior, se pretende desarrollar un sistema de información web para la gestión de fidelización de clientes para la empresa Datafone Panamá, en la que de forma continua buscan soluciones para la atracción de nuevos clientes.

Ver árbol de problemas en [Anexo1: Árbol de problemas](#)

IV. Justificación

El Sistema de información web para la gestión de fidelización de clientes será diseñado a la medida para la empresa Datafone Panamá, el cual ayudará a tener un mejor control y seguimiento a los clientes nuevos o prospectos; El sistema estará basado en un programa de referidos, el que consiste en un esquema de recompensas orientado a la cantidad de prospectos que se unan a la compañía y realicen compras.

Fidelizar clientes fortalecerá la competitividad de la empresa frente a la competencia, incrementar las utilidades, así como, mejorar la experiencia de los prospectos; Será una herramienta para el uso de los colaboradores internos y externos que fomentará el adecuado seguimiento de los nuevos clientes. De esta manera la empresa tendrá un mayor alcance de clientes ampliando su cobertura en el mercado.

La implementación del sistema de información contribuirá a brindar un servicio de calidad, permitirá a los colaboradores tener un amplio conocimiento del estado de cada prospecto, mejorar la atención al cliente y reducir las debilidades que se tienen actualmente.

Ver árbol de objetivos en [Anexo2: Árbol de objetivos](#)

V. Objetivos

General

Desarrollar un sistema de información web para la gestión de fidelización de clientes en la empresa Datafone Panamá

Específicos

- Realizar análisis de los requerimientos funcionales y no funcionales del sistema de información.
- Diseñar el sistema de información en base a los requerimientos utilizando la metodología UWE.
- Construir el sistema de información basado en los requerimientos y diseño establecidos, utilizando un SPA con .NET 7, Angular v15 y SQL.

VI. Marco teórico

Los sistemas de información web son los que ayudan a la buena administración y manejo de datos de una empresa, datos que facilitan la toma de decisiones y que pueden brindar proyecciones futuras para mejorar los procesos ya existentes dentro de la empresa, por otro lado la fidelización de clientes aporta garantía que los clientes existentes estén satisfechos con los servicios brindados y a tener una mayor captación de clientes nuevos, posicionando a la empresa dentro del mercado y dando frente a la competencia.

6.1. Sistema

Un sistema se encuentra compuesto por varios componentes que se encuentran frecuentemente comunicados para lograr un objetivo en común. Según (Oz, 2008):

Un sistema es una matriz de componentes que colaboran para alcanzar una meta común o varias, al aceptar entradas, procesarlas y producir salidas de una manera organizada.

No todos los sistemas tienen una sola meta. A menudo, un sistema está formado por varios subsistemas (componentes de uno mayor) con metas secundarias, todas las cuales contribuyen a alcanzar la meta principal. Los subsistemas pueden recibir entradas y transferir salidas a y de otros sistemas o subsistemas. (p 9)

Se podría decir que una organización es un sistema, ya que incluye diferentes subsistemas o departamentos como lo son administración, tecnología, contabilidad, recursos humanos etc. Todos estos, trabajan en conjunto para crear utilidades tangibles y no tangibles. Estos sistemas dependen de una entidad abstracta conocida como Sistema de Información.

6.1.1. Sistema de información

Un sistema de información es la forma en que los diferentes subsistemas comparten información entre ellos mismos. Según (Oz, 2008):

Un sistema de información está formado por todos los componentes que colaboran para procesar los datos y producir información. Casi todos los sistemas de información empresariales están integrados por muchos subsistemas con metas

secundarias, todas las cuales contribuyen a la meta principal de la organización. (p 11)

Para las organizaciones, los sistemas de información son esenciales para su capacidad de competir u obtener una ventaja competitiva. Dado que se han liberado al mercado nuevas tecnologías estas nos permiten tener acceso a los sistemas desde la web.

6.1.1.1. Sistema de información Web

Los sistemas de información basados en la web, son sistemas basados en tecnologías y estándares web, orientados al contenido y se requiere un buen diseño; Estos poseen características que los hacen únicos como: datos de usuarios demográficos variables, dispositivos y navegadores, cambian constantemente y son multidisciplinarios. Según (LEE, 2019):

Una aplicación web es un sistema de software basado en tecnologías y estándares de World Wide Web Consortium (W3C) (Kappel et al. 2004). En muchas maneras los sistemas web son similares a los sistemas tradicionales, sin embargo, el ambiente único con muchas características en el que estos se hospedan lo separa de un sistema tradicional. A diferencia de los sistemas tradicionales, los sistemas web son orientados al contenido y son altamente visuales.

Datos de usuarios demográficos variables: Los usuarios de una aplicación web son ampliamente variables y diversos, esto es debido a que el internet es abierto y accedido a nivel mundial. Cuando un sitio está disponible en la web, es accesible para el mundo entero de usuarios de internet. Las condiciones demográficas de los usuarios son altamente impredecibles, así como, el número de visitantes. Esto hace que los sistemas sean construidos de forma óptima para el uso fácil, diversidad y escalabilidad.

Dispositivos y navegadores: Los desarrolladores hoy en día deben considerar el esquema visual y estructural de las aplicaciones, para acomodar las páginas web en diversas resoluciones (laptops, tablets, celulares, etc). Del mismo modo el ancho de banda y las capacidades de estos dispositivos pueden variar significativamente.

Multidisciplinarios: Kappel describe la ingeniería web como un campo multidisciplinario (Kappel et al. 2004). Los tipos de experiencia requerida para una

aplicación web son numerosos, se podría mencionar los siguientes roles publicistas, programadores, diseñadores, expertos en tecnologías front-end, expertos en tecnología back-end y otros. Entender un sistema web es considerablemente más complejo comparado con otros sistemas.

Cambian constantemente: *Tanto la tecnología empleada para el desarrollo de un sistema web y los sistemas son caracterizados como elementos que cambian constantemente. El desarrollo ocurrirá frecuentemente e incrementalmente bajo los siguientes escenarios: defectos en el sistema reportados por los usuarios, cambios en la tecnología, modificación o creación de características. (pp 258 - 260)*

6.2. Terminología del negocio

Es un listado de términos que se utilizan en el negocio, este listado incluye las definiciones. Esto se realiza con el objetivo de que las diferentes áreas involucradas utilicen los mismos conceptos.

6.2.1. Distribuidor

Los distribuidores se encargan de la reventa de un producto o servicio y no en la fabricación del mismo. Según (Martínez Briones, Centeno Medina, & Martínez Úbeda, 2020):

Los distribuidores mayoristas se encargan de vender bienes y servicios dirigidos a la reventa y entre sus actividades no incluyen al fabricante ni a los agricultores ya que estos se encargan exclusivamente de la producción, tampoco incluyen a minoristas. (p 19)

Los distribuidores son los encargados de que el producto llegue al cliente final en excelentes condiciones y prestar un servicio de calidad para mantener la fidelización del cliente.

6.2.2. Programa de referidos

Un programa de referidos es una estrategia para la adquisición y fidelización de clientes cuyo objetivo es que los clientes actuales recomiendan los servicios de la compañía a las personas allegadas. Según (Santos, 2021):

Un programa de referidos es una estrategia de adquisición y de fidelización de clientes que busca que tus clientes actuales recomiendan tus servicios y productos

a sus personas allegadas. Para ello debes definir una serie de estímulos tanto para tu cliente actual, como para tu cliente nuevo.

Emplear un sistema de referidos tiene dos ventajas para hacer crecer el negocio. La primera, es que los clientes adquiridos por esta vía tienen un costo significativamente menor frente a otras fuentes de adquisición. La segunda, que tienen un Lifetime Value, entre un 16% y 37% superior al resto de tus clientes.

Existen diferentes modelos de programas de referidos, entre ellos tenemos el modelo piramidal.

6.2.2.1. Modelo Piramidal

Este modelo se enfoca en recompensar al cliente que refiere. Según (Santos, 2021):

Los esquemas piramidales, también conocidos como redes de mercadeo, son esquemas utilizados por compañías cuya fuente principal de crecimiento es justamente su modelo de referidos.

Este modelo se enfoca únicamente en darle beneficios al cliente que refiere y no al referido. Adicionalmente, los esquemas piramidales mejoran sus incentivos a medida que el cliente refiere a más personas. Y de esta manera, le dan un estatus dentro de su comunidad de clientes.

6.2.3. Recompensa

Las recompensas son una estrategia de marketing que las empresas usualmente emplean para realizar la atracción de clientes. Según (Martínez Gutiérrez & Pérez Sánchez, 2016):

Regalar un 20% más de producto ya no es suficiente para competir en un mercado tan agresivo. Hay que ofrecer producto tangible, como la renovación móvil que utilizan habitualmente las empresas de telecomunicaciones para robarse clientes. Claro que estos regalos hay que rentabilizarlos buscando relaciones a largo plazo o potenciando la venta cruzada de productos. (p 47)

Las recompensas y el regalar producto, emplean el mismo concepto sobre brindar a los clientes un determinado servicio o producto con el objetivo de premiarlos y de esta manera obtener lo que se conoce como fidelización.

6.2.4. Fidelización

La fidelización de clientes en una empresa representa la lealtad que ellos poseen hacia un producto o servicio. Según (Martínez Gutiérrez & Pérez Sánchez, 2016):

La fidelización es la retención de los clientes actuales de una empresa. La retención expresa lealtad de estos mediante la repetición de sus compras para satisfacer sus necesidades y deseos. La fidelización es la medida de la vinculación de los clientes.
(p 43)

Para posicionar una empresa en el mercado haciéndole frente a la competencia en el mercado, esta debe de contar con ciertas estrategias para identificar y motivar a sus clientes ideales, a lo que llamamos fidelización.

6.3. Requerimientos de Software

Los requerimientos son la parte más importante al momento de construir un software, si estos no se encuentran bien definidos, el resultado final no resolverá las necesidades de la empresa y/o usuarios; los requerimientos ayudan a establecer una dirección para el proyecto. Según (Stephens, 2015):

Los requerimientos son las características que el software debe proveer. Al comienzo de un proyecto, se realiza la recopilación de requerimientos con el objetivo de identificar qué es lo que debemos construir, estos son definidos por los usuarios/clientes. A lo largo del proceso de desarrollo, se usan los requerimientos para guiar el desarrollo y asegurar que se está construyendo en la dirección correcta. Al final del proyecto, los requerimientos son utilizados para verificar que el producto final cumple con lo solicitado por los clientes. (P 54)

Los requerimientos deben ser claros, concisos, consistentes, verificables, fácil de entender y estos no deben ser ambiguos; En dependencia del alcance del proyecto y la complejidad del mismo, se pueden definir unos cuantos o cientos de requerimientos, el número y tipo de requerimientos también depende del nivel de detalle que el usuario/cliente solicite.

6.3.1. Tipos de requerimientos

Es posible categorizar los requerimientos en muchas maneras, estos pueden ser basados en la audiencia o focalizados en los diferentes aspectos que la aplicación

debe cumplir, a continuación, se definirán los conceptos de los cuatro tipos de requerimientos que se utilizarán.

6.3.1.1. Requerimientos de negocio

Los requerimientos de negocio, están orientados a los objetivos del proyecto. Según (Stephens, 2015):

Los requerimientos de negocio establecen los objetivos a alto nivel del proyecto, estos explican lo que un usuario/cliente espera que el software realice. La palabra clave en este concepto es “espera”, a veces los clientes agregan sus esperanzas y deseos en los requerimientos de negocio aparte de los objetivos verificables. Por ejemplo, un cliente podría establecer como requerimiento de negocio “Aumentar las utilidades en un 25%” o “Incrementar la demanda y obtener 10,000 nuevos clientes” a pesar de que estos requerimientos son cuantificables, estos probablemente están fuera del alcance de lo que un proyecto de software podría cumplir por sí mismo. (p 61)

6.3.1.2. Requerimientos funcionales

Los requerimientos funcionales se refieren a lo que el sistema debe ejecutar. Según (Sommerville, 2011):

Son enunciados acerca de servicios que el sistema debe proveer, de cómo debería de reaccionar el sistema a entradas particulares y de cómo debería comportarse el sistema en situaciones específicas. En algunos casos, los requerimientos funcionales también explican lo que no debe hacer el sistema. (pp 84 - 85)

6.3.1.3. Requerimientos no funcionales

Los requerimientos no funcionales, son requerimientos que no se relacionan directamente con los servicios específicos que el sistema entrega a sus usuarios. Según (Sommerville, 2011):

Son limitaciones sobre servicios o funciones que ofrece el sistema. Incluyen restricciones tanto de temporización y del proceso de desarrollo, como impuestas por los estándares. Los requerimientos no funcionales se suelen aplicar al sistema como un todo, más que a características o a servicios individuales del sistema. (p 85)

Los requerimientos no funcionales no poseen una definición descriptiva, sin embargo, los podemos categorizar en atributos de calidad, diseño, condiciones de implementación, propiedad de comportamiento del sistema importantes, entre otros. Primordialmente los requerimientos no funcionales son usados en la fase de diseño. Según (DENIS, WIXON, & ROTH, 2012):

A pesar que el término “No Funcional” no es muy descriptivo, las categorías de este requerimiento incluyen importantes propiedades de comportamiento que el sistema debe tener como rendimiento y usabilidad. La habilidad para acceder al sistema a través de un dispositivo móvil. Los requerimientos no funcionales son principalmente usados en la etapa de diseño cuando las decisiones acerca de la interfaz de usuario, hardware, software y arquitectura son tomadas. (p 107)

Tabla 1: Requerimientos no funcionales

Categoría	Descripción	Ejemplos
Operaciones	<i>Los ambientes físicos y técnicos en los que sistema operará.</i>	<ul style="list-style-type: none"> <i>El sistema puede usarse en dispositivos móviles</i> <i>El sistema se puede integrar con algún otro existente</i>
Rendimiento	<i>Velocidad, capacidad y fiabilidad del sistema.</i>	<ul style="list-style-type: none"> <i>Cualquier interacción entre el usuario y la app no debe exceder los 2 segundos.</i> <i>El sistema debe estar disponible 24 horas por día y 365 días al año</i>
Seguridad	<i>Quien tiene acceso al sistema y bajo qué circunstancias</i>	<ul style="list-style-type: none"> <i>Únicamente los gerentes pueden ver la información de los empleados</i> <i>El sistema posee tecnología que lo protegerá de virus, gusanos, troyanos etc.</i>
Cultura y política	<i>Factores políticos, culturales y/o legales que afectan el sistema</i>	<ul style="list-style-type: none"> <i>El sistema debe manejar diferentes monedas USD NIO</i> <i>La política de la compañía es utilizar tecnología en la nube de Azure</i>

Fuente: (DENIS, WIXON, & ROTH, 2012, pág. 108)

6.3.1.4. Requerimientos de implementación

Los requerimientos de implementación son aquellos que se utilizarán al finalizar el proyecto, y que con el transcurso del tiempo desaparecerán. Según (Stephens, 2015):

Los requerimientos de implementación son características temporales que son necesarias para la transición del nuevo sistema, pero que en el futuro estos requerimientos serán eliminados. Por ejemplo, supongamos que diseñamos un sistema para el rastreo de facturas para reemplazar un sistema existente. Posterior a la etapa de pruebas y despliegue, se necesitará un método para hacer el traslado de información de una base de datos a otra, esto es un requerimiento de implementación. Otros requerimientos de implementación podrían ser contratar nuevo personal, comprar nuevo hardware, preparar material de entrenamiento, y realizar la capacitación a los usuarios del nuevo sistema. (p 63)

6.3.1.5. Requerimientos de usuario

Los requerimientos de usuario definen cómo los usuarios finales utilizarán el sistema. Según (Stephens, 2015):

Los requerimientos de usuario (también son conocidos como requerimientos de las partes interesadas) describen como el proyecto será usado por los eventuales usuarios finales. Usualmente incluyen bocetos de formularios, instrucciones de como los usuarios deben utilizar el sistema para realizar una tarea específica, casos de uso y prototipos. A veces, estos requerimientos son muy detallados, indicando exactamente lo que el sistema debe hacer bajo diferentes circunstancias, en otras ocasiones especifican que necesita el usuario para completar una tarea, pero no necesariamente como el sistema debe ejecutarlo. (p 62)

6.3.2. Levantamiento de requerimientos

El levantamiento de requerimientos emplea diversos elementos como solución de problemas, elaboración, negociación y especificación. Se debe promover un ambiente colaborativo para identificar el problema, proponer elementos de solución, y negocia diferentes enfoques. Según (Pressman & Maxim, 2020):

El levantamiento de requerimientos combina elementos de solución de problemas, elaboración, negociación y especificación. Para fomentar un ambiente colaborativo

y un equipo orientado al enfoque de levantamiento de requerimientos, los involucrados deben trabajar juntos para identificar el problema. Proponiendo elementos de la solución, negociar diferentes enfoques, especificar un conjunto de soluciones preliminares para la solución del requerimiento.

Levantamiento de requerimientos de forma colaborativa

Existen diferentes enfoques para conseguir el levantamiento de requerimientos de forma colaborativa, cada enfoque es utilizado para diferentes escenarios, pero todos aplican alguna variación de las siguientes pautas:

- *Reuniones (presencial o virtual) son dirigidas y atendidas por ambos equipos los involucrados del proyecto y los ingenieros de software.*
- *Reglas para la preparación y la participación son establecidas.*
- *Una agenda lo suficientemente formal como para cubrir todos los puntos importantes para fomentar la lluvia de ideas.*
- *Un “Facilitador”, este puede ser el cliente, un desarrollador o algún otro involucrado del proyecto que dirija las reuniones.*

Escenarios de uso

Como parte del levantamiento de requerimientos necesitamos una visión de las funciones y características que el sistema estará realizando.

Obtención de los productos de trabajo

Los productos de trabajo producidos durante la obtención podrían variar en dependencia del tamaño del sistema o producto a ser construido. Para sistemas grandes, el resultado debería incluir lo siguiente:

- *Una declaración de necesidad y factibilidad.*
- *Una declaración limitada del alcance del sistema o producto.*
- *Una lista de clientes, usuarios, e involucrados que participaron en la obtención de requerimientos.*
- *Una descripción técnica del entorno técnico del sistema.*
- *Una lista de requerimientos (preferiblemente organizada por funciones) y las restricciones del dominio aplicables a cada uno.*

- *Un conjunto de escenarios que permitan profundizar en el uso del sistema bajo diferentes condiciones operativas.*

Cada uno de los productos deberá ser revisado por todas las personas que participaron en el levantamiento de requerimientos. (pp 111 - 115)

Una vez finalizada la etapa de levantamiento de requerimientos, el resultado o producto final es el que se utilizará para realizar el análisis de cada requerimiento.

6.3.3. Análisis de requerimiento

El proceso de análisis de requerimientos nos permitirá proporcionar de forma iterativa una descripción de la información, funciones y comportamiento del sistema.

Según (Pressman & Maxim, 2020):

La intención del modelo de análisis es proveer una descripción de los dominios de la información, funciones y comportamientos de un sistema de computadoras. A medida que se conoce más información acerca del sistema, el modelo irá cambiando dinámicamente, así como, las personas involucradas en el proyecto entenderán más acerca de lo que necesitan. Por esto es que el análisis es una imagen de los requerimientos en un momento dado, la expectativa es que este cambie.

A medida que el análisis evoluciona algunos elementos se estabilizarán, generando una base sólida para la etapa de diseño. Es posible que otros elementos se hagan más volátiles, indicando que el equipo involucrado no está completamente claro acerca de los requerimientos del sistema. (p 119)

A lo largo del análisis, se debe prestar atención a lo que debemos hacer y no el cómo.

- *Las interacciones de los usuarios.*
- *Que objetos el sistema manipula.*
- *Que funciones el sistema ejecuta.*
- *Que comportamientos el sistema ejecuta.*
- *Que comportamientos el sistema exhibe.*
- *Que interfaces están definidas.*

El cliente/usuario probablemente no está seguro acerca de que es requerido en ciertos aspectos del sistema, el desarrollador probablemente no está seguro acerca de que enfoque debe tomar para completar las funciones y actuación, estas realidades se mitigan a favor de un enfoque iterativo para el análisis y modelado del sistema. El analista debe modelar lo que es conocido y utilizar este modelo como base para la etapa de diseño.

Se deben considerar varias reglas generales al momento de generar el análisis:

- *Enfocarse en el problema o requerimiento de negocio manteniendo un alto nivel de abstracción.*
- *Reconocer que el análisis de requerimiento debe proveer el dominio de la información, la función y el comportamiento del software.*
- *Retrasar una considerable parte de la arquitectura de software y requerimientos no funcionales hasta la etapa de modelado.*
- *Es importante tener conocimiento de las formas en que algunos elementos están interconectados con otros elementos. (p 129)*

El análisis de requerimientos generará las características operativas del sistema, establece las restricciones que debe cumplir e indica la interfaz del software con otros elementos del sistema. Todos estos elementos serán utilizados en el modelado de requerimientos.

6.3.4. Modelado

Una vez finalizada la etapa de levantamiento de requerimientos, se procede con la elaboración del modelo de requerimientos, el cual podría dar como resultado uno o más tipos de modelo (modelos basados en el escenario, modelos de datos, modelos orientados a clases, modelos orientados al flujo, modelos de comportamiento).

Según (Pressman R. S., 2010):

El análisis de requerimientos permite al profesional (sin importar si se llama ingeniero de software, analista o modelista) construir sobre los requerimientos básicos establecidos durante las tareas de concepción, indagación y negociación, que son parte de la ingeniería de requerimientos.

La acción de modelar los requerimientos da como resultado uno o más de los siguientes tipos de modelo:

- *Modelos basados en el escenario de los requerimientos desde el punto de vista de distintos “actores” del sistema.*
- *Modelos de datos, que ilustran el dominio de información del problema.*
- *Modelos orientados a clases, que representan clases orientadas a objetos (atributos y operaciones) y la manera en la que las clases colaboran para cumplir con los requerimientos del sistema.*
- *Modelos orientados al flujo, que representan los elementos funcionales del sistema y la manera como transforman los datos a medida que se avanza a través del sistema.*
- *Modelos de comportamiento, que ilustran el modo en el que se comparte el software como consecuencia de “eventos” externos. (p 127)*

Cada tipo de modelo es utilizado para escenarios específicos, por ejemplo, el modelo basado en datos es más especializado y/o apropiado cuando se crea una aplicación o se manipula un espacio complejo de información; y el modelo basado en escenarios que es la técnica que se ha venido popularizando entre la comunidad de la ingeniería de software.

6.3.4.1. Modelo basado en escenarios

El éxito de un sistema se puede medir de muchas maneras, sin embargo, la satisfacción del usuario ocupa el primer lugar en la lista, el modelo basado en escenarios está compuesto de tres etapas: creación de un caso preliminar de uso, Mejora de un caso de uso preliminar y Escritura de un caso de uso formal. Según (Pressman R. S., 2010):

El modelo de requerimientos con UML, empieza con la creación de escenarios en forma de casos de uso, diagrama de actividades y diagramas de tipo carril de natación.

Escritura de un caso de uso formal

Para el modelado de requerimientos es suficiente con los casos de uso informales, sin embargo, cuando un caso de uso involucra una actividad crítica o cuando describe un conjunto complejo de etapas con un número significativo de excepciones, es deseable un enfoque más formal.

Los casos de uso formales deben contener las precondiciones estas son las condiciones que se deben cumplir antes de iniciar el caso de uso; El disparador o trigger este es el evento o condición que activa el caso de uso; Los escenarios enumeran las acciones específicas que requiere el actor y las respuestas apropiadas del sistema; Las excepciones identificas las situaciones detectadas cuando se mejora el caso de uso preliminar.

Pueden incluirse o no encabezados adicionales y se deben explicar por sí mismos en forma razonable. (pp 131 - 136)

6.3.5. Tabulación (Plantillas Coleman)

Las plantillas de Coleman se utilizan para documentar los casos de uso.

Tabla 2: Plantilla modelo

Caso de Uso			
DEFINICION:			
PRIORIDAD:	<input checked="" type="radio"/> (1) Vital	<input type="radio"/> (2) Importante	<input type="radio"/> (3) Conveniente
URGENCIA:	<input checked="" type="radio"/> (1) Inmediata	<input type="radio"/> (2) Necesario	<input type="radio"/> (3) Puede esperar
ACTORES			
NOMBRE	DEFINICION		
ESCENARIO			
Nombre:			
Precondiciones:			
Iniciado por:			
Finalizado por:			
Post-condiciones:			
Operaciones:			
Excepciones:			
ESCENARIO			
Nombre:			
Precondiciones:			
Iniciado por:			
Finalizado por:			
Post-condiciones:			
Operaciones:			
Excepciones:			

Fuente: (<https://jmpovedar.files.wordpress.com/2014/03/plantillas-de-coleman.docx>, 2014)

6.4. Diseño de Software

El diseño de software es un proceso creativo que es perfeccionado con la experiencia, cuyo objetivo es la construcción de un modelo de lo que es requerido

por el sistema. El modelo debe ser preciso y comprensible. Según (Foster & Towle Jr., 2022):

El diseño de software es un proceso creativo que se va perfeccionando a través de la experiencia, este no puede ser aprendido a través de los libros, sin embargo, estos proveen lineamiento que deben ser seguidos. El objetivo del diseño de software es construir un modelo que represente el sistema requerido. El modelo debe ser preciso y comprensivo.

El proceso de diseño inicia con un esquema de diseño informal, el cual a través de varias etapas de refinamiento este se convierte en un modelo final. El resultado clave de la etapa de diseño es un documento llamado “Especificaciones de diseño”.

(p 193)

Para la elaboración del modelo se utilizan varias herramientas o plataformas en línea, sin embargo, todas estas trabajan sobre un conjunto de diagramas pertenecientes a un lenguaje conocido como UML.

6.4.1. UML

El lenguaje Unificado de Modelado (UML) crea una forma para que cualquier persona en el proyecto de software (programadores, analista de negocios y clientes) puedan compartir un mismo entendimiento y vocabulario al momento de discutir los detalles del software. Según (LEE, 2019):

El Lenguaje Unificado de Modelado (UML) está compuesto por un conjunto de exitosas prácticas de ingeniería de software que han sido compiladas en una especificación de modelado integral. Booch describe UML como “Un lenguaje estandarizado para especificar, visualizar, construir y documentar los elementos de un software, así como para modelado de negocios y otros sistemas no relacionados al software” (Booch et al. 1999). Este sistema de modelado permite que en la ingeniería de software orientada a objetos se visualice y diagramen en una forma estandarizada, permitiendo a los ingenieros de software comunicarse mejor acerca del diseño del sistema que se está creando.

UML fue diseñado para crear un lenguaje de modelado estandarizado que sigue los siguientes principios:

- *Lenguaje de programación independiente.*
- *Proceso de desarrollo independiente.*
- *Podría admitir conceptos de desarrollo de nivel superior*
- *Extensible y capaz de ser especializado a proyectos específicos*
- *Se deben integrar una variedad de las mejores prácticas del campo de la ingeniería de software*
- *Expresivo y listo para usarse, capaz de prever la creación de modelos útiles significativos.*

UML ha evolucionado desde su creación hasta la versión actual, UML 2.5.1, que incluye 13 diferentes tipos de diagramas divididos en 2 grupos principales:

- *Diagramas de estructura: Diagramas de Clases, Objetos, Paquetes, Componentes, Despliegue, Estructura compuesta y Perfil.*
- *Diagramas de comportamiento: Diagramas de Casos de uso, Actividad, Maquinas de estado y de Interacción. (pp 60 - 61)*

El Lenguaje Unificado de Modelado es una herramienta popular para expresar gráficamente las necesidades y comportamiento de un software, el cual tiene una extensión para los sistemas web llamado UWE.

6.4.2. UWE

El enfoque de la Ingeniería Web basada en UML (UWE) provee una notación específica de dominio, un proceso de desarrollo basado en modelos y el soporte de herramientas para la ingeniería de software para aplicaciones o sistemas web. Según (LMU – Ludwig-Maximilians-Universität München, 2016):

El enfoque de UWE proporciona una notación específica de dominio, un proceso de desarrollo basado en modelos y el soporte de herramientas para la ingeniería de las aplicaciones web. La característica de UWE es el hecho de ser un enfoque basado en estándares que no se limita al uso de UML, puede utilizar XMI como formato de intercambio de modelos, MOF para el meta modelado, los principios basados en modelos del enfoque MDA, los lenguajes de transformación de modelos QVT y XML.

La principal razón para utilizar UWE son la aceptación de UML en el desarrollo de software, la flexibilidad para la definición de un lenguaje para la ingeniería de software para aplicaciones web.

UWE utiliza una notación UML pura y utiliza los diagramas de UML en todo momento que sea posible para las etapas de análisis y diseño de una aplicación web (Sin extensiones de ningún tipo). Para las características específicas de la web como nodos y enlaces UWE proporciona estereotipos, valores etiquetados y restricciones definidas para los elementos del modelo. UWE cubre los modelos de navegación, presentación, procesos comerciales y aspectos de adaptación.

UWE se define como una extensión “Ligera” de UML.

UWE al ser una extensión de UML también considera diferentes diagramas en las etapas de análisis y diseño, sin embargo, en UWE estos diagramas se hacen llamar modelos y existen cinco tipos de modelos.

6.4.3. Tipos de modelo

Para modelar las características adaptativas de las aplicaciones web de forma no invasiva, UWE utiliza técnicas de modelado orientado a aspectos, siguiendo el principio de separación de responsabilidades, en el que cada modelo se encarga de una responsabilidad. Según (LMU – Ludwig-Maximilians-Universität München, 2016):

El enfoque de UWE para el modelado de procesos del negocio en la web consiste en introducir clases específicas que forman parte de un modelo de proceso, separado con una interfaz definida para el modelo de navegación. Para modelar las características adaptativas de las aplicaciones web, UWE utiliza técnicas de modelo orientado a aspectos (AOM), siguiendo el principio de separación de preocupaciones, UWE propone construir un modelo adaptativo para sistemas personalizados o dependientes del contexto y tejer los modelos después.

6.4.3.1. Modelo de Requerimientos.

El modelo de requerimientos consta de dos partes los casos de uso y las actividades. Según (LMU – Ludwig-Maximilians-Universität München, 2016):

En UWE el modelo de requerimientos consiste en 2 partes:

- Casos de uso de la aplicación y sus relaciones.
- Actividades describiendo los casos de uso a detalle.

Casos de uso

En UWE los casos de uso se distinguen por los estereotipos “browsing” y “processing” para indicar si la aplicación modifica la data persistente de la aplicación o no. Por ejemplo, una función de búsqueda usaría el estereotipo “browsing” debido a que este ejecuta una acción de lectura de datos y una función agregar o modificar utilizará el estereotipo “processing” ya que modificará los datos en la base de datos.

Actividad

Dado que en los casos de uso podemos detallar una limitada información, cada caso de uso debe ser descrito detalladamente en un flujo de proceso. Por lo tanto, se pueden modelar las acciones dentro de un caso de uso, así como los datos presentados al usuario y la entrada requerida.

Los estereotipos a utilizar son los siguientes:

userAction: se utilizará para indicar las interacciones de un usuario en la página web, iniciando un proceso o respondiendo a un requerimiento de información.

systemAction: se utilizará para indicar las acciones que ejecutará el sistema.

displayAction: se utilizará para los grupos de presentación.

displayPin & interactionPin: se utilizarán para los elementos de salida y entrada.

navigationAction: se utilizará para indicar las opciones de navegación y los elementos de presentación asociados.

6.4.3.2. Modelo de Contenido

El modelo de contenido es un diagrama de clases de UML normal, en el que debemos pensar acerca de las clases que necesitamos para ejecutar el software (LMU – Ludwig-Maximilians-Universität München, 2016):

6.4.3.3. Modelo de Navegación

El modelo de navegación es la representación gráfica de como las páginas pertenecientes a una aplicación web están conectadas entre sí. según (LMU – Ludwig-Maximilians-Universität München, 2016):

Para un sistema basado en páginas web, lo ideal sería conocer como las páginas se encuentran conectadas. Esto significa que se necesita un diagrama que contenga los nodos y enlaces.

Los nodos son las unidades de navegación conectadas por enlaces. Los nodos pueden ser presentados en una diferente o misma página. Para nodos y enlaces se utilizan los estereotipos “navigationClass” y “navigationLink”.

Existen otros estereotipos para determinar la funcionalidad de un nodo, se detallan a continuación, según (Ludwig-Maximilians-Universität München, 2008):

menu: se utilizará para manejar rutas de navegación alternativas.

index: se utilizará para mostrar un numero de objetos del mismo tipo.

query: se utilizará para obtener información de una fuente de datos.

guidedTour: se utilizará para proveer un proceso secuencial a las instancias de una clase de navegación.

processClass: se utilizará para integrar procesos de negocios en el modelo de navegación y para definir los datos que se intercambian con el usuario durante el proceso. (pp 9 - 20)

6.4.3.4. Modelo de presentación

En el modelo de presentación es la representación gráfica de las clases de navegación y las clases de procesos que pertenecen a una página web. Según (LMU – Ludwig-Maximilians-Universität München, 2016):

El modelo de navegación no muestra que clases de navegación y proceso pertenecen a que parte de la página web. El modelo de presentación nos provee esta información. Las propiedades pueden ser anidadas.

Los estereotipos a utilizar son los siguientes. Segundo (Ludwig-Maximilians-Universität München, 2008):

presentationGroup: se utilizará para definir un conjunto de clases de presentación cuyo contenido se muestra en la misma página, dependiendo de la navegación.

presentationPage: tiene la misma semántica de una clase de presentación, con la excepción de que no puede ser incluido dentro de otra clase de presentación.

text: se utilizará para mostrar textos estáticos.

textInput: se utilizará para indicar al usuario que debe ingresar texto.

anchor: permitirá al usuario activar una transición en el modelo de navegación junto con un determinado enlace.

button: se utilizará para identificar que el usuario puede iniciar una acción en el sitio web.

image: se utilizará para mostrar imágenes estáticas. (pp 14 - 18)

6.4.3.5. Modelo de procesos

El modelo de procesos muestra las acciones de las clases de proceso, este está compuesto por el modelo de estructura de proceso y el modelo de flujo de proceso. Según (LMU – Ludwig-Maximilians-Universität München, 2016):

El modelo de proceso muestra cómo se verán las acciones de nuestras clases de proceso, el modelo de proceso comprende:

- *El modelo de estructura de proceso que describe las relaciones entre las diferentes clases de procesos.*
- *El modelo de flujo de proceso es un diagrama de actividades que describe el comportamiento de las clases de proceso. Utiliza los estereotipos **userAction** y **systemAction**.*

6.4.4. Patrones de diseño

Un patrón de diseño puede ser caracterizado como una combinación de 3 reglas que expresan relación entre ellas: contexto, problema y solución. Según (Pressman & Maxim, 2020):

Un patrón de diseño puede ser caracterizado como “una regla de tres partes que expresa una relación entre un cierto contexto, un problema y una solución” [Ale79]. Para el diseño de software, el contexto permite al lector entender el ambiente en el que el problema reside y la solución que podría ser apropiada según el ambiente. Un conjunto de requerimientos incluyendo limitaciones y restricciones, actúa como un sistema de fuerzas que influencia como el problema puede ser interpretado dentro del contexto y como la solución puede ser aplicada efectivamente.

Un patrón de diseño efectivo, cumple con lo siguiente:

- *Captura una solución específica para un problema específico.*

- *Proporciona una solución que ha sido probada en la práctica.*
- *Identifica el enfoque de un problema que no es obvio.*
- *Identifica la relación entre el diseño y otros componentes de arquitectura.*
- *Es elegante en su enfoque y utilidad. (p 291)*

Existe una amplia variedad de patrones de diseño que podrían ser categorizados por su creación, estructura y comportamiento.

6.4.4.1. Patrón CQRS

El patrón de diseño CQRS se utiliza para separar las responsabilidades de comandos y consultas, es decir, separa las operaciones de lectura de los de registro y/o actualización. Según (Microsoft Learn, 2023):

CQRS Significa segregación de responsabilidades de comandos y consultar, un patrón que separa las operaciones de lectura y actualización de un almacén de datos. La implementación de CQRS en la aplicación puede maximizar el rendimiento, escalabilidad y la seguridad. La flexibilidad creada al utilizar CQRS permite que un sistema evolucione mejor con el tiempo y evita que los comandos de actualización provoquen conflictos de combinación en el nivel de dominio.

Contexto y problema

En las arquitecturas tradicionales, se utiliza el mismo modelo de datos para consultar y actualizar una base de datos. Es sencillo y funciona bien para las operaciones CRUD básicas. Sin embargo, en aplicaciones más complejas, este enfoque puede resultar difícil de manejar. Por ejemplo, en el lado de lectura, la aplicación puede realizar muchas consultas diferentes y devolver objetos de transferencia de datos (DTO) con distintas formas. La asignación de objetos puede llegar a ser algo complicado. En el lado de escritura, el modelo puede implementar una validación y una lógica de negocios complejas. En consecuencia, puede acabar con un modelo excesivamente complejo que haga demasiado.

Solución

CQRS separa las lecturas y las escrituras en diferentes modelos, usa comandos para actualizar los datos y consultas para leer los datos.

- Los comandos se deben basar en tareas, en lugar de centrarse en los datos. ("Book hotel room", no "set ReservationStatus to Reserved").
- Los comandos se pueden colocar en una cola para su procesamiento asíncrono, en lugar de procesarse sincrónicamente.
- Las consultas nunca modifican la base de datos. Una consulta devuelve un DTO que no encapsula ningún conocimiento del dominio.

6.4.5. Herramientas de Modelado

Las herramientas de modelado se utilizan para la creación de modelos de sistemas existentes o que se construirán, brindando los beneficios de crear una visualización gráfica del sistema a bajo costo y con riesgo mínimo. Existen diferentes opciones que han sido adaptadas a cada necesidad.

6.4.5.1. MagicDraw

MagicDraw está diseñado para los analistas de negocio, sistemas, programadores, ingenieros de QA y escritores de documentación. MagicDraw soporta UML, las últimas versiones del estándar XMI para almacenamiento de datos y los más populares lenguajes de programación para implementación. Según (3DS CATiA, s.f.):

MagicDraw fue diseñado para los analistas de negocio, analistas de software, programadores, ingenieros de calidad y escritores de documentación, esta versátil y dinámica herramienta de desarrollo facilita el análisis y diseño de sistemas y bases de datos orientados a objetos, proporciona el mejor mecanismo de ingeniería de código (con soporte total para los lenguajes de programación JAVA, C++, CL(MSIL) y CORBA IDL), así como modelado de esquemas de bases de datos, generación de DLL e ingeniería inversa.

MagicDraw soporta el modelo de UML 2, las versiones más recientes del estándar XMI para el almacenamiento de datos y los lenguajes de programación más populares para implementación.

MagicDraw aporta independencia de cualquier proceso de desarrollo específico, y se ajusta muy bien al proceso de la empresa; permitiendo la centralización del modelado de procesos y negocios, la captura y diseño de los requerimientos.

6.4.5.2. MagicUWE

MagicUWE es un plugin para la herramienta MagicDraw para el modelado de aplicaciones Web. Según (MagicUWE - UWE Plugin for MagicDraw, 2016):

La herramienta MagicUWE fue desarrollada para el diseño de aplicaciones Web, utilizando la metodología de ingeniería web basada en UML (UWE). MagicUWE fue construido como un complemento para MagicDraw.

6.4.5.3. BPMN

El Modelo y Notación de Procesos de Negocio (BPMN) es una notación gráfica para el modelado de negocios. Según (Wiegers & Beatty, 2013):

El Modelado y Notación de Procesos de Negocio (BPMN) es una notación gráfica para el modelado de procesos de negocio (OMG 2011). BPMN puede ser aplicado a cualquier enfoque en el proceso de modelado de negocio. Es un robusto lenguaje de símbolos que puede ser útil cuando el diagrama de carriles no es suficiente.

6.5. Construcción de Software

El desarrollo de software es la etapa donde se puede ver el fruto del trabajo diligente realizado en el levantamiento y diseño de software. Según (Foster & Towle Jr., 2022):

El desarrollo de software es divertido, representa el comienzo de ver los frutos de un trabajo diligente realizado en las etapas de planeación, análisis y diseño del software. Vale la pena, recordar la siguiente afirmación:

“Es posible obtener un software de baja calidad basado en un excelente diseño (gracias a la incompetencia o descuidada etapa de desarrollo) y también podemos obtener un software de excelente calidad de un diseño de baja calidad”.

En el desarrollo de software se utilizará el siguiente enfoque:

Modelo de desarrollo Ágil: combinación de ideas para la creación de prototipos por fases y desarrollo iterativo, enfocándose en la construcción y entrega de un producto con las características esenciales mínimas requeridas. Luego refinar o expandir en subsecuentes versiones según sea necesario.

Durante el desarrollo de software, el equipo de desarrolladores utilizará un conjunto de herramientas para la creación o generación de código del sistema, basado en las especificaciones del diseño.

Para esta etapa del ciclo de desarrollo la entrada es el documento de especificaciones de diseño y la salida es el software como producto. (pp 297 - 298)

6.5.1. Aplicación de página única (SPA)

Una aplicación de página única es una aplicación o sitio web que cabe en una sola página con el propósito de tener un mejor rendimiento. Según (Mikowski & Powell, 2014):

Un SPA es una aplicación que se despliega en el navegador web y que no necesita recargarse durante su uso. Como toda aplicación, su intención es ayudar a los usuarios a completar tareas como “escribir un documento” o “administrar un servidor web”. Se podría pensar que un SPA es un cliente obeso que es cargado desde un navegador. (p 4)

Toda aplicación web debe estar alojado en un servidor ya sea legacy o en la nube, puede ser utilizando una máquina virtual como servidor o bien tecnología que no necesita servidores (*Serverless*).

6.5.2. Alojamiento del sistema Web

El alojamiento web es un servicio que almacena su sitio o aplicación web y lo hace accesible desde las computadoras de escritorio, web, celulares o tablets. Según (AWS, s.f.):

El alojamiento web es un servicio que almacena su sitio web o aplicación web y lo hace fácilmente accesible a través de diferentes dispositivos como el escritorio, el móvil y las tabletas. Cualquier aplicación web o sitio web suele estar formado por muchos archivos, como imágenes, vídeos, texto y código, que hay que almacenar en ordenadores especiales llamados servidores. El proveedor de servicios de alojamiento web mantiene, configura y ejecuta los servidores físicos que usted puede alquilar para sus archivos. Los servicios de alojamiento de sitios y aplicaciones web también proporcionan apoyo adicional, como seguridad, copias de

seguridad y rendimiento del sitio web, con lo que se ahorra tiempo para poder centrarse en las funciones principales del sitio web.

6.5.2.1. Computación en la nube

La computación en la nube es un término que se refiere acceso por demanda a servicios de IT que se componen de cómputo, internet, almacenamiento y servicios de software de proveedores externos. Según (Daswani, 2021):

El termino computación en la nube es usado para describir los accesos por demanda a servicios de IT que están compuestos por capacidad de cómputo, internet, almacenamiento y servicios de software provistos por un proveedor externo, usualmente a través de la internet pública o algún acceso directo a través de Wide-Area Network (WAN). Las compañías pueden proporcionar las aplicaciones de TI necesarias para su organización sin tener que adquirir y administrar su propia infraestructura para alojar estas aplicaciones. Por el contrario, la compañía puede rentar la infraestructura de TI requerida de los proveedores externos. (p 4)

Existe una variedad de proveedores de computación en la nube en el mercado, entre los más comunes o conocidos se encuentra la plataforma de Azure.

6.5.2.2. Plataforma Azure

Azure es la plataforma publica en la nube creada por Microsoft, en la que se ofrecen un gran número de servicios IaaS, PaaS y SaaS, todo esto se encuentra implementado en uno o más centros de datos para proveer los servicios a los clientes. Según (Microsoft Learn, 2023):

Azure es la plataforma publica en la nube de Microsoft. Azure ofrece una gran colección de servicios que incluyen plataforma como servicio (PaaS), infraestructura como servicio (IaaS), software como servicio (SaaS) y servicio de bases de datos administradas.

Azure es como cualquier otra plataforma en la nube, toda su tecnología está basado en lo que conocemos como virtualización. Casi todas las configuraciones de hardware las podemos emular en software.

Esencialmente, la nube es un conjunto de servidores físicos en uno o más centros de datos. Los centros de datos ejecutan hardware virtualizado para los clientes.

Azure es una enorme colección de servidores y equipos de red, que ejecutan un complejo conjunto de aplicaciones distribuidas. Estas aplicaciones organizan la configuración y el funcionamiento del hardware y el software virtualizados. Con Azure, los usuarios no tienen que mantener ni actualizar su hardware, ya que Azure lo hace en segundo plano.

Azure es una plataforma muy completa que ofrece diferentes servicios para la administración de base datos, balanceador de carga, seguridad en los API, y para el hospedaje en la nube, como lo son los Azure App Service.

6.5.2.3. Azure App Service

Una app Service es un servicio de Azure, en el que podemos desplegar aplicaciones de .NET, Node.JS, Java, Rubi, etc. Estos pueden ser creadas con un sistema operativo Windows o Linux, y también es posible agregar otras características de DevOps como lo es CI/CD, agregar un dominio personalizado, entre otros. Según (Microsoft Learn, 2022):

Azure App Service es un servicio basado en HTTP para el hospedar aplicaciones web, REST APIs y back-ends móviles. Se puede utilizar los siguientes lenguajes .NET, .NET Core, Java, Rubi, Node.js o Python. Las aplicaciones corren y se pueden escalar fácilmente tanto en Windows como en Linux.

Los App Service no solo agregan el poder de Azure a la aplicación, como seguridad, balanceo de carga, auto-escalamiento y administración automatizada. También puede aprovechar sus capacidades de DevOps, como la implementación continua desde Azure DevOps, GitHub, Docker Hub y otras fuentes, administración de paquetes, ambientes de prueba, dominio personalizado y certificados TLS/SSL.

Los Azure App Service, son muy ventajosos al momento de hospedar una aplicación web, este servicio tiene una característica que nos permite ejecutar tareas y que no tienen costo adicional, llamada WebJob.

6.5.2.4. Azure WebJob

Un Azure WebJob es una funcionalidad de los Azure App Service que permite ejecutar tareas de forma calendarizada o continua. Según (Microsoft Learn, 2022):

Un WebJob es una característica de Azure app Service que permite ejecutar programas o scripts en la misma instancia de una aplicación web, REST API o una aplicación móvil. Se puede utilizar los Azure WebJobs para simplificar muchas tareas de programación.

Los tipos de archivos permitidos son los siguientes:

- *.cmd, .bat, .exe (usando la consola de Windows)*

Los WebJobs son soportados únicamente en el sistema operativo Windows.

6.5.3. Protocolos de red

Los protocolos de red son la tecnología, reglas y procedimientos que definen como empaquetar y formatear un mensaje para enviar a través de internet. Según (Singh, 2022):

Los protocolos de red son simplemente la tecnología, las reglas y los procedimientos subyacentes que definen cómo un remitente puede empaquetar y formatear un mensaje para enviarlo a través de una red a un host de destino. sin protocolos o reglas para la comunicación en una red, los dispositivos no formatearán ni direccionarán un mensaje correctamente antes de colocar el mensaje en la red para su entrega. (p 224)

6.5.3.1. HTTP

El Protocolo de Transferencia de Hipertexto (HTTP) se utiliza para realizar la navegación en los sitios web. Es un protocolo que no tiene seguridad, por lo tanto, los mensajes se envían en texto plano. Según (Singh, 2022):

Cuando un usuario realiza interacción con un sitio web a través de una red privada o la internet, el usuario accederá al navegador que utiliza HTTP para crear un mensaje que es reconocible para la aplicación web que está ejecutándose en un servidor web. HTTP es un protocolo que no tiene seguridad, no provee seguridad o privacidad de los datos, envía los mensajes en texto plano a través de la red hacia el servidor web. Este protocolo por defecto usa el puerto 80. (p 240)

6.5.3.2. HTTPS

El Protocolo de Transferencia de Hipertexto Seguro (HTTPS) surgió para resolver los problemas de seguridad de HTTP, con HTTPS la comunicación es cifrada. Según (Singh, 2022):

El Protocolo de Transferencia de Hipertexto Seguro (HTTPS) es la versión segura de HTTP, establece la comunicación segura entre el navegador web y el servidor web en la red, por defecto utiliza el puerto 443. HTTPS puede utilizar SSL o TLS para conectarse a un servidor web. (pp 240 -241)

6.5.3.3. SSL/ TLS

La Capa de Sockets Seguro (SSL) es un protocolo que provee los servicios de encriptación y autenticación entre dos partes durante la comunicación. Según (Meyers & Jernigan, 2021):

La Capa de Sockets Seguro (SSL) es un protocolo que provee servicios de encriptación y autenticación durante la transmisión de datos.

El servicio de encriptación encripta la sesión completa entre dos equipos en un túnel SSL, cualquier tráfico que es enviado entre los dispositivos es cifrado y protegido de cualquier intercepción no autorizada.

El servicio de autenticación normalmente utiliza certificados digitales para autenticar el servidor para un cliente, sin embargo, SSL también soporta una autenticación mutua donde el cliente y servidor intercambian certificados. (p 510)

La Capa de Transporte Seguro (TLS) es un protocolo que fue desarrollado para reemplazar SSL hace un tiempo atrás. Según (Meyers & Jernigan, 2021):

Debido a las debilidades en los algoritmos de cifrado usados en varias versiones de SSL, el protocolo de Capa de Transporte Seguro (TLS) fue creado, para reemplazar a SSL. Hoy día ha reemplazado casi por completo el uso de SSL en las páginas web. Actualmente todos los sistemas operativos modernos, navegadores, aplicaciones web soportan TLS.

Desde un punto de vista funcional, TLS y SSL son muy similares. De hecho, dado que TLS ha suplantado a SSL, es muy común ver referencias como SSL/TLS. SSL y TLS utilizan el puerto 443.

TLS asegura la transferencia de datos sensibles durante las sesiones web. Adicionalmente TLS es utilizado para proteger otros protocolos que requieren sesiones seguras como SFTP. (pp 510 - 511)

6.5.3.4. SFTP

El protocolo de transporte de archivos seguros (SFTP) opera en un modelo cliente-servidor, que permite a los usuarios cargar y descargar archivos a través de internet.

Según (Singh, 2022):

El protocolo de transporte de archivos seguro (SFTP) es otro protocolo de transferencia de archivos que permite a los usuarios de forma segura transferir archivos entre el cliente y el servidor usando internet. SFTP usa SSL o TLS para encriptar el mensaje FTP que se enviará a través de internet entre el usuario y el servidor.

El cliente encripta el mensaje FTP usando SSL o TLS antes de enviar el mensaje a través de internet hacia el servidor FTP destino. Dado que los mensajes se envían a través de la red, no hay conexión segura, cada paquete SFTP se cifra individualmente mediante SSL/TLS, por lo tanto, el servidor FTP descifrará cada mensaje SFTP a medida que se reciban y volverá a ensamblar los mensajes en datos. (p 228)

6.5.4. REST API

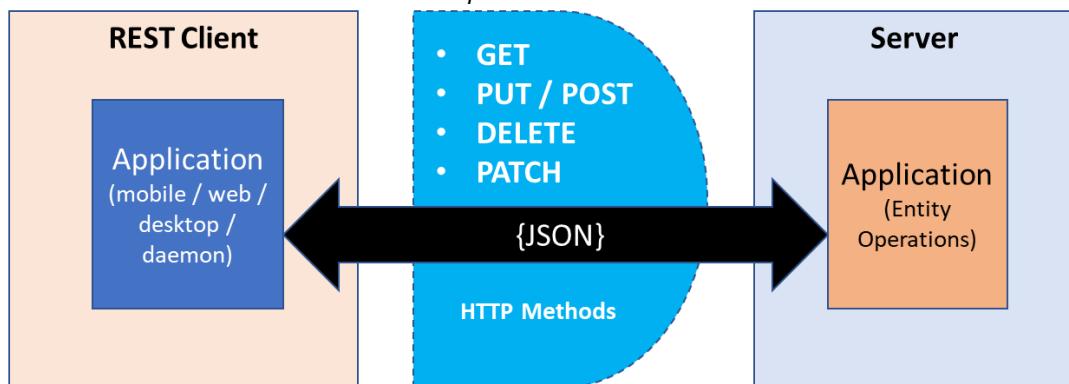
Los RESTful APIs son un estilo de arquitectura para la creación de servicios Web que deben cumplir con un conjunto de reglas y restricciones. Estas constan de dos partes, el cliente quien hace las peticiones y el servidor quien tiene los recursos.

Según (Chatterjee, 2021):

El concepto REST es un acrónimo de Transferencia de Estado Representacional, es un estilo de arquitectura para la creación de servicios web que cumple con un conjunto de reglas y restricciones. REST usa únicamente un subconjunto del protocolo estándar HTTP.

Una arquitectura de sistema basada en REST se puede visualizar como dos partes; primero el cliente que es quien realiza las peticiones, y el servidor quien posee los recursos. Los servicios web basados en HTTP que se adhieren a las pautas REST se conocen como RESTful. (pp 108 -109)

Ilustración 1: Arquitectura estilo REST



Fuente: (Chatterjee, 2021, pág. 109)

Para procesar las peticiones de un cliente se utilizan los verbos HTTP, los más utilizados son POST, GET, PUT, PATCH y DELETE que hacen correspondencia con el CRUD.

6.5.4.1. Verbos HTTP/REST

Las operaciones de los servicios REST API son modeladas como una operación HTTP, y debemos mantener una correspondencia entre la acción y el verbo. Según (Chatterjee, 2021):

Las operaciones REST API siempre son modeladas como una operación del protocolo HTTP. Existe una relación entre el verbo HTTP y la acción respectiva del CRUD. Debemos mantener esto en mente al momento de preparar la definición de la interfaz del API. (p 110)

Tabla 3: Verbos HTTP y su uso

Verbo HTTP	Acción del CRUD	Propósito	Código de estado esperado
POST	Insertar	<i>Insertar un nuevo registro usando los datos disponibles en el cuerpo de la petición</i>	<i>201: Creado 404: No encontrado 409: Conflicto (Si el recurso ya existe)</i>
GET	Leer	<i>Retorna uno o más registros. No debe modificar datos</i>	<i>200: Ok (Información solicitada) 404: No encontrado</i>

			<i>Usa paginación, filtros, orden para navegar listas grandes.</i>
<i>PUT</i>	<i>Actualizar / Reemplazar</i>	<i>Actualizar o reemplazar un registro existente</i>	<i>200: OK 204: Sin contenido 404: No encontrado 405: No permitido</i>
<i>PATCH</i>	<i>Actualizar / Modificar</i>	<i>Actualizar o modificar un registro existente</i>	<i>200: OK 204: Sin contenido 404: No encontrado 405: No permitido</i>
<i>DELETE</i>	<i>Borrar</i>	<i>Borrar o inactivar un registro existente; para determinar si el registro fue borrado se utilizan los códigos HTTP</i>	<i>200: OK 404: No encontrado 405: No permitido</i>

Fuente: (Chatterjee, 2021, pág. 111)

REST APIs puede utilizar una gran variedad de formatos como XML, YAML o cualquier otro legible por máquina, sin embargo, el formato más utilizado y preferido es JSON.

6.5.4.2. JSON

La Notación de Objetos JavaScript (JSON) es una notación liviana, basada en texto y que es utilizada para el intercambio de información independientemente del lenguaje. Según (RFC, 2006):

La Notación de Objetos JavaScript (JSON) es un formato de texto que se utiliza en la serialización de datos estructurados. Es derivado de los objetos literales de JavaScript definidos en el lenguaje de programación estándar ECMAScript (tercera edición).

JSON puede representar cuatro tipos primitivos (strings, numbers, booleans y null) y dos tipos de datos estructurados (objetos y arreglos).

Los objetivos del diseño de JSON son minimalista, portátil, textual y un subconjunto de JavaScript.

6.5.5. Tecnologías Front-End

Las tecnologías Front-End son las que se utilizan del lado del cliente, se utilizan en los distintos dispositivos que se utilizan para conectarse al servidor a través de internet. Normalmente estas tecnologías vienen implementadas en los navegadores. Las que se utilizaran en el proyecto son HTML5, CSS, JavaScript y Angular.

6.5.5.1. HTML5

El lenguaje de marcado de hipertexto (HTML) es un lenguaje de computadoras que está diseñado para especificar el contenido como la estructura de las páginas web. Según (Deitel, Deitel, & Deitel, 2014):

HTML (Lenguaje de marcado de hipertexto): un tipo especial de lenguaje de computadora que se cómo lenguaje de marcado y está diseñado para especificar tanto el contenido como la estructura de páginas web (también conocidas como documentos) de una manera portable.

Este lenguaje nos permite crear contenido que se despliega de manera apropiada a través de todo el extraordinario rango de dispositivos conectados a internet: smartphones, computadoras tipo Tablet, computadoras tipo notebook, computadoras de escritorio, dispositivos de propósito especial como pantallas gigantes en auditorios de conciertos y estados deportivos, y muchos más. (p 6)

Aunque HTML5 puede controlar la presentación de un documento, es mejor no mezclar la presentación con el contenido, para controlar la presentación se utilizará hojas de estilo en cascada (CSS)

6.5.5.2. Hojas de estilo en cascada (CSS)

Las hojas de estilo (CSS) se utilizan para controlar la presentación de un documento HTML. Según (Deitel, Deitel, & Deitel, 2014):

Las hojas de estilo en cascada (CSS) se usan para especificar la presentación o el estilo de los elementos en una página web (fuentes, espaciado, tamaños, colores, posicionamiento). La tecnología CSS se diseñó para aplicar estilo a las páginas webs portables en forma independiente a su contenido y estructura. Al separar el

estilo de las páginas de su contenido y estructura, es posible cambiar con facilidad la apariencia visual en un sitio web completo, o en una parte de un sitio web. (p 7)

Para agregar dinamismo a las páginas web, se utiliza un lenguaje llamado JavaScript.

6.5.5.3. JavaScript

JavaScript es un lenguaje que ayuda a crear páginas web dinámicas. Según (Deitel, Deitel, & Deitel, 2014):

JavaScript es un lenguaje que nos ayuda a crear páginas web dinámicas (es decir, páginas que pueden modificarse “al instante” en respuesta a ciertos eventos, como la entrada del usuario, cambios en el tiempo, etcétera) y aplicaciones de computadora. Nos permite realizar la programación del lado del cliente de las aplicaciones Web. (p 7)

6.5.5.4. Frameworks de Front-End

Los Frameworks de Front-End son un conjunto de lenguajes para crear aplicaciones web. HTML es el responsable de los diseños, CSS administra el formato y la estructura visual, JavaScript para mantener la interactividad y funcionalidad. (Osman, 2022):

La mayoría de las veces, los desarrolladores front-end necesitan usar un conjunto combinado de lenguajes para crear sus aplicaciones web front-end. HTML es responsable de los diseños básicos en una página web, CSS administra el formato y la estructura visual, y JavaScript se usa para mantener la interactividad y la funcionalidad.

Los Frameworks de front-end facilitan el trabajo de los desarrolladores web al proporcionarles módulos de código reutilizables, tecnologías front-end estandarizadas y bloques de interfaz listos para usar que facilitan el desarrollo de aplicaciones e interfaces de usuario sin la necesidad de codificar cada función u objeto desde cero.

6.5.5.4.1. Angular

Angular es una plataforma de desarrollo basada en TypeScript. Según (Angular, 2022):

Angular es una plataforma de desarrollo basada en TypeScript. Como plataforma, Angular incluye:

- *Un framework basado en componentes para construir aplicaciones web escalables.*
- *Una colección de librerías bien integradas que cubren una gran variedad de características, incluyendo: ruteo, administración de formularios, comunicación cliente servidor y más.*
- *Un conjunto de herramientas de desarrollo que ayudará a desarrollar, compilar, probar y actualizar el código.*

Al utilizar Angular se tienen las ventajas de una plataforma que puede escalar de un único desarrollador de proyectos a un nivel de aplicaciones empresariales. Angular está diseñado para realizar actualizaciones de la forma más sencilla, es decir, se pueden tomar las ventajas de las últimas versiones con un esfuerzo mínimo. Lo mejor de todo, el ecosistema de Angular consiste en un grupo de alrededor de 1.7 millones de desarrolladores, creadores de librerías y creadores de contenido.

6.5.6. Tecnologías Back-End

Las tecnologías Back-End son las que se utilizan del lado del servidor, se utilizan para gestionar las diferentes peticiones de información y para gestionar las bases de datos. Las que se utilizarán en el proyecto son .NET y SQL

6.5.6.1. .NET

.NET es una plataforma para desarrolladores de código abierto, multiplataforma y gratuita para crear muchos tipos de aplicaciones diferentes. Según (Microsoft, 2023):

.NET es una plataforma para desarrolladores de código abierto, multiplataforma y gratuita para crear muchos tipos de aplicaciones.

Puede escribir aplicaciones en .NET en C#, F# o Visual Basic; C# es un lenguaje de programación simple, moderno, orientado a objetos y con seguridad de tipos.

Tanto si trabaja en C#, F# o Visual Basic, el código se ejecutará de forma nativa en cualquier sistema operativo compatible.

.NET proporciona un conjunto estándar de API y bibliotecas de clases base que son comunes a todas las aplicaciones .NET

6.5.6.2. Principios SOLID

Los principios SOLID son un subconjunto de muchos principios, estos principios se han convertido en estándares de la programación orientada a objetos y también se han convertido en la filosofía principal de otras metodologías y paradigmas. Según (Akella, Kumar Tamirisa, Kumar Kunani, & Guptha Muthiyalu, 2021):

Los principios SOLID son un subconjunto de muchos principios promovidos por el ingeniero de software e instructor estadounidense Robert C. Martin. Estos principios se han convertido en los principios estándar del paradigma de la programación orientada a objetos y se ha convertido en el fundamento de la filosofía de otras metodologías y paradigmas.

SOLID es un acrónimo de los siguientes 5 principios:

1. **Principio de responsabilidad única (SRP):** una entidad o módulo de software debe tener una responsabilidad única. Se debe evitar otorgar a una entidad múltiples responsabilidades.
2. **Principio de abierto-cerrado (OCP):** las entidades deben ser diseñadas de manera que estas estén abiertas a extenderse y cerradas para modificarse. Esto significa que se puede evitar las pruebas de regresión y ejecutar únicamente las pruebas para validar las extensiones.
3. **Principio de sustitución de Liskov (LSP):** las instancias de clase principal o base deben ser reemplazables con instancias de sus clases derivadas o subtipos sin alterar la sanidad del programa.
4. **Principio de segregación de interfaces (ISP):** en vez de utilizar una interfaz de uso común, se debe planificar múltiples interfaces basadas en escenarios específicos para un mejor desacoplamiento y mantenibilidad del software.
5. **Principio de inversión de dependencias (DIP):** Se debe evitar tener dependencias directas a implementaciones. Módulos de alto y bajo nivel no deben depender directamente unos de otros, sino que ambos deben depender de abstracciones tanto como sea posible. (pp 5 - 6)

6.5.6.3. JSON Web Token

Los JSON Web Token (JWT) es un estándar abierto que utiliza una forma compacta de compartir información segura entre dos entidades. Este token es firmado digitalmente. Según (OKTA, s.f.):

Los JSON Web Token (JWT) es un estándar abierto (<https://www.rfc-editor.org/rfc/rfc7519>) que define de manera compacta y autónoma un mecanismo de transmisión de información en forma de objeto JSON entre dos partes en modo seguro. Esta información compartida es verificable y confiable ya que se encuentran firmadas digitalmente. Los JWT pueden ser firmados utilizando una palabra clave (usando el algoritmo HMAC) o utilizando una combinación de llaves pública y privada usando RSA o ECDSA.

Los tokens firmados pueden verificar la integridad de la información contenida en él, mientras que los tokens cifrados ocultan la información para cualquier otra entidad. Cuando utilizamos tokens firmados con llaves públicas o privadas.

Se puede utilizar JWT en los siguientes escenarios:

Autorización: una vez el usuario ha iniciado sesión, cada petición al servicio debe enviar en su encabezado el JWT, permitiendo a los usuarios acceder a los recursos, rutas y servicios a los que tiene acceso.

Intercambio de información: los JWT son una excelente forma de compartir información de forma segura entre entidades. Dado que los JWT se pueden firmar, se puede estar seguro que la persona que envía la información es el que dice ser.

Los JWT en su forma compacta consiste en tres partes separadas por puntos

Header: este consiste en 2 partes, el tipo de token que es JWT y el algoritmo utilizado para firmar.

Payload: este contiene toda la información que agregamos al token (nombre de usuario, rol, correo electrónico, etc.). Existen 3 tipos de información: información registrada estos no son requeridos, pero es preferible que estén presentes (issuer, exp time, subject); información pública esta puede ser definida a voluntad en el JWT; información privada esta son llaves creadas para compartir información entre entidades.

Signature: es utilizado para verificar que el mensaje no ha sido modificado en el transcurso del tiempo.

6.5.6.4. Inyecciones de Dependencia

Las inyecciones de dependencia (DI) son una técnica en la que los objetos reciben los objetos de los que depende. Según (Akella, Kumar Tamirisa, Kumar Kunani, & Guptha Muthiyalu, 2021):

Las inyecciones de dependencia son una técnica en la que los objetos reciben los objetos de los que depende. Las DI cumplen el principio de inversión de dependencias de los principios SOLID. Con el uso de las DI el código será mantenible, leíble, fácil de hacer pruebas y fácil para realizar extensiones.

DI es uno de los métodos más conocidos que ayudan a obtener un código de fácil mantenimiento.

DI tiene tres entidades inmersas el inyector, el servicio y el cliente. El inyector crea una instancia del servicio y lo inyecta en el objeto cliente. Los clientes dependen de los servicios inyectados para ejecutar operaciones. (p 134)

En las inyecciones de dependencias existen múltiples formas de realizar las inyecciones, esto es conocido como Tipos de DI.

6.5.6.4.1. Tipos de DI

Los tipos de inyecciones de dependencia son la forma en la que los servicios son inyectados como dependencia. Según (Akella, Kumar Tamirisa, Kumar Kunani, & Guptha Muthiyalu, 2021):

Existen diversas formas en la que un servicio puede ser inyectado como dependencia. Basado en la forma que los servicios son inyectados en los objetos clientes, las DI son categorizadas en tres tipos:

Inyección de constructor: las dependencias son inyectadas a través de un constructor y al mismo tiempo se inicializa la dependencia.

Inyección de setter: las dependencias exponen un método o propiedad set que el inyector utiliza para inyectar la dependencia.

Inyección de método: En estos casos la dependencia es establecida a través de un parámetro de un método. (pp 135 - 136)

Cada dependencia tiene un tiempo de vida definido según el tipo utilizado al momento de registrarla.

6.5.6.4.2. Tiempos de vida de las DI

Cuando se registra un servicio es necesario especificar el tiempo de vida del mismo, el contenedor eliminará la instancia del servicio basado en el tipo especificado al momento de registrar la DI. Según (Akella, Kumar Tamirisa, Kumar Kunani, & Guptha Muthiyalu, 2021):

Al momento de registrar un servicio como DI se especifica un tiempo de vida, el contenedor eliminará automáticamente la instancia del servicio de acuerdo al tiempo de vida que se estableció. Existen tres tipos de tiempos de vida disponibles para uso con el contenedor DI de Microsoft:

Transient: los objetos son creados cada vez que es solicitado del contenedor de servicios. Este tipo de tiempo de vida se utiliza para servicios livianos o stateless

Singleton: los objetos son creados una única vez por ciclo de vida de la aplicación. Es decir, por cada petición que realizamos obtendremos el mismo objeto.

Scope: los objetos son creados una única vez en el alcance de la petición del cliente. Por ejemplo, cada vez que realicemos una petición http al servicio se creará un nuevo objeto. (p 139)

6.5.6.5. Sistema Gestor de Base de datos

Un Sistema Gestor de Bases de Datos (SGBD) es un software dirigido a crear, gestionar y administrar la información que se encuentra en la base de datos Según (Universidad Europea, 2022):

Un sistema gestor de bases de datos (SGBD) es un software constituido por una serie de programas dirigidos a crear, gestionar y administrar la información que se encuentra en la base de datos. Su principal objetivo es servir de interfaz entre los usuarios y las aplicaciones para facilitar la organización de los datos, garantizar su accesibilidad, calidad e integridad, brindando a su vez una manera eficaz de administrar esa información.

El lenguaje de manipulación, un lenguaje de definición de datos y un lenguaje de consulta son los tres componentes de un gestor de base de datos; por lo que puede trabajar a diferentes niveles, pero es invisible para el usuario final.

6.5.6.5.1. Servidor SQL Lógico

Un servidor lógico es una construcción lógica que actúa como punto administrativo central para una colección de bases de datos SQL de Azure. Según (Microsoft Learn, 2022):

Un servidor es una construcción lógica que actúa como punto administrativo central para una colección de bases de datos. En el nivel de servidor lógico, puede administrar inicio de sesión, reglas de firewall, reglas de auditoría, directivas de detección de amenazas y grupos de conmutación por error automático.

El servidor lógico debe existir antes de poder crear una base de datos en Azure SQL Database. Todas las bases de datos que administra un servidor lógico se crean en la misma región que ese servidor lógico.

6.5.6.5.2. Azure SQL Database

Azure SQL Database es el motor de base de datos SQL que es de tipo Software como servicio, en la que la plataforma provee el sistema operativo, lo actualiza y aplica los parches necesarios para que la base de datos se ejecute sobre un ambiente actualizado. Según (Microsoft Learn, 2023):

Base de datos SQL en Azure es un motor de base de datos de plataforma como servicio (SaaS) completamente administrado. Este maneja la mayoría de funciones de administración como actualización, copias de seguridad y la supervisión sin supervisión de un usuario.

Las bases de datos SQL en Azure, se ejecutan en la última versión estable del motor de SQL Server y el sistema operativo parchado con un 99.99% de disponibilidad. Las capacidades de PaaS integradas en Azure SQL Database le permiten concentrarse en las actividades de optimización y administración de bases de datos específicas del dominio que son críticas para el negocio.

Con Azure SQL Database, puede crear una capa de almacenamiento de datos de alta disponibilidad y alto rendimiento para las aplicaciones y soluciones en Azure. SQL Database puede ser la elección correcta para una variedad de aplicaciones modernas en la nube porque le permite procesar datos relacionales y estructuras no relacionales, como gráficos, JSON, espaciales y XML

6.5.6.6. Entity Framework

Entity Framework es el enfoque de Microsoft para el acceso a datos desde las aplicaciones de software. Según (Driscoll, Gupta, Vettor, Hirani, & Tenny, 2013):

Entity Framework es el enfoque estratégico de Microsoft para la tecnología de acceso a datos para crear aplicaciones web. Entity Framework, junto con Visual Studio, ofrece un completo ecosistema basado en modelos que le permitirá desarrollar una amplia gama de aplicaciones orientadas a datos. Esto incluye aplicaciones de escritorio, web, nube y aplicaciones basadas en servicios. (p 2)

Existen varios enfoques para hacer uso de Entity Framework, cada uno tiene una perspectiva, nivel de madurez y usos diferentes.

6.5.6.6.1. Enfoque DB First

El enfoque base de datos primero se utiliza cuando tenemos la base de datos creada previamente. Según (Driscoll, Gupta, Vettor, Hirani, & Tenny, 2013):

El primer enfoque soportado por Entity Framework es base de datos primero, usando este enfoque el desarrollador empieza con una base de datos existente que es usada para crear el modelo conceptual inicial. Este modelo inicial sirve como punto de inicio para desarrollos futuros. A medida que la base de datos cambia, el modelo puede ser cambiado a partir de los cambios en la base de datos. Este enfoque es el mejor enfoque cuando deseamos migrar una aplicación existente al uso de EF. (p xxxiii)

Capítulo I: Análisis de requerimientos

El análisis se basa en la recopilación de información y en un estudio de la situación actual de la empresa y conocer su problemática; todo esto utilizando técnicas y procedimientos que permitirán conocer los elementos necesarios para el análisis, con el fin de poder dar una solución sistematizada y confiable a la situación que tienen hoy en día. Para ello se hará una descripción de procesos, el alcance que tendría el sistema y definir requerimientos e involucrados al sistema web de referidos.

1.1. Perfil del negocio

Actualmente no se cuenta con una Misión, Visión y Valores claros y específicos como empresa, por lo que se realizó la siguiente propuesta dando respuesta a las siguientes preguntas:

- ¿Qué hacemos? ¿Quién es nuestro público objetivo?
- ¿Qué queremos lograr? ¿Para quién lo hare?
- ¿Cómo somos? ¿En qué creemos?

Misión

Somos una empresa que a través del intercambio de información, experiencia y coordinación de esfuerzos ofrecemos soluciones tecnológicas, innovadoras y agiles con la ayuda de un equipo altamente capacitado, enfocados en la satisfacción del cliente y la provisión de servicios de telecomunicaciones para el desarrollo regional.

Visión

Buscar constantemente nuevos talentos con ideas frescas para consolidar un lugar como el proveedor de telecomunicaciones móviles líder en la región, satisfaciendo las expectativas de nuestros clientes en calidad, respuesta y costos.

Valores

Actualmente la empresa no cuenta con los valores específicos y claros, por lo que se les hace la propuesta siguiente:

- **Innovación:** Buscamos constantemente hacer las cosas de manera diferente para agregar valor a nuestros clientes a través de nuevos productos y servicios.
- **Enfoque en resultados:** Todas nuestras acciones buscan maximizar el valor como empresa, teniendo como base la eficiencia como una constante del día a día, proyectos y procesos.
- **Desarrollo de empleados:** Reconocemos y apoyamos el desarrollo de nuestro personal a través de capacitación continua y con un liderazgo participativo de la mano de nuestros gerentes.
- **Pasión:** Es lo que nos mueve y ayuda a ser agiles y responsables con nuestros clientes, para cumplir con los compromisos adquiridos como empresa.
- **Integridad:** Trabajamos en apego a la ley y la ética profesional con respeto a las diferencias de cada persona.
- **Transparencia:** Uno de nuestros pilares principales para la tranquilidad de nuestros clientes y competencia del mercado, buscando estrategias claras y concisas.
- **Compromiso:** Por cumplir las expectativas que tienen nuestros clientes y mejorar nuestra posición en el mercado, con personal capacitado.

1.2. Descripción de los procesos de negocio

El proceso para la ejecución de recompensas es activado cuando el equipo de marketing recibe un correo electrónico con los datos de uno o varias personas que se encuentran interesadas en formar parte de la empresa, el equipo de marketing crea un archivo Excel que contiene la siguiente información:

- **Referente:** Número telefónico de la persona (sin código de país)
- **Referido:** Nombre y apellido, correo electrónico, provincia, ciudad y número telefónico (sin código de país)

Al finalizar la elaboración del archivo, el equipo de marketing envía el documento generado al equipo de ventas.

Una vez que el equipo de ventas recibe el reporte de referidos, este se dispone a ingresar la información recibida en su base de datos (una plantilla en Excel), para que los datos sean ingresados en el archivo Excel es necesario que la persona asigne un vendedor para que este realice las llamadas pertinentes al referido. Al finalizar cada llamada realizada a un referido el vendedor debe ingresar el detalle obtenido en la llamada en el campo Comentarios y se debe indicar si habrá un nuevo contacto o si el cliente decidió no aceptar la nueva línea. En caso que el vendedor realice nuevos contactos, este debe actualizar el documento indicando el nuevo detalle de la llamada realizada.

El equipo de marketing periódicamente realiza monitoreo a los referidos que se han ingresado, es por ello que todos los días deben enviar un correo al equipo de ventas solicitando actualización sobre los referidos, el equipo de ventas al recibir esta solicitud debe realizar los contactos u obtener la información más reciente de su base de datos y enviarla al equipo de marketing.

En este momento, el equipo de marketing es el encargado de realizar el proceso de la ejecución de recompensas. Es por ello, que deben solicitar al equipo de ventas un reporte de las ventas realizadas en el último periodo (desde la última ejecución de recompensas), el equipo de ventas recibe la solicitud y procede a generar el reporte que contiene la información de los clientes nuevos y la compra realizada

(esta puede ser una recarga telefónica o un plan pospago adquirido), una vez generado el reporte se revisa y se procede a enviar los datos al equipo de marketing.

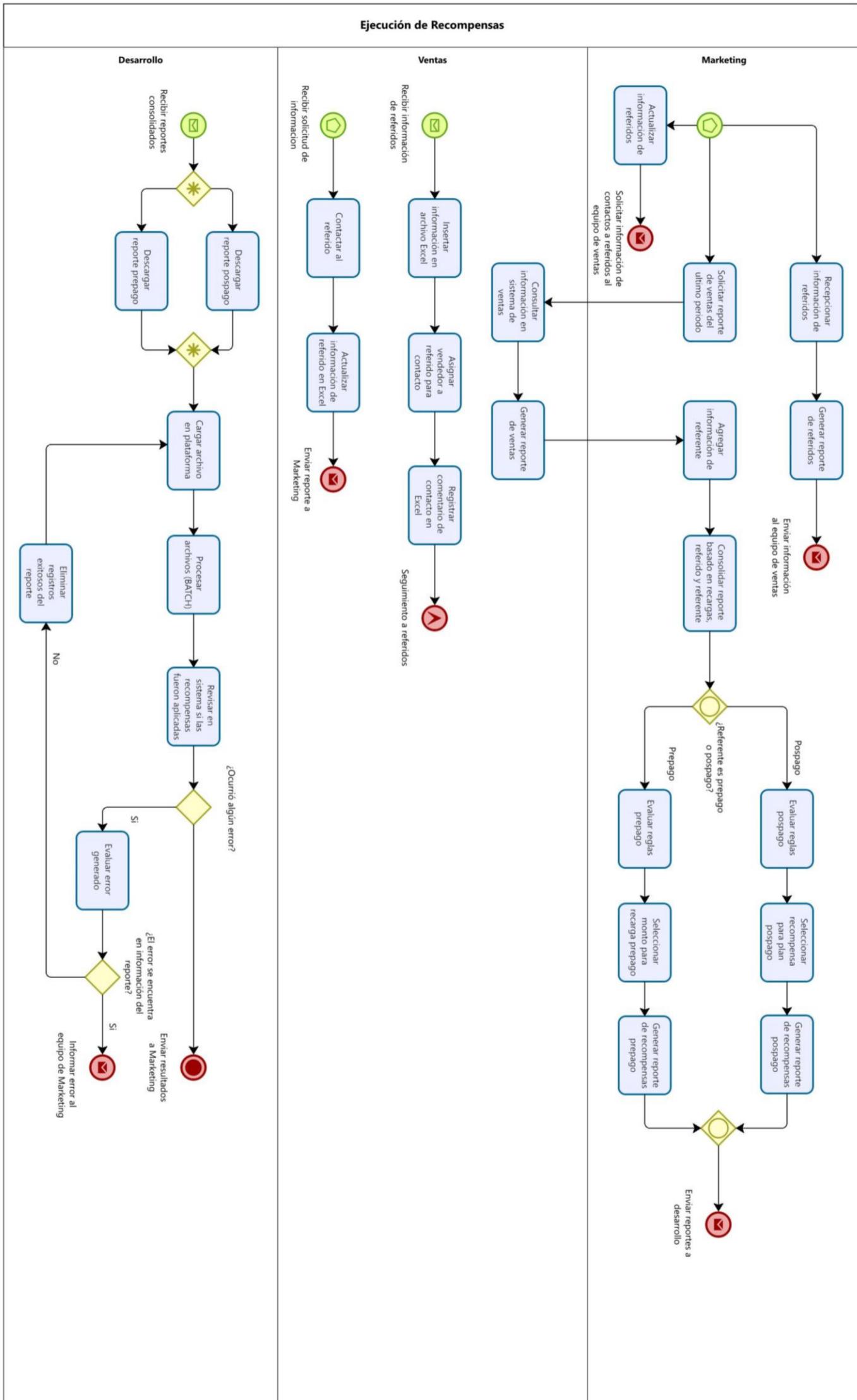
El equipo de marketing recibe el reporte de ventas y agrega los datos del referente de cada cliente en el reporte, dado que es un reporte que podría contener más de ocho días es necesario realizar una consolidación de los datos, esta consolidación se realiza en base al cliente, tipo de línea adquirida y el referente, posterior se procede a evaluar las reglas de cada tipo de recompensa y se asigna la recompensa que recibirá el referente, como paso final se debe segmentar la información en dos reportes basados en el tipo de referente (pospago o prepago). Estos reportes deben contener los datos del referente, referido y recompensa a aplicar para enviarla al equipo de desarrollo.

El equipo de desarrollo recibe a través del correo electrónico los reportes de recompensas pospago y prepago, y procede a descargarlos para posteriormente cargarlos en la plataforma de Datafone e indica el tipo de archivo, basados en el tipo de archivo el sistema procede a realizar las recargas correspondientes o bien debe abonar crédito a los planes pospago definidos en el reporte. Una vez que el archivo es procesado, el sistema retorna un listado de mensajes (que podrían ser exitosos o indican error) estos hacen correspondencia a cada registro del archivo, por lo que, si el error es relacionado a algún tipo controlado por el equipo de desarrollo, estos proceden a generar un nuevo archivo y reprocesar esos registros; En caso, que el error se encuentre en los datos proveídos por el equipo de marketing se procede a generar un reporte y enviarlo al equipo de marketing.

El equipo de marketing recibe el reporte con los errores y procede a realizar las correcciones a los datos, para posteriormente enviarlo nuevamente al equipo de desarrollo.

Ver diagrama BPMN del proceso de ejecución de recompensas en ilustración 2.

Ilustración 2: Diagrama BPMN – Proceso de ejecución de recompensas



Fuente: Elaboración propia – Utilizando Bizagi

1.3. Objetivos del sistema

Los objetivos de un sistema son comúnmente conocidos como propósitos, finalidades, misiones o visiones y estos ayudan a definir el funcionamiento de un sistema. Los objetivos identificados en el sistema de gestión para la fidelización de clientes son los siguientes:

OBJ-01: Administrar los referidos

OBJ-02: Administrar los contactos

OBJ-03: Administrar las recompensas

OBJ-04: Administrar los motivos de declinación

OBJ-05: Sincronizar la información con la plataforma principal de Datafone

OBJ-06: Mostrar los datos correspondientes

OBJ-07: Ejecutar las recompensas una vez por semana

OBJ-08: Garantizar accesos seguros al sistema web

OBJ-09: Garantizar la disponibilidad del sitio web

OBJ-10: Notificar el uso de la aplicación

OBJ-11: Facilitar el mantenimiento de la aplicación utilizando tecnologías conocidas por el equipo de desarrollo

OBJ-12: Facilitar uso de la aplicación

1.4. Alcance del sistema

El alcance del sistema de información para la gestión de fidelización de clientes incluye la administración de referidos, contactos, motivos de declinación y recompensas, así como aspectos relacionados a la seguridad como el uso de JWT y un segundo factor de autenticación. Adicionalmente, se deben ejecutar procesos calendarizados para la sincronización de información y ejecución del proceso de recompensas. El alojamiento del sistema debe realizarse en la nube, por lo que es necesario que el código y la base de datos se desplieguen en elementos de Azure.

1.5. Beneficios del sistema

Los beneficios de un sistema web pueden ser de tipo tangibles e intangibles, los beneficios tangibles son usualmente medidos en dinero, y están típicamente asociados a un incremento directo de ingresos o un decrecimiento de costos; los beneficios intangibles son más difíciles de medir, y no pueden ser directamente traducidos a dinero. Los beneficios identificados en el sistema de gestión para la fidelización de clientes son los siguientes:

Beneficios tangibles:

BT-01: Incremento en ventas

BT-02: Incremento en el cumplimiento de recompensas

BT-03: Incremento en la participación en el mercado

BT-04: Reducción de tiempo en las interacciones entre diferentes departamentos

Beneficios intangibles:

BI-01: Mejora en el control de referidos

BI-02: Mejora en el proceso de recompensas

BI-03: Incremento en la satisfacción de los referentes (clientes o vendedores)

BI-04: Mejora en el contacto hacia los referidos

BI-05: Ayudará a medir las razones por las que los clientes rechazan unirse a la compañía

1.6. Personal o actores involucrados

El proceso de desarrollo de software requiere de un número de personas que contribuyen para completar cada una de las etapas, para el presente proyecto las personas involucradas son:

- **Gerente de desarrollo:** Es la persona responsable de proveer la información relacionada a los servicios externos, guiar y coordinar las reuniones durante

las etapas de análisis, gestionar la aprobación del análisis y diseño realizado, así como también debe realizar revisiones periódicas al trabajo realizado

- **Gerente de ventas:** Es la persona responsable de proveer la información relacionada al proceso deseado para el registro de clientes, contactos y motivos de declinación, establecer prioridades sobre los requerimientos, así como también debe establecer la frecuencia para la sincronización de datos.
- **Supervisor de ventas:** Debe dar seguimiento continuo a la priorización establecida por el gerente de ventas y debe certificar que los procesos desarrollados.
- **Gerente de marketing:** Es la persona responsable de proveer la información relacionada a la configuración de recompensas, negociar prioridades de requerimientos y también debe establecer la frecuencia para el proceso de recompensas
- **Supervisor de marketing:** Es la persona responsable de proveer la paleta de colores de la empresa, logotipos, banners, etc. Debe dar seguimiento continuo a la priorización establecida por el gerente de marketing y debe certificar que los procesos desarrollados.
- **Analista de software:** Es la persona encargada de reunirse con el equipo de negocio para establecer los requerimientos y el diseño de la aplicación, modelar ambas etapas y realizar revisiones continuas de los modelos.
- **Desarrollador de software:** Es la persona responsable de construir el sistema

1.7. Definición de usuarios

Usuario: Es una entidad abstracta que puede tomar un nombre, nombre de usuario, correo electrónico, contraseña y rol únicos. Basados en el rol se determinará el nivel de acceso al sistema web. Los diferentes roles son los siguientes:

- **Administrador:** Este tipo de rol tendrá acceso a todas las opciones disponibles. Este podrá ver la toda la información registrada incluyendo la privada para distribuidores y vendedores.

- **Distribuidor:** Este tipo de rol tendrá acceso únicamente a las pestañas de reporte, referidos, motivos de declinación, contactos y cierre de sesión. En los casos de referidos y contactos, podrá ver la información registrada de cada uno de sus vendedores.
- **Marketing:** Este tipo de rol tendrá acceso únicamente a las pestañas de reporte, recompensas y cerrar sesión. Este acceso es habilitado únicamente para visualizar la información resumida y gestionar las recompensas.
- **Vendedor:** Este tipo de rol podrá gestionar los referidos y gestionar los contactos, es por ello que solo tendrá acceso a las pestañas referidos, contactos y cerrar sesión.

Sistema: Este es un actor que será utilizado para realizar las funciones internas necesarias para el funcionamiento como por ejemplo crear conexiones a la base de datos, envío de correo electrónico, crear conexiones a SFTP, ejecutar los procesos calendarizados, entre otros.

1.8. Requerimientos de negocio

Los requerimientos de negocio son utilizados para establecer los objetivos del proyecto, deben explicar lo que el usuario o cliente espera que el sistema realice. Los objetivos deben ser claros y verificables.

RN-01: Gestionar referidos.

RN-02: Gestionar contactos.

RN-03: Gestionar motivos de declinación.

RN-04: Gestionar configuración de recompensas.

RN-05: Mostrar indicadores en reportes.

RN-06: Cambiar visibilidad de los referidos

RN-07: Actualizar estado de los referidos

RN-08: Sincronizar referidos nuevos con la plataforma de Datafone

RN-09: Enviar reporte de vendedores a distribuidores

RN-10: Ejecutar proceso de recompensas

1.9. Requerimientos funcionales

Los requerimientos funcionales detallan los servicios que un sistema debe cumplir, el comportamiento del sistema cuando se realizan entradas particulares y comportamientos en situaciones específicas.

RF-01: Iniciar sesión con doble autenticación

RF-02: Recuperar contraseña

RF-03: Cargar menu de navegación basado en roles

RF-04: Crear conexión a la base de datos

RF-05: Crear conexión a SFTP

RF-06: Enviar correo electrónico

RF-07: Mostrar nombre de usuario y correo electrónico del usuario

RF-08: Mostrar los referidos basados en el rol

RF-09: Cerrar sesión

RF-10: Consumir servicios externos para recargar

RF-11: Consumir servicios externos para abonar a un plan

RF-12: Recordar usuario

RF-13: Mostrar cantidad de registros en las pestañas

1.10. Requerimientos no funcionales

Los requerimientos no funcionales son limitaciones sobre las funciones que el sistema debe cumplir, se suelen aplicar al sistema como un todo, más que a características o funciones individuales del sistema.

RNF-01: Guardar registros de los archivos descargados en base de datos

RNF-02: BackEnd se debe realizar utilizando patrón de diseño CQRS

RNF-03: FrontEnd se debe realizar con Angular

RNF-04: BackEnd se debe realizar con .NET 7

RNF-05: Reglas de seguridad para establecer una contraseña

RNF-06: Cifrado de contraseña

RNF-07: Plataforma web debe poder utilizarse en cualquier dispositivo

RNF-08: Servicios externos serán de tipo WCF

RNF-09: La aplicación se desplegará en una App Service de Azure

1.11. Requerimientos de implementación

Los requerimientos de implementación son características temporales que son necesarias para la implementación o transición de un sistema, pero que en el futuro estos serán eliminados.

RI-01: Crear manual de usuario

RI-02: Crear la cuenta en Azure

RI-03: Crear las subscripciones de QA y Producción en Azure

RI-04: Crear Azure SQL Database

RI-05: Crear App Service

RI-06: Desplegar WebJobs

1.12. Requerimientos de usuario

Los requerimientos de usuario describen cómo el proyecto será utilizado por los eventuales usuarios finales.

RU-01: Establecer estado por defecto a referidos

RU-02: Mostrar restricciones de UI en recompensas

1.13. Plantillas REM

La matriz de trazabilidad. Las plantillas REM (Requirements Management) son una herramienta para la gestión de requisitos diseñada para soportar la etapa de ingeniería de requerimientos.

Tabla 4: Plantilla REM RN-01

RN-01	Gestionar referidos
Versión	1.0 (29/07/2023)
Autores	Cristopher Josue Zepeda Tellez; Gloria Esther Mayorga Mena; Magaly Magdalena Mayorga Maradiaga
Fuentes	Ventas y Marketing (Gerente y Supervisor)
Descripción	El sistema debe estar habilitado para realizar las actividades de mostrar la información de los referidos previamente guardados, insertar nuevos referidos, actualizar la información de cualquier referido, eliminar referidos y también debe tener la capacidad de filtrar los registros por nombre y/o correo electrónico.
Importancia	Vital
Urgencia	Inmediata
Comentarios	NA

Fuente: Elaboración propia

Tabla 5: Plantilla REM RN-02

RN-02	Gestionar contactos
Versión	1.0 (29/07/2023)
Autores	Cristopher Josue Zepeda Tellez; Gloria Esther Mayorga Mena; Magaly Magdalena Mayorga Maradiaga
Fuentes	Ventas y Marketing (Gerente y Supervisor)
Descripción	El sistema debe estar habilitado para realizar las actividades de mostrar la información de los contactos previamente guardados, insertar nuevos contactos, actualizar la información de cualquier contacto, eliminar contactos y también debe tener la capacidad de filtrar los registros por referido.
Importancia	Vital
Urgencia	Necesario
Comentarios	NA

Fuente: Elaboración propia

Tabla 6: Plantilla REM RN-03

RN-03	Gestionar motivos de declinación
Versión	1.0 (29/07/2023)
Autores	Cristopher Josue Zepeda Tellez; Gloria Esther Mayorga Mena; Magaly Magdalena Mayorga Maradiaga
Fuentes	Ventas y Marketing (Gerente y Supervisor)
Descripción	El sistema debe estar habilitado para realizar las actividades de mostrar los motivos de declinación previamente guardados, insertar nuevos motivos de declinación, actualizar los motivos de declinación, eliminar registros y también debe tener la capacidad de filtrar los registros por nombre del motivo de declinación.
Importancia	Importante
Urgencia	Necesario
Comentarios	NA

Fuente: Elaboración propia

Ver continuación de plantillas [REM en Anexo](#)[Tabla1: Plantillas REM](#)

1.14. Matriz de trazabilidad

La matriz de trazabilidad es una herramienta que se utiliza para relacionar cada uno de los requerimientos con el entregable que se haya solicitado, esta tabla es en doble sentido; Permite identificar qué objetivo se alcanza a través de cada requerimiento, y a la vez, permite conocer que requerimientos son necesarios para obtener un objetivo. Ver tabla #7.

Tabla 7: Matriz de trazabilidad

	OBJ-01	OBJ-02	OBJ-03	OBJ-04	OBJ-05	OBJ-06	OBJ-07	OBJ-08	OBJ-09	OBJ-10	OBJ-11	OBJ-12
RN-01	X											
RN-02		X										
RN-03				X								
RN-04			X									
RN-05						X						
RN-06												X
RN-07					X							
RN-08					X							
RN-09										X		
RN-10							X					
RF-01								X				
RF-02								X				
RF-03							X					
RF-04	X	X	X	X	X	X	X	X		X		
RF-05					X							
RF-06										X		
RF-07						X						
RF-08						X						
RF-09												X
RF-10					X							
RF-11					X							
RF-12												X
RF-13						X						
RNF-01					X						X	
RNF-02											X	
RNF-03											X	
RNF-04											X	
RNF-05								X				
RNF-06								X				
RNF-07									X			
RNF-08					X							
RNF-09									X			
RI-01												X
RI-02										X		
RI-03										X		
RI-04										X		
RI-05										X		
RI-06										X		
RU-01	X											
RU-02			X									

Fuente: Elaboración propia

Capítulo II: Diseño del sistema

El diseño del sistema web se encuentra basado en toda la información recopilada en el análisis de requerimientos. La etapa de diseño tiene como objetivo construir un modelo de lo que es requerido por el sistema, para la elaboración de los modelos se utilizará la metodología UWE que se encuentra basada UML, así como también se establecerá la arquitectura del sistema a utilizar en la etapa de construcción.

2.1. Diseño de la arquitectura del sistema

La arquitectura de un sistema de software es la forma que le dan a ese sistema quienes lo construyen. La forma del sistema esta divida en componentes, la disposición de esos componentes y como estos se comunican entre sí.

El propósito de este modelo es facilitar el desarrollo, despliegue, operación y mantenimiento del sistema de software contenido en él.

Existe una gran variedad de estilos de arquitectura a implementar en un sistema, la decisión de que arquitectura implementar puede estar basada en los requerimientos o bien basados en las inquietudes del dominio. En el caso del sistema para la fidelización de clientes, la decisión sobre que estilo de arquitectura utilizar está basado en los requerimientos.

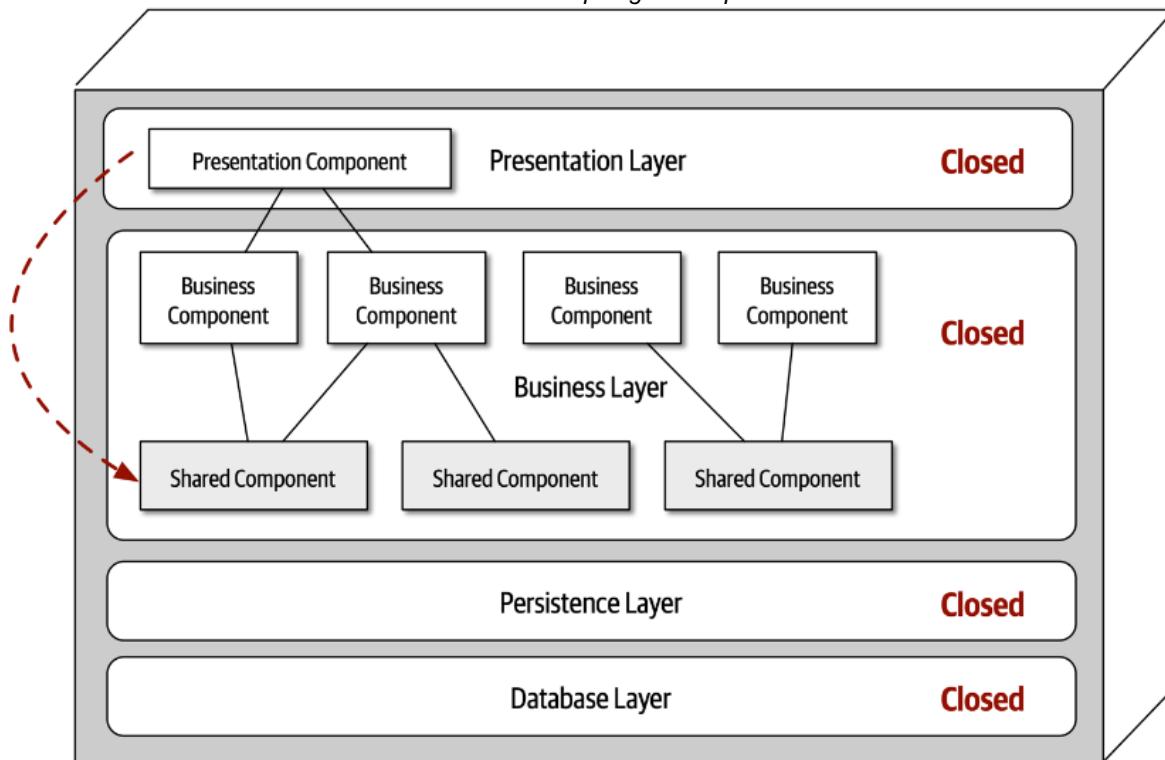
2.1.1. Estilo de arquitectura en capas

La arquitectura de capas, también conocida como n-capas, es una de los más comunes estilos de arquitectura. Es de hecho el estándar para la mayoría de aplicaciones, principalmente debido a su simplicidad, familiaridad y bajo costo.

Topología

Todos los componentes dentro del estilo de arquitectura en capas son organizados en capas horizontales y cada capa desempeña una función específica dentro de la aplicación (como lógica de presentación o lógica comercial). Aunque no existen restricciones específicas en cuanto al número y tipos de capas que deben existir, la mayoría de las arquitecturas en capas constan de cuatro capas estándar: presentación, negocio, persistencia y base de datos.

Ilustración 3: Topología N-Capas



Fuente: (Richards & Ford, 2020):

Aislamiento de capas

El aislamiento de capas consiste en que los cambios realizados en una capa de la arquitectura no impactan o afectan los componentes en otras capas. siempre que los contratos entre esas capas permanezcan sin cambios. Cada capa es independiente de las otras capas, por lo que tiene poco o ningún conocimiento del funcionamiento interno de otras capas en la arquitectura

Cada capa dentro del estilo de arquitectura puede ser cerrado o abierto. Una capa cerrada significa que una petición se mueve de arriba hacia abajo dentro de las capas, la petición no puede omitir capas.

Una capa abierta significa que una petición puede saltar capas y llegar directamente a la capa que necesita.

2.2. Patrón de diseño de la arquitectura

Un patrón de diseño de arquitectura es una solución general y reutilizable a un problema común dentro de un contexto dado. Los patrones arquitectónicos son similares al patrón de diseño de software, pero tienen un alcance más amplio.

El patrón de diseño a implementar en el sistema para la gestión de fidelización de clientes, será el Clean Architecture ideal para el desarrollo de aplicaciones modernas.

2.2.1. Clean Architecture

Es un conjunto de principios cuya finalidad principal es ocultar los detalles de implementación a la lógica de dominio de la aplicación, de esta manera se mantiene aislada la lógica, consiguiendo tener una lógica mucho más mantenible y escalable en el tiempo.

Características de clean architecture

Independencia de Frameworks: La arquitectura no depende de la existencia de alguna librería. Esto permite utilizar tanto Frameworks como herramientas sin limitar al desarrollador.

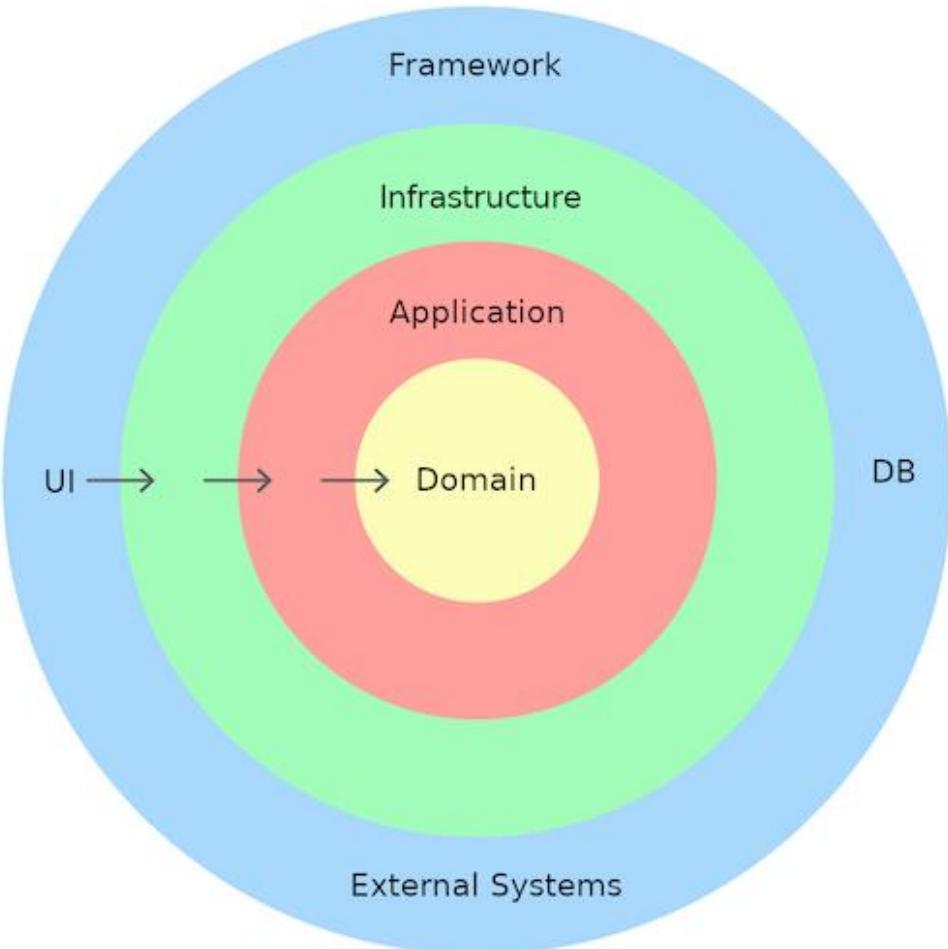
Testable: Las reglas de negocio pueden ser probadas sin la necesidad de una interfaz de usuario, base de datos, servidor web o cualquier otro elemento externo.

Independencia de UI: La interfaz de usuario puede cambiar con facilidad, sin la necesidad de cambiar el resto del sistema. Una interfaz de usuario web podría ser reemplazado con una consola, sin necesidad de cambiar las reglas de negocio.

Independencia de base de datos: Es posible realizar intercambios de bases de datos Oracle por SQL Server, MongoDB, CouchDB o cualquier otro gestor. Las reglas de negocio no están limitadas a la base de datos.

Independencia de cualquier agente externo: Las reglas de negocio no tienen conocimiento de las interfaces expuestas.

Ilustración 4: Diseño Clean Architecture



Fuente: (Nayan, 2021)

Regla de dependencia:

La regla de dependencia indica que un círculo interior nunca debe conocer nada sobre un círculo exterior. Sin embargo, los círculos exteriores si pueden conocer los círculos interiores.

Capas de clean architecture:

Domino: El dominio es el corazón de una aplicación y debe estar totalmente aislado de cualquier dependencia a la lógica o los datos de negocio.

Infrastructure: Se encarga de transformar la información (desde el modelo de petición del API hacia el modelo requerido en la base datos o servicio externo).

Application: Encapsula e implementa todas las reglas de negocio de un sistema, estos casos de uso gestionan el flujo de datos desde y hacia la base de datos.

2.3. Modelos UWE

UWE es un enfoque de la ingeniería de software orientado a las aplicaciones web y cuyo principal objetivo es cubrir todo el ciclo de vida de desarrollo de los sistemas web.

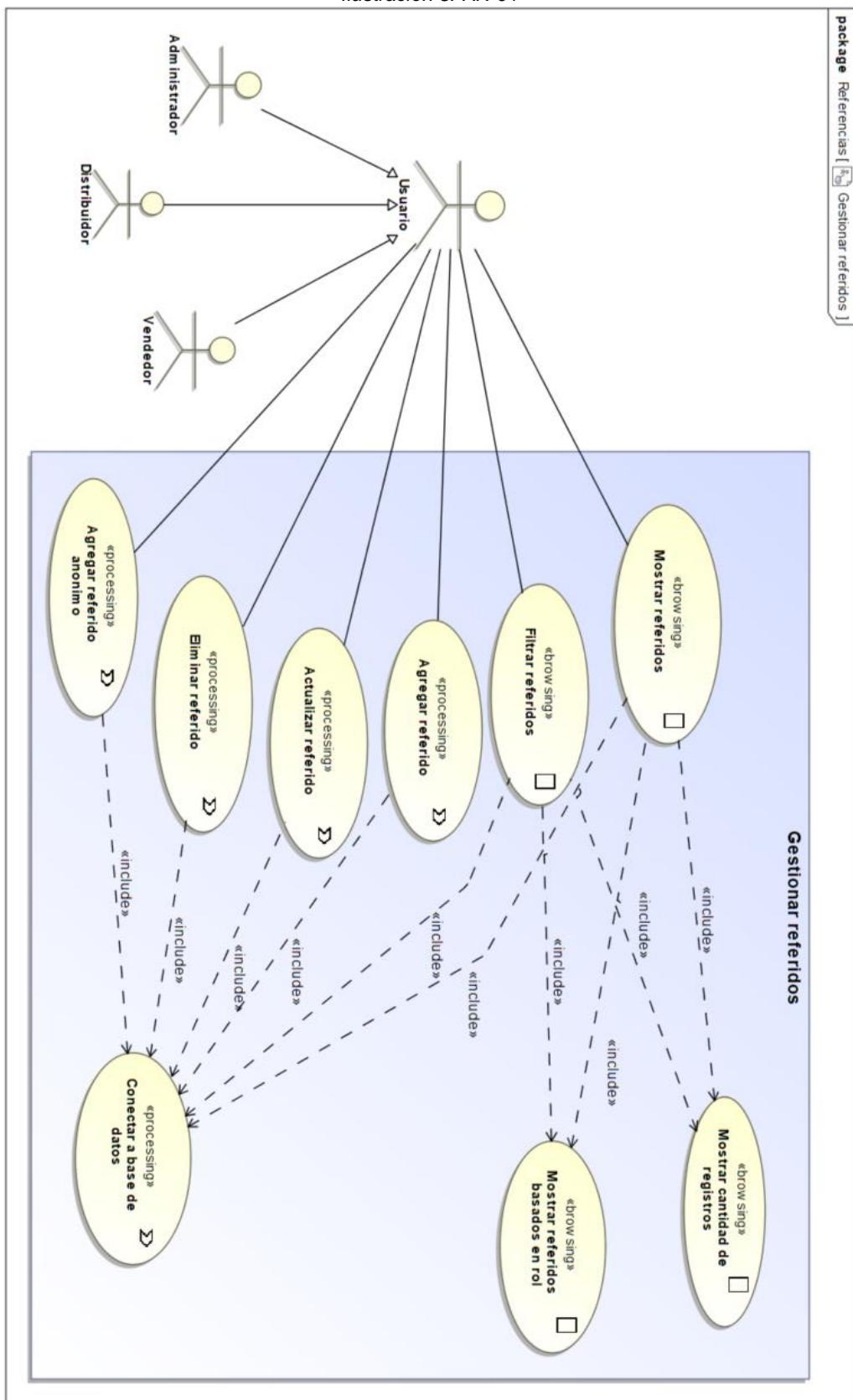
2.3.1. Modelo de requerimientos

El modelo de requerimientos consiste en dos partes, los casos de uso y los diagramas de actividades que detallan los casos de uso.

Diagrama de Casos de uso

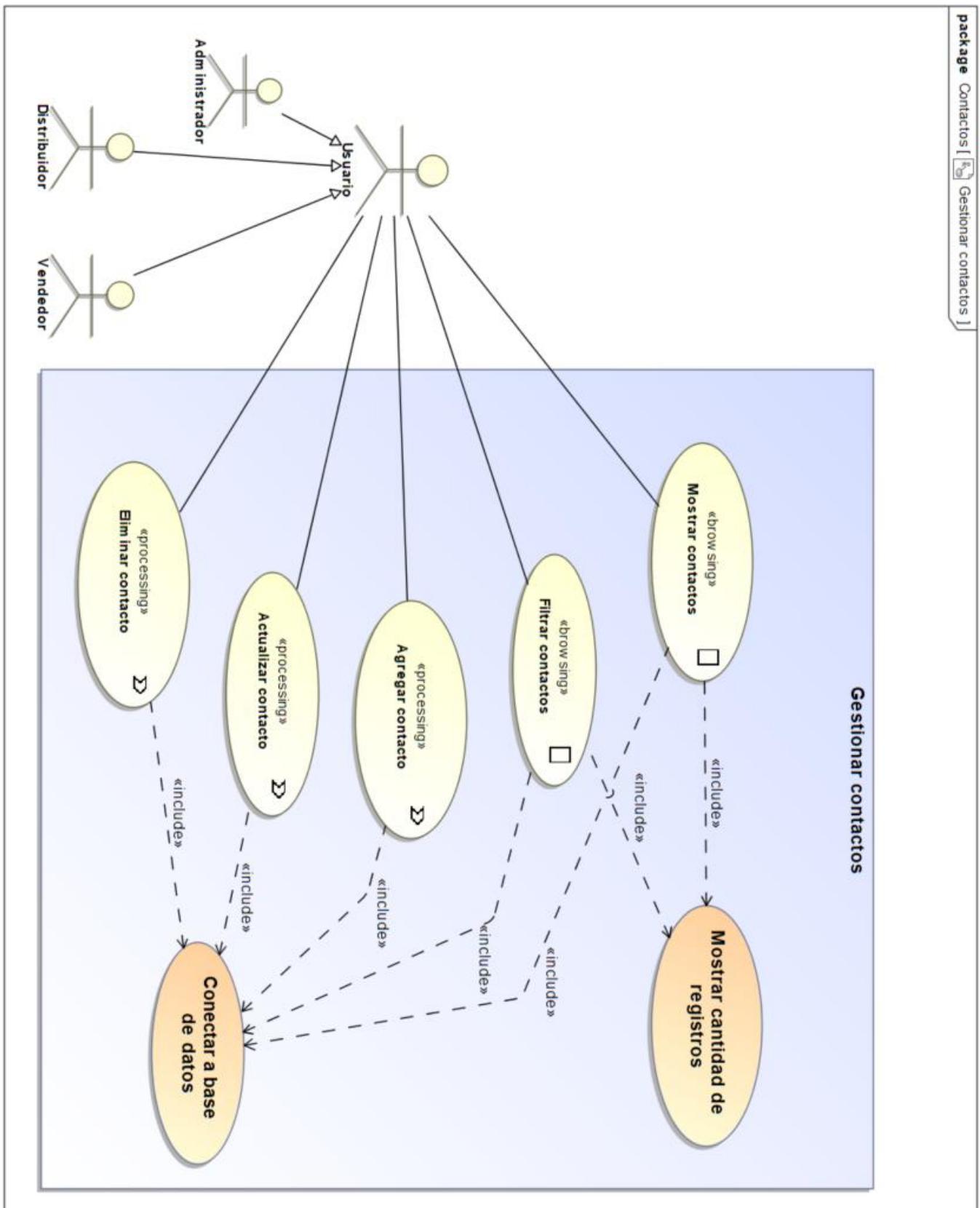
Los casos de uso definen una secuencia de acciones que dan como resultado un valor observable, en UW los casos de uso se distinguen por los estereotipos browsing y processing. En el sistema de gestión para la fidelización de clientes se definieron los siguientes casos de uso, ver ilustraciones #05 - #07

Ilustración 5: RN-01



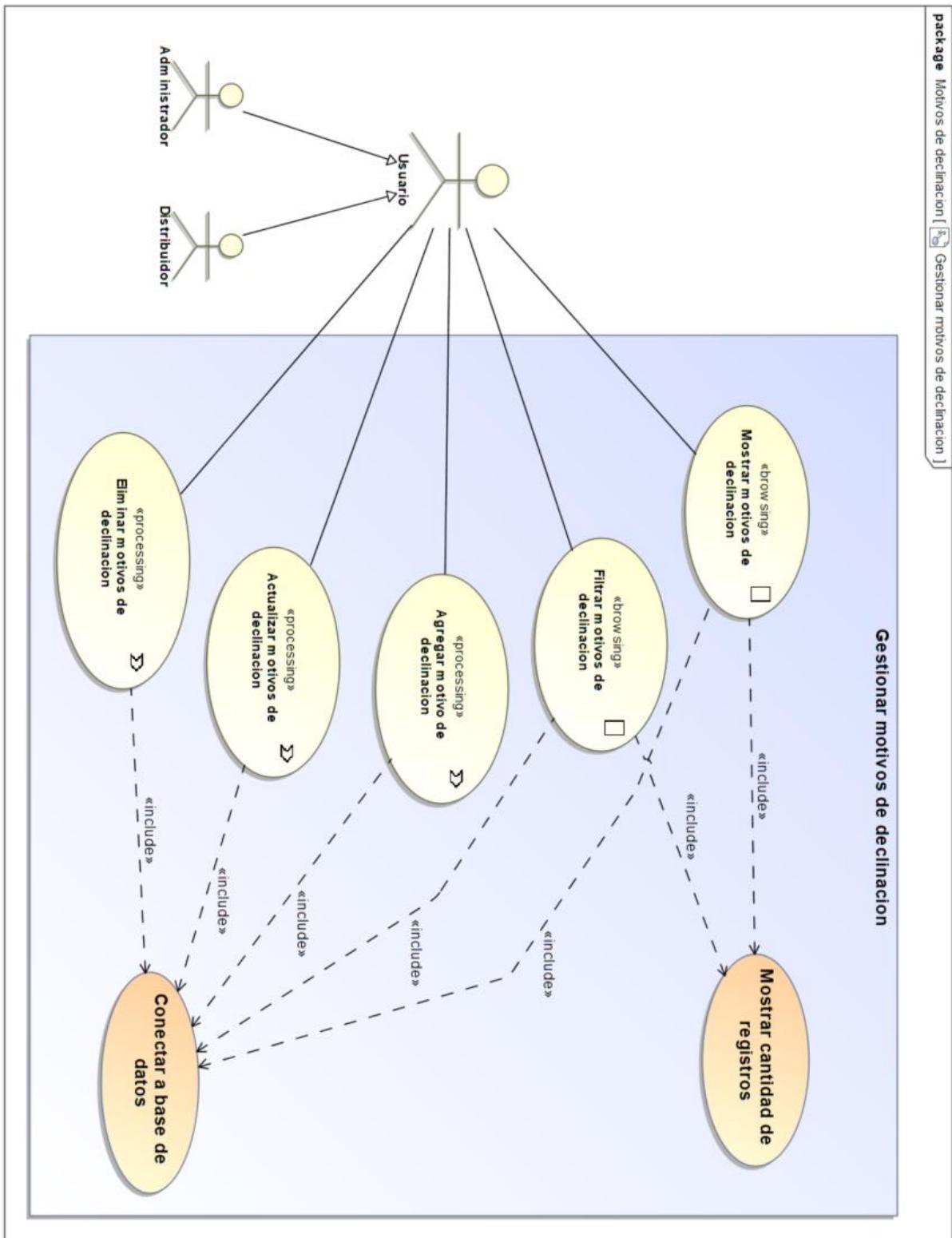
Fuente: Elaboración propia

Ilustración 6: RN-02



Fuente: Elaboración propia

Ilustración 7: RN-03



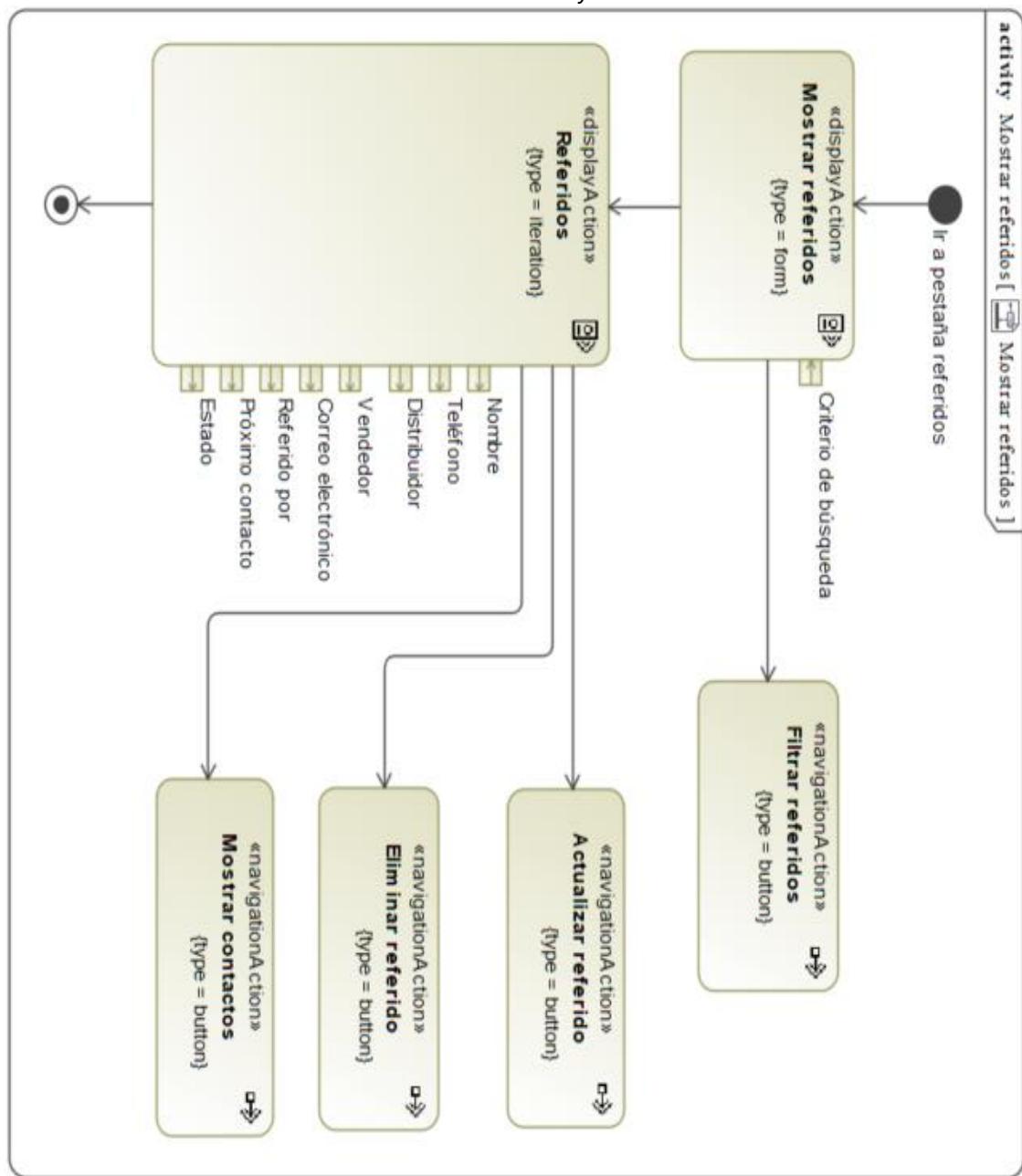
Fuente: Elaboración propia

Ver continuación de diagramas de casos de uso en [Anexo1: Ilustración 1: Modelo de requerimientos – Diagramas de casos de uso](#)

Diagrama de Actividades

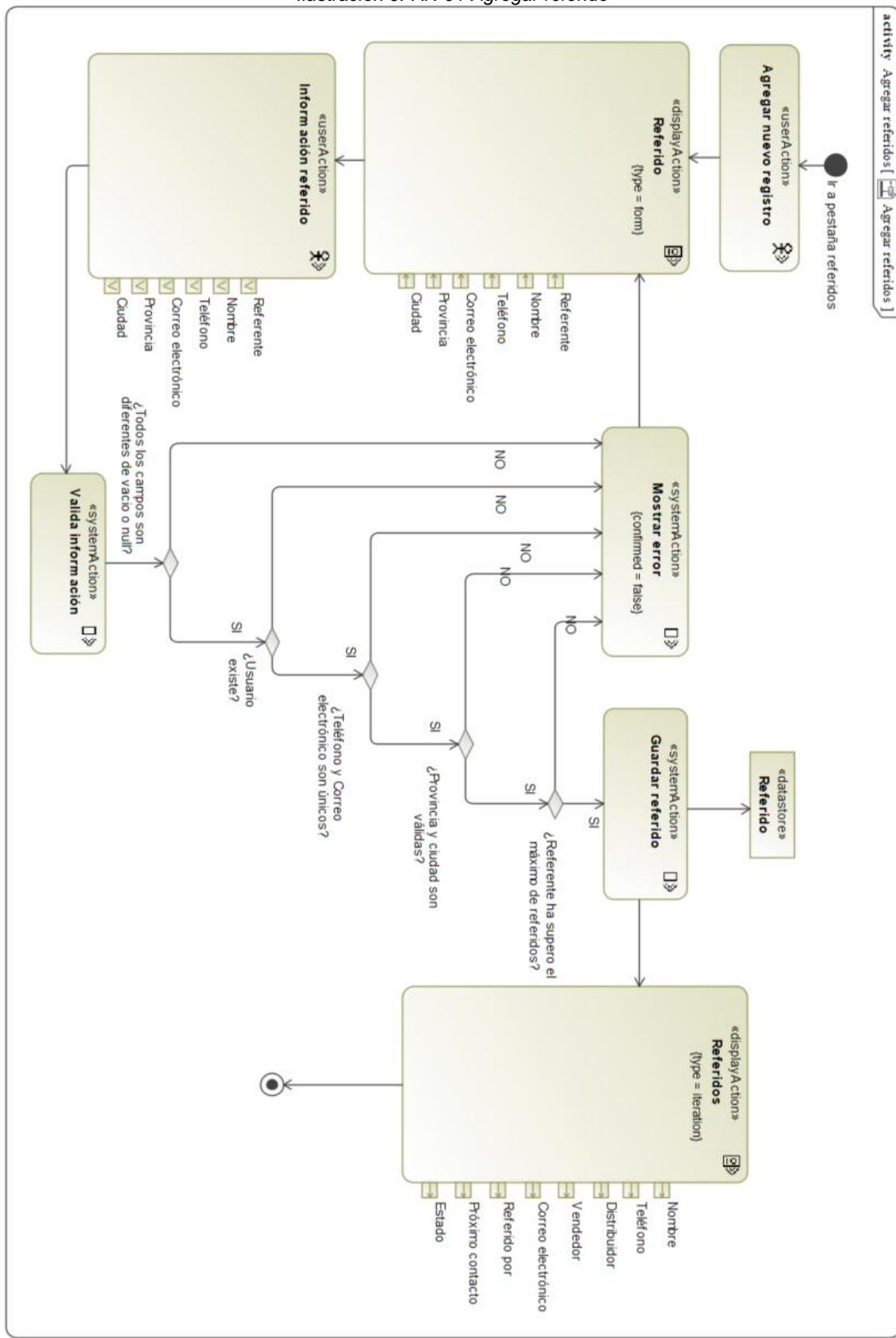
Los diagramas de actividades son un diagrama de flujo que muestra las actividades ejecutadas por un sistema, en UWE estos son un complemento de los casos de uso con la finalidad de describir detalladamente un flujo de proceso. En el sistema de gestión para la fidelización de clientes se definieron los siguientes diagramas de actividades, ver ilustraciones #08 - #10.

Ilustración 8: RN-01 Mostrar y filtrar referidos



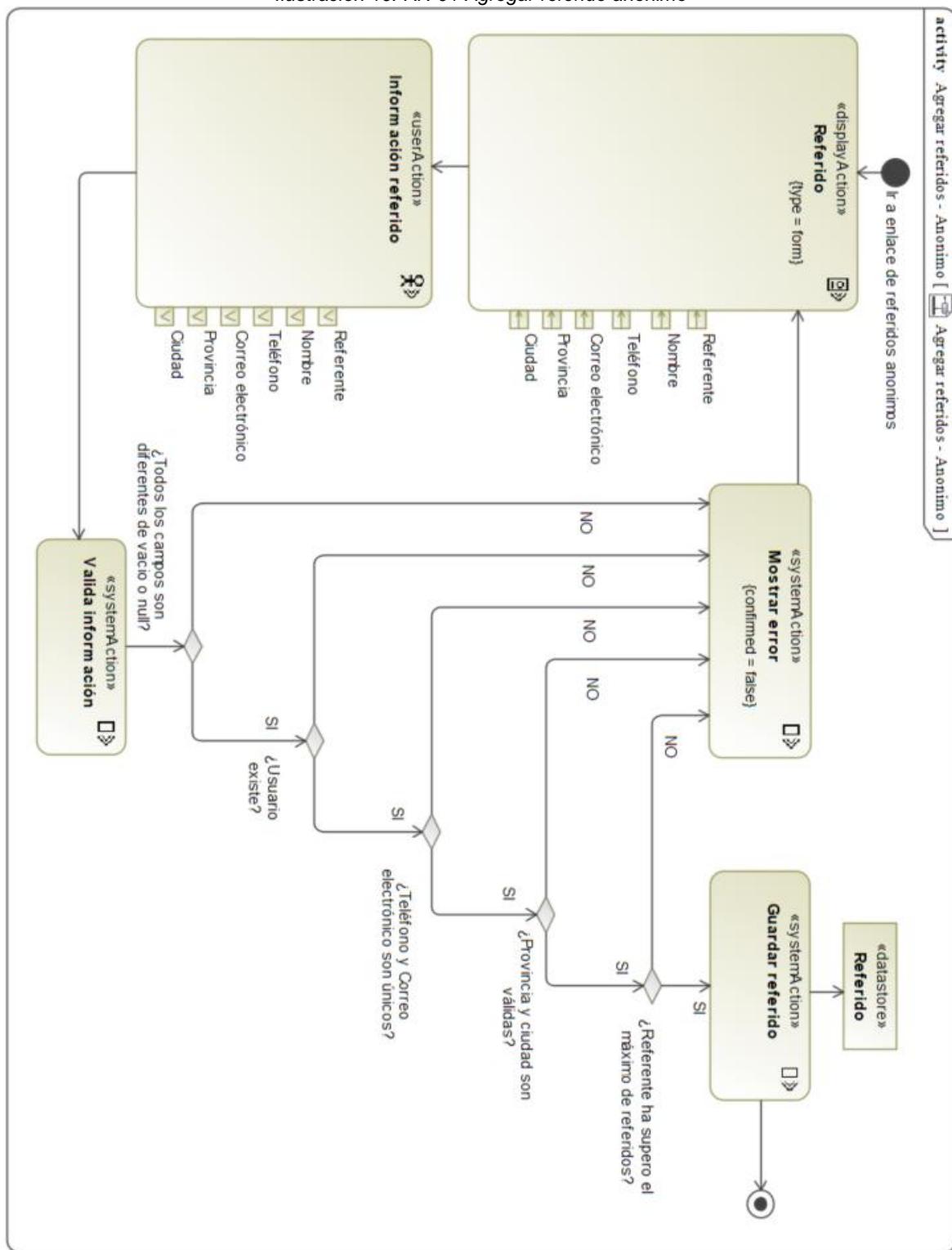
Fuente: Elaboración propia

Ilustración 9: RN-01 Agregar referido



Fuente: Elaboración propia

Ilustración 10: RN-01 Agregar referido anónimo



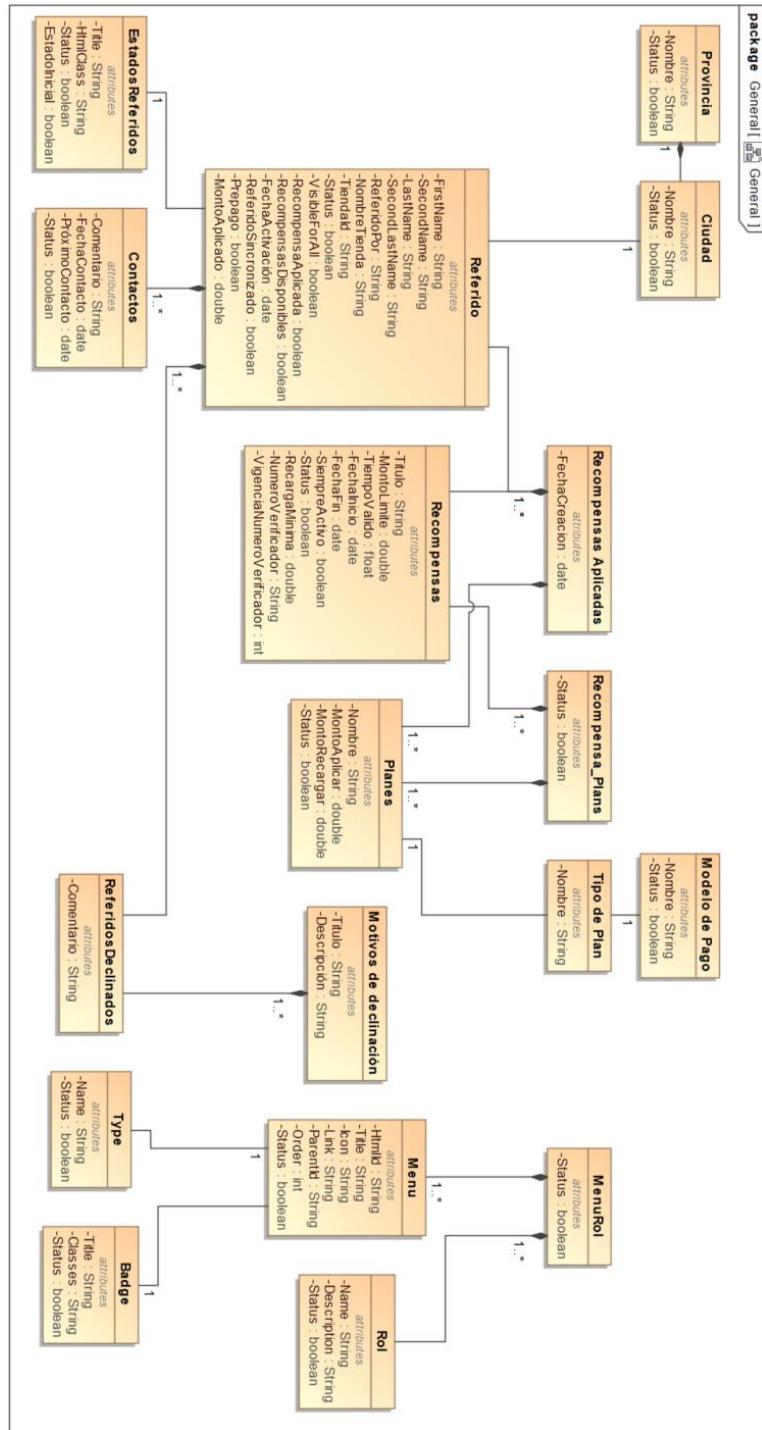
Fuente: Elaboración propia

Ver continuación de diagramas de actividades en [Anexo Ilustración 2: Modelo de requerimientos – Diagramas de actividades](#)

2.3.2. Modelo de contenido

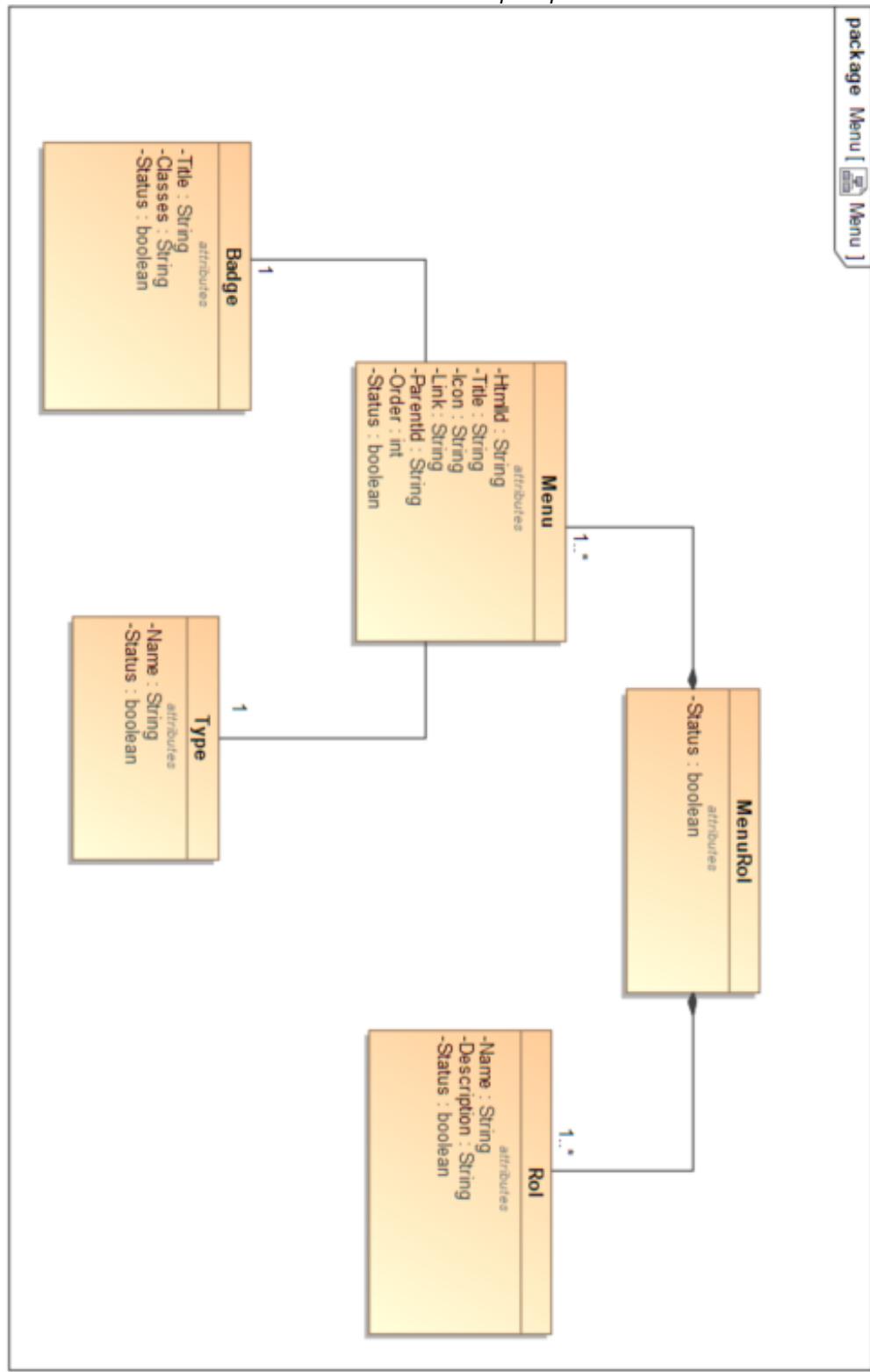
El modelo de contenido es utilizado para simbolizar las clases que son necesarias para la aplicación. En el sistema de gestión para la fidelización de clientes se definieron los siguientes modelos de contenido, ver ilustraciones #11 - #12.

Ilustración 11: General



Fuente: Elaboración propia

Ilustración 12: Menú principal



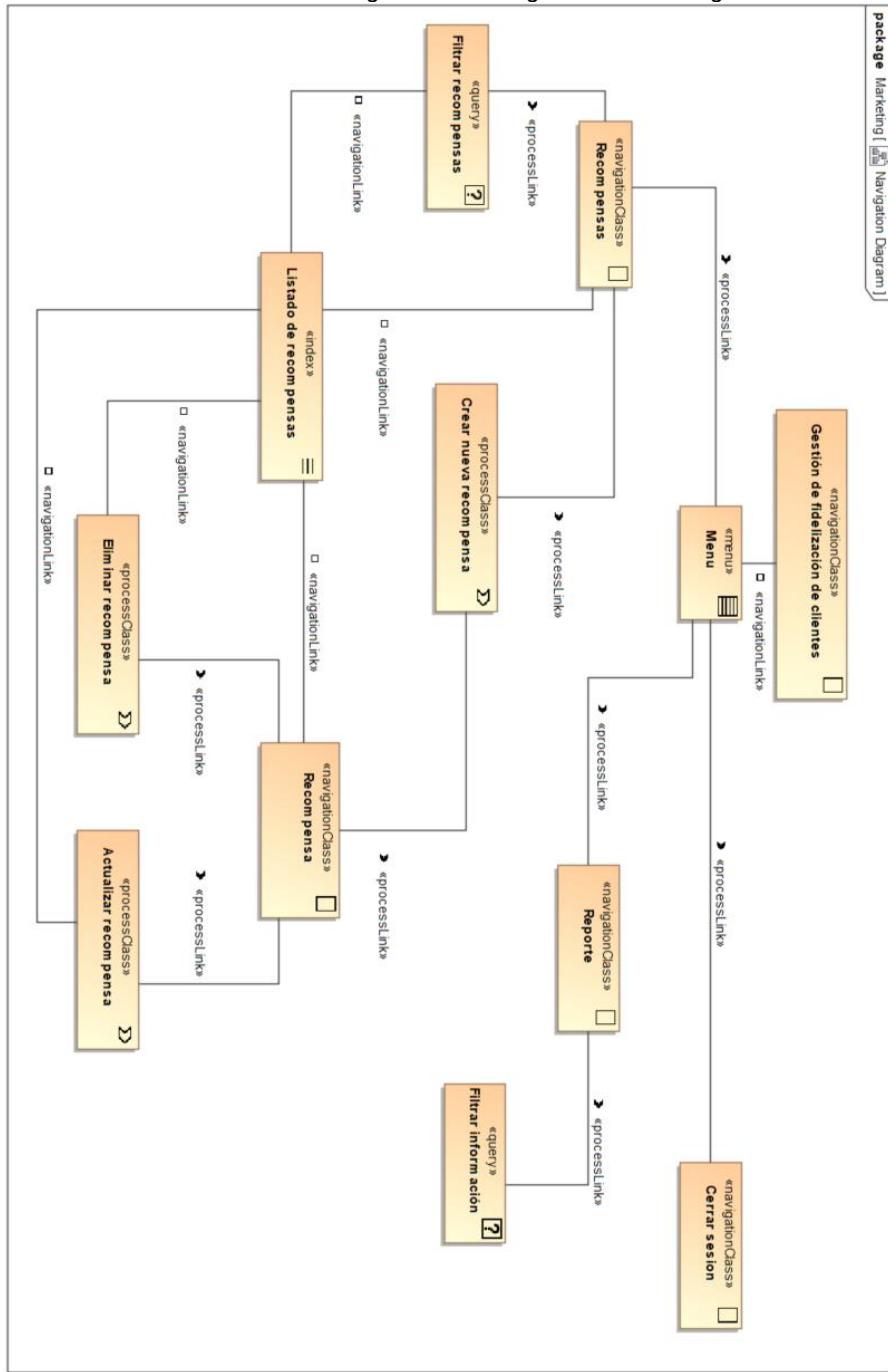
Fuente: Elaboración propia

Ver continuación de modelos de contenido en [Anexos](#)Ilustración 3: Modelo de contenido

2.3.3. Modelo de navegación

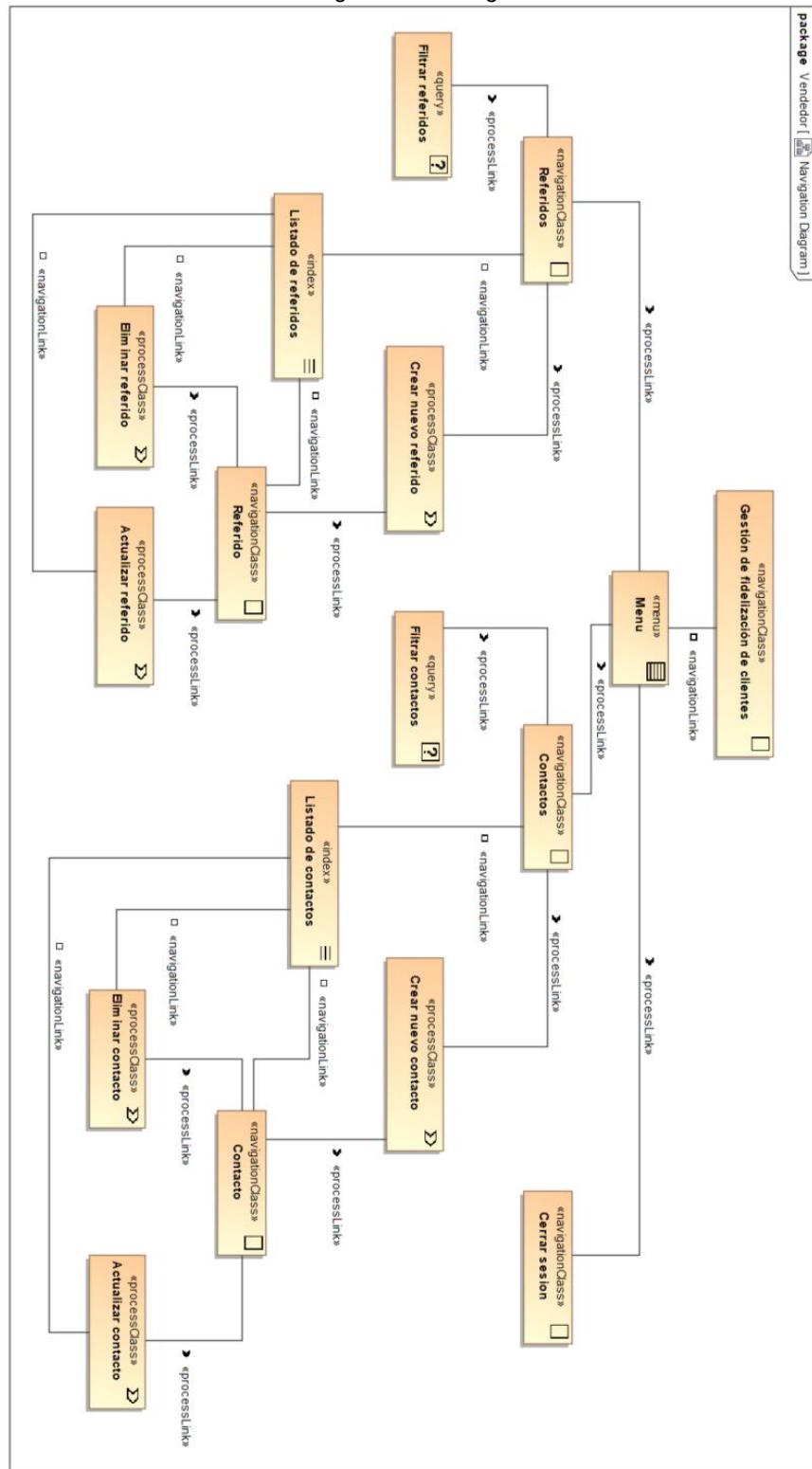
El modelo de navegación es utilizado para mostrar las relaciones entre las distintas páginas web que contiene un sistema web. En el sistema de gestión para la fidelización de clientes se definieron los siguientes modelos de navegación, ver ilustraciones #13 - #14

Ilustración 13: Diagrama de navegación - Marketing



Fuente: Elaboración propia

Ilustración 14: Diagrama de navegación - Vendedor



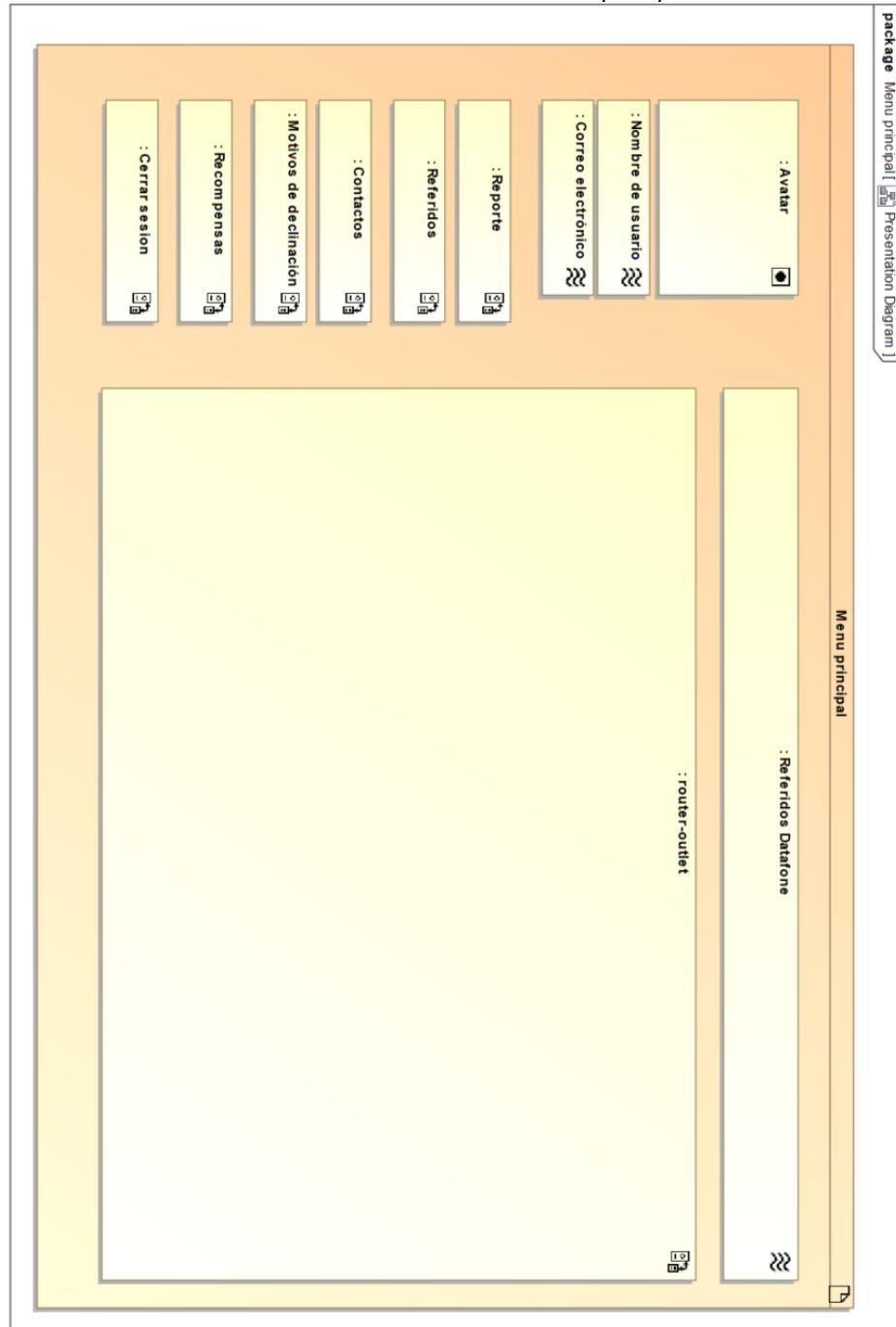
Fuente: Elaboración propia

Ver continuación de modelos de navegación en [Anexos Ilustración 4: Modelo de navegación](#)

2.3.4. Modelo de presentación

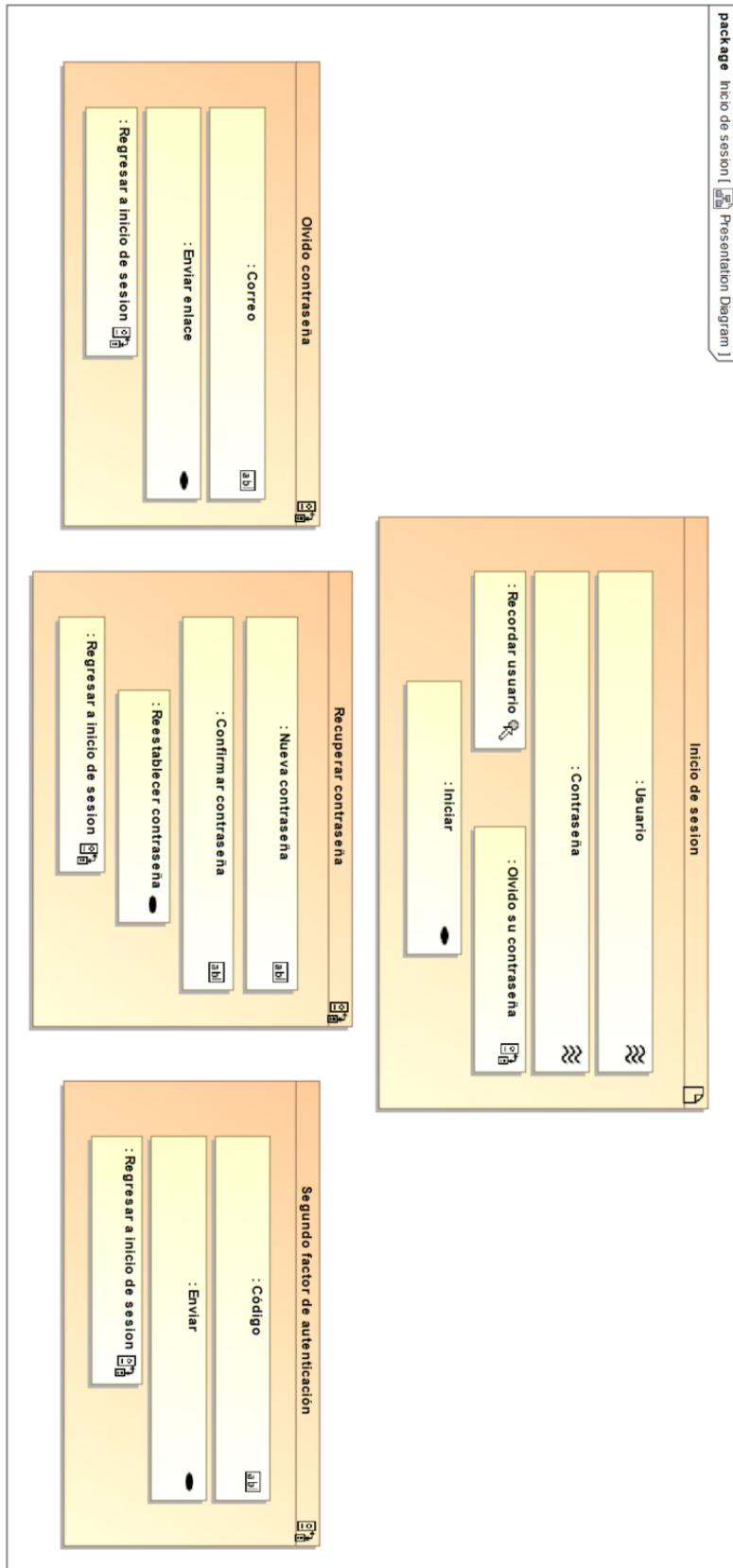
El modelo de presentación ayudará a representar de manera grafica la ubicación o pertenencia de cada clase de proceso o navegación a una determinada página web. En el sistema de gestión para la fidelización de clientes se definieron los siguientes modelos de presentación, ver ilustraciones #15 - #17.

Ilustración 15: Presentación – Menu principal



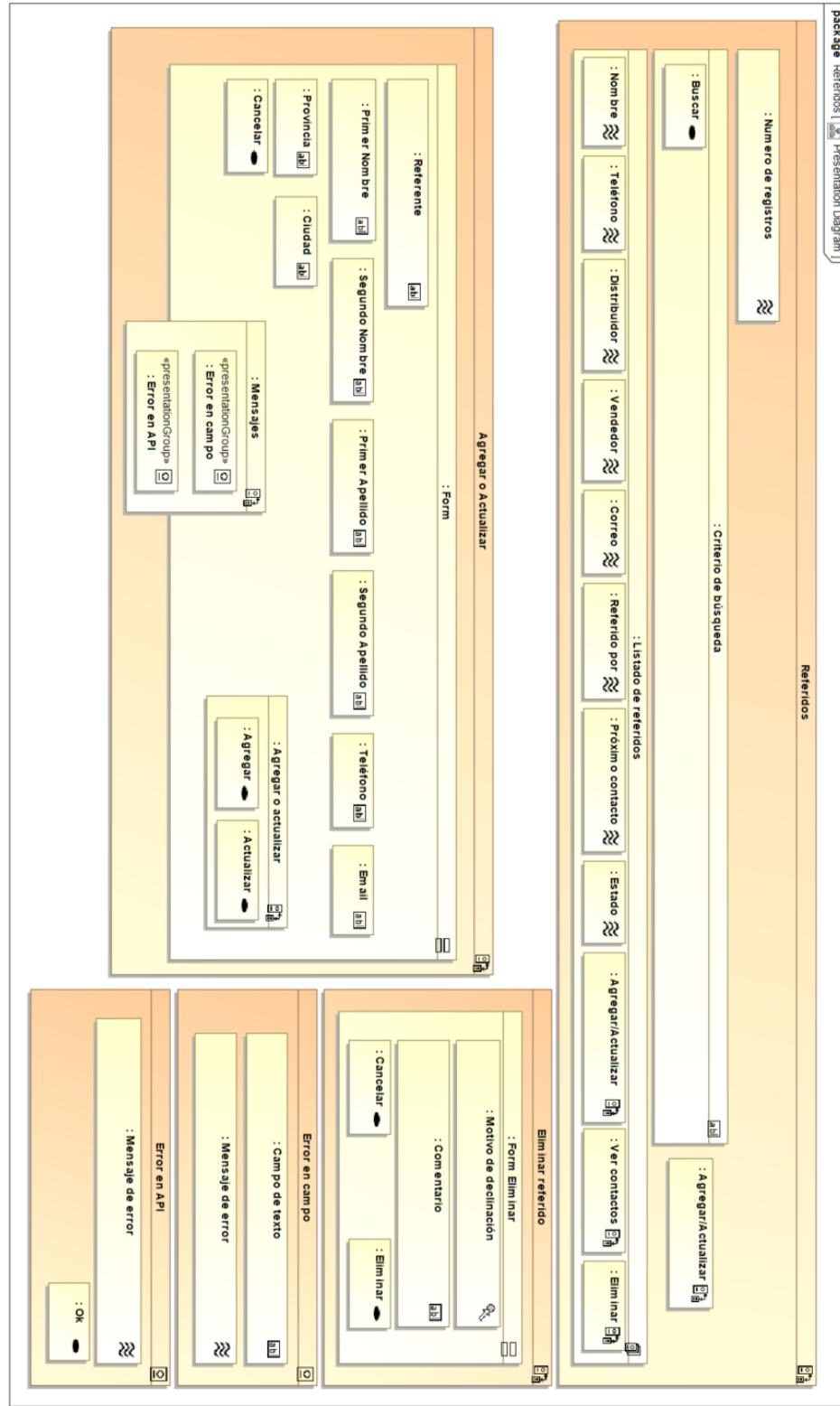
Fuente: Elaboración propia

Ilustración 16: Presentación - Inicio de sesión



Fuente: Elaboración propia

Ilustración 17: Presentación - Referidos



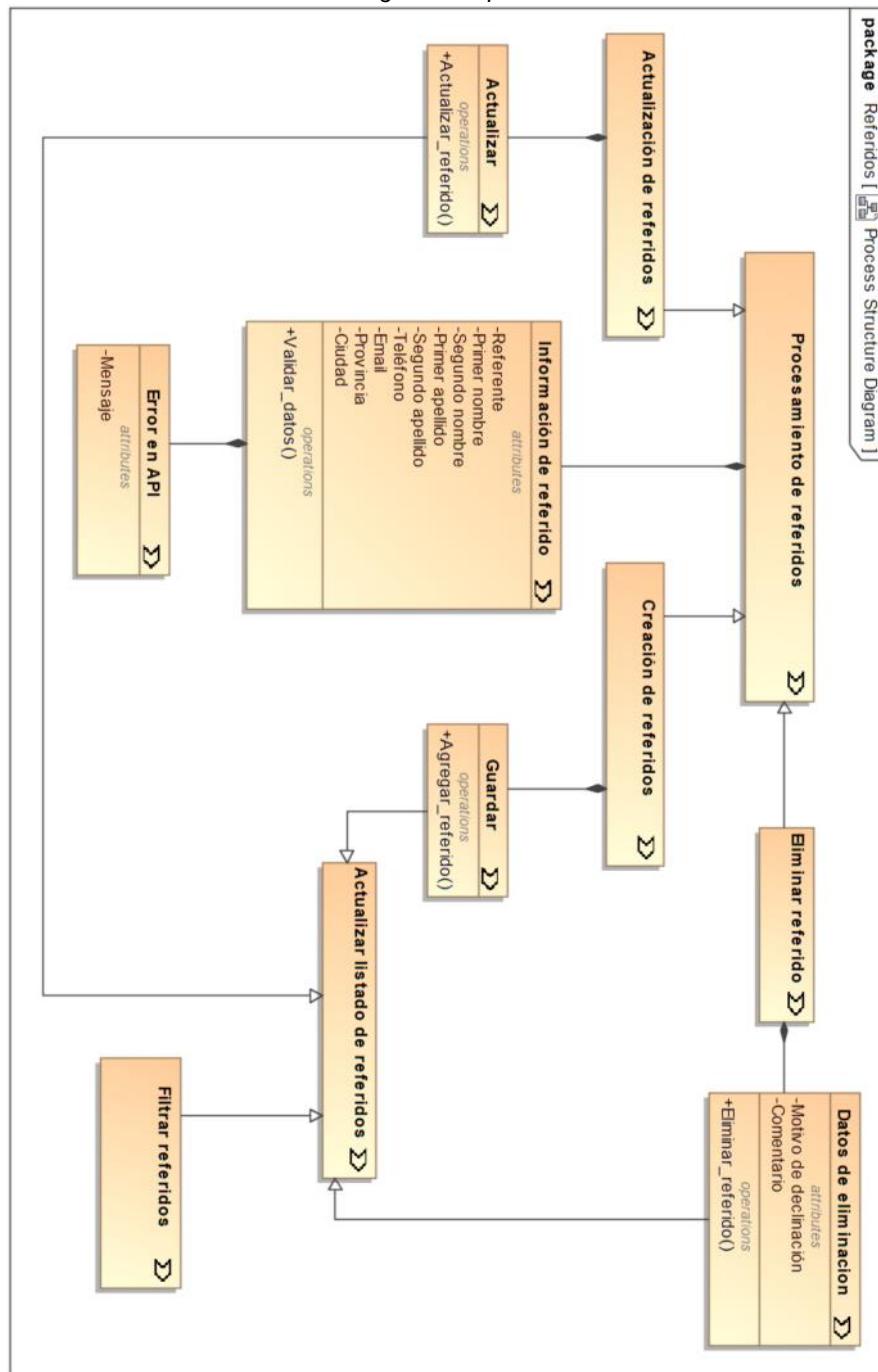
Fuente: Elaboración propia

Ver continuación de modelos de presentación en [Anexos Ilustración 5: Modelo de presentación](#)

2.3.5. Modelo de procesos

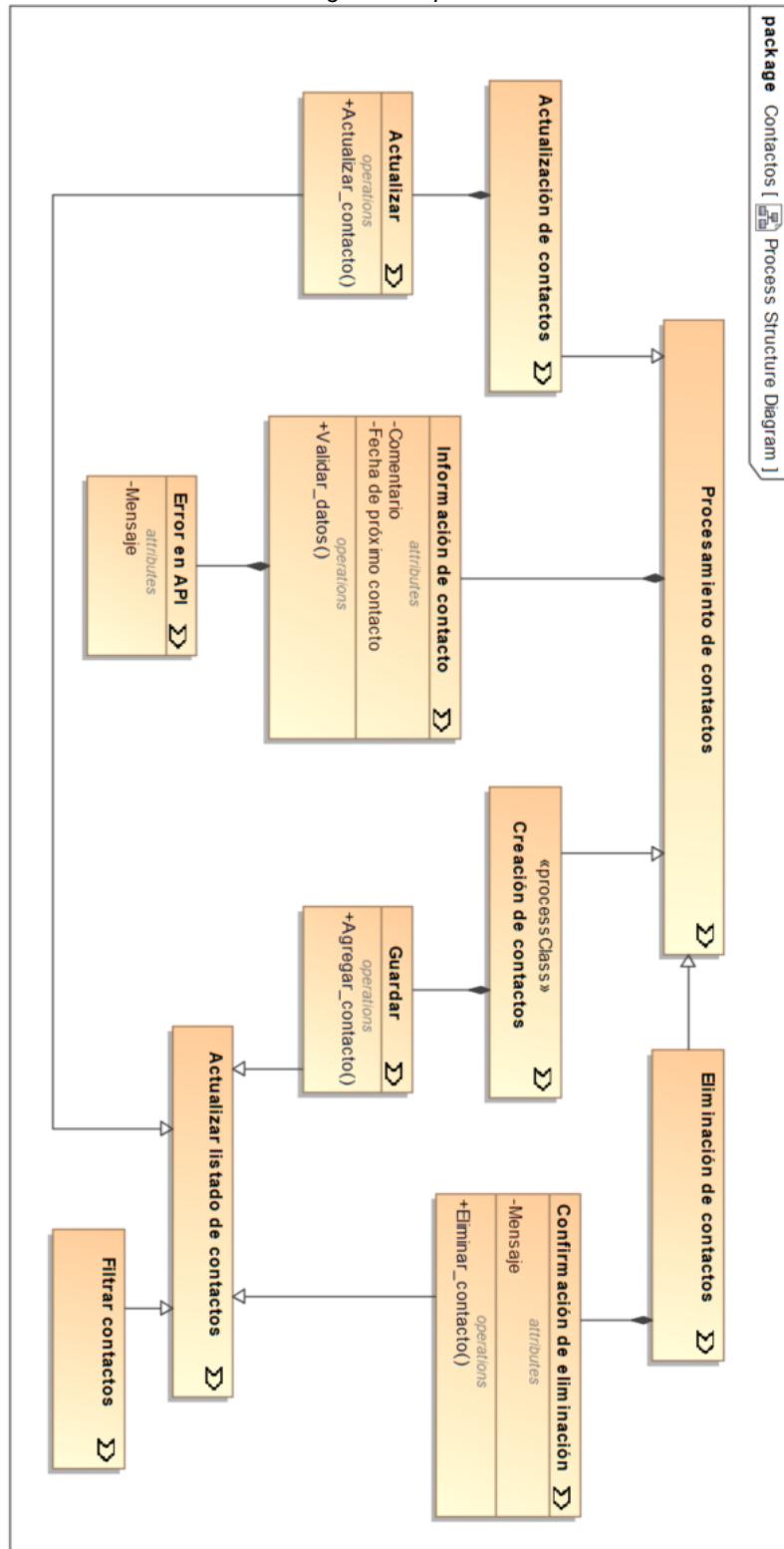
El modelo de procesos es utilizado para describir las relaciones entre las diferentes clases de procesos, este modelo está basado en el modelo de navegación. En el sistema de gestión para la fidelización de clientes se definieron los siguientes modelos de procesos, ver ilustraciones #18 - #19.

Ilustración 18: Diagrama de procesos - Referidos



Fuente: Elaboración propia

Ilustración 19: Diagrama de procesos - Contactos



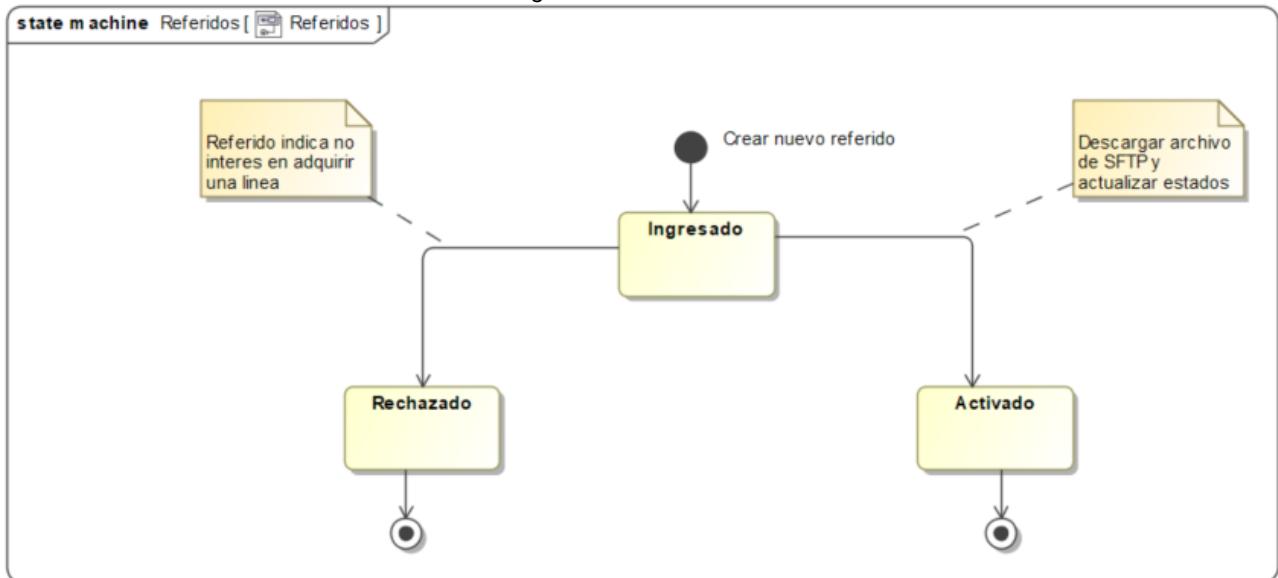
Fuente: Elaboración propia

Ver continuación de modelos de procesos en [Anexos](#) Ilustración 6: Modelo de procesos

2.4. Diagrama de estados

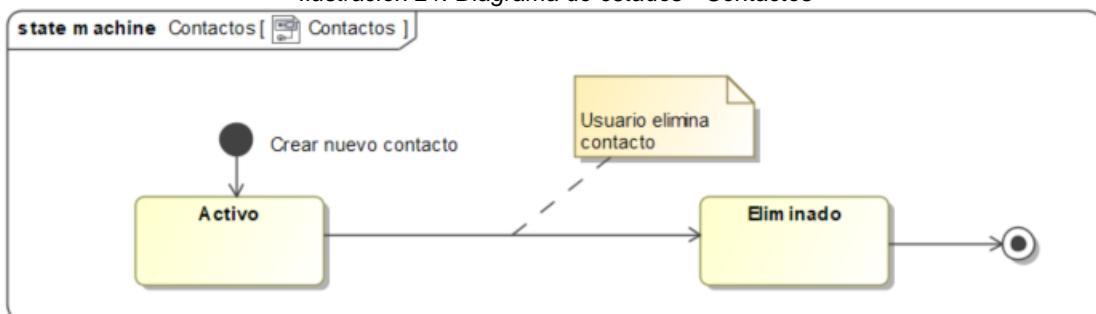
El diagrama de estados es utilizado para demostrar los estados por los que una entidad pasa, es decir, un modelo de comportamiento que consiste en acciones y estados o transiciones a otros estados. Proporciona un estado inicial y uno final. En el sistema de gestión para la fidelización de clientes se definieron los siguientes diagramas de estado, ver ilustraciones #20 - 23

Ilustración 20: Diagrama de estados - Referidos



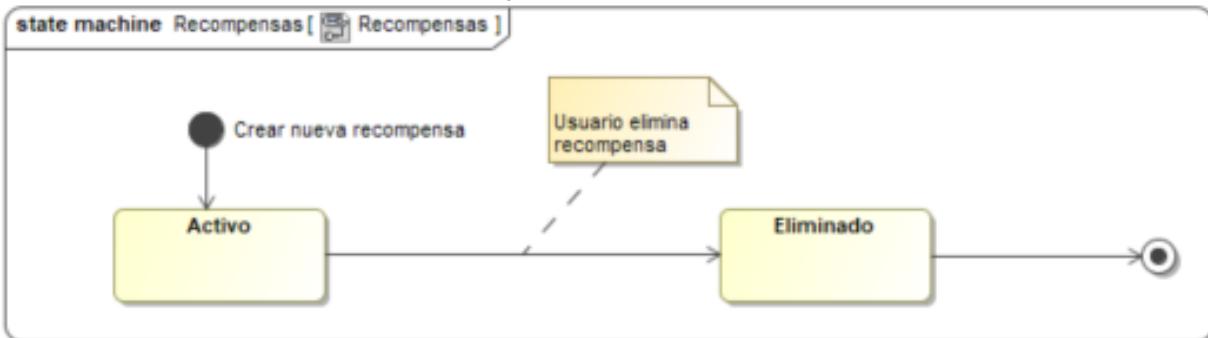
Fuente: Elaboración propia

Ilustración 21: Diagrama de estados - Contactos



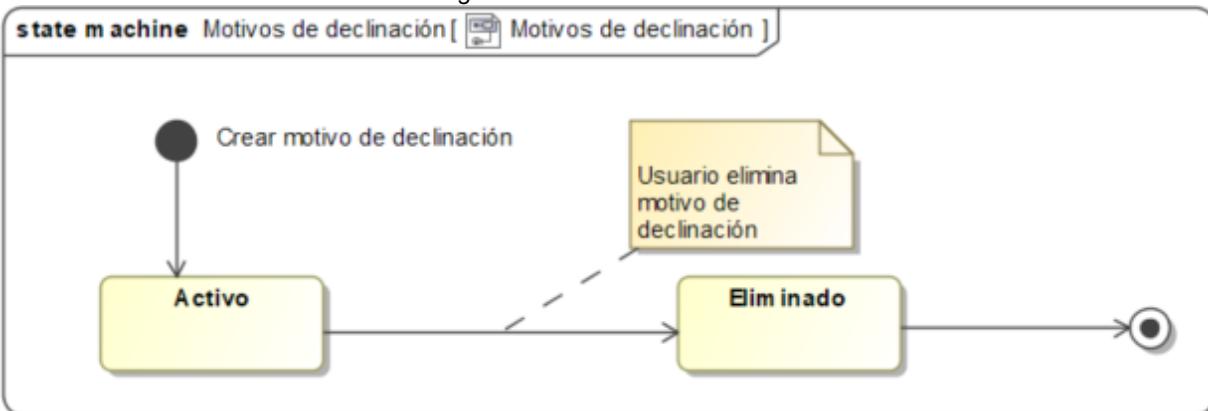
Fuente: Elaboración propia

Ilustración 22: Diagrama de estados - Recompensas



Fuente: Elaboración propia

Ilustración 23: Diagrama de estados - Motivos de declinación

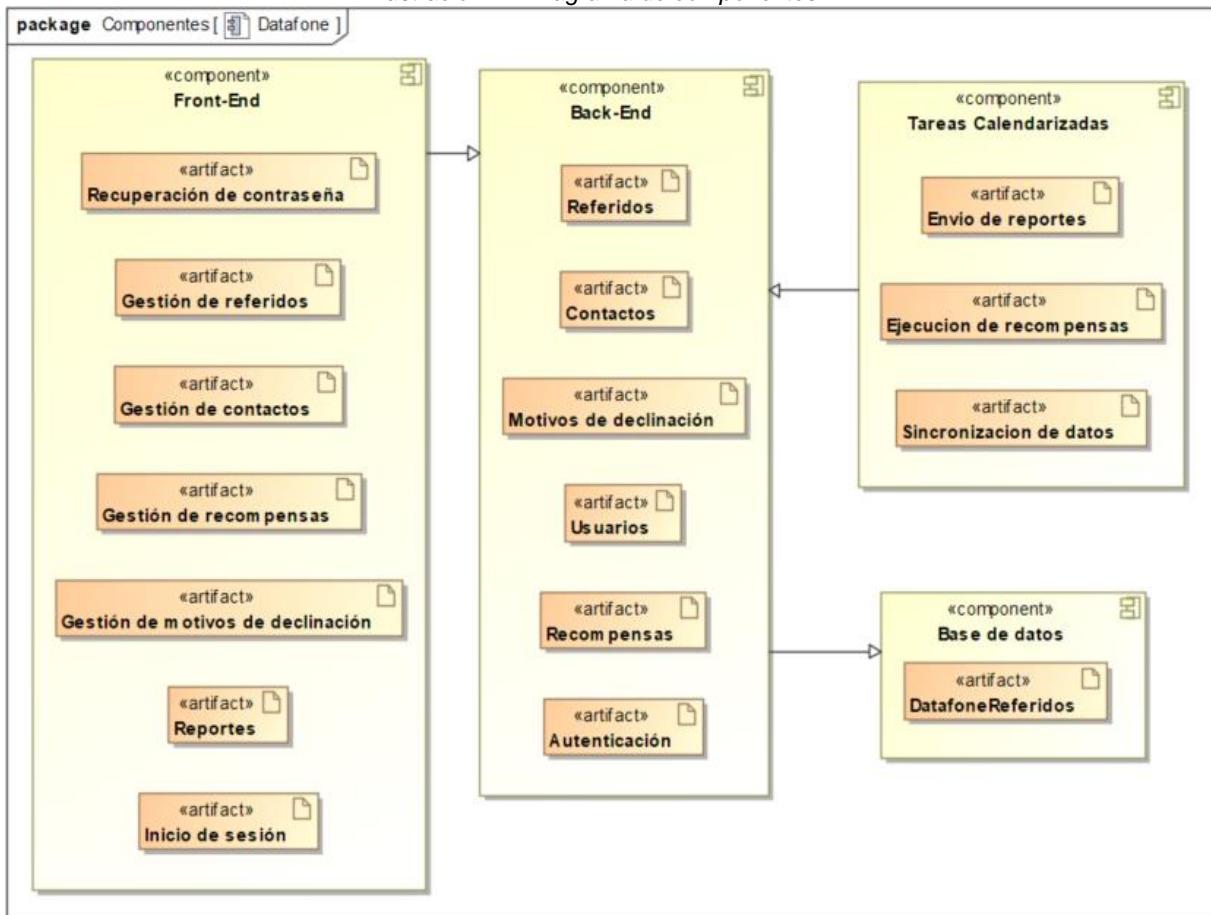


Fuente: Elaboración propia

2.5. Diagrama de componentes

El diagrama de componentes es utilizado para ilustrar a alto nivel los componentes que conforman un sistema. Los componentes pueden ser un componente de software como una base de datos o una interfaz de usuario, o un componente de hardware como un circuito, microchip, etc; así como también podría ser una unidad de negocio como un proveedor, nomina o envío.

Ilustración 24: Diagrama de componentes

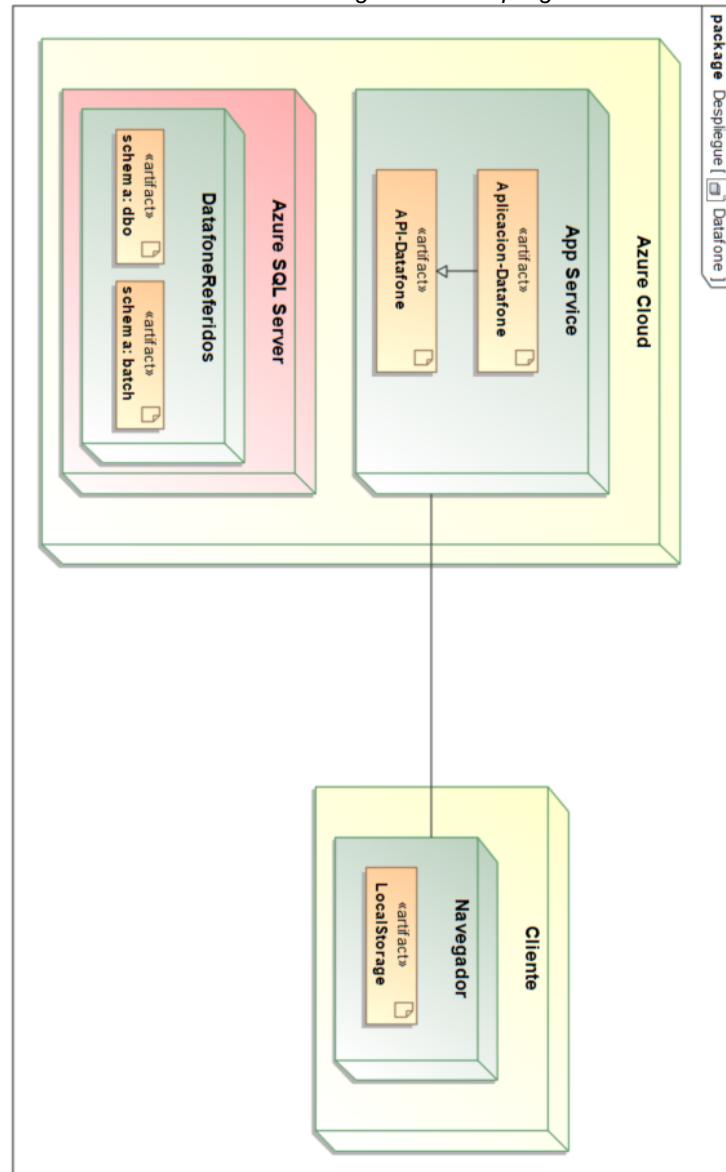


Fuente: Elaboración propia

2.6. Diagrama de despliegue

El diagrama de despliegue es utilizado para ilustrar el hardware y software empleado por un sistema. Ayuda a visualizar los procesadores, nodos y todos los dispositivos vinculados a la arquitectura del sistema.

Ilustración 25: Diagrama de despliegue



Fuente: Elaboración propia

2.7. Plantillas de Coleman

La plantilla de Coleman es utilizada para documentar el comportamiento de los requerimientos. En esta se deben establecer de forma obligatoria los siguientes campos: nombre, definición, actores, escenarios, precondiciones y Post-condiciones, así como la secuencia de pasos que se deben ejecutar para que el requerimiento funcione. Ver tablas #4 - #6

Tabla 8: Plantilla RN-01

Caso de Uso	RN-01: Gestionar referidos.		
DEFINICION:	El sistema debe estar habilitado para realizar las actividades de mostrar la información de los referidos previamente guardados, insertar nuevos referidos, actualizar la información de cualquier referido, eliminar referidos y también debe tener la capacidad de filtrar los registros por nombre y/o correo electrónico.		
PRIORIDAD:	<input checked="" type="checkbox"/> Vital	<input type="checkbox"/> Importante	<input type="checkbox"/> Conveniente
URGENCIA:	<input checked="" type="checkbox"/> Inmediata	<input type="checkbox"/> Necesario	<input type="checkbox"/> Puede esperar
AUTORES:	Christopher Josue Zepeda Tellez Gloria Esther Mayorga Mena Magaly Magdalena Mayorga Maradiaga		
ACTORES			
NOMBRE	DEFINICION		
Administrador, Distribuidor y Vendedor	Los diferentes roles de usuario tendrán acceso a la opción de gestión de referidos.		
ESCENARIO			
Nombre:	Mostrar referidos		
Precondiciones:	<ul style="list-style-type: none"> • El usuario debió haber iniciado sesión • El usuario debe tener acceso a la opción 		
Iniciado por:	Usuario		
Finalizado por:	Sistema		
Post-condiciones:	Mostrar listado de referidos al acceder a la pestaña		
Operaciones:	<ol style="list-style-type: none"> 1. Acceder a la url del sistema web 2. Ingresar las credenciales del usuario 3. Realizar proceso de segundo factor de autenticación 4. Ingresar a pestaña de referidos 5. El sistema consulta el listado de referidos 6. El sistema carga el listado en la pantalla 		
Excepciones:	No hay referidos registrados		
ESCENARIO			
Nombre:	Agregar referido		
Precondiciones:	<ul style="list-style-type: none"> • El usuario debió haber iniciado sesión • El usuario debe tener acceso a la opción • Debe precargar la información de Provincia/Ciudad 		

Iniciado por:	Usuario
Finalizado por:	Sistema
Post-condiciones:	<ul style="list-style-type: none"> Se actualiza el listado de referidos, se agrega el nuevo referido Estado por defecto es Ingresado Los referidos son visibles únicamente para el usuario que los creo
Operaciones:	<ol style="list-style-type: none"> Acceder a la url del sistema web Ingresar las credenciales del usuario Realizar proceso de segundo factor de autenticación Ingresar a pestaña de referidos Accionar botón Agregar Completar información de referido Accionar botón guardar Se realiza verificación de excepciones y se almacena información en base de datos Retornar listado de referidos nuevos Actualizar pantalla de referidos
Excepciones:	<ul style="list-style-type: none"> Que el usuario no ingrese los datos obligatorios en pantalla Existe otro referido con el mismo número de teléfono Existe otro referido con el mismo correo electrónico
ESCENARIO	
Nombre:	Actualizar referido
Precondiciones:	<ul style="list-style-type: none"> El usuario debió haber iniciado sesión El usuario debe tener acceso a la opción. Debe precargar la información del referido que ha sido guardado previamente
Iniciado por:	Usuario
Finalizado por:	Sistema
Post-condiciones:	Se actualiza el listado de referidos, se agrega el nuevo referido
Operaciones:	<ol style="list-style-type: none"> Acceder a la url del sistema web Ingresar las credenciales del usuario Realizar proceso de segundo factor de autenticación Ingresar a pestaña de referidos Buscar el referido a actualizar Ingresar nueva información Accionar botón Actualizar Se realiza verificación de excepciones y se almacena información en base de datos Retornar listado de referidos Actualizar pantalla de referidos
Excepciones:	<ul style="list-style-type: none"> Que el usuario no ingrese los datos obligatorios en pantalla Existe otro referido con el mismo número de teléfono Existe otro referido con el mismo correo electrónico
ESCENARIO	
Nombre:	Eliminar referido
Precondiciones:	<ul style="list-style-type: none"> El usuario debió haber iniciado sesión

	<ul style="list-style-type: none"> • El usuario debe tener acceso a la opción
Iniciado por:	Usuario
Finalizado por:	Sistema
Post-condiciones:	<ul style="list-style-type: none"> • Se actualiza estado del referido a Rechazado • Se establece motivo de declinación
Operaciones:	<ol style="list-style-type: none"> 1. Acceder a la url del sistema web 2. Ingresar las credenciales del usuario 3. Realizar proceso de segundo factor de autenticación 4. Ingresar a pestaña de referidos 5. Buscar el referido a eliminar 6. Accionar botón de eliminar/declinar 7. Se despliega pantalla en la que se debe establecer un motivo de declinación y un comentario. 8. El sistema verifica la existencia del referido y el motivo de declinación 9. El sistema actualiza y almacena la relación entre referido y motivos de declinación 10. Actualizar pantalla de referidos
Excepciones:	<ul style="list-style-type: none"> • Referido no existe • Motivo de declinación no existe
ESCENARIO	
Nombre:	Filtrar referido
Precondiciones:	<ul style="list-style-type: none"> • El usuario debió haber iniciado sesión • El usuario debe tener acceso a la opción
Iniciado por:	Usuario
Finalizado por:	Sistema
Post-condiciones:	<ul style="list-style-type: none"> • Se actualiza el listado de referidos en pantalla
Operaciones:	<ol style="list-style-type: none"> 1. Acceder a la url del sistema web 2. Ingresar las credenciales del usuario 3. Realizar proceso de segundo factor de autenticación 4. Ingresar a pestaña de referidos 5. Establecer nombre o correo electrónico en caja de texto para la búsqueda 6. Accionar botón de buscar 7. El sistema filtra los registros basados en los parámetros de búsqueda
Excepciones:	<ul style="list-style-type: none"> • Referido no existe
ESCENARIO	
Nombre:	Agregar referido - Anónimo
Precondiciones:	<ul style="list-style-type: none"> • El usuario debe acceder al enlace para registro de referidos de forma anónima • Debe precargar la información de Provincia/Ciudad
Iniciado por:	Usuario
Finalizado por:	Sistema
Post-condiciones:	<ul style="list-style-type: none"> • Estado por defecto es Ingresado

	<ul style="list-style-type: none"> Los referidos son visibles para todos los vendedores o distribuidores
Operaciones:	<ol style="list-style-type: none"> Acceder al enlace del sistema web anónimo Accionar botón Agregar Completar información de referido Accionar botón guardar Se realiza verificación de excepciones y se almacena información en base de datos Mostrar mensaje de error o éxito
Excepciones:	<ul style="list-style-type: none"> Que el usuario no ingrese los datos obligatorios en pantalla Existe otro referido con el mismo número de teléfono Existe otro referido con el mismo correo electrónico

Fuente: Elaboración propia

Tabla 9: Plantilla RN-02

Caso de Uso	RN-02: Gestionar contactos		
DEFINICION:	El sistema debe estar habilitado para realizar las actividades de mostrar la información de los contactos previamente guardados, insertar nuevos contactos, actualizar la información de cualquier contacto, eliminar contactos y también debe tener la capacidad de filtrar los registros por referido.		
PRIORIDAD:	<input checked="" type="checkbox"/> Vital	<input type="checkbox"/> Importante	<input type="checkbox"/> Conveniente
URGENCIA:	<input type="checkbox"/> Inmediata	<input checked="" type="checkbox"/> Necesario	<input type="checkbox"/> Puede esperar
AUTORES:	Christopher Josue Zepeda Tellez Gloria Esther Mayorga Mena Magaly Magdalena Mayorga Maradiaga		
ACTORES			
NOMBRE	DEFINICION		
Administrador, Distribuidor y Vendedor	Los diferentes roles de usuario tendrán acceso a la opción de gestión de contactos.		
ESCENARIO			
Nombre:	Mostrar contactos		
Precondiciones:	<ul style="list-style-type: none"> El usuario debió haber iniciado sesión El usuario debe tener acceso a la opción Debe existir al menos 1 referido 		
Iniciado por:	Usuario		
Finalizado por:	Sistema		
Post-condiciones:	Mostrar listado de contactos del día al acceder a la pestaña		
Operaciones:	<ol style="list-style-type: none"> Acceder a la url del sistema web Ingresar las credenciales del usuario Realizar proceso de segundo factor de autenticación Ingresar a pestaña de contactos El sistema consulta el listado de referidos El sistema carga el listado de referidos en lista de selección Se muestran los contactos realizados durante el día 		

Excepciones:	<ul style="list-style-type: none"> • No hay referidos registrados • No hay contactos realizados durante el día
ESCENARIO	
Nombre:	Filtrar contactos por referido
Precondiciones:	<ul style="list-style-type: none"> • El usuario debió haber iniciado sesión • El usuario debe tener acceso a la opción • Debe existir al menos 1 referido
Iniciado por:	Usuario
Finalizado por:	Sistema
Post-condiciones:	Mostrar listado de contactos realizados al referido
Operaciones:	<ol style="list-style-type: none"> 1. Acceder a la url del sistema web 2. Ingresar las credenciales del usuario 3. Realizar proceso de segundo factor de autenticación 4. Ingresar a pestaña de contactos 5. El sistema consulta el listado de referidos 6. El sistema carga el listado de referidos en lista de selección 7. Se muestran los contactos realizados durante el día
Excepciones:	No hay contactos realizados durante el día
ESCENARIO	
Nombre:	Agregar contacto
Precondiciones:	<ul style="list-style-type: none"> • El usuario debió haber iniciado sesión • El usuario debe tener acceso a la opción • Debe existir al menos 1 referido
Iniciado por:	Usuario
Finalizado por:	Sistema
Post-condiciones:	Se debe mostrar el listado de contactos realizados al referido seleccionado
Operaciones:	<ol style="list-style-type: none"> 1. Acceder a la url del sistema web 2. Ingresar las credenciales del usuario 3. Realizar proceso de segundo factor de autenticación 4. Ingresar a pestaña de contactos 5. El sistema consulta el listado de referidos 6. El sistema carga el listado de referidos en lista de selección 7. Se muestran los contactos realizados durante el día 8. Seleccionar referido 9. Accionar botón agregar 10. Ingresar información de contacto 11. Accionar botón guardar 12. El sistema valida la información ingresada 13. Se almacena información en la base de datos 14. Se muestra listado de contactos actualizado en orden descendente
Excepciones:	<ul style="list-style-type: none"> • La fecha de próximo contacto es inferior al día de registro • Fecha de próximo registro no cumple con formato de fecha • El comentario no puede estar vacío
ESCENARIO	

Nombre:	Actualizar contacto
Precondiciones:	<ul style="list-style-type: none"> • El usuario debió haber iniciado sesión • El usuario debe tener acceso a la opción • Debe existir al menos 1 referido • Debe existir al menos 1 contacto
Iniciado por:	Usuario
Finalizado por:	Sistema
Post-condiciones:	Se debe mostrar el listado de contactos realizados al referido seleccionado
Operaciones:	<ol style="list-style-type: none"> 1. Acceder a la url del sistema web 2. Ingresar las credenciales del usuario 3. Realizar proceso de segundo factor de autenticación 4. Ingresar a pestaña de contactos 5. El sistema consulta el listado de referidos 6. El sistema carga el listado de referidos en lista de selección 7. Se muestran los contactos realizados durante el día 8. Seleccionar referido 9. Seleccionar contacto a actualizar 10. Ingresar información de contacto 11. Accionar botón actualizar 12. El sistema valida la información ingresada 13. Se almacena información en la base de datos 14. Se muestra listado de contactos actualizado en orden descendente
Excepciones:	<ul style="list-style-type: none"> • La fecha de próximo contacto es inferior al día de actualización • Fecha de próximo registro no cumple con formato de fecha • El comentario no puede estar vacío
ESCENARIO	
Nombre:	Eliminar contacto
Precondiciones:	<ul style="list-style-type: none"> • El usuario debió haber iniciado sesión • El usuario debe tener acceso a la opción • Debe existir al menos 1 referido • Debe existir al menos 1 contacto
Iniciado por:	Usuario
Finalizado por:	Sistema
Post-condiciones:	Se debe mostrar el listado de contactos realizados al referido seleccionado
Operaciones:	<ol style="list-style-type: none"> 1. Acceder a la url del sistema web 2. Ingresar las credenciales del usuario 3. Realizar proceso de segundo factor de autenticación 4. Ingresar a pestaña de contactos 5. El sistema consulta el listado de referidos 6. El sistema carga el listado de referidos en lista de selección 7. Se muestran los contactos realizados durante el día 8. Seleccionar referido 9. Seleccionar contacto a eliminar

	10. Confirmar eliminación de contacto 11. Sistema valida que contacto existe 12. Se desactiva registro de contacto 13. Se muestra listado de contactos actualizado
Excepciones:	<ul style="list-style-type: none"> • Contacto no existe

Fuente: Elaboración propia

Tabla 10: Plantilla RN-03

Caso de Uso	RN-03: Gestionar motivos de declinación	
DEFINICION:	El sistema debe estar habilitado para realizar las actividades de mostrar los motivos de declinación previamente guardados, insertar nuevos motivos de declinación, actualizar los motivos de declinación, eliminar registros y también debe tener la capacidad de filtrar los registros por nombre del motivo de declinación.	
PRIORIDAD:	<input type="checkbox"/> Vital <input checked="" type="checkbox"/> Importante <input type="checkbox"/> Conveniente	
URGENCIA:	<input type="checkbox"/> Inmediata <input checked="" type="checkbox"/> Necesario <input type="checkbox"/> Puede esperar	
AUTORES:	Cristopher Josue Zepeda Tellez Gloria Esther Mayorga Mena Magaly Magdalena Mayorga Maradiaga	
ACTORES		
NOMBRE	DEFINICION	
Administrador y Distribuidor	Los diferentes roles de usuario tendrán acceso a la opción de gestión de motivos de declinación.	
ESCENARIO		
Nombre:	Mostrar motivos de declinación	
Precondiciones:	<ul style="list-style-type: none"> • El usuario debió haber iniciado sesión • El usuario debe tener acceso a la opción 	
Iniciado por:	Usuario	
Finalizado por:	Sistema	
Post-condiciones:	Mostrar listado de motivos de declinación activos al acceder a la pestaña	
Operaciones:	<ol style="list-style-type: none"> 1. Acceder a la url del sistema web 2. Ingresar las credenciales del usuario 3. Realizar proceso de segundo factor de autenticación 4. Ingresar a pestaña motivos de declinación 5. El sistema consulta los motivos de declinación 6. El sistema muestra los motivos de declinación 	
Excepciones:	<ul style="list-style-type: none"> • No hay motivos de declinación registrados 	
ESCENARIO		
Nombre:	Filtrar motivos de declinación	
Precondiciones:	<ul style="list-style-type: none"> • El usuario debió haber iniciado sesión • El usuario debe tener acceso a la opción • Debe existir al menos 1 motivo de declinación registrado 	
Iniciado por:	Usuario	
Finalizado por:	Sistema	

Post-condiciones:	Mostrar listado de motivos de declinación basado en el nombre buscado
Operaciones:	<ol style="list-style-type: none"> 1. Acceder a la url del sistema web 2. Ingresar las credenciales del usuario 3. Realizar proceso de segundo factor de autenticación 4. Ingresar a pestaña motivos de declinación 5. El sistema consulta los motivos de declinación 6. El sistema muestra los motivos de declinación 7. Establecer texto de búsqueda en caja de texto 8. Accionar botón de búsqueda 9. Buscar motivos de declinación basado en el filtro 10. Mostrar motivos de declinación
Excepciones:	No hay motivos de declinación registrados que cumplan el parámetro de búsqueda.
ESCENARIO	
Nombre:	Agregar motivos de declinación
Precondiciones:	<ul style="list-style-type: none"> • El usuario debió haber iniciado sesión El usuario debe tener acceso a la opción
Iniciado por:	Usuario
Finalizado por:	Sistema
Post-condiciones:	Mostrar listado de motivos de declinación activos al acceder a la pestaña
Operaciones:	<ol style="list-style-type: none"> 1. Acceder a la url del sistema web 2. Ingresar las credenciales del usuario 3. Realizar proceso de segundo factor de autenticación 4. Ingresar a pestaña motivos de declinación 5. El sistema consulta los motivos de declinación 6. El sistema muestra los motivos de declinación 7. Accionar botón agregar 8. Ingresar información de motivo de declinación 9. El sistema valida la información ingresada 10. El sistema ingresa la información a la base de datos 11. Se retorna nuevo listado de motivos de declinación 12. Se muestra el nuevo listado
Excepciones:	<ul style="list-style-type: none"> • Titulo ya existe
ESCENARIO	
Nombre:	Actualizar motivos de declinación
Precondiciones:	<ul style="list-style-type: none"> • El usuario debió haber iniciado sesión • El usuario debe tener acceso a la opción • Debe existir al menos un 1 registro • Al momento de seleccionar un registro, debe precargar la información del registro seleccionado
Iniciado por:	Usuario
Finalizado por:	Sistema
Post-condiciones:	Mostrar listado de motivos de declinación activos
Operaciones:	<ol style="list-style-type: none"> 1. Acceder a la url del sistema web

	<ol style="list-style-type: none"> 2. Ingresar las credenciales del usuario 3. Realizar proceso de segundo factor de autenticación 4. Ingresar a pestaña motivos de declinación 5. El sistema consulta los motivos de declinación 6. El sistema muestra los motivos de declinación 7. Seleccionar registro a actualizar 8. Ingresar nueva información 9. Accionar botón actualizar 10. El sistema valida la información ingresada 11. El sistema actualiza la información en la base de datos 12. Se retorna nuevo listado de motivos de declinación 13. Se muestra el nuevo listado
Excepciones:	Titulo ya existe
ESCENARIO	
Nombre:	Eliminar motivos de declinación
Precondiciones:	<ul style="list-style-type: none"> • El usuario debió haber iniciado sesión • El usuario debe tener acceso a la opción • Debe existir al menos un 1 registro
Iniciado por:	Usuario
Finalizado por:	Sistema
Post-condiciones:	Mostrar listado de motivos de declinación activos
Operaciones:	<ol style="list-style-type: none"> 1. Acceder a la url del sistema web 2. Ingresar las credenciales del usuario 3. Realizar proceso de segundo factor de autenticación 4. Ingresar a pestaña motivos de declinación 5. El sistema consulta los motivos de declinación 6. El sistema muestra los motivos de declinación 7. Seleccionar registro a eliminar 8. Accionar botón eliminar 9. Confirmar eliminación de registro 10. El sistema valida la información ingresada 11. El sistema actualiza la información a la base de datos 12. Se retorna nuevo listado de motivos de declinación 13. Se muestra el nuevo listado
Excepciones:	Titulo no existe

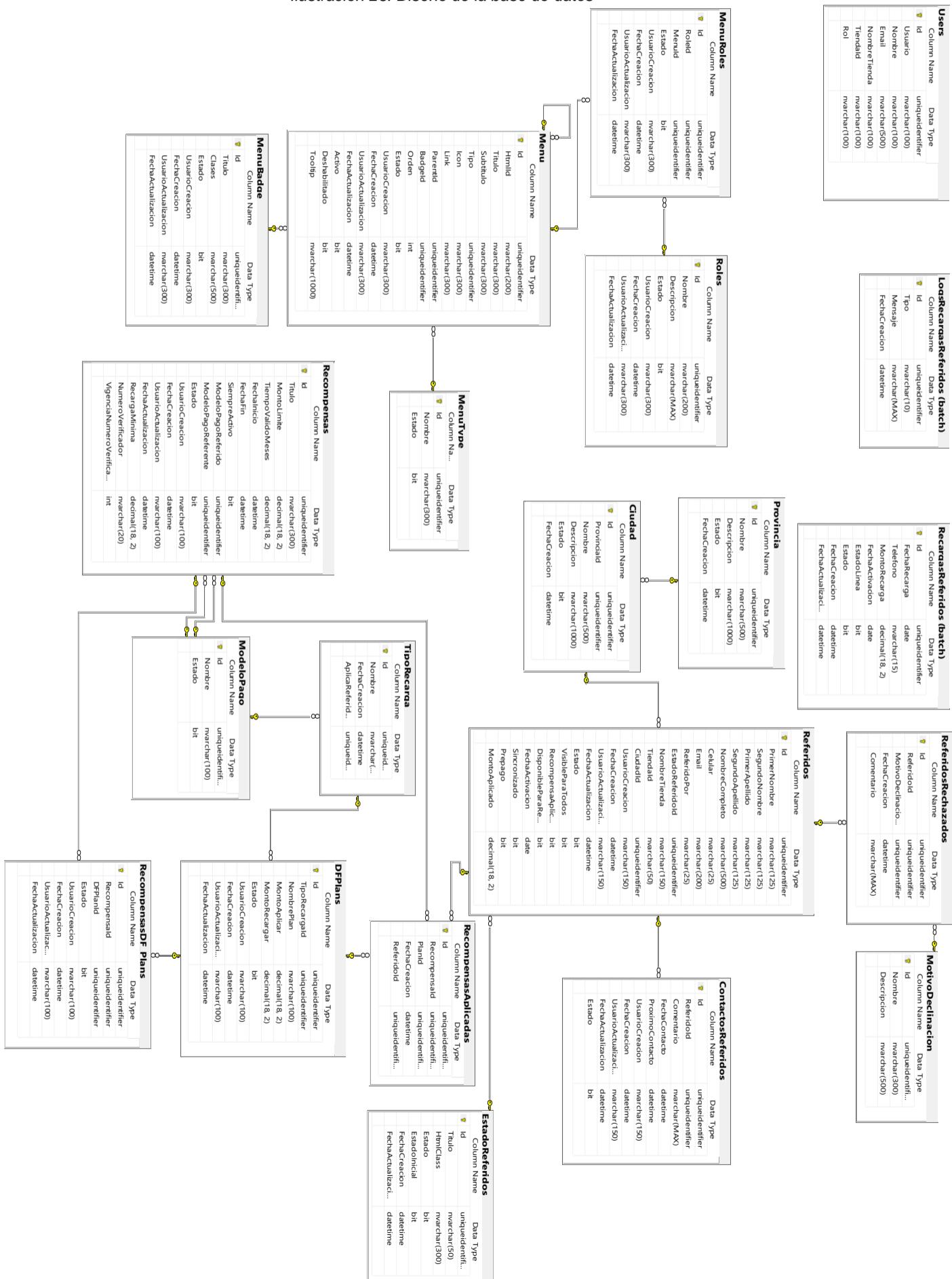
Fuente: Elaboración propia

Ver continuación de tabulaciones en [AnexoTabla1: Tabulación de requerimientos](#)

2.8. Diseño lógico y físico de la base de datos (3FN)

El modelo de bases de datos físico muestra todas las estructuras de tablas, incluidos los nombres de columnas, tipo de dato, claves principales, claves externas y las relaciones entre las tablas. En el sistema de gestión para la fidelización de clientes se definió el siguiente diseño de la base de datos, ver ilustración #24

Ilustración 26: Diseño de la base de datos



Fuente: Elaboración propia

Capítulo III: Construcción del sistema

La construcción del sistema se encuentra basado en las etapas de análisis y diseño. La etapa de construcción tiene como objetivo construir la aplicación, utilizando el patrón de diseño y la arquitectura de software especificada en la etapa de diseño. Así como, también escribir el código utilizando las tecnologías .NET 7, Angular 15 y bases de datos SQL.

3.1. Hardware utilizado para la construcción

Hardware es la parte que se puede ver y tocar de los dispositivos. Es decir, todos los componentes de su estructura física como pantallas y teclados. Para la construcción del sistema para la gestión de fidelización de clientes, se utilizarán equipos con las siguientes características (Ver tablas #08 - #10):

Tabla 11: Equipo de desarrollo 1

Característica	Descripción
Procesador	Core I5-10300H 2.5GHZ
RAM	32GB
Tipo de disco	Disco estado sólido (SSD)
Capacidad de disco	256GB
Tipo de sistema	64Bits
Sistema operativo	Windows 10
Marca	Dell

Fuente: Elaboración propia

Tabla 12: Equipo de desarrollo 2

Característica	Descripción
Procesador	Core I5-1035G1 1.00GHZ
RAM	8GB
Tipo de disco	Disco estado sólido (SSD)
Capacidad de disco	512GB
Tipo de sistema	64Bits
Sistema operativo	Windows 11
Marca	Dell

Fuente: Elaboración propia

Tabla 13: Equipo de desarrollo 3

Característica	Descripción
Procesador	Core I5-1035G1 CPU 1.00 GHz
RAM	8GB
Tipo de disco	Disco Duro (HDD)
Capacidad de disco	512GB
Tipo de sistema	64Bits

Sistema operativo	Windows
Marca	Dell

Fuente: Elaboración propia

3.2. Software utilizado para la construcción

El software son los programas informáticos que hacen posible la ejecución de tareas específicas dentro de un computador. Para la construcción del sistema para la fidelización de clientes, se utilizarán los siguientes programas (Ver tablas #11 - #13):

Tabla 14: Software - Visual Studio 2022

Información de programa	Descripción
Nombre	Visual Studio 2022
Versión	17.5.4
Edición	Community
Función	Crear el proyecto principal, así como también los subproyectos que funcionaran como librerías o capas.

Fuente: Elaboración propia

Tabla 15: Software - Visual Studio 2019

Información de programa	Descripción
Nombre	Visual Studio 2019
Versión	16.11.21
Edición	Community
Función	Crear servicio WCF para simular el servicio de Datafone

Fuente: Elaboración propia

Tabla 16: Software – Visual Studio Code

Información de programa	Descripción
Nombre	Visual Studio Code
Versión	1.77.3
Función	Editar y compilar la aplicación en angular

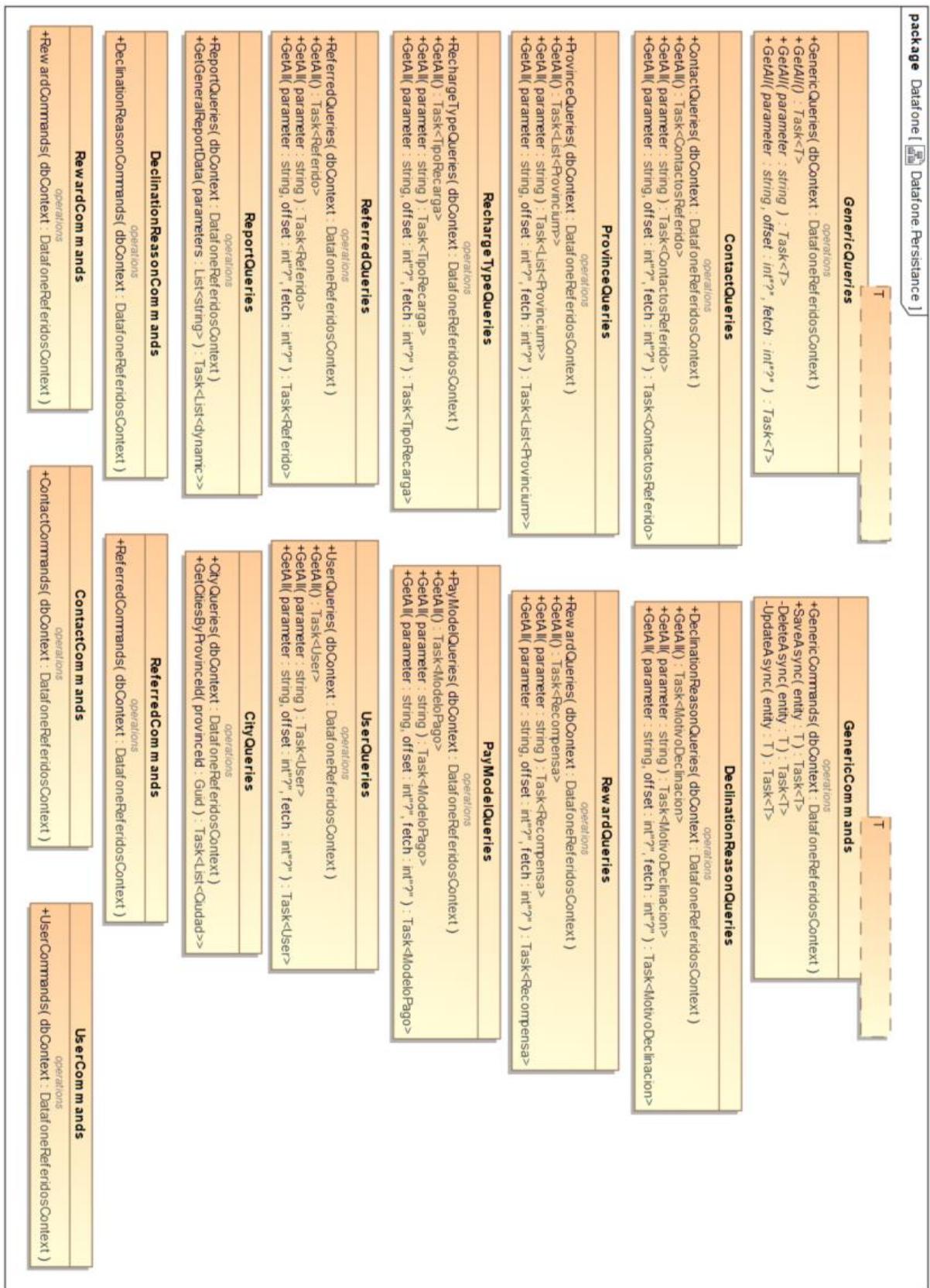
Fuente: Elaboración propia

Ver continuación de software utilizado en [AnexoTabla2: Software utilizado para la construcción](#)

3.3. Especificación de las clases del sistema

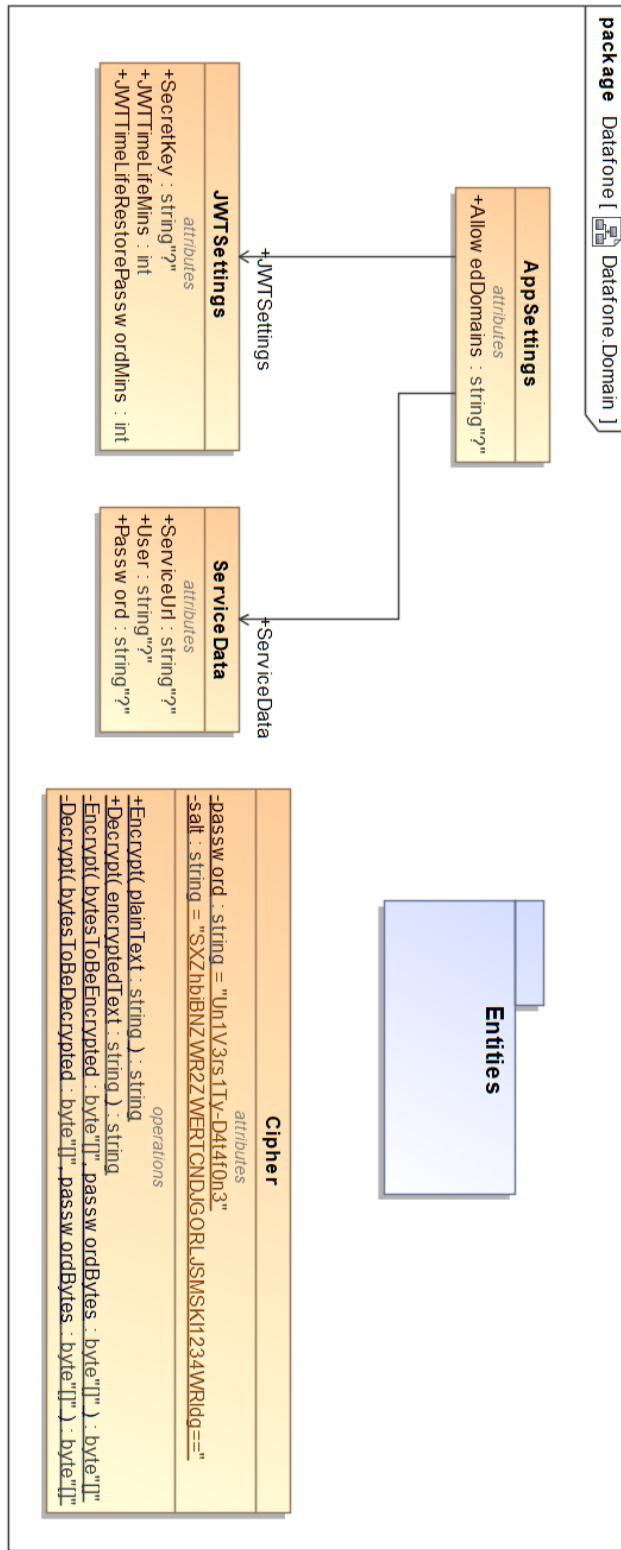
Los diagramas de clases son un tipo de diagrama UML que describen un sistema, visualizando los diferentes tipos de objetos dentro de un sistema y los tipos de relaciones que existen entre ellos, también ilustra las operaciones y atributos que posee cada clase. Durante la construcción del sistema para la fidelización de clientes, se crearon las siguientes clases en C# (Ver ilustraciones #25 - #26):

Ilustración 27: Clases - Datafone.Persistence



Fuente: Elaboración propia

Ilustración 28: Clases - Datafone.Domain



Fuente: Elaboración propia

Ver continuación de diagramas de clases en [Anexo Ilustración 7: Especificación de las clases del sistema](#)

3.4. Diseño y uso de APIs

Las interfaces de programación de aplicaciones (API) son un estilo de arquitectura para la creación de servicios web que están estrechamente enlazados a las acciones leer, crear, actualizar y borrar (CRUD). Cada operación tiene su verbo HTTP reservado GET, POST, UPDATE y DELETE. Durante la construcción del sistema para la fidelización de clientes, se crearon las siguientes APIs (Ver ilustraciones #27 - #29):

Ilustración 29: Swagger – Referidos Anónimos

The screenshot shows the Swagger UI interface for the 'AnonymousReferred' API. At the top right is an 'Authorize' button with a lock icon. Below it, there's a section header 'AnonymousReferred' with a collapse arrow. A list of operations is shown in a table:

Method	Path	Lock
POST	/api/AnonymousReferred/validateRequest	🔓
GET	/api/AnonymousReferred/getbyreferent	🔓
POST	/api/AnonymousReferred	🔓
PUT	/api/AnonymousReferred	🔒
DELETE	/api/AnonymousReferred	🔒

Fuente: Elaboración propia

Ilustración 30: Swagger – Autenticación

The screenshot shows the Swagger UI interface for the 'Authentication' API. At the top right is a collapse arrow. A list of operations is shown in a table:

Method	Path	Lock
POST	/api/Authentication/login	🔓
POST	/api/Authentication/refreshtoken	🔓
POST	/api/Authentication/resetpassword	🔓
POST	/api/Authentication/forgotpassword	🔓
POST	/api/Authentication/validatepin	🔓

Fuente: Elaboración propia

Ilustración 31: Swagger - Ciudades

The screenshot shows the Swagger UI interface for the 'City' API. At the top right is a collapse arrow. A list of operations is shown in a table:

Method	Path	Lock
GET	/api/City	🔓

Fuente: Elaboración propia

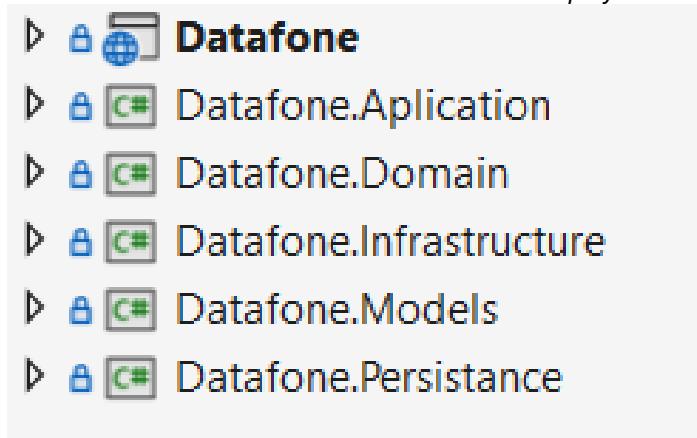
Ver continuación de diseños de APIs en [Anexo Ilustración 8: Diseño y uso de APIs](#)

3.5. Implementación de patrones de diseño

Los patrones de diseño son soluciones habituales a problemas comunes, durante la etapa de diseño se planteó utilizar una combinación de patrones y arquitecturas para la construcción (CQRS y Clean Architecture), así como también se planteó que el estilo de arquitectura sería en capas. Durante la construcción del sistema para la fidelización de clientes, los patrones se implementaron de la siguiente forma (Ver ilustraciones #30 - #33):

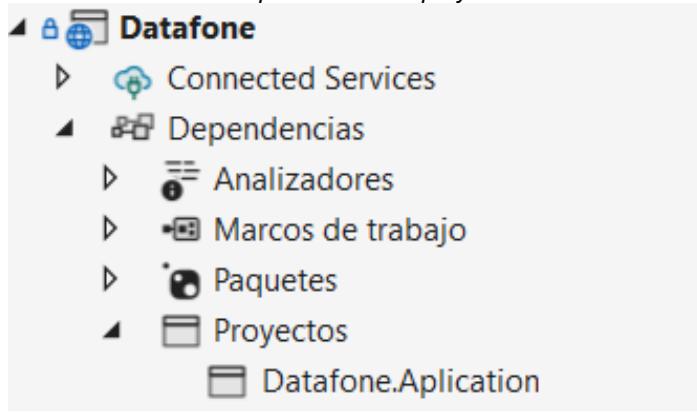
Clean Architecture

Ilustración 32: Clean Architecture - Estructura de proyectos



Fuente: Elaboración propia

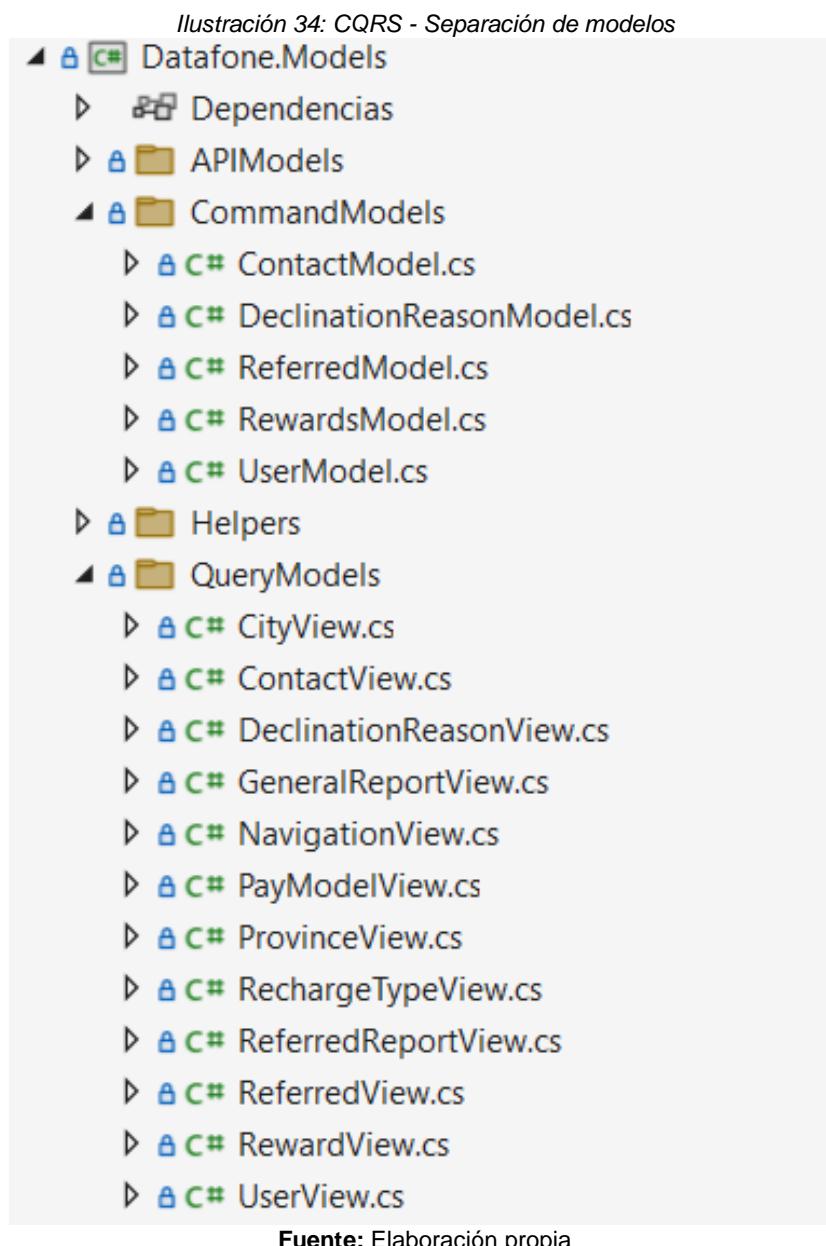
Ilustración 33: Dependencias de proyectos - Datafone



Fuente: Elaboración propia

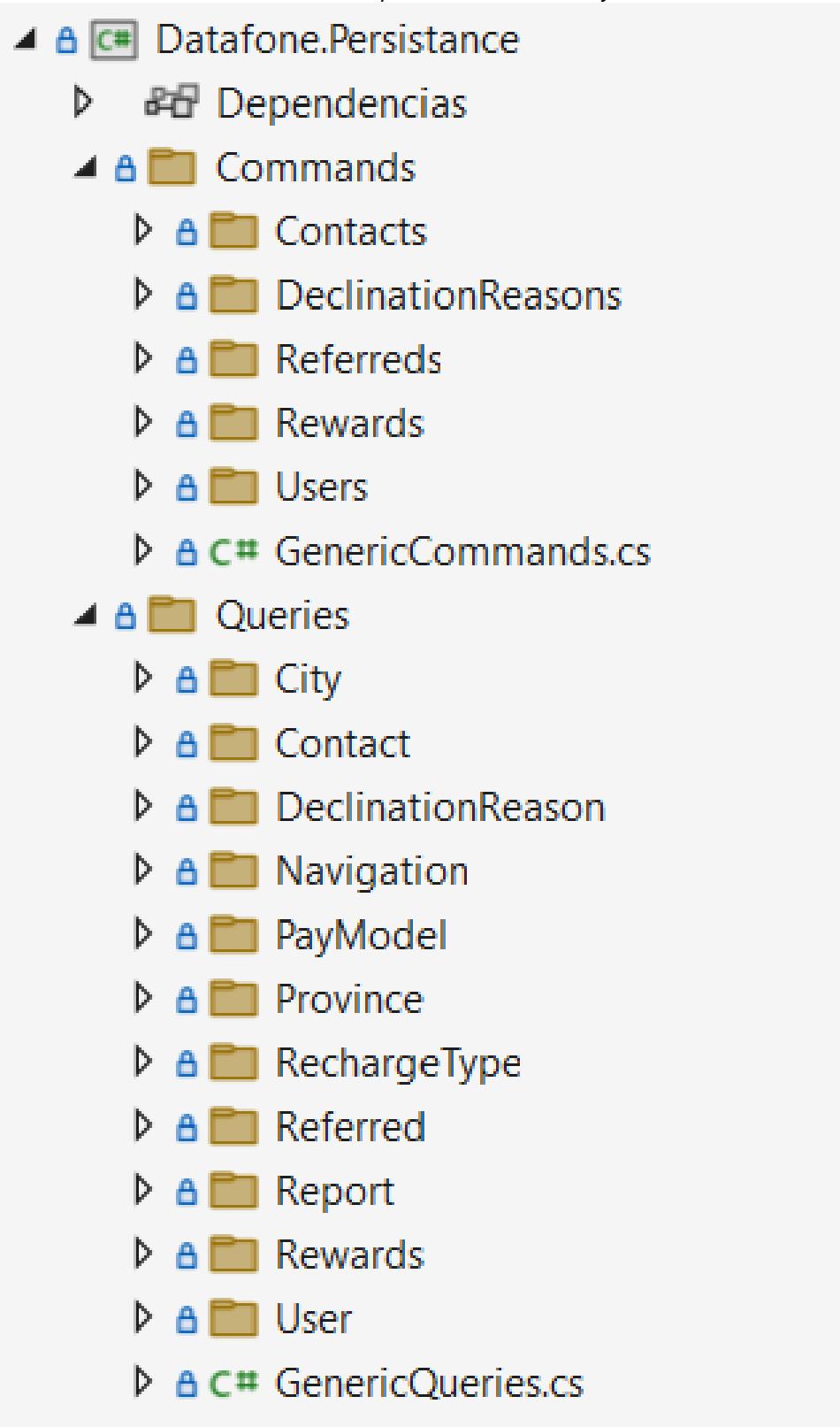
Ver continuación de implementación Clean Architecture en [Anexos Ilustración 9: Implementación patrones de diseño – Clean Architecture](#)

CQRS



Fuente: Elaboración propia

Ilustración 35: CQRS - Separación de comandos y consultas

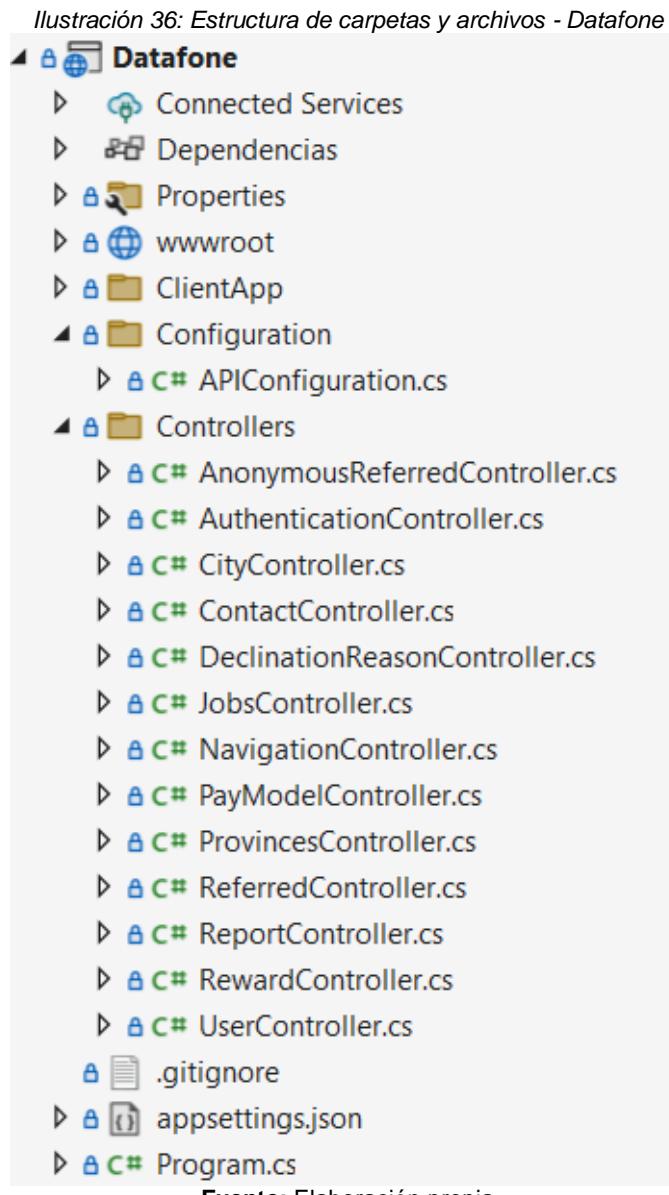


Fuente: Elaboración propia

3.6. Organización de los archivos del proyecto

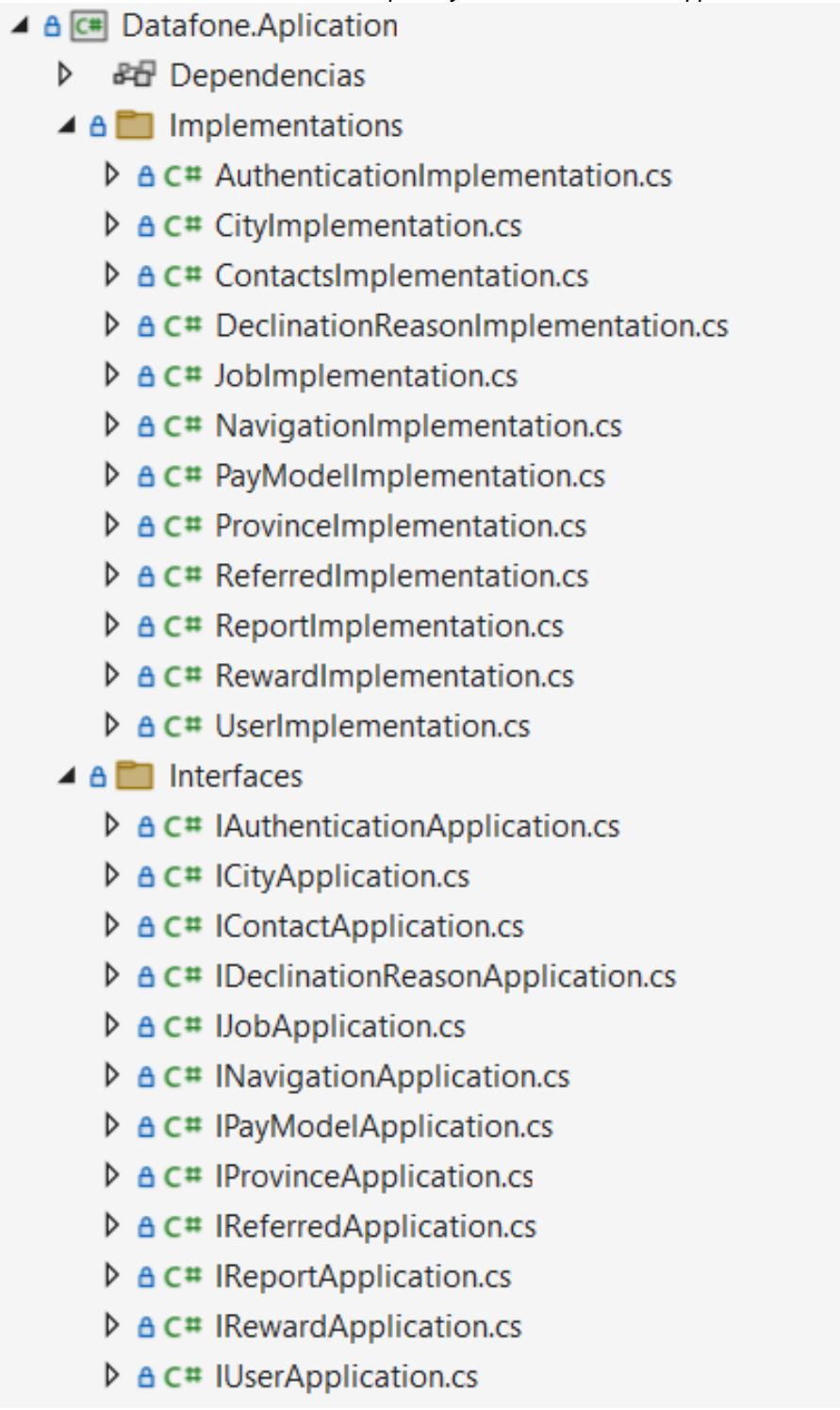
Al implementar patrones de diseño y estilos de arquitectura es obligatorio cumplir con estructuras de carpetas o proyectos que facilitarán la comprensión del proyecto, adicional esto facilitará el escalamiento y mantenimiento de la aplicación. Durante la construcción del sistema para la fidelización de clientes, se definió la siguiente estructura de proyecto (Ver ilustraciones #34 - #37):

Estructura de carpetas Back-End



Fuente: Elaboración propia

Ilustración 37: Estructura de carpetas y archivos - Datafone.Application

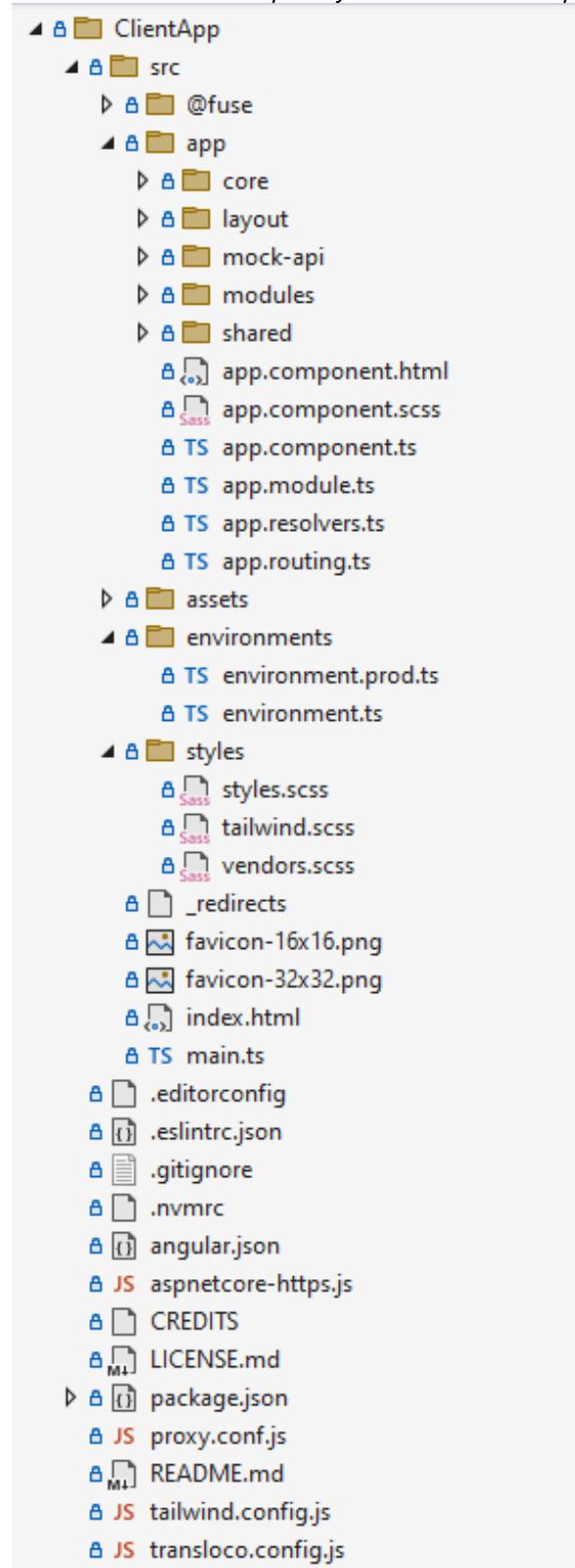


Fuente: Elaboración propia

Ver continuación de estructuras de carpetas Back-End en [Anexo Ilustración 10: Organización de los archivos del proyecto – Back-End](#)

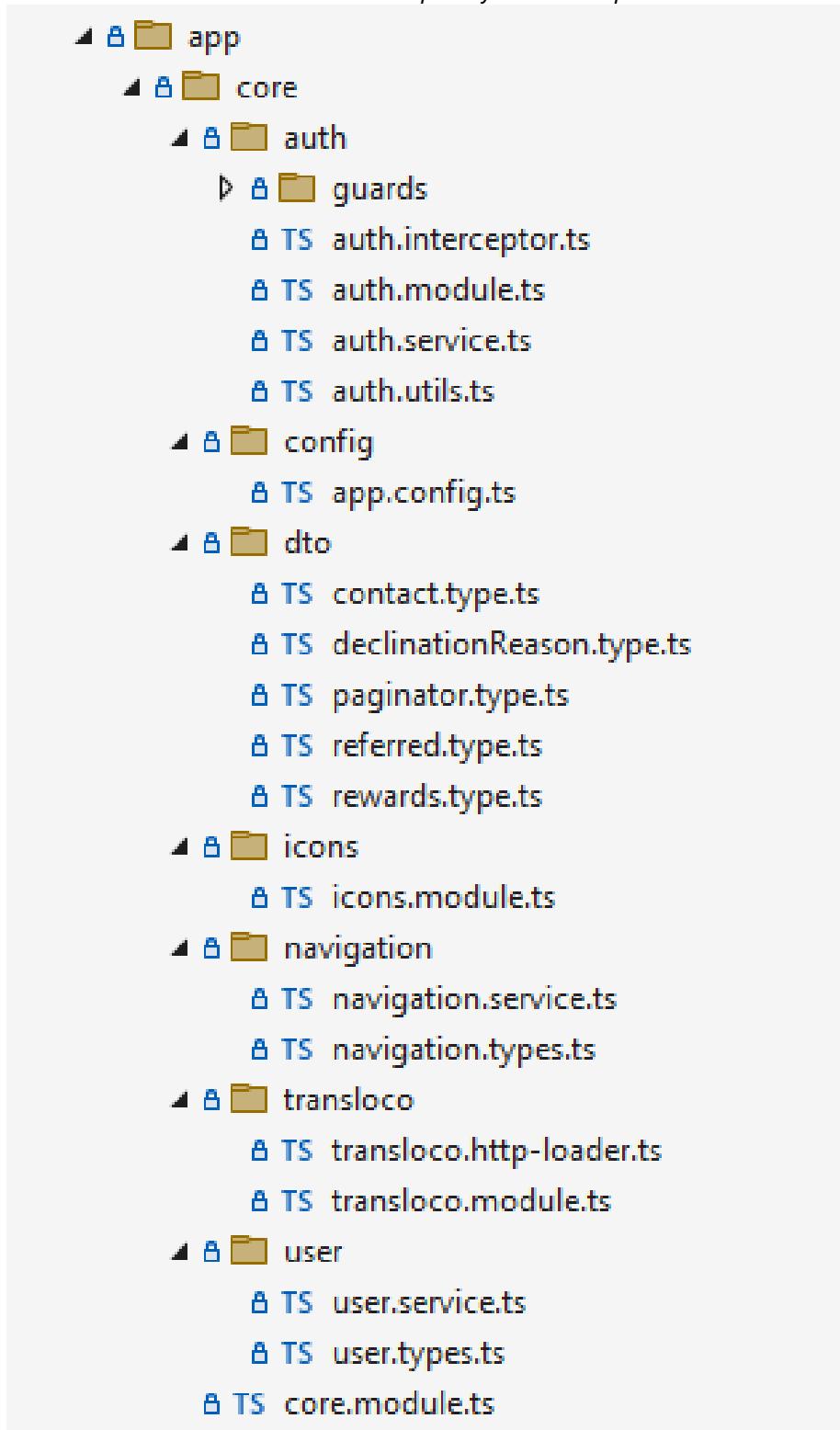
Estructura de carpetas Front-End

Ilustración 38: Estructura de carpetas y archivos - Raíz de aplicación



Fuente: Elaboración propia

Ilustración 39: Estructura de carpetas y archivos - Aplicacion.core



Fuente: Elaboración propia

Ver continuación de estructuras de carpetas Front-End en [Anexos Ilustración 11: Organización de los archivos del proyecto – Front-End](#)

3.7. Diseños de las interfaces del sistema

Las interfaces de usuarios son el medio que permite la comunicación entre un usuario y el sistema. Durante la construcción del sistema para la fidelización de clientes, se crearon las siguientes interfaces:

Interfaces: Gestión de accesos

Propósito: Gestionar los accesos en el sitio web

Acciones o eventos: Iniciar sesión con doble autenticación, recuperar contraseña y reestablecer contraseña.

Elementos gráficos que la componen:

- Cajas de texto: Permite mostrar o editar texto sin formato
- Casillas de verificación: Las casillas de verificación permiten que el usuario seleccione una o más opciones de un conjunto
- Botones: Permite accionar los eventos en un sitio web
- Enlaces: Son hipervínculos que redireccionan a otro componente del sitio web

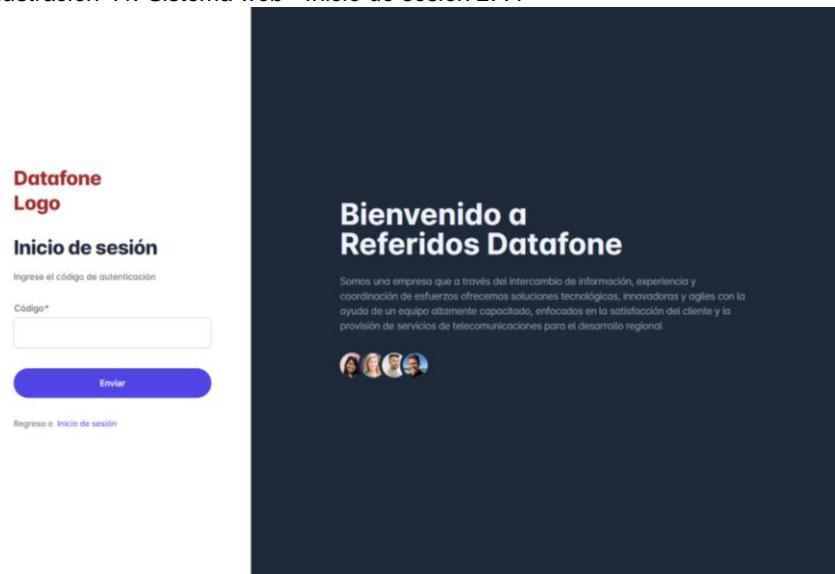
Capturas de pantalla (Ver ilustraciones #38 - #41):

Ilustración 40: Sistema web - Inicio de sesión



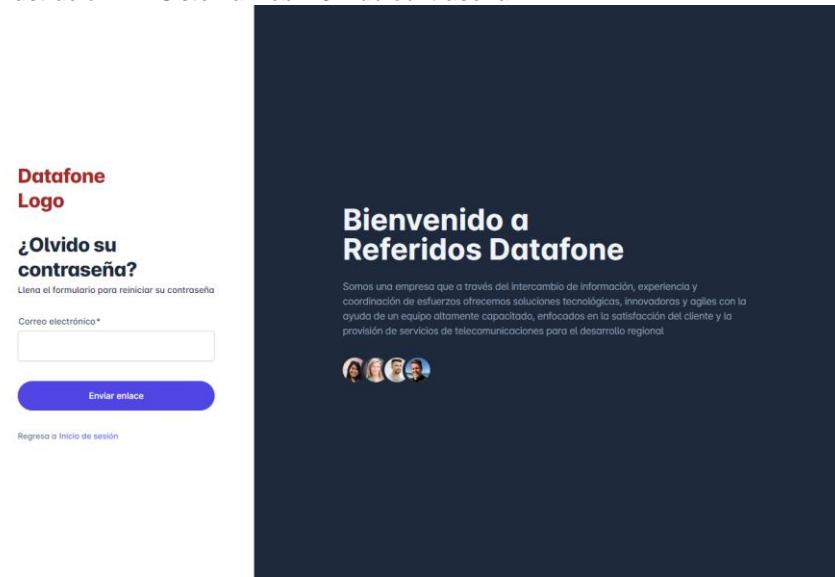
Fuente: Elaboración propia

Ilustración 41: Sistema web - Inicio de sesión 2FA



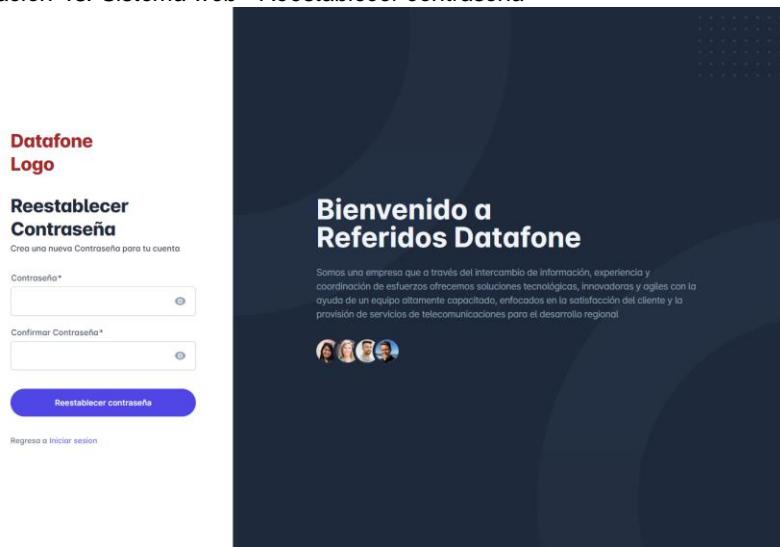
Fuente: Elaboración propia

Ilustración 42: Sistema web - Olvido contraseña



Fuente: Elaboración propia

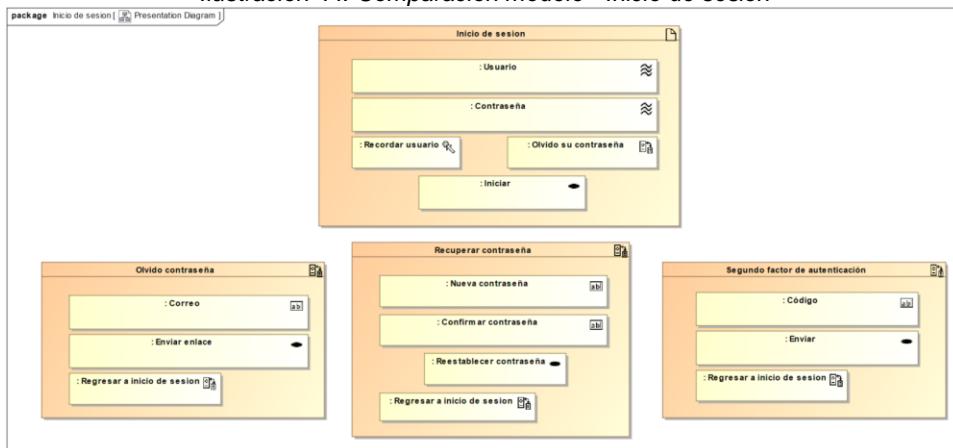
Ilustración 43: Sistema web - Reestablecer contraseña



Fuente: Elaboración propia

Comparación con modelo de presentación (Ver ilustración #42):

Ilustración 44: Comparación modelo - Inicio de sesión



Datafone
Logo

Inicio de sesión

Ingrese los datos para iniciar sesión

Usuario*	<input type="text" value="hughes.brian@company.com"/>
Contraseña*	<input type="password" value="*****"/>
<input type="checkbox"/> Recuérdame ¿Olvidó su contraseña?	
<input type="button" value="Iniciar"/>	

Datafone
Logo

¿Olvido su contraseña?

Llene el formulario para reiniciar su contraseña

Correo electrónico*	<input type="text"/>
<input type="button" value="Enviar enlace"/>	

Datafone
Logo

Reestablecer Contraseña

Crea una nueva Contraseña para tu cuenta

Contraseña*	<input type="password"/>
Confirmar Contraseña*	<input type="password"/>
<input type="button" value="Reestablecer contraseña"/>	

Datafone
Logo

Inicio de sesión

Ingrese el código de autenticación

Código*	<input type="text"/>
<input type="button" value="Enviar"/>	

Fuente: Elaboración propia

Ver continuación de diseños de interfaces en [Anexo Ilustración 12: Diseños de las interfaces del sistema](#)

VII. Conclusiones

El sistema de información web para la gestión de fidelización de clientes creado en este proyecto puede impactar positivamente en la manera en que actualmente el equipo de Datafone desarrolla sus requerimientos. Esto debido a la implementación de los diferentes patrones de diseños y arquitectura utilizados, así como también por la facilidad de uso para el usuario final.

El sistema web para la fidelización de clientes incluye las siguientes características:

- Gestiona la información de los referidos de una manera sencilla y organizada, asigna y actualiza los estados de los referidos.
- Gestiona los contactos realizados a referidos.
- Gestiona los motivos de declinación que serán utilizados al momento de eliminar un referido.
- Gestiona las configuraciones de una recompensa.
- Ejecuta procesos automáticos para la sincronización de datos y procesos de recompensas.

Durante las etapas de análisis y diseño, se crearon una serie de diagramas basados en la metodología para sistemas web UWE que incluye los siguientes modelos: Modelo de requerimientos, de contenido, de navegación, de presentación y de procesos, que pudieron ser utilizados por el equipo de desarrolladores para una mejor comprensión de las funcionalidades, diseño, arquitectura, estructura del proyecto y de esta forma se redujo considerablemente el tiempo de desarrollo del sistema.

Como sistema web para la fidelización de clientes es una idea innovadora que actualmente no se encuentra disponible y que puede ser fácilmente distribuida y utilizada en la compañía.

VIII. Recomendaciones

Al momento de implementar el presente proyecto en un ambiente productivo, se recomienda aprovechar las funcionalidades que fueron desarrolladas y expandirlas utilizando los mismos patrones y arquitectura previamente utilizados. Esencialmente se debe continuar con el uso del patrón de diseño CQRS, ya que este facilita el cambio de origen de datos tanto al momento de modificar información en la base de datos como leer la información.

Mantener la dependencia de proyectos es vital para la implementación de Clean Architecture, por lo que al momento de implementar nuevos módulos se debe evitar realizar saltos entre las capas (los desarrolladores suelen realizar estos saltos para reducir la cantidad de código escrito y reducir el tiempo de desarrollo de un nuevo requerimiento).

Implementar un origen de datos para los comandos de consulta a la base de datos mejorará el rendimiento del sistema web, esto implicaría que al momento de realizar una inserción, modificación o eliminación de registro en la base de datos se deba replicar la información en el segundo origen de datos (se recomienda que este origen de datos sea una base de datos NoSQL por el rendimiento que estos tienen para ser consultadas).

Realizar la separación de ambientes (Desarrollo, pruebas y producción) ayudará a una mejor gestión para la solución de errores encontrados en las diferentes etapas, así como también facilitará la implementación de nuevas características con la menor cantidad de errores posibles en producción.

Por último, se recomienda el uso de versionamiento de código (utilizando las diferentes ramas respectivas master/producción, uat/pruebas, y las ramas nuevas creadas por requerimiento) para un mejor control del código fuente; y el uso de Azure DevOps para la integración continua y despliegue continuo, esto reducirá considerablemente el tiempo utilizado en el despliegue de la aplicación.

IX. Bibliografía

- 3DS CATiA. (s.f.). *MAGICDRAW*. Recuperado el 02 de 02 de 2023, de
<https://www.3ds.com/products-services/catia/products/no-magic/magicdraw/>
- Akella, R., Kumar Tamirisa, A., Kumar Kunani, S., & Guptha Muthiyalu, B. (2021). *Enterprise Application Development with C# 9 and .NET 5*. Birmingham: Packt Publishing.
- Angular. (28 de 02 de 2022). *What is Angular?* Obtenido de <https://angular.io/guide/what-is-angular>
- AWS. (s.f.). *¿En qué consiste el alojamiento web?* Recuperado el 01 de 02 de 2023, de
<https://aws.amazon.com/es/what-is/web-hosting/>
- BERNAL, C. (2010). *Metodología de la investigación administración, economía, humanidades y ciencias sociales*. Colombia: PEARSON EDUCACIÓN.
- bnamericas. (02 de 04 de 2022). *bn americas*. Recuperado el 18 de 01 de 2023, de
<https://www.bnamericas.com/es/reportajes/que-le-espera-al-mercado-de-panama-tras-la-salida-de-compania#:~:text=compania%20Panam%C3%A1%20es%20el%20operador,y%20una%20penetraci%C3%B3n%20de%20138%25>.
- Chatterjee, S. (2021). *Designing API-First Enterprise Architectures*. Birmingham: Packt Publishing Ltd.
- Daswani, R. (2021). *AWS Certified Cloud Practitioner Exam Guide*. Birmingham: Packt Publishing Ltd.
- Deitel, P., Deitel, H., & Deitel, A. (2014). *Cómo programar Internet & World Wide Web*. México: PEARSON EDUCACIÓN.
- DENIS, A., WIXON, B. H., & ROTH, R. (2012). *SYSTEM ANALYSIS AND DESIGN*. United States of America: John Wiley & Sons, Inc. .
- Driscoll, B., Gupta, N., Vettor, R., Hirani, Z., & Tenny, L. (2013). *Entity Framework 6 Recipes*. Apress Media, LLC .
- Foster, E., & Towle Jr., B. (2022). *Software Engineering A Methodical Approach*. Boca Raton London New York: Taylor & Francis Group, LLC.
- González Guzmán, K. M., Mendoza González, A. R., & Barberena Martínez, Y. D. (2015). *Implementación de un prototipo de Sistemas en gestión de relaciones con los clientes para la Pyme durante el primer semestre del 2014*. Juigalpa, Chontales.
- Hernández Sampieri, R., Fernández Collado, C., & Baptista Lucio, M. d. (2014). *METODOLOGÍA DE LA INVESTIGACIÓN*. México: McGRAW-HILL / INTERAMERICANA EDITORES, S.A. DE C.V.
- <https://jmpovedar.files.wordpress.com/2014/03/plantillas-de-coleman.docx>. (03 de 2014). Recuperado el 01 de 02 de 2023

- huaman, W. C. (7 de Septiembre de 2018). *Medium*. Obtenido de Los 10 patrones comunes de arquitectura de software: <https://medium.com/@maniakhitoccori/los-10-patrones-comunes-de-arquitectura-de-software-d8b9047edf0b>
- huaman, W. C. (7 de Septiembre de 2018). *Medium*. Obtenido de Los 10 patrones comunes de arquitectura de software: <https://medium.com/@maniakhitoccori/los-10-patrones-comunes-de-arquitectura-de-software-d8b9047edf0b>
- LEE, R. Y. (2019). *OBJECT-ORIENTED SOFTWARE ENGINEERING WITH UML A HANDS-ON APPROACH*. New York: Nova Science Publishers, Inc.
- LMU – Ludwig-Maximilians-Universität München. (10 de 08 de 2016). *UWE – UML-based Web Engineering*. Recuperado el 29 de 01 de 2023, de <https://uwe.pst.ifi.lmu.de/teachingTutorialProcess.html>
- Ludwig-Maximilians-Universität München. (2008). *UWE Metamodel and Profile User Guide and Reference*. Germany: Ludwig-Maximilians-Universität München.
- MagicUWE - UWE Plugin for MagicDraw*. (10 de 08 de 2016). Recuperado el 02 de 02 de 2023, de LMU – Ludwig-Maximilians-Universität München: <https://uwe.pst.ifi.lmu.de/toolMagicUWE.html>
- Martin, R. C. (2018). *Clean Architecture A CRAFTSMAN'S GUIDE TO SOFTWARE STRUCTURE AND DESIGN*. United States of America: Pearson Education, Inc.
- Martínez Briones, A. Y., Centeno Medina, F. J., & Martínez Úbeda, G. T. (2020). *Análisis de la fidelización de los clientes para el incremento de ventas de la empresa Agua Roca en el II semestre del año 2020 en la ciudad de Estelí*. Estelí.
- Martínez Gutiérrez, M. G., & Pérez Sánchez, M. F. (2016). *Instrumentos de mercadeo de servicio para lograr la captación de nuevos clientes proporcionarles valor, satisfacción, retención y fidelización*. Managua.
- Meyers, M., & Jernigan, S. (2021). *Mike Meyers' CompTIA Security+™ Certification Guide*. United States: McGraw Hill.
- Microsoft. (2023). *¿Qué es .NET?* Recuperado el 01 de 02 de 2023, de <https://dotnet.microsoft.com/es-es/learn/dotnet/what-is-dotnet>
- Microsoft Learn. (20 de 12 de 2022). *¿Qué es un servidor SQL lógico en Azure SQL Database y Azure Synapse?* Recuperado el 01 de 02 de 2023, de <https://learn.microsoft.com/es-es/azure/azure-sql/database/logical-servers?view=azuresql&tabs=portal>
- Microsoft Learn. (26 de 10 de 2022). *App Service overview*. Recuperado el 26 de 01 de 2023, de <https://learn.microsoft.com/en-us/azure/app-service/overview>
- Microsoft Learn. (22 de 02 de 2022). *Run background tasks with WebJobs in Azure App Service*. Recuperado el 26 de 01 de 2023, de <https://learn.microsoft.com/en-us/azure/app-service/webjobs-create>

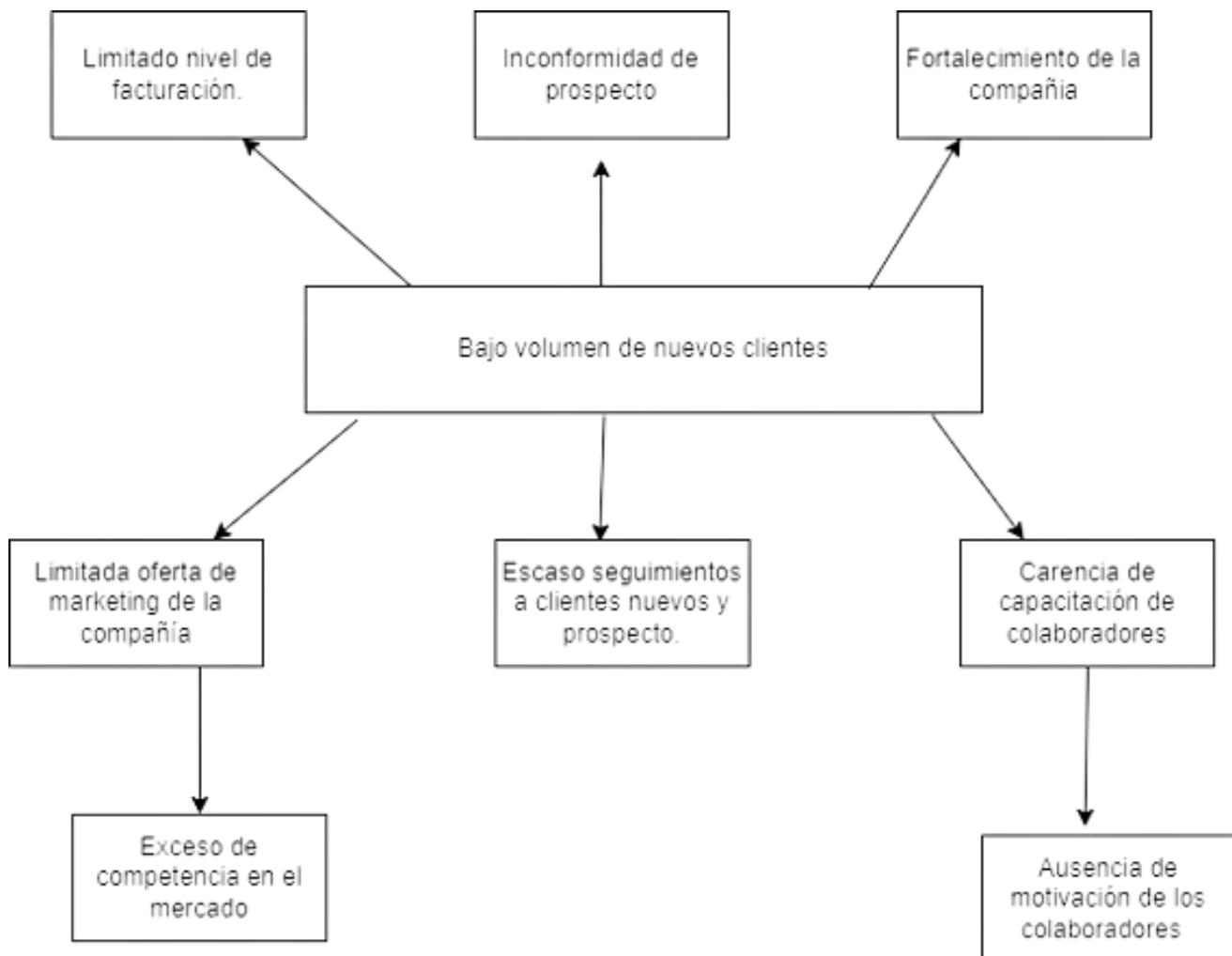
- Microsoft Learn. (31 de 01 de 2023). *How does Azure work?* Recuperado el 31 de 01 de 2023, de <https://learn.microsoft.com/en-us/azure/cloud-adoption-framework/get-started/what-is-azure>
- Microsoft Learn. (2023). *Patrón CQRS*. Recuperado el 31 de 01 de 2023, de <https://learn.microsoft.com/es-es/azure/architecture/patterns/cqrs>
- Microsoft Learn. (13 de 01 de 2023). *What is Azure SQL Database?* Recuperado el 26 de 01 de 2023, de <https://learn.microsoft.com/en-us/azure/azure-sql/database/sql-database-paas-overview?view=azuresql>
- Mikowski, M., & Powell, J. (2014). *Single Page Web Applications*. New York: Manning Publications Co.
- Nayan, B. D. (02 de Agosto de 2021). *C# Corner*. Obtenido de <https://www.c-sharpcorner.com/article/introduction-to-clean-architecture-and-implementation-with-asp-net-core/>
- OKTA. (s.f.). *Introduction to JSON Web Tokens*. Recuperado el 30 de 01 de 2023, de <https://jwt.io/introduction>
- Osman, J. (22 de 9 de 2022). *AppMaster*. Recuperado el 01 de 02 de 2023, de Los 10 mejores marcos frontend populares para usar en 2023: <https://appmaster.io/es/blog/marcos-frontend-populares>
- Oz, E. (2008). *ADMINISTRACIÓN DE LOS SISTEMAS DE INFORMACIÓN*. Distrito Federal: Cengage Learning Editores, S.A.
- Pressman, R. S. (2010). *INGENIERÍA DEL SOFTWARE. UN ENFOQUE PRÁCTICO*. Distrito Federal: McGRAW-HILL INTERAMERICANA EDITORES, S.A. DE C.V.
- Pressman, R., & Maxim, B. (2020). *Software Engineering A PRACTITIONER'S APPROACH*. New York: McGraw-Hill Education.
- RFC. (07 de 2006). *The application/json Media Type for JavaScript Object Notation (JSON)*. Recuperado el 01 de 02 de 2023, de <https://www.rfc-editor.org/rfc/rfc4627>
- Richards, M., & Ford, N. (2020). *Fundamentals of Software Architecture*. Sebastopol, CA: O'Reilly Media, Inc.
- Santos, J. (31 de 07 de 2021). *Jorge Santos – Growth & Marketing*. Recuperado el 02 de 02 de 2023, de ¿Qué Es Y Cómo Funciona Un Programa De Referidos?: <https://jorgesantos.co/que-es-y-como-funciona-un-programa-de-referidos/>
- Singh, G. D. (2022). *CompTIA Network+ N10-008 Certification Guide*. Birmingham: Packt Publishing Ltd.
- Sommerville, I. (2011). *INGENIERÍA DE SOFTWARE*. Naucalpan de Juárez: Pearson Educación de México, S.A. de C.V.
- Stephens, R. (2015). *Beginning Software Engineering*. Indianapolis: John Wiley & Sons, Inc.

Universidad Europea. (22 de 03 de 2022). *¿Para qué sirve un gestor de base de datos?* Recuperado el 02 de 02 de 2023, de <https://universidadeuropea.com/blog/para-que-sirve-gestor-base-datos/#:~:text=Un%20sistema%20gestor%20de%20bases,en%20la%20base%20de%20datos>.

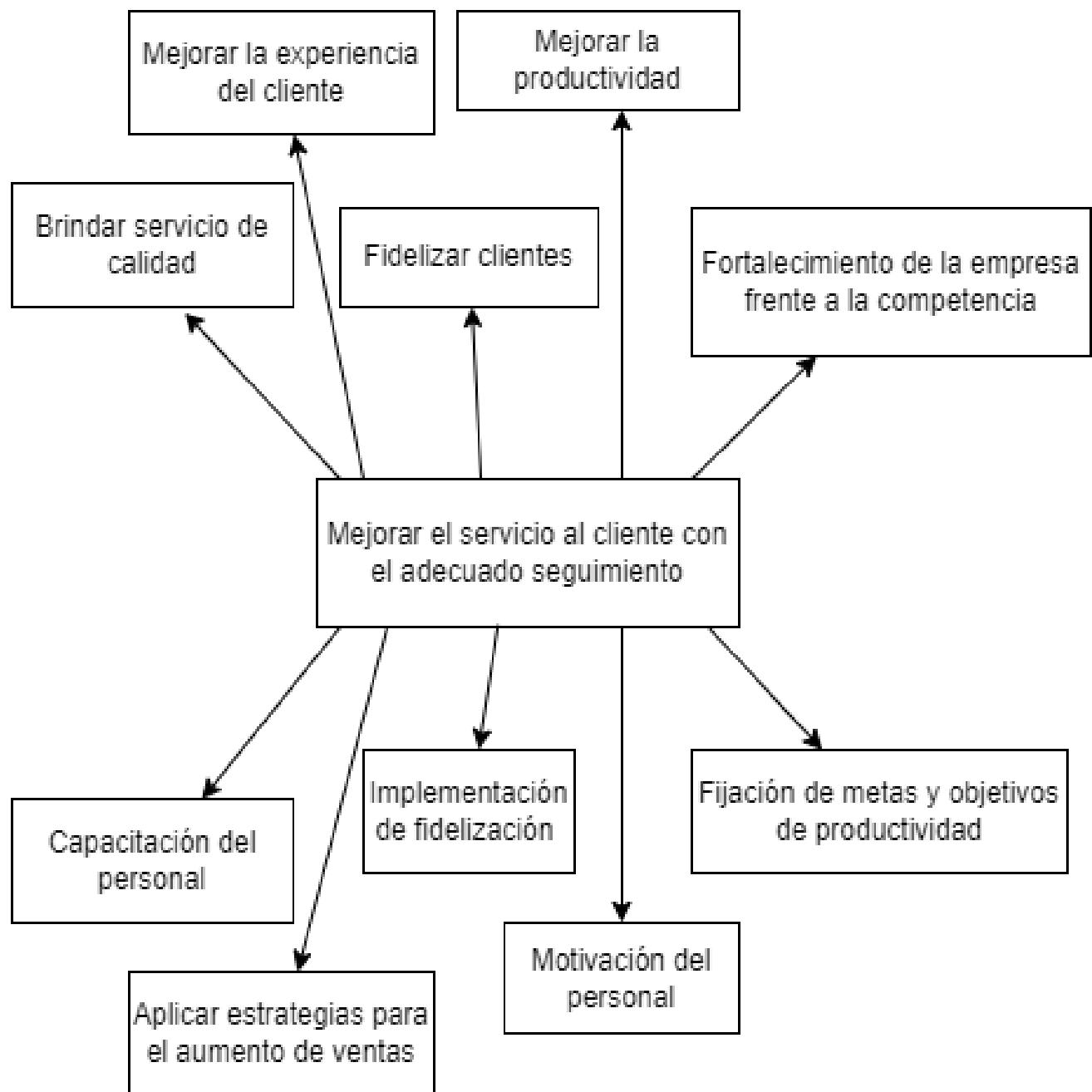
Wiegers, K., & Beatty, J. (2013). *Software Requirements, Third Edition*. Washington: Microsoft Press.

X. Anexos

Anexo1: Árbol de problemas



Anexo2: Árbol de objetivos



Anexo3: Modelo de entrevista

Somos estudiantes de V año de la carrera de Ingeniería de Sistemas.

Actualmente realizamos un proyecto para la construcción de un sistema de información Web para la gestión de fidelización de clientes.

El objetivo principal es recolectar información útil, referente al tema de investigación, el cual permita dar solución a los objetivos planteados.

Entrevista

Datos Generales: _____

Personas involucradas:

1. Cargo
2. Cargo

Preguntas:

1. ¿Cuál es el objetivo del sistema?
2. ¿Tienen preferencias de alguna plataforma o lenguaje específico?
3. ¿Qué información necesitamos recopilar?
4. ¿Será un sistema externo o estará integrado a alguno ya existente?
5. ¿Cuáles son las funcionalidades que tendrá el sistema?
6. ¿Cuáles serán las restricciones que debe cumplir el sistema?
7. ¿Existe algún estándar de desarrollo?
8. ¿Tienen preferencia por algún patrón de diseño para la construcción del sistema?
9. ¿Cuándo proveerán las definiciones de los servicios?
10. ¿Cómo se registrará el proceso de seguimiento?
11. ¿Cómo se configurarán las reglas para la premiación?
12. ¿Cómo debemos enviar los reportes?
13. ¿Qué indicadores debemos mostrar en el dashboard?
14. ¿Cómo se empleará el programa de referidos?
15. ¿Qué información necesitamos de los referentes y referidos?

Basado en (Martínez Briones, Centeno Medina, & Martínez Úbeda, 2020, págs. 65-66)

Anexos de tablas

AnexoTabla1: Plantillas REM

Requerimientos de Negocio

Tabla 17: Plantilla REM RN-04

RN-04	Gestionar configuración de recompensas
Versión	1.0 (29/07/2023)
Autores	Cristopher Josue Zepeda Tellez; Gloria Esther Mayorga Mena; Magaly Magdalena Mayorga Maradiaga
Fuentes	Ventas y Marketing (Gerente y Supervisor)
Descripción	El sistema debe estar habilitado para realizar las actividades de mostrar la información de la configuración de recompensas previamente guardadas, insertar nuevas recompensas, actualizar la información de cualquier recompensa, eliminar recompensas y también debe tener la capacidad de filtrar los registros.
Importancia	Vital
Urgencia	Inmediata
Comentarios	NA

Fuente: Elaboración propia

Tabla 18: Plantilla REM RN-05

RN-05	Mostrar indicadores en reportes
Versión	1.0 (29/07/2023)
Autores	Cristopher Josue Zepeda Tellez; Gloria Esther Mayorga Mena; Magaly Magdalena Mayorga Maradiaga
Fuentes	Ventas y Marketing (Gerente y Supervisor)
Descripción	El sistema debe mostrar indicadores en la pestaña de reportes, estos incluyen uso de los motivos de declinación, referidos ingresados y referidos aceptados, recompensas aplicas y pendientes, ingreso de referidos por día, todos estos indicadores se deben poder filtrar por un rango de fechas, distribuidor, vendedor estado del referido y si es pospago o prepago
Importancia	Conveniente
Urgencia	Necesario
Comentarios	NA

Fuente: Elaboración propia

Tabla 19: Plantilla REM RN-06

RN-06	Cambiar visibilidad de los referidos
Versión	1.0 (29/07/2023)
Autores	Cristopher Josue Zepeda Tellez; Gloria Esther Mayorga Mena; Magaly Magdalena Mayorga Maradiaga
Fuentes	Marketing (Gerente y Supervisor)

Descripción	Inicialmente los referidos son visibles únicamente para el vendedor que lo registro, sin embargo, posterior a 30 días de su registro los referidos deben ser visibles para todos los vendedores y distribuidores
Importancia	Vital
Urgencia	Inmediata
Comentarios	NA

Fuente: Elaboración propia

Tabla 20: Plantilla REM RN-07

RN-07	Actualizar estado de los referidos
Versión	1.0 (29/07/2023)
Autores	Cristopher Josue Zepeda Tellez; Gloria Esther Mayorga Mena; Magaly Magdalena Mayorga Maradiaga
Fuentes	Marketing (Gerente y Supervisor)
Descripción	Inicialmente el estado de un referido es Ingresado, sin embargo, cuando un cliente ha adquirido una línea telefónica debemos actualizar el estado a Aceptado. Esto se realizará a través de los procesos automáticos.
Importancia	Vital
Urgencia	Necesario
Comentarios	NA

Fuente: Elaboración propia

Tabla 21: Plantilla REM RN-08

RN-08	Sincronizar referidos nuevos con la plataforma de Datafone
Versión	1.0 (29/07/2023)
Autores	Cristopher Josue Zepeda Tellez; Gloria Esther Mayorga Mena; Magaly Magdalena Mayorga Maradiaga
Fuentes	Desarrollo (Gerente y Supervisor)
Descripción	El sistema debe sincronizar de forma automática el listado de referidos nuevos, esto se debe realizar a través de la carga de un archivo con formato CSV y utilizando el separador “ ”.
Importancia	Importante
Urgencia	Necesario
Comentarios	NA

Fuente: Elaboración propia

Tabla 22: Plantilla REM RN-09

RN-09	Enviar reporte de vendedores a distribuidores
Versión	1.0 (29/07/2023)
Autores	Cristopher Josue Zepeda Tellez; Gloria Esther Mayorga Mena; Magaly Magdalena Mayorga Maradiaga
Fuentes	Marketing (Gerente y Supervisor)
Descripción	El sistema debe enviar el reporte de los referidos registrados durante el día, este reporte se debe

	enviar únicamente a los distribuidores. El reporte debe incluir la información de recompensas aplicadas en el último periodo.
Importancia	Importante
Urgencia	Puede esperar
Comentarios	NA

Fuente: Elaboración propia

Tabla 23: Plantilla REM RN-10

RN-10	Ejecutar proceso de recompensas
Versión	1.0 (29/07/2023)
Autores	Cristopher Josue Zepeda Tellez; Gloria Esther Mayorga Mena; Magaly Magdalena Mayorga Maradiaga
Fuentes	Desarrollo (Gerente y Supervisor)
Descripción	El sistema debe ejecutar de forma automática el proceso de recompensas, considerando únicamente los referidos que han sido activados, evaluar las reglas de recompensas y aplicarlas cuando sea necesario.
Importancia	Vital
Urgencia	Inmediata
Comentarios	NA

Fuente: Elaboración propia

Requerimientos Funcionales

Tabla 24: Plantilla REM RF-01

RF-01	Iniciar sesión con doble autenticación
Versión	1.0 (29/07/2023)
Autores	Cristopher Josue Zepeda Tellez; Gloria Esther Mayorga Mena; Magaly Magdalena Mayorga Maradiaga
Fuentes	Desarrollo (Gerente y Supervisor)
Descripción	Para acceder al sistema se debe realizar un proceso de autenticación indicando correo electrónico y contraseña, el sistema validará las credenciales y si estas son válidas se procederá a enviar el segundo factor de autenticación para validar la autenticidad del acceso.
Importancia	Vital
Urgencia	Inmediata
Comentarios	NA

Fuente: Elaboración propia

Tabla 25: Plantilla REM RF-02

RF-02	Recuperar contraseña
Versión	1.0 (29/07/2023)
Autores	Cristopher Josue Zepeda Tellez; Gloria Esther Mayorga Mena; Magaly Magdalena Mayorga Maradiaga
Fuentes	Desarrollo, Ventas y Marketing (Gerente y Supervisor)
Descripción	Los usuarios deben tener la posibilidad de reestablecer su contraseña, para ello recibirán un correo electrónico de acceso único en el que podrán establecer una nueva contraseña.
Importancia	Importante
Urgencia	Necesario
Comentarios	NA

Fuente: Elaboración propia

Tabla 26: Plantilla REM RF-03

RF-03	Cargar menu de navegación basado en roles
Versión	1.0 (29/07/2023)
Autores	Cristopher Josue Zepeda Tellez; Gloria Esther Mayorga Mena; Magaly Magdalena Mayorga Maradiaga
Fuentes	Desarrollo, Ventas y Marketing (Gerente y Supervisor)
Descripción	El menu de opciones debe ser completamente basado en el rol, y este debe ser configurable desde la base de datos, es decir, para aplicar un nuevo permiso a un usuario no será necesario que se recompile y publique el código nuevamente.
Importancia	Vital
Urgencia	Inmediata
Comentarios	NA

Fuente: Elaboración propia

Tabla 27: Plantilla REM RF-04

RF-04	Crear conexión a la base de datos
Versión	1.0 (29/07/2023)
Autores	Cristopher Josue Zepeda Tellez; Gloria Esther Mayorga Mena; Magaly Magdalena Mayorga Maradiaga
Fuentes	Desarrollo (Gerente y Supervisor)
Descripción	El sistema debe estar preparado para crear una conexión a base de datos tipo SQL, y esta debe poder cambiarse por ambiente (Desarrollo, UAT y Producción)
Importancia	Vital
Urgencia	Inmediata
Comentarios	NA

Fuente: Elaboración propia

Tabla 28: Plantilla REM RF-05

RF-05	Crear conexión a SFTP
Versión	1.0 (29/07/2023)
Autores	Cristopher Josue Zepeda Tellez; Gloria Esther Mayorga Mena; Magaly Magdalena Mayorga Maradiaga
Fuentes	Desarrollo (Gerente y Supervisor)
Descripción	Para realizar las actividades de cargar o descargar archivos en un SFTP es necesario abrir una conexión, y posterior a su uso debemos cerrar la conexión.
Importancia	Vital
Urgencia	Necesario
Comentarios	NA

Fuente: Elaboración propia

Tabla 29: Plantilla REM RF-06

RF-06	Enviar correo electrónico
Versión	1.0 (29/07/2023)
Autores	Cristopher Josue Zepeda Tellez; Gloria Esther Mayorga Mena; Magaly Magdalena Mayorga Maradiaga
Fuentes	Desarrollo (Gerente y Supervisor)
Descripción	Para realizar el envío de reporte a los distribuidores es necesario crear un cliente SMTP con los parámetros en el appsettings
Importancia	Importante
Urgencia	Necesario
Comentarios	NA

Fuente: Elaboración propia

Tabla 30: Plantilla REM RF-07

RF-07	Mostrar nombre de usuario y correo electrónico del usuario
Versión	1.0 (29/07/2023)
Autores	Cristopher Josue Zepeda Tellez; Gloria Esther Mayorga Mena; Magaly Magdalena Mayorga Maradiaga
Fuentes	Ventas (Gerente y Supervisor)
Descripción	Una vez el usuario accedió a la plataforma, en el menu principal se debe mostrar el nombre de usuario y el correo electrónico del mismo.
Importancia	Conveniente
Urgencia	Inmediata
Comentarios	NA

Fuente: Elaboración propia

Tabla 31: Plantilla REM RF-08

RF-08	Mostrar los referidos basados en el rol
Versión	1.0 (29/07/2023)
Autores	Cristopher Josue Zepeda Tellez; Gloria Esther Mayorga Mena; Magaly Magdalena Mayorga Maradiaga

Fuentes	Ventas (Gerente y Supervisor)
Descripción	El sistema debe mostrar los referidos basados en el rol, es decir, un vendedor puede únicamente ver los referidos que registro, un distribuidor puede ver los referidos registrados de todos los vendedores y el administrador puede ver todos los referidos
Importancia	Vital
Urgencia	Inmediata
Comentarios	NA

Fuente: Elaboración propia

Tabla 32: Plantilla REM RF-09

RF-09	Cerrar sesión
Versión	1.0 (29/07/2023)
Autores	Cristopher Josue Zepeda Tellez; Gloria Esther Mayorga Mena; Magaly Magdalena Mayorga Maradiaga
Fuentes	Desarrollo (Gerente y Supervisor)
Descripción	Todos los usuarios deben poder cerrar sesión
Importancia	Vital
Urgencia	Inmediata
Comentarios	NA

Fuente: Elaboración propia

Tabla 33: Plantilla REM RF-10

RF-10	Consumir servicios externos para recargar
Versión	1.0 (29/07/2023)
Autores	Cristopher Josue Zepeda Tellez; Gloria Esther Mayorga Mena; Magaly Magdalena Mayorga Maradiaga
Fuentes	Desarrollo (Gerente y Supervisor)
Descripción	Cuando la recompensa es de tipo prepago, se deben consumir los servicios de verificación y recarga telefónica.
Importancia	Vital
Urgencia	Inmediata
Comentarios	NA

Fuente: Elaboración propia

Tabla 34: Plantilla REM RF-11

RF-11	Consumir servicios externos para abonar a un plan
Versión	1.0 (29/07/2023)
Autores	Cristopher Josue Zepeda Tellez; Gloria Esther Mayorga Mena; Magaly Magdalena Mayorga Maradiaga
Fuentes	Desarrollo (Gerente y Supervisor)
Descripción	Cuando la recompensa es de tipo pospago, se deben consumir los servicios de verificación y acreditación a un plan
Importancia	Vital
Urgencia	Inmediata

Comentarios	NA
-------------	----

Fuente: Elaboración propia

Tabla 35: Plantilla REM RF-12

RF-12	Recordar usuario
Versión	1.0 (29/07/2023)
Autores	Cristopher Josue Zepeda Tellez; Gloria Esther Mayorga Mena; Magaly Magdalena Mayorga Maradiaga
Fuentes	Marketing (Gerente y Supervisor)
Descripción	El usuario debe tener la facilidad de marcar una opción para recordar su usuario y contraseña en el navegador, y que este no las esté ingresando a cada momento
Importancia	Importante
Urgencia	Puede esperar
Comentarios	NA

Fuente: Elaboración propia

Tabla 36: Plantilla REM RF-13

RF-13	Mostrar cantidad de registros en las pestañas
Versión	1.0 (29/07/2023)
Autores	Cristopher Josue Zepeda Tellez; Gloria Esther Mayorga Mena; Magaly Magdalena Mayorga Maradiaga
Fuentes	Ventas y Marketing (Gerente y Supervisor)
Descripción	El sistema debe establecer de forma automática en cada pestaña el número de registros almacenados
Importancia	Vital
Urgencia	Inmediata
Comentarios	NA

Fuente: Elaboración propia

Requerimientos No Funcionales

Tabla 37: Plantilla REM RNF-01

RNF-01	Guardar registros de los archivos descargados en base de datos
Versión	1.0 (29/07/2023)
Autores	Cristopher Josue Zepeda Tellez; Gloria Esther Mayorga Mena; Magaly Magdalena Mayorga Maradiaga
Fuentes	Desarrollo (Gerente y Supervisor)
Descripción	Para realizar validaciones de las recompensas aplicadas y estado de los referidos, es necesario almacenar una copia en una tabla SQL del archivo que recibimos del SFTP
Importancia	Importante
Urgencia	Puede esperar
Comentarios	NA

Fuente: Elaboración propia

Tabla 38: Plantilla REM RNF-02

RNF-02	BackEnd se debe realizar utilizando patrón de diseño CQRS
Versión	1.0 (29/07/2023)
Autores	Cristopher Josue Zepeda Tellez; Gloria Esther Mayorga Mena; Magaly Magdalena Mayorga Maradiaga
Fuentes	Desarrollo (Gerente y Supervisor)
Descripción	El API debe emplear un patrón de diseño llamado CQRS, el que consiste en separar los modelos de afectación de registros en la base de datos y los modelos para realizar consultas a la base de datos.
Importancia	Vital
Urgencia	Inmediata
Comentarios	NA

Fuente: Elaboración propia

Tabla 39: Plantilla REM RNF-03

RNF-03	FrontEnd se debe realizar con Angular
Versión	1.0 (29/07/2023)
Autores	Cristopher Josue Zepeda Tellez; Gloria Esther Mayorga Mena; Magaly Magdalena Mayorga Maradiaga
Fuentes	Desarrollo (Gerente y Supervisor)
Descripción	La tecnología a utilizar en FrontEnd debe ser Angular, utilizando la versión 13. Este es un framework utilizado por el equipo de desarrollo de Datafone, de esta forma les será sencillo realizar ajustes o mejoras al código
Importancia	Vital
Urgencia	Inmediata
Comentarios	NA

Fuente: Elaboración propia

Tabla 40: Plantilla REM RNF-04

RNF-04	BackEnd se debe realizar con .NET 7
Versión	1.0 (29/07/2023)
Autores	Cristopher Josue Zepeda Tellez; Gloria Esther Mayorga Mena; Magaly Magdalena Mayorga Maradiaga
Fuentes	Desarrollo (Gerente y Supervisor)
Descripción	La tecnología a utilizar en BackEnd debe ser .NET, utilizando la versión 7. Este es un framework utilizado por el equipo de desarrollo de Datafone, de esta forma les será sencillo realizar ajustes o mejoras al código
Importancia	Vital
Urgencia	Inmediata
Comentarios	NA

Fuente: Elaboración propia

Tabla 41: Plantilla REM RNF-05

RNF-05	Reglas de seguridad para establecer una contraseña
Versión	1.0 (29/07/2023)
Autores	Cristopher Josue Zepeda Tellez; Gloria Esther Mayorga Mena; Magaly Magdalena Mayorga Maradiaga
Fuentes	Desarrollo (Gerente y Supervisor)
Descripción	<p>La contraseña debe cumplir con los siguientes parámetros:</p> <ol style="list-style-type: none"> 1. Longitud mayor o igual a 8 caracteres 2. Debe incluir una mayúscula 3. Debe incluir un número 4. Debe incluir un carácter especial <p>No debe contener secuencias de letras o números mayor a 3</p>
Importancia	Vital
Urgencia	Inmediata
Comentarios	NA

Fuente: Elaboración propia

Tabla 42: Plantilla REM RNF-06

RNF-06	Cifrado de contraseña
Versión	1.0 (29/07/2023)
Autores	Cristopher Josue Zepeda Tellez; Gloria Esther Mayorga Mena; Magaly Magdalena Mayorga Maradiaga
Fuentes	Desarrollo (Gerente y Supervisor)
Descripción	Toda contraseña almacenada en las bases de datos debe estar cifrada, así como también la contraseña enviada desde el front-end hacia el back-end.
Importancia	Vital
Urgencia	Inmediata
Comentarios	NA

Fuente: Elaboración propia

Tabla 43: Plantilla REM RNF-07

RNF-07	Plataforma web debe poder utilizarse en cualquier dispositivo
Versión	1.0 (29/07/2023)
Autores	Cristopher Josue Zepeda Tellez; Gloria Esther Mayorga Mena; Magaly Magdalena Mayorga Maradiaga
Fuentes	Desarrollo (Gerente y Supervisor)
Descripción	Todas las pantallas deben ser adaptables al tamaño de la pantalla del dispositivo, en el caso de las pantallas pequeñas las tablas deben mostrar un máximo de 3 columnas y las acciones que se deben ejecutar

Importancia	Vital
Urgencia	Inmediata
Comentarios	NA

Fuente: Elaboración propia

Tabla 44: Plantilla REM RNF-08

RNF-08	Servicios externos serán de tipo WCF
Versión	1.0 (29/07/2023)
Autores	Cristopher Josue Zepeda Tellez; Gloria Esther Mayorga Mena; Magaly Magdalena Mayorga Maradiaga
Fuentes	Desarrollo (Gerente y Supervisor)
Descripción	Toda comunicación entre el sistema web y el servicio de Datafone se realizará a través de servicios WCF. Para ello se debe crear un cliente de forma dinámica previo al consumo del servicio
Importancia	Vital
Urgencia	Inmediata
Comentarios	NA

Fuente: Elaboración propia

Tabla 45: Plantilla REM RNF-09

RNF-09	La aplicación se desplegará en una App Service de Azure
Versión	1.0 (29/07/2023)
Autores	Cristopher Josue Zepeda Tellez; Gloria Esther Mayorga Mena; Magaly Magdalena Mayorga Maradiaga
Fuentes	Desarrollo (Gerente y Supervisor)
Descripción	El sistema web será desplegado en el ambiente de Azure, para ello se utilizará un App Service (uno para UAT y otro para PROD), dado que es un SPA solo será necesario realizar un único despliegue
Importancia	Vital
Urgencia	Inmediata
Comentarios	NA

Fuente: Elaboración propia

Requerimientos de Implementación

Tabla 46: Plantilla REM RI-01

RI-01	Crear manual de usuario
Versión	1.0 (29/07/2023)
Autores	Cristopher Josue Zepeda Tellez; Gloria Esther Mayorga Mena; Magaly Magdalena Mayorga Maradiaga
Fuentes	Ventas (Gerente y Supervisor)
Descripción	Una vez completadas las pruebas de usuario, y los gerentes de desarrollo, ventas y marketing hayan aprobado el sistema, se debe crear un manual de

	usuario sobre las funcionalidades que el sistema contempla
Importancia	Importante
Urgencia	Puede esperar
Comentarios	NA

Fuente: Elaboración propia

Tabla 47: Plantilla REM RI-02

RI-02	Crear la cuenta en Azure
Versión	1.0 (29/07/2023)
Autores	Cristopher Josue Zepeda Tellez; Gloria Esther Mayorga Mena; Magaly Magdalena Mayorga Maradiaga
Fuentes	Desarrollo (Gerente y Supervisor)
Descripción	El sistema estará desplegado en un ambiente en la nube, es por ello que es necesario crear una cuenta en Azure
Importancia	Vital
Urgencia	Inmediata
Comentarios	NA

Fuente: Elaboración propia

Tabla 48: Plantilla REM RI-03

RI-03	Crear las subscripciones de QA y Producción en Azure
Versión	1.0 (29/07/2023)
Autores	Cristopher Josue Zepeda Tellez; Gloria Esther Mayorga Mena; Magaly Magdalena Mayorga Maradiaga
Fuentes	Desarrollo (Gerente y Supervisor)
Descripción	Se debe tener control sobre los costos empleados por ambientes de QA y Producción, es por ello que es necesario crear dos subscripciones, una para el ambiente de QA y otra para el ambiente productivo
Importancia	Vital
Urgencia	Inmediata
Comentarios	NA

Fuente: Elaboración propia

Tabla 49: Plantilla REM RI-04

RI-04	Crear Azure SQL Database
Versión	1.0 (29/07/2023)
Autores	Cristopher Josue Zepeda Tellez; Gloria Esther Mayorga Mena; Magaly Magdalena Mayorga Maradiaga
Fuentes	Desarrollo (Gerente y Supervisor)
Descripción	Para el almacenamiento de los datos se utilizará Azure SQL Database, por lo que debemos crear dos Azure SQL Database una para el ambiente de QA y otra para el ambiente de Producción. Al momento de crear las bases de datos será necesario crear un servidor lógico y la región debe ser EAST-US
Importancia	Vital

Urgencia	Inmediata
Comentarios	NA

Fuente: Elaboración propia

Tabla 50: Plantilla REM RI-05

RI-05	Crear App Service
Versión	1.0 (29/07/2023)
Autores	Cristopher Josue Zepeda Tellez; Gloria Esther Mayorga Mena; Magaly Magdalena Mayorga Maradiaga
Fuentes	Desarrollo (Gerente y Supervisor)
Descripción	La aplicación web será desplegada en un App Service en Azure, es necesario crear un App Service para cada ambiente. Cada App Service debe tener configurado el entorno de .NET 7, se debe publicar en formato de código y la región debe ser EAST-US
Importancia	Vital
Urgencia	Inmediata
Comentarios	NA

Fuente: Elaboración propia

Tabla 51: Plantilla REM RI-06

RI-06	Desplegar WebJobs
Versión	1.0 (29/07/2023)
Autores	Cristopher Josue Zepeda Tellez; Gloria Esther Mayorga Mena; Magaly Magdalena Mayorga Maradiaga
Fuentes	Desarrollo (Gerente y Supervisor)
Descripción	Todos los procesos calendarizados estarán en diferentes WebJobs, estos serán desplegados desde la herramienta de visual studio y es necesario el App Service tenga configurada la opción “Siempre Activado” en Activado
Importancia	Vital
Urgencia	Inmediata
Comentarios	NA

Fuente: Elaboración propia

Requerimientos de Usuario

Tabla 52: Plantilla REM RU-01

RU-01	Establecer estado por defecto a referidos
Versión	1.0 (29/07/2023)
Autores	Cristopher Josue Zepeda Tellez; Gloria Esther Mayorga Mena; Magaly Magdalena Mayorga Maradiaga
Fuentes	Ventas (Gerente y Supervisor)
Descripción	Cada vez que se registre un nuevo referido, su estado por defecto debe ser Ingresado
Importancia	Vital

Urgencia	Inmediata
Comentarios	NA

Fuente: Elaboración propia

Tabla 53: Plantilla REM RU-02

RU-02	Mostrar restricciones de UI en recompensas
Versión	1.0 (29/07/2023)
Autores	Cristopher Josue Zepeda Tellez; Gloria Esther Mayorga Mena; Magaly Magdalena Mayorga Maradiaga
Fuentes	Ventas (Gerente y Supervisor)
Descripción	En la pestaña de recompensas se deben mostrar algunos campos basados en el tipo de referente y tipo de referido, el sistema debe controlar esto en base a la selección del usuario
Importancia	Vital
Urgencia	Inmediata
Comentarios	NA

Fuente: Elaboración propia

AnexoTabla2: Plantillas de Coleman

Tabla 54: Plantilla Coleman RN-04

Caso de Uso	RN-04: Gestionar configuración de recompensas		
DEFINICION:	El sistema debe estar habilitado para realizar las actividades de mostrar la información de la configuración de recompensas previamente guardadas, insertar nuevas recompensas, actualizar la información de cualquier recompensa, eliminar recompensas y también debe tener la capacidad de filtrar los registros.		
PRIORIDAD:	<input checked="" type="checkbox"/> Vital	<input type="checkbox"/> Importante	<input type="checkbox"/> Conveniente
URGENCIA:	<input checked="" type="checkbox"/> Inmediata	<input type="checkbox"/> Necesario	<input type="checkbox"/> Puede esperar
AUTORES:	Cristopher Josue Zepeda Tellez Gloria Esther Mayorga Mena Magaly Magdalena Mayorga Maradiaga		
ACTORES			
NOMBRE	DEFINICION		
Administrador, Marketing	Los diferentes roles de usuario tendrán acceso a la opción de gestión de configuración de recompensas.		
ESCENARIO			
Nombre:	Mostrar configuración de recompensas		
Precondiciones:	<ul style="list-style-type: none"> • El usuario debió haber iniciado sesión • El usuario debe tener acceso a la opción 		
Iniciado por:	Usuario		
Finalizado por:	Sistema		
Post-condiciones:	Mostrar listado de configuraciones de recompensas activas al acceder a la pestaña		
Operaciones:	<ol style="list-style-type: none"> 1. Acceder a la url del sistema web 2. Ingresar las credenciales del usuario 3. Realizar proceso de segundo factor de autenticación 4. Ingresar a pestaña recompensas 5. El sistema consulta las configuraciones de recompensas activas 6. El sistema muestra las configuraciones de recompensas 		
Excepciones:	No hay configuraciones que mostrar		
ESCENARIO			
Nombre:	Filtrar configuración de recompensas		
Precondiciones:	<ul style="list-style-type: none"> • El usuario debió haber iniciado sesión • El usuario debe tener acceso a la opción • Debe existir al menos 1 registro 		
Iniciado por:	Usuario		
Finalizado por:	Sistema		
Post-condiciones:	Mostrar listado de configuraciones de recompensas activas al acceder a la pestaña		
Operaciones:	<ol style="list-style-type: none"> 1. Acceder a la url del sistema web 2. Ingresar las credenciales del usuario 3. Realizar proceso de segundo factor de autenticación 		

	4. Ingresar a pestaña recompensas 5. El sistema consulta las configuraciones de recompensas activas 6. El sistema muestra las configuraciones de recompensas 7. Establecer texto de búsqueda en caja de texto 8. Accionar botón de búsqueda 9. Buscar configuraciones de recompensas basado en el filtro 10. Mostrar configuraciones
Excepciones:	No hay configuraciones que mostrar
ESCENARIO	
Nombre:	Agregar configuración de recompensas
Precondiciones:	<ul style="list-style-type: none"> • El usuario debió haber iniciado sesión • El usuario debe tener acceso a la opción
Iniciado por:	Usuario
Finalizado por:	Sistema
Post-condiciones:	Mostrar listado de configuraciones de recompensas actualizado
Operaciones:	1. Acceder a la url del sistema web 2. Ingresar las credenciales del usuario 3. Realizar proceso de segundo factor de autenticación 4. Ingresar a pestaña recompensas 5. El sistema consulta las configuraciones de recompensas activas 6. El sistema muestra las configuraciones de recompensas 7. Accionar botón agregar 8. Ingresar información de recompensa 9. Accionar botón guardar 10. El sistema valida la información ingresada 11. El sistema ingresa la información a la base de datos 12. Se muestra el listado de configuraciones actualizado
Excepciones:	<ul style="list-style-type: none"> • Campos obligatorios faltantes • Existe otra recompensa con la misma configuración
ESCENARIO	
Nombre:	Actualizar configuración de recompensas
Precondiciones:	<ul style="list-style-type: none"> • El usuario debió haber iniciado sesión • El usuario debe tener acceso a la opción • El sistema debe precargar la información de la configuración seleccionada
Iniciado por:	Usuario
Finalizado por:	Sistema
Post-condiciones:	Mostrar listado de configuraciones de recompensas actualizado
Operaciones:	1. Acceder a la url del sistema web 2. Ingresar las credenciales del usuario 3. Realizar proceso de segundo factor de autenticación 4. Ingresar a pestaña recompensas 5. El sistema consulta las configuraciones de recompensas activas

	<ol style="list-style-type: none"> 6. El sistema muestra las configuraciones de recompensas 7. Seleccionar configuración a actualizar 8. Ingresar información recompensas 9. Accionar botón actualizar 10. El sistema valida la información ingresada 11. El sistema actualiza la información en la base de datos 12. Se retorna el nuevo listado de configuraciones 13. Se muestra el nuevo listado
Excepciones:	<ul style="list-style-type: none"> • Campos obligatorios faltantes • Existe otra recompensa con la misma configuración
ESCENARIO	
Nombre:	Eliminar configuración de recompensas
Precondiciones:	<ul style="list-style-type: none"> • El usuario debió haber iniciado sesión • El usuario debe tener acceso a la opción • Debe existir al menos un 1 registro
Iniciado por:	Usuario
Finalizado por:	Sistema
Post-condiciones:	Mostrar listado de motivos de declinación activos
Operaciones:	<ol style="list-style-type: none"> 1. Acceder a la url del sistema web 2. Ingresar las credenciales del usuario 3. Realizar proceso de segundo factor de autenticación 4. Ingresar a pestaña recompensas 5. El sistema consulta los motivos de declinación 6. El sistema muestra los motivos de declinación 7. Seleccionar registro a eliminar 8. Accionar botón eliminar 9. Confirmar eliminación de registro 10. El sistema valida la información ingresada 11. El sistema actualiza la información a la base de datos 12. Se retorna nuevo listado de configuraciones 13. Se muestra el nuevo listado
Excepciones:	Configuración no existe

Fuente: Elaboración propia

Tabla 55: Plantilla Coleman RN-05

Caso de Uso	RN-05: Mostrar indicadores en reportes.		
DEFINICION:	EL sistema debe mostrar indicadores en la pestaña de reportes, estos incluyen uso de los motivos de declinación, referidos ingresados y referidos aceptados, recompensas aplicas y pendientes, ingreso de referidos por día, todos estos indicadores se deben poder filtrar por un rango de fechas, distribuidor, vendedor estado del referido y si es pospago o prepago		
PRIORIDAD:	<input type="checkbox"/> Vital	<input type="checkbox"/> Importante	<input checked="" type="checkbox"/> Conveniente
URGENCIA:	<input type="checkbox"/> Inmediata	<input checked="" type="checkbox"/> Necesario	<input type="checkbox"/> Puede esperar
AUTORES:	Cristopher Josue Zepeda Tellez Gloria Esther Mayorga Mena		

	Magaly Magdalena Mayorga Maradiaga
ACTORES	
NOMBRE	DEFINICION
Administrador, Distribuidor y Marketing	Los diferentes roles de usuario tendrán acceso a la opción de reportes.
ESCENARIO	
Nombre:	Mostrar indicadores
Precondiciones:	<ul style="list-style-type: none"> • El usuario debió haber iniciado sesión • El usuario debe tener acceso a la opción
Iniciado por:	Usuario
Finalizado por:	Sistema
Post-condiciones:	Todos los indicadores son mostrados
Operaciones:	<ol style="list-style-type: none"> 1. Acceder a la url del sistema web 2. Ingresar las credenciales del usuario 3. Realizar proceso de segundo factor de autenticación 4. Ingresar a pestaña reporte 5. El sistema obtiene la información de referidos, vendedores, recompensas 6. Mostrar información en pantalla
Excepciones:	No hay información para mostrar
ESCENARIO	
Nombre:	Filtrar indicadores
Precondiciones:	<ul style="list-style-type: none"> • El usuario debió haber iniciado sesión • El usuario debe tener acceso a la opción
Iniciado por:	Usuario
Finalizado por:	Sistema
Post-condiciones:	Mostrar la información basada en los indicadores
Operaciones:	<ol style="list-style-type: none"> 1. Acceder a la url del sistema web 2. Ingresar las credenciales del usuario 3. Realizar proceso de segundo factor de autenticación 4. Ingresar a pestaña reporte 5. Indicar información en los filtros 6. Accionar botón filtrar 7. El sistema busca la información basada en los filtros establecidos 8. El sistema muestra la información obtenida
Excepciones:	No hay información para mostrar

Fuente: Elaboración propia

Tabla 56: Plantilla Coleman RN-06

Caso de Uso	RN-06: Cambiar visibilidad de los referidos
DEFINICION:	Inicialmente los referidos son visibles únicamente para el vendedor que lo registro, sin embargo, posterior a 30 días de su

	registro los referidos deben ser visibles para todos los vendedores y distribuidores		
PRIORIDAD:	<input checked="" type="checkbox"/> Vital	<input type="checkbox"/> Importante	<input type="checkbox"/> Conveniente
URGENCIA:	<input checked="" type="checkbox"/> Inmediata	<input type="checkbox"/> Necesario	<input type="checkbox"/> Puede esperar
AUTORES:	Cristopher Josue Zepeda Tellez Gloria Esther Mayorga Mena Magaly Magdalena Mayorga Maradiaga		
ACTORES			
NOMBRE	DEFINICION		
Sistema	Es un proceso que se ejecutará de forma automática todos los días		
ESCENARIO			
Nombre:	Cambiar visibilidad		
Precondiciones:	<ul style="list-style-type: none"> El AppService debe estar configurado para no apagarse El WebJob debe tener configurado la url del api El WebJob debe tener configurado una expresión CRON para las 8:30pm 		
Iniciado por:	Sistema		
Finalizado por:	Sistema		
Post-condiciones:	Los referidos que cumplen la condición deben ser visibles para todos los vendedores y distribuidores		
Operaciones:	<ol style="list-style-type: none"> El WebJob se ejecuta automáticamente en la hora especificada por el CRON El sistema obtiene el listado de referidos que tengan el flag VisibleForAll en false y la fecha de registro es superior a 30 días El sistema actualiza el campo VisibleForAll en true El sistema actualiza de forma masiva los registros afectados El sistema retorna la cantidad de registros afectados 		
Excepciones:	No hay referidos con fecha mayor a 30 días de su registro		

Fuente: Elaboración propia

Tabla 57: Plantilla RN-07

Caso de Uso	RN-07: Actualizar estado de los referidos		
DEFINICION:	Inicialmente el estado de un referido es Ingresado, sin embargo, cuando un cliente ha adquirido una línea telefónica debemos actualizar el estado ha Aceptado. Esto se realizará a través de		
PRIORIDAD:	<input checked="" type="checkbox"/> Vital	<input type="checkbox"/> Importante	<input type="checkbox"/> Conveniente
URGENCIA:	<input type="checkbox"/> Inmediata	<input checked="" type="checkbox"/> Necesario	<input type="checkbox"/> Puede esperar
AUTORES:	Cristopher Josue Zepeda Tellez Gloria Esther Mayorga Mena Magaly Magdalena Mayorga Maradiaga		
ACTORES			
NOMBRE	DEFINICION		
Sistema	Es un proceso que se ejecutará de forma automática todos los días		

ESCENARIO	
Nombre:	Actualizar estado de referidos
Precondiciones:	<ul style="list-style-type: none"> El AppService debe estar configurado para no apagarse El WebJob debe tener configurado la url del api El WebJob debe tener configurado una expresión CRON para las 8:30pm El WebJob debe tener configurado la ubicación para la descarga del archivo El archivo a descargar debe cumplir la siguiente nomenclatura: REFERIDOS_yyyyMMdd.dat
Iniciado por:	Sistema
Finalizado por:	Sistema
Post-condiciones:	El estado de los referidos es actualizado
Operaciones:	<ol style="list-style-type: none"> El WebJob se ejecuta automáticamente en la hora especificada por el CRON Abrir conexión a SFTP Establecer nombre del archivo por día de ejecución Descargar archivo en la ruta especificada Crear copia del archivo en carpeta de históricos Leer archivo Transformar archivo en variable tipo tabla Ejecutar proceso en BD para la actualización de estados
Excepciones:	<ul style="list-style-type: none"> La conexión al SFTP no es realizada El archivo descargado no cumple con la estructura esperada El archivo CSV no está usando el separador esperado El archivo CSV está vacío

Fuente: Elaboración propia

Tabla 58: Plantilla RN-08

Caso de Uso	RN-08: Sincronizar referidos nuevos con la plataforma de Datafone		
DEFINICION:	El sistema debe sincronizar de forma automática el listado de referidos nuevos, esto se debe realizar a través de la carga de un archivo con formato CSV y utilizando el separador " ".		
PRIORIDAD:	<input type="checkbox"/> Vital	<input checked="" type="checkbox"/> Importante	<input type="checkbox"/> Conveniente
URGENCIA:	<input type="checkbox"/> Inmediata	<input checked="" type="checkbox"/> Necesario	<input type="checkbox"/> Puede esperar
AUTORES:	Christopher Josue Zepeda Tellez Gloria Esther Mayorga Mena Magaly Magdalena Mayorga Maradiaga		
ACTORES			
NOMBRE	DEFINICION		
Sistema	Es un proceso que se ejecutará de forma automática todos los días		
ESCENARIO			
Nombre:	Sincronizar referidos		
Precondiciones:	<ul style="list-style-type: none"> El AppService debe estar configurado para no apagarse 		

	<ul style="list-style-type: none"> • El WebJob debe tener configurado la url del api • El WebJob debe tener configurado una expresión CRON para las 8:30pm • El WebJob debe tener configurado la ubicación para la carga del archivo • El archivo a descargar debe cumplir la siguiente nomenclatura: REFERIDOS_yyyyMMdd.dat
Iniciado por:	Sistema
Finalizado por:	Sistema
Post-condiciones:	Se carga el archivo dat en la ubicación esperada para los nuevos referidos
Operaciones:	<ol style="list-style-type: none"> 1. El WebJob se ejecuta automáticamente en la hora especificada por el CRON 2. El sistema obtiene el listado de referidos que son activos y que no han sido enviados previamente 3. El sistema genera un archivo dat 4. El sistema agrega el listado de referidos al archivo generado 5. El sistema se conecta al SFTP 6. El sistema carga el archivo en la ubicación indicada 7. El sistema actualiza los referidos a enviado
Excepciones:	<ul style="list-style-type: none"> • La conexión al SFTP no es realizada • No hay referidos para cargar

Fuente: Elaboración propia

Tabla 59: Plantilla RN-09

Caso de Uso	RN-09: Enviar reporte de vendedores a distribuidores		
DEFINICION:	El sistema debe enviar el reporte de los referidos registrados durante el día, este reporte se debe enviar únicamente a los distribuidores. El reporte debe incluir la información de recompensas aplicadas en el último periodo.		
PRIORIDAD:	<input type="checkbox"/> Vital	<input checked="" type="checkbox"/> Importante	<input type="checkbox"/> Conveniente
URGENCIA:	<input type="checkbox"/> Inmediata	<input type="checkbox"/> Necesario	<input checked="" type="checkbox"/> Puede esperar
AUTORES:	Christopher Josue Zepeda Tellez Gloria Esther Mayorga Mena Magaly Magdalena Mayorga Maradiaga		
ACTORES			
NOMBRE	DEFINICION		
Sistema	Es un proceso que se ejecutará de forma automática todos los días		
ESCENARIO			
Nombre:	Enviar reporte		
Precondiciones:	<ul style="list-style-type: none"> • El AppService debe estar configurado para no apagarse • El WebJob debe tener configurado la url del api • El WebJob debe tener configurado una expresión CRON para las 8:30pm 		
Iniciado por:	Sistema		

Finalizado por:	Sistema
Post-condiciones:	Se envía el correo electrónico a los distribuidores
Operaciones:	<ol style="list-style-type: none"> 1. El WebJob se ejecuta automáticamente en la hora especificada por el CRON 2. El sistema obtiene el listado de distribuidores 3. El sistema obtiene el total de referidos ingresados 4. El sistema obtiene el total de referidos rechazados 5. El sistema obtiene el total de referidos activados 6. El sistema obtiene el total de recompensas aplicadas y pendientes de aplicar 7. El sistema obtiene el listado de referidos ingresados 8. El sistema mapea los datos obtenidos anteriormente en la plantilla de HTML 9. El sistema realiza el envío de correo electrónico
Excepciones:	<ul style="list-style-type: none"> • El envío de correo falló • No hay referidos para enviar • No hay información de recompensas • No hay distribuidores

Fuente: Elaboración propia

Tabla 60: Plantilla RN-10

Caso de Uso	RN-10: Ejecutar proceso de recompensas		
DEFINICION:	El sistema debe ejecutar de forma automática el proceso de recompensas, considerando únicamente los referidos que han sido activados, evaluar las reglas de recompensas y aplicarlas cuando sea necesario.		
PRIORIDAD:	<input checked="" type="checkbox"/> Vital	<input type="checkbox"/> Importante	<input type="checkbox"/> Conveniente
URGENCIA:	<input checked="" type="checkbox"/> Inmediata	<input type="checkbox"/> Necesario	<input type="checkbox"/> Puede esperar
AUTORES:	Cristopher Josue Zepeda Tellez Gloria Esther Mayorga Mena Magaly Magdalena Mayorga Maradiaga		
ACTORES			
NOMBRE	DEFINICION		
Sistema	Es un proceso que se ejecutará de forma automática todos los días		
ESCENARIO			
Nombre:	Procesar recompensas pospago		
Precondiciones:	<ul style="list-style-type: none"> • El AppService debe estar configurado para no apagarse • El WebJob debe tener configurado la url del api • El WebJob debe tener configurado una expresión CRON para ejecutarse cada miércoles a las 9:00PM • El sistema debe crear el cliente para los servicios WCF de forma dinámica • Los referidos están disponibles para aplicar recompensas • Los referidos deben estar activos 		
Iniciado por:	Sistema		

Finalizado por:	Sistema
Post-condiciones:	El sistema aplica las recompensas pospago
Operaciones:	<ol style="list-style-type: none"> 1. El WebJob se ejecuta automáticamente en la hora especificada por el CRON 2. El sistema obtiene el listado de referidos activos y que aún está disponible para aplicación de recompensas 3. El sistema utiliza el servicio para determinar si el referido es pospago 4. Obtener recompensas pospago 5. El sistema utiliza el servicio para determinar si el referente es prepago o pospago 6. Si el referente es pospago se activa el plan seleccionado en la recompensa 7. Si el referente es prepago se envía la recarga 8. Guardar registro de recompensa en la base de datos 9. Guardar bitácora de la recompensa aplicada
Excepciones:	<ul style="list-style-type: none"> • No hay referidos para procesar • Los servicios WCF no envían las respuestas esperadas
ESCENARIO	
Nombre:	Procesar recompensas prepago
Precondiciones:	<ul style="list-style-type: none"> • El AppService debe estar configurado para no apagarse • El WebJob debe tener configurado la url del api • El WebJob debe tener configurado una expresión CRON para ejecutarse cada miércoles a las 9:00PM • El sistema debe crear el cliente para los servicios WCF de forma dinámica • Los referidos están disponibles para aplicar recompensas Los referidos deben estar activos
Iniciado por:	Sistema
Finalizado por:	Sistema
Post-condiciones:	El sistema aplica las recompensas prepago
Operaciones:	<ol style="list-style-type: none"> 1. El WebJob se ejecuta automáticamente en la hora especificada por el CRON 2. El sistema obtiene el listado de referidos activos y que aún está disponible para aplicación de recompensas 3. El sistema utiliza el servicio para determinar si el referido es pospago 4. Obtener recompensas prepago 5. El sistema utiliza el servicio para determinar si el referente es prepago o pospago 6. Si el referente es pospago se activa el plan seleccionado en la recompensa 7. Si el referente es prepago se envía la recarga 8. Guardar registro de recompensa en la base de datos 9. Guardar bitácora de la recompensa aplicada
Excepciones:	<ul style="list-style-type: none"> • No hay referidos para procesar

- | | |
|--|--|
| | <ul style="list-style-type: none"> Los servicios WCF no envían las respuestas esperadas |
|--|--|

Fuente: Elaboración propia

Tabla 61: Plantilla RF-01

Caso de Uso	RF-01: Iniciar sesión con doble autenticación		
DEFINICION:	Para acceder al sistema se debe realizar un proceso de autenticación indicando correo electrónico y contraseña, el sistema validará las credenciales y si estas son válidas se procederá a enviar el segundo factor de autenticación para validar la autenticidad del acceso.		
PRIORIDAD:	<input checked="" type="checkbox"/> Vital	<input type="checkbox"/> Importante	<input type="checkbox"/> Conveniente
URGENCIA:	<input checked="" type="checkbox"/> Inmediata	<input type="checkbox"/> Necesario	<input type="checkbox"/> Puede esperar
AUTORES:	Cristopher Josue Zepeda Tellez Gloria Esther Mayorga Mena Magaly Magdalena Mayorga Maradiaga		
ACTORES			
NOMBRE	DEFINICION		
Administrador, Distribuidor, Marketing Vendedor y	Los diferentes roles de usuario podrán iniciar sesión		
Sistema	El sistema se conectará a los servicios externos para enviar el segundo factor de autenticación		
ESCENARIO			
Nombre:	Iniciar sesión		
Precondiciones:			
Iniciado por:	Usuario		
Finalizado por:	Sistema		
Post-condiciones:	El usuario accede al sistema		
Operaciones:	1. Ingresar credenciales 2. Accionar botón Iniciar Sesión 3. El sistema valida las credenciales 4. Si las credenciales son válidas consume el servicio para enviar el pin al correo electrónico del usuario 5. El usuario recibe el PIN en su correo electrónico 6. Ingresa el PIN 7. Accionar botón Validar PIN 8. El sistema valida el PIN 9. Si el PIN es válido, se genera un token JWT que será utilizado en las subsecuentes peticiones que se realizaran al API 10. El usuario accede al sistema		
Excepciones:	<ul style="list-style-type: none"> Correo electrónico no existe Contraseña incorrecta PIN incorrecto Servicio externo responde con error 		

Fuente: Elaboración propia

Tabla 62: Plantilla RF-02

Caso de Uso	RF-02: Recuperar contraseña		
DEFINICION:	Los usuarios deben tener la posibilidad de reestablecer su contraseña, para ello recibirán un correo electrónico de acceso único en el que podrán establecer una nueva contraseña.		
PRIORIDAD:	<input type="checkbox"/> Vital	<input checked="" type="checkbox"/> Importante	<input type="checkbox"/> Conveniente
URGENCIA:	<input type="checkbox"/> Inmediata	<input checked="" type="checkbox"/> Necesario	<input type="checkbox"/> Puede esperar
AUTORES:	Christopher Josue Zepeda Tellez Gloria Esther Mayorga Mena Magaly Magdalena Mayorga Maradiaga		
ACTORES			
NOMBRE	DEFINICION		
Administrador, Distribuidor, Marketing Vendedor y	Los diferentes roles de usuario podrán iniciar sesión		
Sistema	El sistema se conectará a los servicios externos para enviar el segundo factor de autenticación		
ESCENARIO			
Nombre:	Reestablecer contraseña		
Precondiciones:	El usuario debe existir		
Iniciado por:	Usuario		
Finalizado por:	Sistema		
Post-condiciones:	La contraseña es reestablecida		
Operaciones:	1. Accionar botón olvide contraseña 2. Indicar correo electrónico 3. El sistema consume servicio externo que realiza el envío de correo electrónico 4. El usuario recibe el correo electrónico 5. Accede al enlace 6. Establece la nueva contraseña 7. Acciona el botón guardar 8. El sistema valida el estándar de seguridad de la contraseña 9. El sistema consume servicio externo para reestablecer contraseña 10. El usuario recibe indicación que la contraseña fue cambiada exitosamente 11. El usuario es redirigido a la pantalla de inicio de sesión		
Excepciones:	<ul style="list-style-type: none"> • El correo electrónico no existe • Contraseña no cumple estándar de seguridad • El enlace enviado ha expirado • El envío de correo electrónico fallo 		

Fuente: Elaboración propia

Tabla 63: Plantilla RF-03

Caso de Uso	RF-03: Cargar menu de navegación basado en roles		
DEFINICION:	El menu de opciones debe ser completamente basado en el rol, y este debe ser configurable desde la base de datos, es decir, para aplicar un nuevo permiso a un usuario no será necesario que se recompile y publique el código nuevamente.		
PRIORIDAD:	<input checked="" type="checkbox"/> Vital	<input type="checkbox"/> Importante	<input type="checkbox"/> Conveniente
URGENCIA:	<input checked="" type="checkbox"/> Inmediata	<input type="checkbox"/> Necesario	<input type="checkbox"/> Puede esperar
AUTORES:	Cristopher Josue Zepeda Tellez Gloria Esther Mayorga Mena Magaly Magdalena Mayorga Maradiaga		
ACTORES			
NOMBRE	DEFINICION		
Sistema	El sistema se conectará al API para obtener la información		
ESCENARIO			
Nombre:	Cargar menu		
Precondiciones:	<ul style="list-style-type: none"> • Debe haber iniciado sesión • El usuario debe tener un rol asignado • El rol debe tener las opciones de menu asignadas • Debe existir un JWT de autenticación 		
Iniciado por:	Sistema		
Finalizado por:	Sistema		
Post-condiciones:	Se carga el menu		
Operaciones:	1. El sistema envía la petición al api de navegación 2. El sistema obtiene el listado de opciones basados en el rol 3. Retorna el listado de opciones 4. El sistema muestra las opciones		
Excepciones:	No hay opciones para el rol asignado		

Fuente: Elaboración propia

Tabla 64: Plantilla RF-04

Caso de Uso	RF-04: Crear conexión a la base de datos		
DEFINICION:	El sistema debe estar preparado para crear una conexión a base de datos tipo SQL, y esta debe poder cambiarse por ambiente (Desarrollo, UAT y Producción)		
PRIORIDAD:	<input checked="" type="checkbox"/> Vital	<input type="checkbox"/> Importante	<input type="checkbox"/> Conveniente
URGENCIA:	<input checked="" type="checkbox"/> Inmediata	<input type="checkbox"/> Necesario	<input type="checkbox"/> Puede esperar
AUTORES:	Cristopher Josue Zepeda Tellez Gloria Esther Mayorga Mena Magaly Magdalena Mayorga Maradiaga		
ACTORES			
NOMBRE	DEFINICION		
Sistema	El sistema creara la conexión a la base de datos como una inyección de dependencia		
ESCENARIO			
Nombre:	Crear conexión SQL		

Precondiciones:	Debe existir un parámetro en el archivo appsettings que contenga la cadena de conexión, indicando servidor, base de datos, usuario y contraseña
Iniciado por:	Sistema
Finalizado por:	Sistema
Post-condiciones:	La conexión es creada
Operaciones:	<ol style="list-style-type: none"> 1. Se lee el archivo appsettings 2. obtiene el parámetro ConnectionString 3. Genera la conexión a la base de datos 4. Crea la inyección de dependencia
Excepciones:	<ul style="list-style-type: none"> • No existe el parámetro en el appsettings • El servidor indicado no existe • Las credenciales indicadas no existen • La base de datos indicada no existe

Fuente: Elaboración propia

Tabla 65: Plantilla RF-05

Caso de Uso	RF-05: Crear conexión a SFTP		
DEFINICION:	Para realizar las actividades de cargar o descargar archivos en un SFTP es necesario abrir una conexión, y posterior a su uso debemos cerrar la conexión.		
PRIORIDAD:	<input checked="" type="checkbox"/> Vital	<input type="checkbox"/> Importante	<input type="checkbox"/> Conveniente
URGENCIA:	<input type="checkbox"/> Inmediata	<input checked="" type="checkbox"/> Necesario	<input type="checkbox"/> Puede esperar
AUTORES:	Christopher Josue Zepeda Tellez Gloria Esther Mayorga Mena Magaly Magdalena Mayorga Maradiaga		
ACTORES			
NOMBRE	DEFINICION		
Sistema	El sistema se conectará al servidor SFTP		
ESCENARIO			
Nombre:	Crear conexión SFTP		
Precondiciones:	<ul style="list-style-type: none"> • Debe existir la llave en la ubicación ./sftpkey/ • En el appsettings deben existir los parámetros usuario, contraseña, puerto y host del SFTP 		
Iniciado por:	Sistema		
Finalizado por:	Sistema		
Post-condiciones:	La conexión al SFTP es creada		
Operaciones:	<ol style="list-style-type: none"> 1. Lee los parámetros del appsettings 2. Obtiene la llave en ./sftpkey/ 3. Crea la conexión usando el host, usuario, contraseña y la llave 4. Se crea un cliente de la conexión 5. Se retorna el cliente 		
Excepciones:	La conexión no es realizada		

Fuente: Elaboración propia

Tabla 66: Plantilla RF-06

Caso de Uso	RF-06: Enviar correo electrónico		
DEFINICION:	Para realizar el envío de reporte a los distribuidores es necesario crear un cliente SMTP con los parámetros en el appsettings		
PRIORIDAD:	<input type="checkbox"/> Vital	<input checked="" type="checkbox"/> Importante	<input type="checkbox"/> Conveniente
URGENCIA:	<input type="checkbox"/> Inmediata	<input checked="" type="checkbox"/> Necesario	<input type="checkbox"/> Puede esperar
AUTORES:	Cristopher Josue Zepeda Tellez Gloria Esther Mayorga Mena Magaly Magdalena Mayorga Maradiaga		
ACTORES			
NOMBRE	DEFINICION		
Sistema	El sistema se encarga de crear el cliente SMTP y enviar el correo		
ESCENARIO			
Nombre:	Enviar correo electrónico		
Precondiciones:	<ul style="list-style-type: none"> • Debe existir parámetros de SMTP en el appsettings • Debe existir al menos un distribuidor para enviar correo • El correo debe estar activo 		
Iniciado por:	Sistema		
Finalizado por:	Sistema		
Post-condiciones:	Envía correo electrónico en formato HTML		
Operaciones:	1. Se obtiene el listado de correos de los distribuidores 2. Se obtienen las credenciales del SMTP 3. Se crea el cliente SMTP 4. Se establece el asunto del correo electrónico 5. Se establece el formato del cuerpo en HTML 6. Se agregan los destinatarios 7. Se envía el correo electrónico		
Excepciones:	<ul style="list-style-type: none"> • Credenciales de SMTP son invalidas • No hay correo de destinatario 		

Fuente: Elaboración propia

Tabla 67: Plantilla RF-07

Caso de Uso	RF-07: Mostrar nombre de usuario y correo electrónico del usuario		
DEFINICION:	Una vez el usuario accedió a la plataforma, en el menu principal se debe mostrar el nombre de usuario y el correo electrónico del mismo.		
PRIORIDAD:	<input type="checkbox"/> Vital	<input type="checkbox"/> Importante	<input checked="" type="checkbox"/> Conveniente
URGENCIA:	<input checked="" type="checkbox"/> Inmediata	<input type="checkbox"/> Necesario	<input type="checkbox"/> Puede esperar
AUTORES:	Cristopher Josue Zepeda Tellez Gloria Esther Mayorga Mena Magaly Magdalena Mayorga Maradiaga		
ACTORES			
NOMBRE	DEFINICION		
Sistema	El sistema establece la información necesaria en el JWT		
ESCENARIO			

Nombre:	Mostrar información de usuario
Precondiciones:	El usuario debe iniciar sesión
Iniciado por:	Sistema
Finalizado por:	Sistema
Post-condiciones:	El sistema muestra el nombre y correo electrónico del usuario
Operaciones:	<ol style="list-style-type: none"> 1. El api de verificación del PIN (2FA) retorna el JWT 2. El sistema obtiene la información de correo electrónico y nombre de usuario 3. Almacena la información en memoria 4. Muestra la información en la pantalla
Excepciones:	

Fuente: Elaboración propia

Tabla 68: Plantilla RF-08

Caso de Uso	RF-08: Mostrar los referidos basados en el rol		
DEFINICION:	El sistema debe mostrar los referidos basados en el rol, es decir, un vendedor puede únicamente ver los referidos que registro, un distribuidor puede ver los referidos registrados de todos los vendedores y el administrador puede ver todos los referidos		
PRIORIDAD:	<input checked="" type="checkbox"/> Vital	<input type="checkbox"/> Importante	<input type="checkbox"/> Conveniente
URGENCIA:	<input checked="" type="checkbox"/> Inmediata	<input type="checkbox"/> Necesario	<input type="checkbox"/> Puede esperar
AUTORES:	Christopher Josue Zepeda Tellez Gloria Esther Mayorga Mena Magaly Magdalena Mayorga Maradiaga		
ACTORES			
NOMBRE	DEFINICION		
Administrador, Distribuidor y Vendedor	Los diferentes roles de usuario tendrán acceso a la opción de gestión de referidos.		
ESCENARIO			
Nombre:	Consultar referidos		
Precondiciones:	<ul style="list-style-type: none"> • El usuario debió haber iniciado sesión • El usuario debe tener acceso a la opción 		
Iniciado por:	Usuario		
Finalizado por:	Sistema		
Post-condiciones:	Se muestra la información correctamente		
Operaciones:	<ol style="list-style-type: none"> 1. Al consultar la información de referidos se envía el rol del usuario 2. El sistema consulta la información de todos los referidos 3. Filtra la información basada en el rol 4. Si es administrador no aplica filtro 5. Si es vendedor, aplica el filtro de vendedor 6. Si es distribuidor, busca todos los vendedores de un distribuidor y lo filtra en la lista de referidos 7. Retorna el listado de referidos 		
Excepciones:	No hay referidos basados en el filtro		

Fuente: Elaboración propia

Tabla 69: Plantilla RF-09

Caso de Uso	RF-09: Cerrar sesión		
DEFINICION:	Todos los usuarios deben poder cerrar sesión		
PRIORIDAD:	<input checked="" type="checkbox"/> Vital	<input type="checkbox"/> Importante	<input type="checkbox"/> Conveniente
URGENCIA:	<input checked="" type="checkbox"/> Inmediata	<input type="checkbox"/> Necesario	<input type="checkbox"/> Puede esperar
AUTORES:	Cristopher Josue Zepeda Tellez Gloria Esther Mayorga Mena Magaly Magdalena Mayorga Maradiaga		
ACTORES			
NOMBRE	DEFINICION		
Administrador, Distribuidor, Marketing Vendedor	y Los diferentes roles de usuario tendrán acceso a la opción de cerrar sesión.		
ESCENARIO			
Nombre:	Cerrar sesión		
Precondiciones:	<ul style="list-style-type: none"> • El usuario debió haber iniciado sesión • El usuario debe tener acceso a la opción 		
Iniciado por:	Usuario		
Finalizado por:	Sistema		
Post-condiciones:	Se borra la información almacenada en el navegador		
Operaciones:	1. El usuario acciona la opción Cerrar Sesión 2. El sistema borra la información almacenada en el navegador 3. El sistema redirige al usuario a la página de inicio de sesión		
Excepciones:			

Fuente: Elaboración propia

Tabla 70: Plantilla RF-10

Caso de Uso	RF-10: Consumir servicios externos para recargar		
DEFINICION:	Cuando la recompensa es de tipo prepago, se deben consumir los servicios de verificación y recarga telefónica.		
PRIORIDAD:	<input checked="" type="checkbox"/> Vital	<input type="checkbox"/> Importante	<input type="checkbox"/> Conveniente
URGENCIA:	<input checked="" type="checkbox"/> Inmediata	<input type="checkbox"/> Necesario	<input type="checkbox"/> Puede esperar
AUTORES:	Cristopher Josue Zepeda Tellez Gloria Esther Mayorga Mena Magaly Magdalena Mayorga Maradiaga		
ACTORES			
NOMBRE	DEFINICION		
Sistema	El sistema consume el servicio WCF		
ESCENARIO			
Nombre:	Recargar Línea telefónica		
Precondiciones:	<ul style="list-style-type: none"> • Debe ejecutarse el proceso de recompensas • Debe haber un usuario activo • El referente debe ser tipo prepago 		
Iniciado por:	Sistema		
Finalizado por:	Sistema		

Post-condiciones:	Se realiza la recarga telefónica
Operaciones:	<ol style="list-style-type: none"> 1. El sistema crea el cliente WCF 2. Consume servicio para obtener información de un número telefónico 3. Consume servicio para recargar el monto 4. Evaluar respuesta del cliente 5. Eliminar cliente WCF
Excepciones:	El servicio WCF responde con error

Fuente: Elaboración propia

Tabla 71: Plantilla RF-11

Caso de Uso	RF-11: Consumir servicios externos para abonar a un plan		
DEFINICION:	Cuando la recompensa es de tipo pospago, se deben consumir los servicios de verificación y acreditación a un plan.		
PRIORIDAD:	<input checked="" type="checkbox"/> Vital	<input type="checkbox"/> Importante	<input type="checkbox"/> Conveniente
URGENCIA:	<input checked="" type="checkbox"/> Inmediata	<input type="checkbox"/> Necesario	<input type="checkbox"/> Puede esperar
AUTORES:	Cristopher Josue Zepeda Tellez Gloria Esther Mayorga Mena Magaly Magdalena Mayorga Maradiaga		
ACTORES			
NOMBRE	DEFINICION		
Sistema	El sistema consume el servicio WCF		
ESCENARIO			
Nombre:	Acreditar plan telefónico		
Precondiciones:	<ul style="list-style-type: none"> • Debe ejecutarse el proceso de recompensas • Debe haber un usuario activo • El referente debe ser tipo pospago 		
Iniciado por:	Sistema		
Finalizado por:	Sistema		
Post-condiciones:	Se realiza la recarga telefónica		
Operaciones:	<ol style="list-style-type: none"> 1. El sistema crea el cliente WCF 2. Consume servicio para obtener información de un número telefónico 3. Consume servicio para obtener información de plan 4. Consume servicio para acreditar saldo al plan 5. Evaluar respuesta del cliente 6. Eliminar cliente WCF 		
Excepciones:	El servicio WCF responde con error		

Fuente: Elaboración propia

Tabla 72: Plantilla RF-12

Caso de Uso	RF-12: Recordar usuario		
DEFINICION:	El usuario debe tener la facilidad de marcar una opción para recordar su usuario y contraseña en el navegador, y que este no las esté ingresando a cada momento		
PRIORIDAD:	<input type="checkbox"/> Vital	<input checked="" type="checkbox"/> Importante	<input type="checkbox"/> Conveniente
URGENCIA:	<input type="checkbox"/> Inmediata	<input type="checkbox"/> Necesario	<input checked="" type="checkbox"/> Puede esperar

AUTORES:	Cristopher Josue Zepeda Tellez
	Gloria Esther Mayorga Mena
	Magaly Magdalena Mayorga Maradiaga
ACTORES	
NOMBRE	DEFINICION
Administrador, Distribuidor, Marketing Vendedor y	Los diferentes roles de usuario tendrán acceso a la opción de recordar usuario.
ESCENARIO	
Nombre:	Marcar opción recordar usuario
Precondiciones:	Las credenciales ingresadas deben ser validas
Iniciado por:	Usuario
Finalizado por:	Sistema
Post-condiciones:	Las credenciales son guardadas en el navegador
Operaciones:	<ol style="list-style-type: none"> El usuario ingresa las credenciales El usuario marca la opción recordar credenciales El sistema almacena las credenciales en el almacenamiento local del navegador El usuario cierra sesión Las credenciales deben estar precargadas y ser funcionales
Excepciones:	<ul style="list-style-type: none"> Las credenciales son incorrectas, no son guardadas Las credenciales son borradas directamente en la pestaña aplicación
ESCENARIO	
Nombre:	Desmarcar opción recordar usuario
Precondiciones:	Las credenciales ingresadas deben ser válidas
Iniciado por:	Usuario
Finalizado por:	Sistema
Post-condiciones:	Las credenciales son eliminadas del navegador
Operaciones:	<ol style="list-style-type: none"> El usuario desmarca la opción recordar credenciales El sistema elimina las credenciales El usuario recarga la página de inicio de sesión Las credenciales no son precargadas
Excepciones:	

Fuente: Elaboración propia

Tabla 73: Plantilla RF-13

Caso de Uso	RF-13: Mostrar cantidad de registros en las pestañas		
DEFINICION:	El sistema debe establecer de forma automática en cada pestaña el número de registros almacenados		
PRIORIDAD:	<input checked="" type="checkbox"/> Vital	<input type="checkbox"/> Importante	<input type="checkbox"/> Conveniente
URGENCIA:	<input checked="" type="checkbox"/> Inmediata	<input type="checkbox"/> Necesario	<input type="checkbox"/> Puede esperar
AUTORES:	Cristopher Josue Zepeda Tellez		
	Gloria Esther Mayorga Mena		
	Magaly Magdalena Mayorga Maradiaga		

ACTORES	
NOMBRE	DEFINICION
Administrador, Distribuidor, Marketing Vendedor y	El usuario tiene acceso a las pestañas Referidos, Contactos, Recompensas o motivos de declinación
Sistema	El sistema obtiene y muestra la información
ESCENARIO	
Nombre:	Mostrar cantidad de registros
Precondiciones:	<ul style="list-style-type: none"> • El usuario debe tener acceso a la opción • El usuario debe estar en la pestaña
Iniciado por:	Usuario
Finalizado por:	Sistema
Post-condiciones:	Se muestra el número de registros
Operaciones:	<ol style="list-style-type: none"> 1. El usuario accede a la opción 2. El sistema consulta el número de registros (Siempre basado en el rol) 3. Retorna el número de registros 4. Muestra el número en la pestaña correspondiente
Excepciones:	Si no hay registros se muestra 0

Fuente: Elaboración propia

Tabla 74: Plantilla RNF-01

Caso de Uso	RNF-01: Guardar registros de los archivos descargados en base de datos		
DEFINICION:	Para realizar validaciones de las recompensas aplicadas y estado de los referidos, es necesario almacenar una copia en una tabla SQL del archivo que recibimos del SFTP		
PRIORIDAD:	<input type="checkbox"/> Vital	<input checked="" type="checkbox"/> Importante	<input type="checkbox"/> Conveniente
URGENCIA:	<input type="checkbox"/> Inmediata	<input type="checkbox"/> Necesario	<input checked="" type="checkbox"/> Puede esperar
AUTORES:	Christopher Josue Zepeda Tellez Gloria Esther Mayorga Mena Magaly Magdalena Mayorga Maradiaga		
ACTORES			
NOMBRE	DEFINICION		
Sistema	Es un proceso que se ejecutará de forma automática al descargar un archivo		
ESCENARIO			
Nombre:	Almacenar registros en BD		
Precondiciones:	Se debió descargar un archivo		
Iniciado por:	Sistema		
Finalizado por:	Sistema		
Post-condiciones:	Se registra información en la base de datos		
Operaciones:	<ol style="list-style-type: none"> 1. Se lee el archivo descargado 2. Se transforma el archivo en una tabla 3. Se almacenan los registros a través de un SP 		

Excepciones:	El archivo está vacío
--------------	-----------------------

Fuente: Elaboración propia

Tabla 75: Plantilla RNF-02

Caso de Uso	RNF-02: BackEnd se debe realizar utilizando patrón de diseño CQRS		
DEFINICION:	El API debe emplear un patrón de diseño llamado CQRS, el que consiste en separar los modelos de afectación de registros en la base de datos y los modelos para realizar consultas a la base de datos.		
PRIORIDAD:	<input checked="" type="checkbox"/> Vital	<input type="checkbox"/> Importante	<input type="checkbox"/> Conveniente
URGENCIA:	<input checked="" type="checkbox"/> Inmediata	<input type="checkbox"/> Necesario	<input type="checkbox"/> Puede esperar
AUTORES:	Cristopher Josue Zepeda Tellez Gloria Esther Mayorga Mena Magaly Magdalena Mayorga Maradiaga		

Fuente: Elaboración propia

Tabla 76: Plantilla RNF-03

Caso de Uso	RNF-03: FrontEnd se debe realizar con Angular		
DEFINICION:	La tecnología a utilizar en FrontEnd debe ser Angular, utilizando la versión 13. Este es un framework utilizado por el equipo de desarrollo de Datafone, de esta forma les será sencillo realizar ajustes o mejoras al código		
PRIORIDAD:	<input checked="" type="checkbox"/> Vital	<input type="checkbox"/> Importante	<input type="checkbox"/> Conveniente
URGENCIA:	<input checked="" type="checkbox"/> Inmediata	<input type="checkbox"/> Necesario	<input type="checkbox"/> Puede esperar
AUTORES:	Cristopher Josue Zepeda Tellez Gloria Esther Mayorga Mena Magaly Magdalena Mayorga Maradiaga		

Fuente: Elaboración propia

Tabla 77: Plantilla RNF-04

Caso de Uso	RNF-04: BackEnd se debe realizar con .NET 7		
DEFINICION:	La tecnología a utilizar en BackEnd debe ser .NET, utilizando la versión 7. Este es un framework utilizado por el equipo de desarrollo de Datafone, de esta forma les será sencillo realizar ajustes o mejoras al código		
PRIORIDAD:	<input checked="" type="checkbox"/> Vital	<input type="checkbox"/> Importante	<input type="checkbox"/> Conveniente
URGENCIA:	<input checked="" type="checkbox"/> Inmediata	<input type="checkbox"/> Necesario	<input type="checkbox"/> Puede esperar
AUTORES:	Cristopher Josue Zepeda Tellez Gloria Esther Mayorga Mena Magaly Magdalena Mayorga Maradiaga		

Fuente: Elaboración propia

Tabla 78: Plantilla RNF-05

Caso de Uso	RNF-05: Reglas de seguridad para establecer una contraseña
DEFINICION:	La contraseña debe cumplir con los siguientes parámetros:

	5. Longitud mayor o igual a 8 caracteres 6. Debe incluir una mayúscula 7. Debe incluir un número 8. Debe incluir un carácter especial 9. No debe contener secuencias de letras o números mayor a 3
PRIORIDAD:	<input checked="" type="checkbox"/> Vital <input type="checkbox"/> Importante <input type="checkbox"/> Conveniente
URGENCIA:	<input checked="" type="checkbox"/> Inmediata <input type="checkbox"/> Necesario <input type="checkbox"/> Puede esperar
AUTORES:	Cristopher Josue Zepeda Tellez Gloria Esther Mayorga Mena Magaly Magdalena Mayorga Maradiaga

Fuente: Elaboración propia

Tabla 79: Plantilla RNF-06

Caso de Uso	RNF-06: Cifrado de contraseña		
DEFINICION:	Toda contraseña almacenada en las bases de datos debe estar cifrada, así como también la contraseña enviada desde el front-end hacia el back-end.		
PRIORIDAD:	<input checked="" type="checkbox"/> Vital <input type="checkbox"/> Importante <input type="checkbox"/> Conveniente		
URGENCIA:	<input checked="" type="checkbox"/> Inmediata <input type="checkbox"/> Necesario <input type="checkbox"/> Puede esperar		
AUTORES:	Cristopher Josue Zepeda Tellez Gloria Esther Mayorga Mena Magaly Magdalena Mayorga Maradiaga		

Fuente: Elaboración propia

Tabla 80: Plantilla RNF-07

Caso de Uso	RNF-07: Plataforma web debe poder utilizarse en cualquier dispositivo		
DEFINICION:	Todas las pantallas deben ser adaptables al tamaño de la pantalla del dispositivo, en el caso de las pantallas pequeñas las tablas deben mostrar un máximo de 3 columnas y las acciones que se deben ejecutar		
PRIORIDAD:	<input checked="" type="checkbox"/> Vital <input type="checkbox"/> Importante <input type="checkbox"/> Conveniente		
URGENCIA:	<input checked="" type="checkbox"/> Inmediata <input type="checkbox"/> Necesario <input type="checkbox"/> Puede esperar		
AUTORES:	Cristopher Josue Zepeda Tellez Gloria Esther Mayorga Mena Magaly Magdalena Mayorga Maradiaga		

Fuente: Elaboración propia

Tabla 81: Plantilla RNF-08

Caso de Uso	RNF-08: Servicios externos serán de tipo WCF		
DEFINICION:	Toda comunicación entre el sistema web y el servicio de Datafone se realizará a través de servicios WCF. Para ello se debe crear un cliente de forma dinámica previo al consumo del servicio.		
PRIORIDAD:	<input checked="" type="checkbox"/> Vital <input type="checkbox"/> Importante <input type="checkbox"/> Conveniente		
URGENCIA:	<input checked="" type="checkbox"/> Inmediata <input type="checkbox"/> Necesario <input type="checkbox"/> Puede esperar		
AUTORES:	Cristopher Josue Zepeda Tellez		

	Gloria Esther Mayorga Mena
	Magaly Magdalena Mayorga Maradiaga

Fuente: Elaboración propia

Tabla 82: Plantilla Coleman RNF-09

Caso de Uso	RNF-09: La aplicación se desplegará en una App Service de Azure		
DEFINICION:	El sistema web será desplegado en el ambiente de Azure, para ello se utilizará un App Service (uno para UAT y otro para PROD), dado que es un SPA solo será necesario realizar un único despliegue.		
PRIORIDAD:	<input checked="" type="checkbox"/> Vital	<input type="checkbox"/> Importante	<input type="checkbox"/> Conveniente
URGENCIA:	<input checked="" type="checkbox"/> Inmediata	<input type="checkbox"/> Necesario	<input type="checkbox"/> Puede esperar
AUTORES:	Cristopher Josue Zepeda Tellez Gloria Esther Mayorga Mena Magaly Magdalena Mayorga Maradiaga		

Fuente: Elaboración propia

Tabla 83: Plantilla Coleman RI-01

Caso de Uso	RI-01: Crear manual de usuario		
DEFINICION:	Una vez completadas las pruebas de usuario, y los gerentes de desarrollo, ventas y marketing hayan aprobado el sistema, se debe crear un manual de usuario sobre las funcionalidades que el sistema contempla		
PRIORIDAD:	<input type="checkbox"/> Vital	<input checked="" type="checkbox"/> Importante	<input type="checkbox"/> Conveniente
URGENCIA:	<input type="checkbox"/> Inmediata	<input type="checkbox"/> Necesario	<input checked="" type="checkbox"/> Puede esperar
AUTORES:	Cristopher Josue Zepeda Tellez Gloria Esther Mayorga Mena Magaly Magdalena Mayorga Maradiaga		

Fuente: Elaboración propia

Tabla 84: Plantilla Coleman RI-02

Caso de Uso	RI-02: Crear la cuenta en Azure		
DEFINICION:	El sistema estará desplegado en un ambiente en la nube, es por ello que es necesario crear una cuenta en Azure.		
PRIORIDAD:	<input checked="" type="checkbox"/> Vital	<input type="checkbox"/> Importante	<input type="checkbox"/> Conveniente
URGENCIA:	<input checked="" type="checkbox"/> Inmediata	<input type="checkbox"/> Necesario	<input type="checkbox"/> Puede esperar
AUTORES:	Cristopher Josue Zepeda Tellez Gloria Esther Mayorga Mena Magaly Magdalena Mayorga Maradiaga		

Fuente: Elaboración propia

Tabla 85: Plantilla Coleman RI-03

Caso de Uso	RI-03: Crear las subscripciones de QA y Producción en Azure		
DEFINICION:	Se debe tener control sobre los costos empleados por ambientes de QA y Producción, es por ello que es necesario crear dos		

	subscripciones, una para el ambiente de QA y otra para el ambiente productivo		
PRIORIDAD:	<input checked="" type="checkbox"/> Vital	<input type="checkbox"/> Importante	<input type="checkbox"/> Conveniente
URGENCIA:	<input checked="" type="checkbox"/> Inmediata	<input type="checkbox"/> Necesario	<input type="checkbox"/> Puede esperar
AUTORES:	Cristopher Josue Zepeda Tellez Gloria Esther Mayorga Mena Magaly Magdalena Mayorga Maradiaga		

Fuente: Elaboración propia

Tabla 86: Plantilla Coleman RI-04

Caso de Uso	RI-04: Crear Azure SQL Database		
DEFINICION:	Para el almacenamiento de los datos se utilizará Azure SQL Database, por lo que debemos crear dos Azure SQL Database una para el ambiente de QA y otra para el ambiente de Producción. Al momento de crear las bases de datos será necesario crear un servidor lógico y la región debe ser EAST-US		
PRIORIDAD:	<input checked="" type="checkbox"/> Vital	<input type="checkbox"/> Importante	<input type="checkbox"/> Conveniente
URGENCIA:	<input checked="" type="checkbox"/> Inmediata	<input type="checkbox"/> Necesario	<input type="checkbox"/> Puede esperar
AUTORES:	Cristopher Josue Zepeda Tellez Gloria Esther Mayorga Mena Magaly Magdalena Mayorga Maradiaga		

Fuente: Elaboración propia

Tabla 87: Plantilla Coleman RI-05

Caso de Uso	RI-05: Crear App Service		
DEFINICION:	La aplicación web será desplegada en un App Service en Azure, es necesario crear un App Service para cada ambiente. Cada App Service debe tener configurado el entorno de .NET 7, se debe publicar en formato de código y la región debe ser EAST-US		
PRIORIDAD:	<input checked="" type="checkbox"/> Vital	<input type="checkbox"/> Importante	<input type="checkbox"/> Conveniente
URGENCIA:	<input checked="" type="checkbox"/> Inmediata	<input type="checkbox"/> Necesario	<input type="checkbox"/> Puede esperar
AUTORES:	Cristopher Josue Zepeda Tellez Gloria Esther Mayorga Mena Magaly Magdalena Mayorga Maradiaga		

Fuente: Elaboración propia

Tabla 88: Plantilla Coleman RI-06

Caso de Uso	RI-06: Desplegar WebJobs		
DEFINICION:	Todos los procesos calendarizados estarán en diferentes WebJobs, estos serán desplegados desde la herramienta de visual studio y es necesario el App Service tenga configurada la opción “Siempre Activado” en Activado		
PRIORIDAD:	<input checked="" type="checkbox"/> Vital	<input type="checkbox"/> Importante	<input type="checkbox"/> Conveniente
URGENCIA:	<input checked="" type="checkbox"/> Inmediata	<input type="checkbox"/> Necesario	<input type="checkbox"/> Puede esperar
AUTORES:	Cristopher Josue Zepeda Tellez Gloria Esther Mayorga Mena		

	Magaly Magdalena Mayorga Maradiaga
--	------------------------------------

Fuente: Elaboración propia

Tabla 89: Plantilla Coleman RU-01

Caso de Uso	RU-01: Establecer estado por defecto a referidos		
DEFINICION:	Cada vez que se registre un nuevo referido, su estado por defecto debe ser Ingresado		
PRIORIDAD:	<input checked="" type="checkbox"/> Vital	<input type="checkbox"/> Importante	<input type="checkbox"/> Conveniente
URGENCIA:	<input checked="" type="checkbox"/> Inmediata	<input type="checkbox"/> Necesario	<input type="checkbox"/> Puede esperar
AUTORES:	Cristopher Josue Zepeda Tellez Gloria Esther Mayorga Mena Magaly Magdalena Mayorga Maradiaga		
ACTORES			
NOMBRE	DEFINICION		
Sistema	Asignar estado a referido		
ESCENARIO			
Nombre:	Asignar estado		
Precondiciones:	<ul style="list-style-type: none"> • El usuario debió haber iniciado sesión • El usuario debe tener acceso a la opción • Debe haber un registro nuevo de referido 		
Iniciado por:	Sistema		
Finalizado por:	Sistema		
Post-condiciones:	Estado asignado al referido		
Operaciones:	1. Referido es creado 2. Estado es asignado al referido		
Excepciones:			

Fuente: Elaboración propia

Tabla 90: Plantilla Coleman RU-02

Caso de Uso	RU-02: Mostrar restricciones de UI en recompensas		
DEFINICION:	En la pestaña de recompensas se deben mostrar algunos campos basados en el tipo de referente y tipo de referido, el sistema debe controlar esto en base a la selección del usuario		
PRIORIDAD:	<input checked="" type="checkbox"/> Vital	<input type="checkbox"/> Importante	<input type="checkbox"/> Conveniente
URGENCIA:	<input checked="" type="checkbox"/> Inmediata	<input type="checkbox"/> Necesario	<input type="checkbox"/> Puede esperar
AUTORES:	Cristopher Josue Zepeda Tellez Gloria Esther Mayorga Mena Magaly Magdalena Mayorga Maradiaga		
ACTORES			
NOMBRE	DEFINICION		
Sistema	Mostrará y ocultará opciones		
ESCENARIO			
Nombre:	Mostrar opciones (referente prepago, referido prepago)		
Precondiciones:	<ul style="list-style-type: none"> • El usuario debió haber iniciado sesión • El usuario debe tener acceso a la opción 		
Iniciado por:	Usuario		

Finalizado por:	Sistema
Post-condiciones:	Se muestran las opciones correctas
Operaciones:	<ol style="list-style-type: none"> 1. El usuario agrega o selecciona una recompensa 2. Selecciona las opciones: referente prepago y referido prepago 3. El sistema muestra los siguientes campos: Tabla para registro de recargas Fechas de inicio y vigencia de la recompensa Recarga mínima para la recompensa Límite de recompensas Parámetros de configuración de la recarga
Excepciones:	
ESCENARIO	
Nombre:	Mostrar opciones (referente prepago, referido pospago)
Precondiciones:	<ul style="list-style-type: none"> • El usuario debió haber iniciado sesión • El usuario debe tener acceso a la opción
Iniciado por:	Usuario
Finalizado por:	Sistema
Post-condiciones:	Se muestran las opciones correctas
Operaciones:	<ol style="list-style-type: none"> 1. El usuario agrega o selecciona una recompensa 2. Selecciona las opciones: referente prepago y referido pospago 3. El sistema muestra los siguientes campos: Tabla para registro de recargas Fechas de inicio y vigencia de la recompensa Parámetros de configuración de la recarga
Excepciones:	
ESCENARIO	
Nombre:	Mostrar opciones (referente pospago, referido pospago)
Precondiciones:	<ul style="list-style-type: none"> • El usuario debió haber iniciado sesión • El usuario debe tener acceso a la opción
Iniciado por:	Usuario
Finalizado por:	Sistema
Post-condiciones:	Se muestran las opciones correctas
Operaciones:	<ol style="list-style-type: none"> 4. El usuario agrega o selecciona una recompensa 5. Selecciona las opciones: referente pospago y referido pospago 6. El sistema muestra los siguientes campos: Campos: Tipo de plan, Valor de la recompensa y código de verificación Fechas de inicio y vigencia de la recompensa
Excepciones:	
ESCENARIO	
Nombre:	Mostrar opciones (referente pospago, referido prepago)
Precondiciones:	<ul style="list-style-type: none"> • El usuario debió haber iniciado sesión • El usuario debe tener acceso a la opción
Iniciado por:	Usuario
Finalizado por:	Sistema
Post-condiciones:	Se muestran las opciones correctas

Operaciones:	7. El usuario agrega o selecciona una recompensa 8. Selecciona las opciones: referente pospago y referido pospago 9. El sistema muestra los siguientes campos: Campos: Tipo de plan, Valor de la recompensa y código de Fechas de inicio y vigencia de la recompensa Recarga mínima para la recompensa Límite de recompensas Parámetros de configuración de la recarga
Excepciones:	

Fuente: Elaboración propia

AnexoTabla3: Software utilizado para la construcción

Tabla 91: Software – NodeJS

Información de programa	Descripción
Nombre	NodeJS
Versión	16.20.0
Función	Runtime para angular

Fuente: Elaboración propia

Tabla 92: Software – Package Manager

Información de programa	Descripción
Nombre	Package Manager
Versión	8.19.4
Función	Administrar paquetes

Fuente: Elaboración propia

Tabla 93: Software – Angular

Información de programa	Descripción
Nombre	Angular
Versión	15.1.2
Función	Crear aplicación

Fuente: Elaboración propia

Tabla 94: Software – SQL Server Management Studio

Información de programa	Descripción
Nombre	SQL Server Management Studio
Versión	15.0.18384.0
Función	Administrar base de datos

Fuente: Elaboración propia

Tabla 95: Software – WinSCP

Información de programa	Descripción
Nombre	WinSCP
Versión	5.19.5
Función	Validar conexión a SFTP

Fuente: Elaboración propia

Tabla 96: Software – Postman

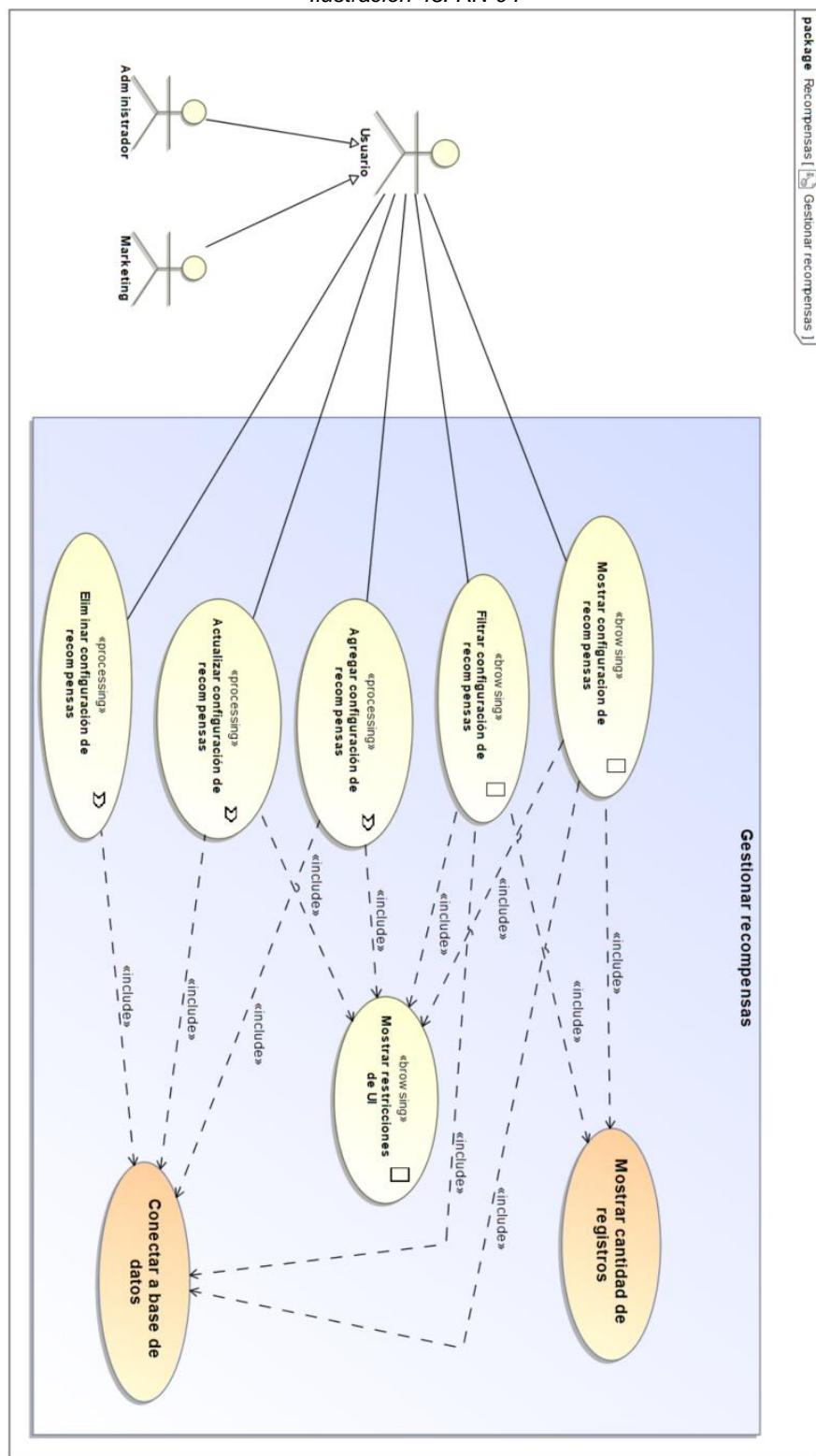
Información de programa	Descripción
Nombre	Postman
Versión	10.12.13
Función	Realizar pruebas sobre el API

Fuente: Elaboración propia

Anexos de ilustraciones

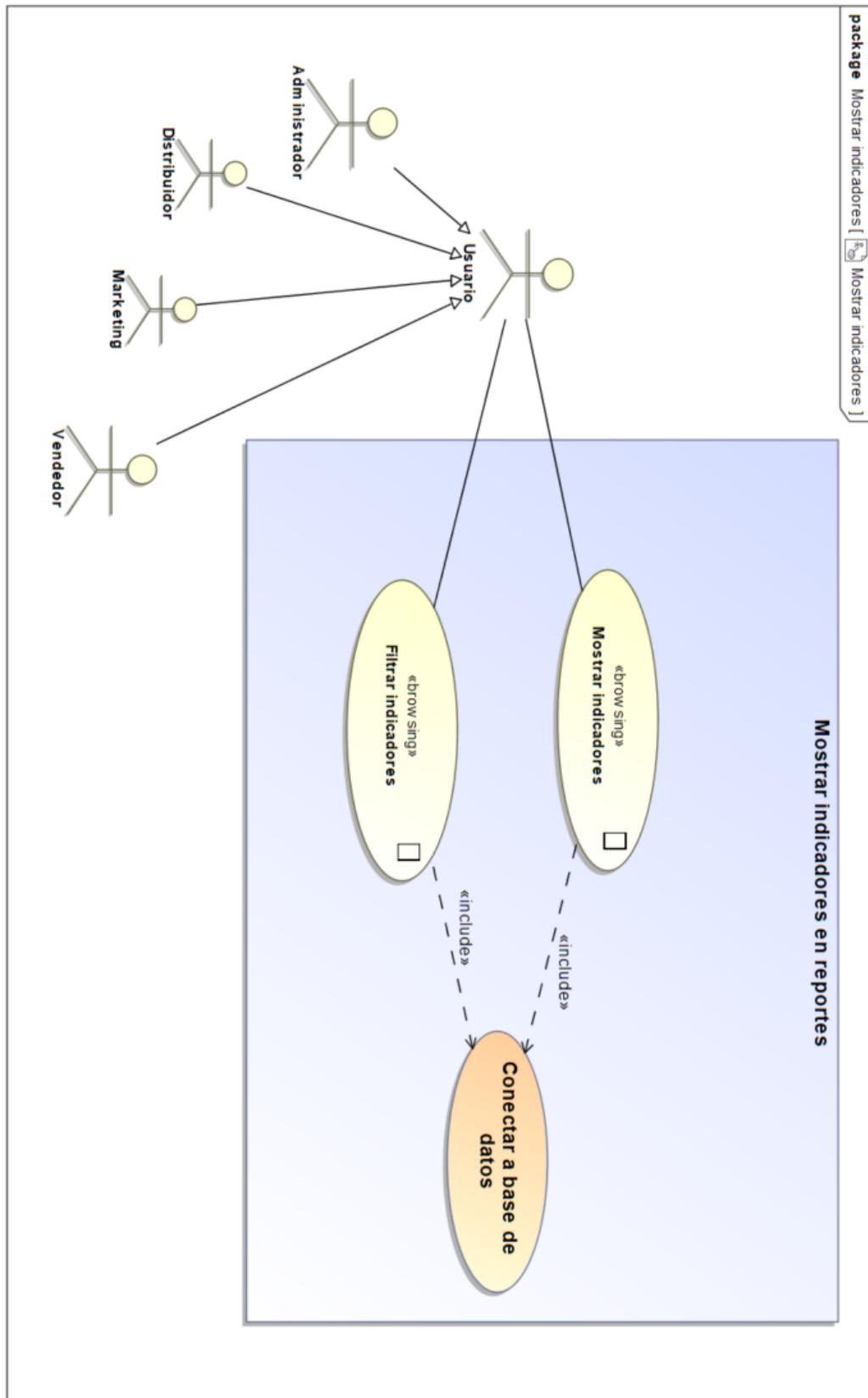
Anexo Ilustración 1: Modelo de requerimientos – Diagramas de casos de uso

Ilustración 45: RN-04



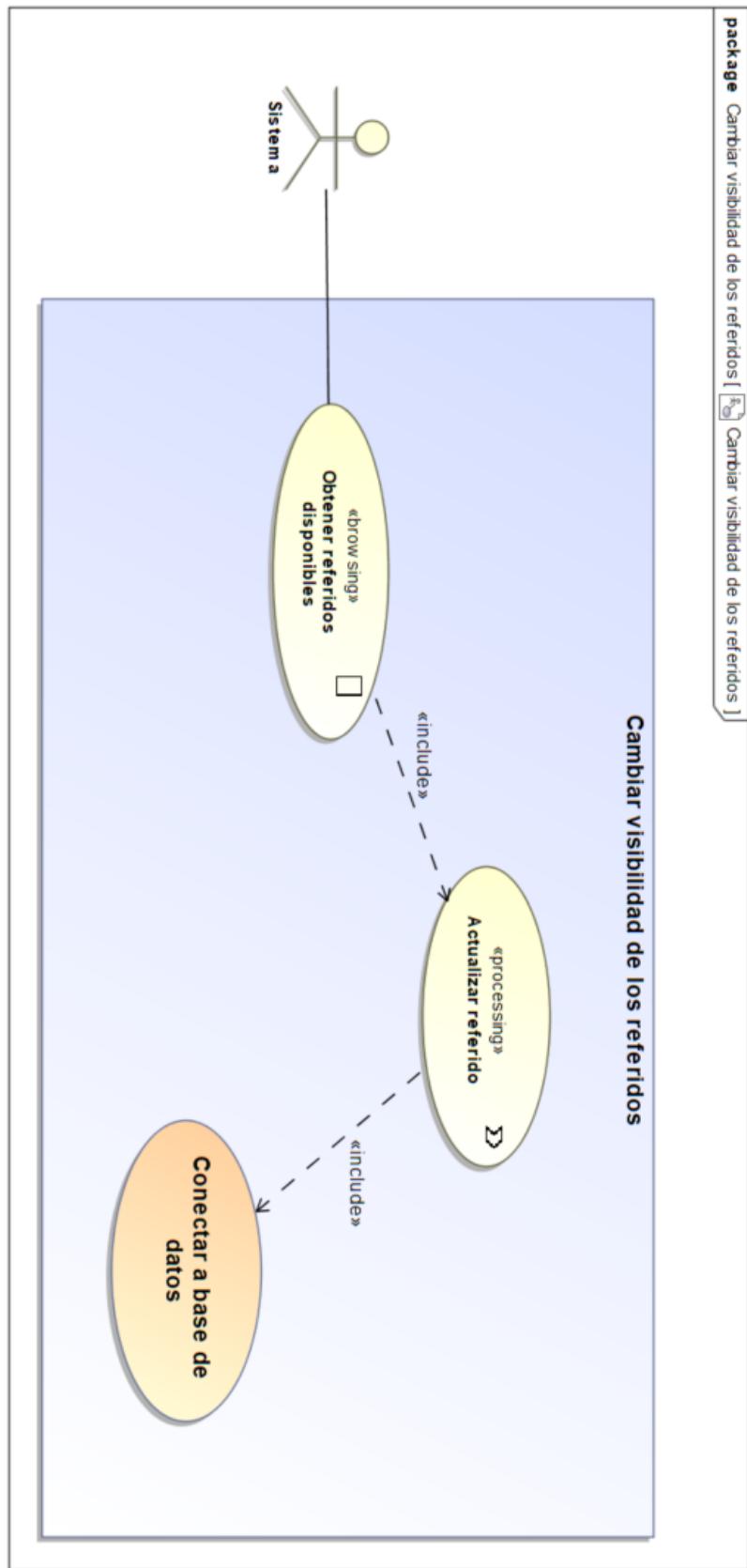
Fuente: Elaboración propia

Ilustración 46: RN-05



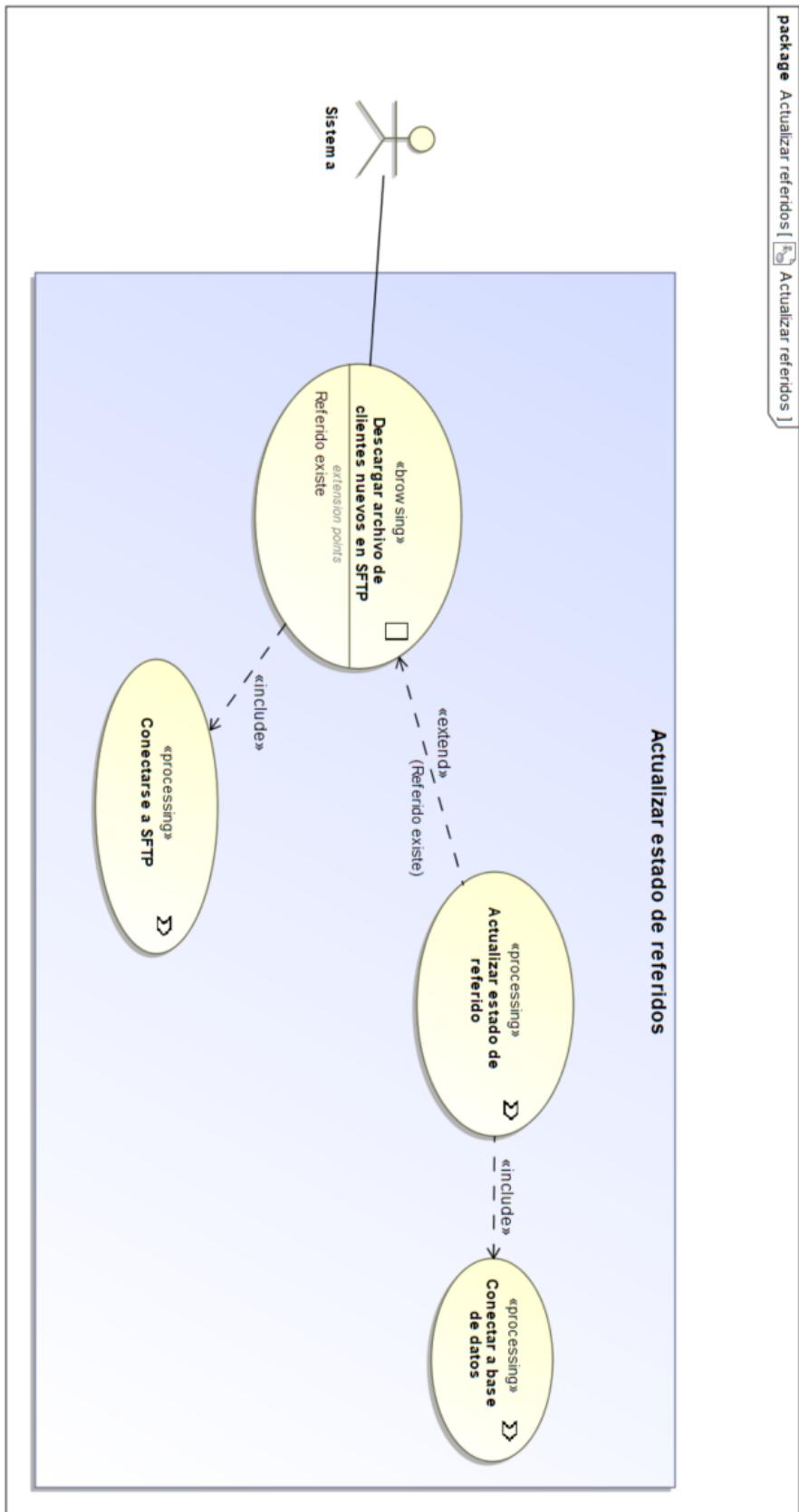
Fuente: Elaboración propia

Ilustración 47: RN-06



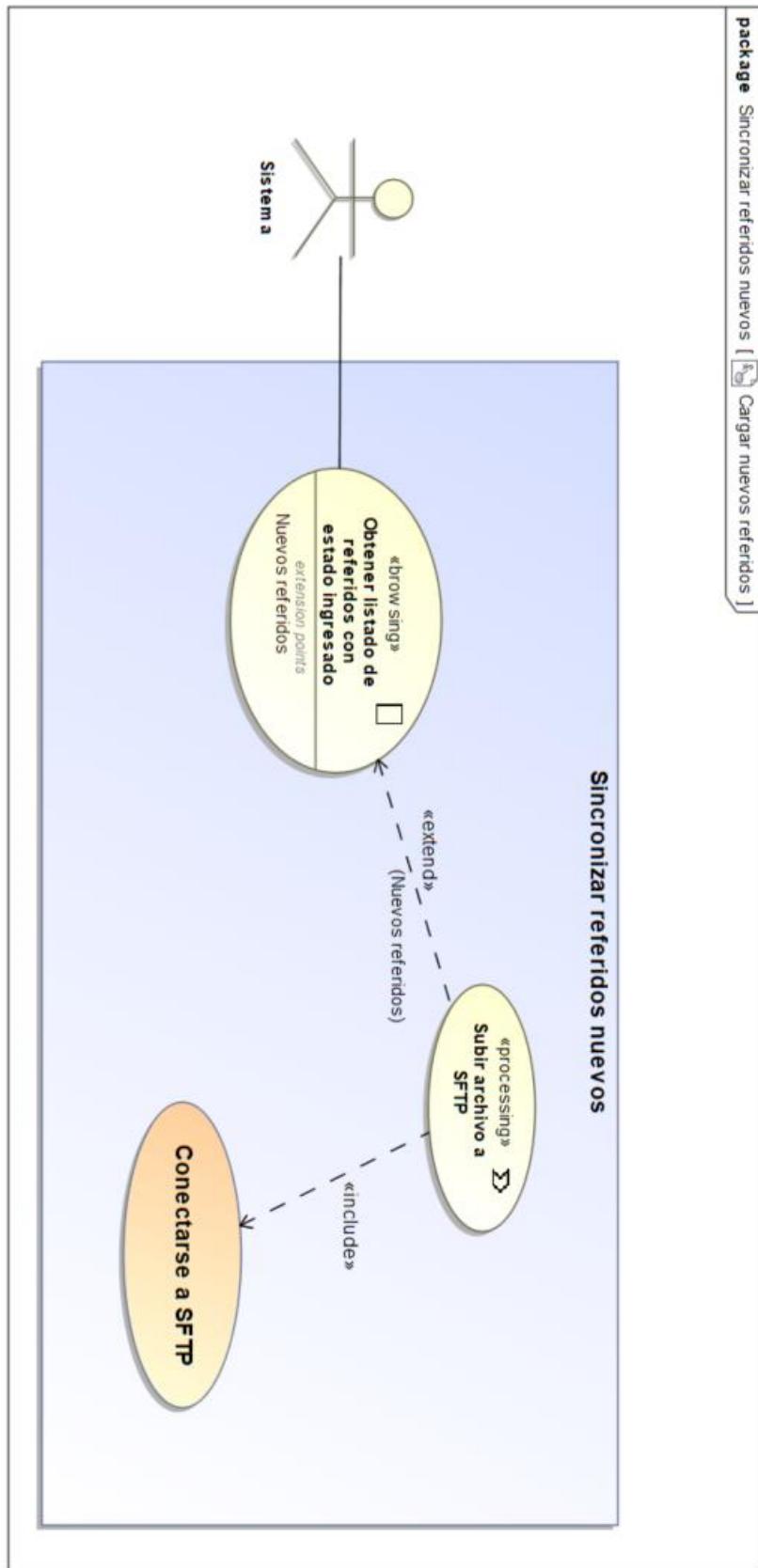
Fuente: Elaboración propia

Ilustración 48: RN-07



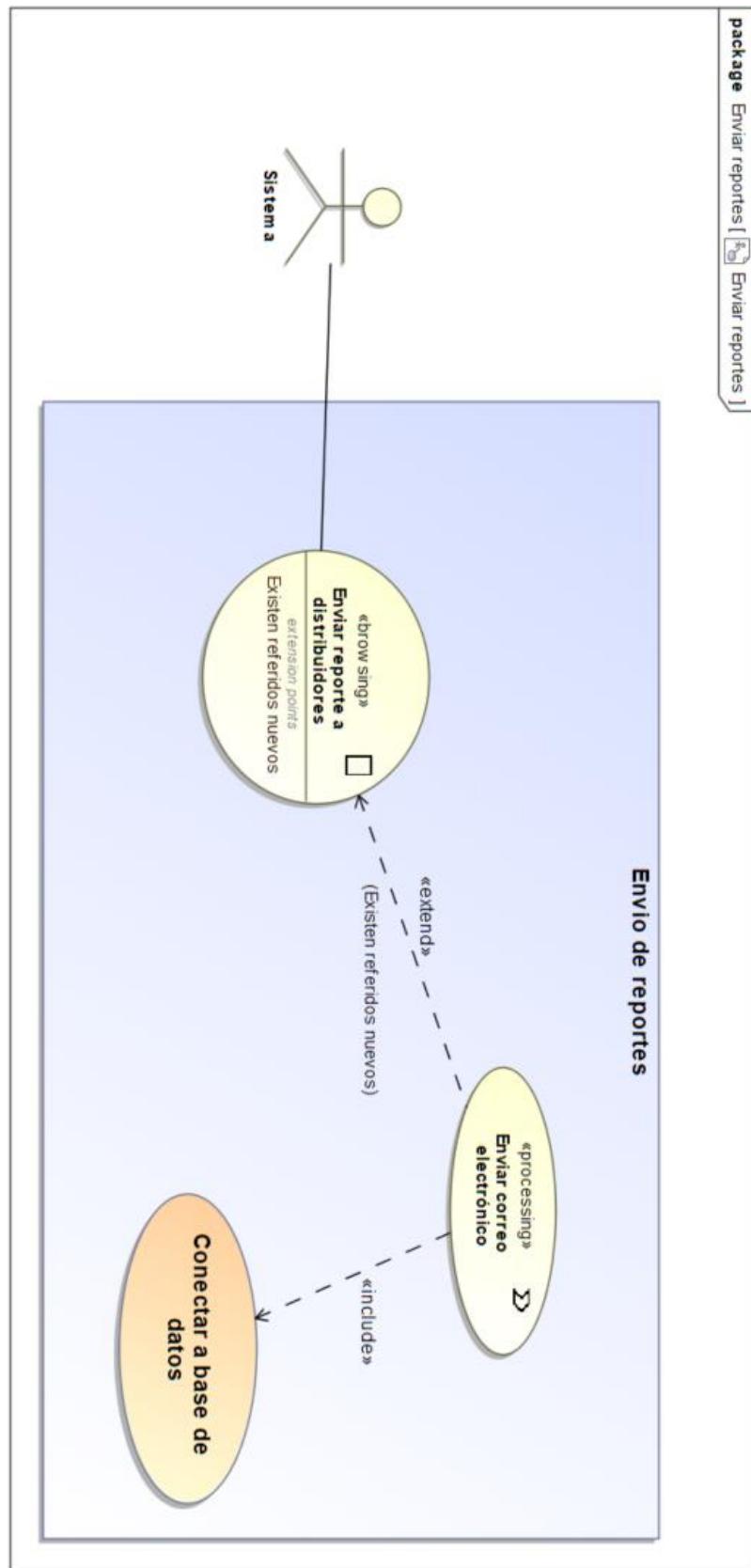
Fuente: Elaboración propia

Ilustración 49: RN-08



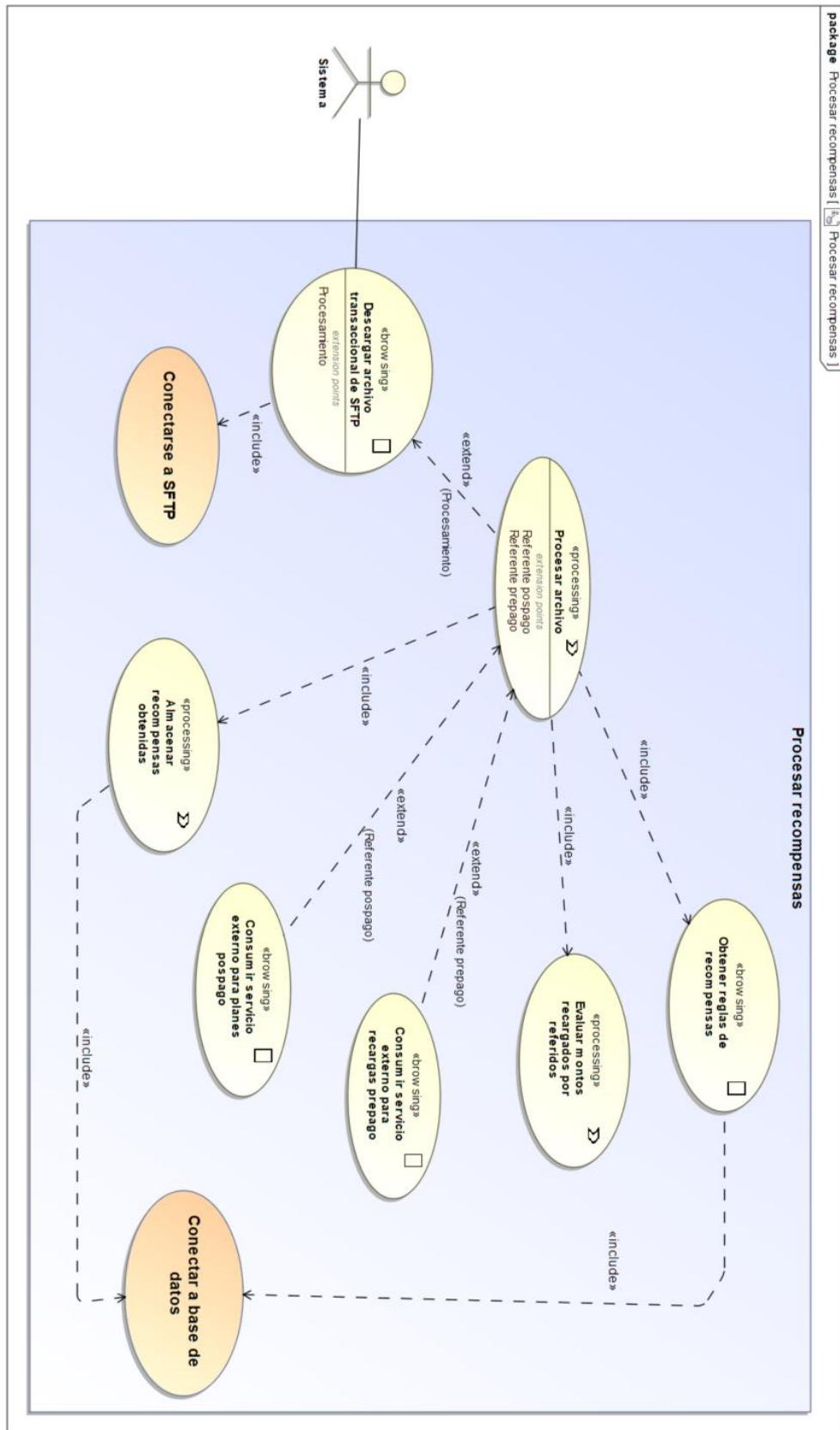
Fuente: Elaboración propia

Ilustración 50: RN-09



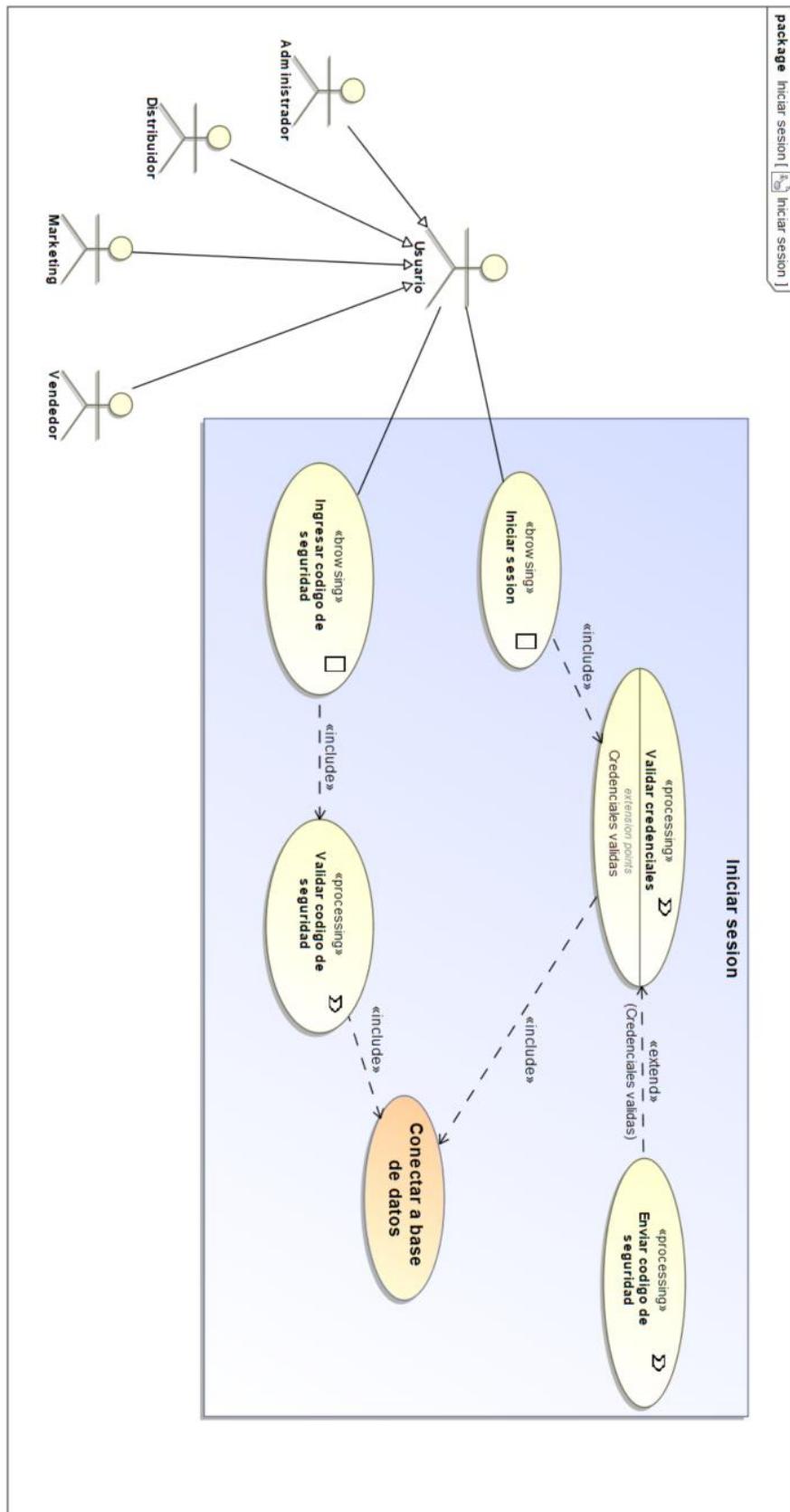
Fuente: Elaboración propia

Ilustración 51: RN-10



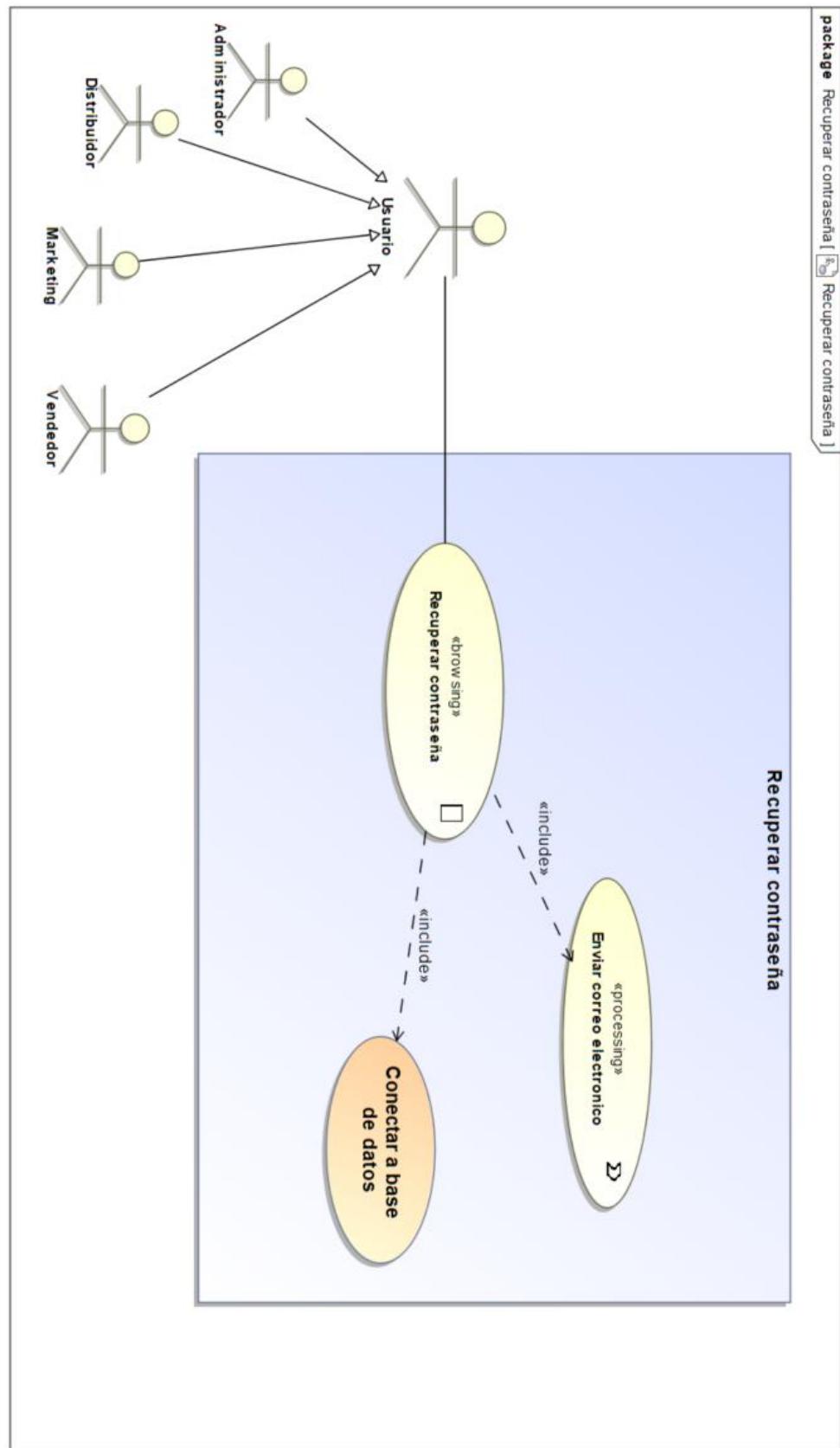
Fuente: Elaboración propia

Ilustración 52: RF-01



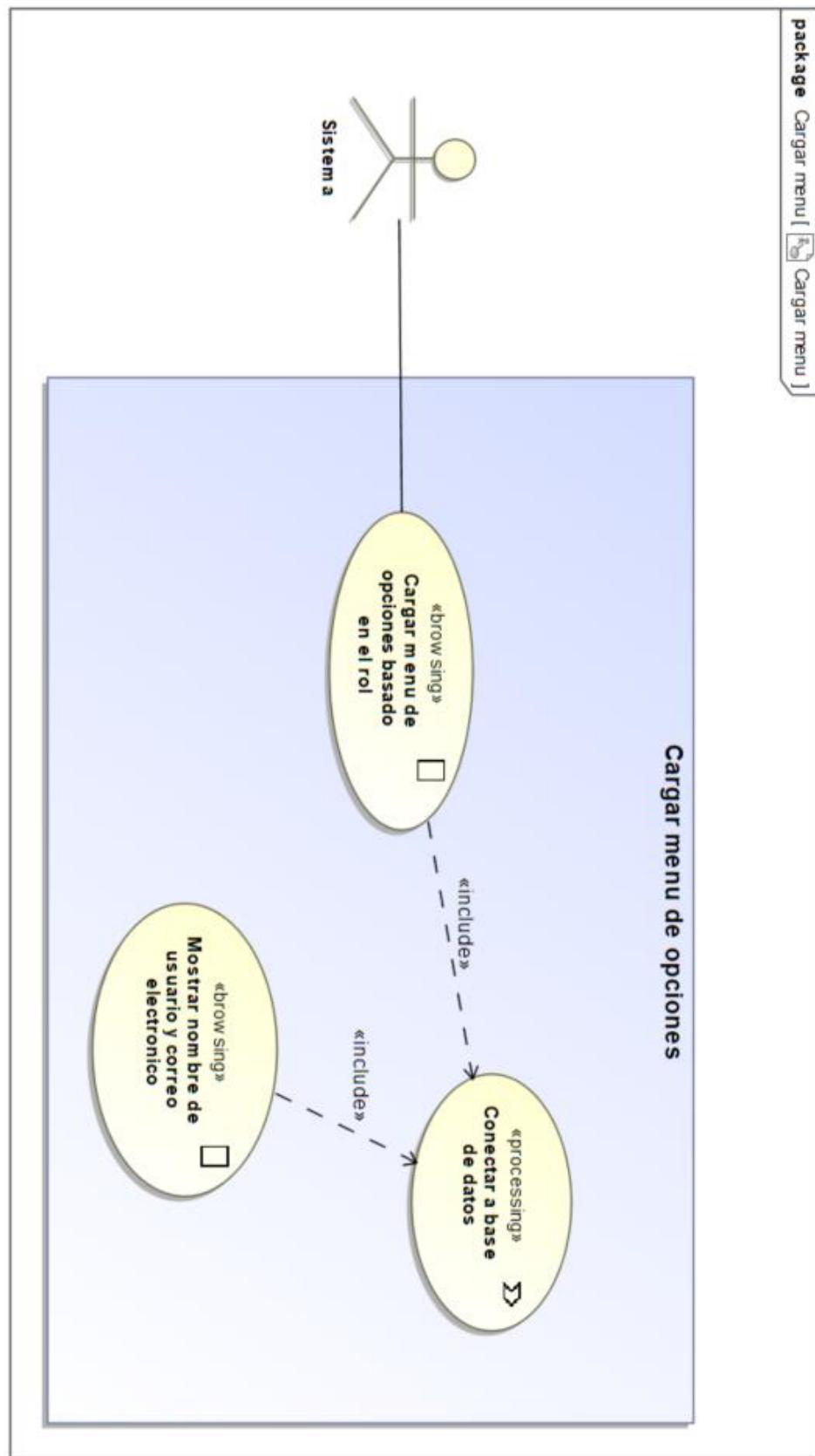
Fuente: Elaboración propia

Ilustración 53: RF-02



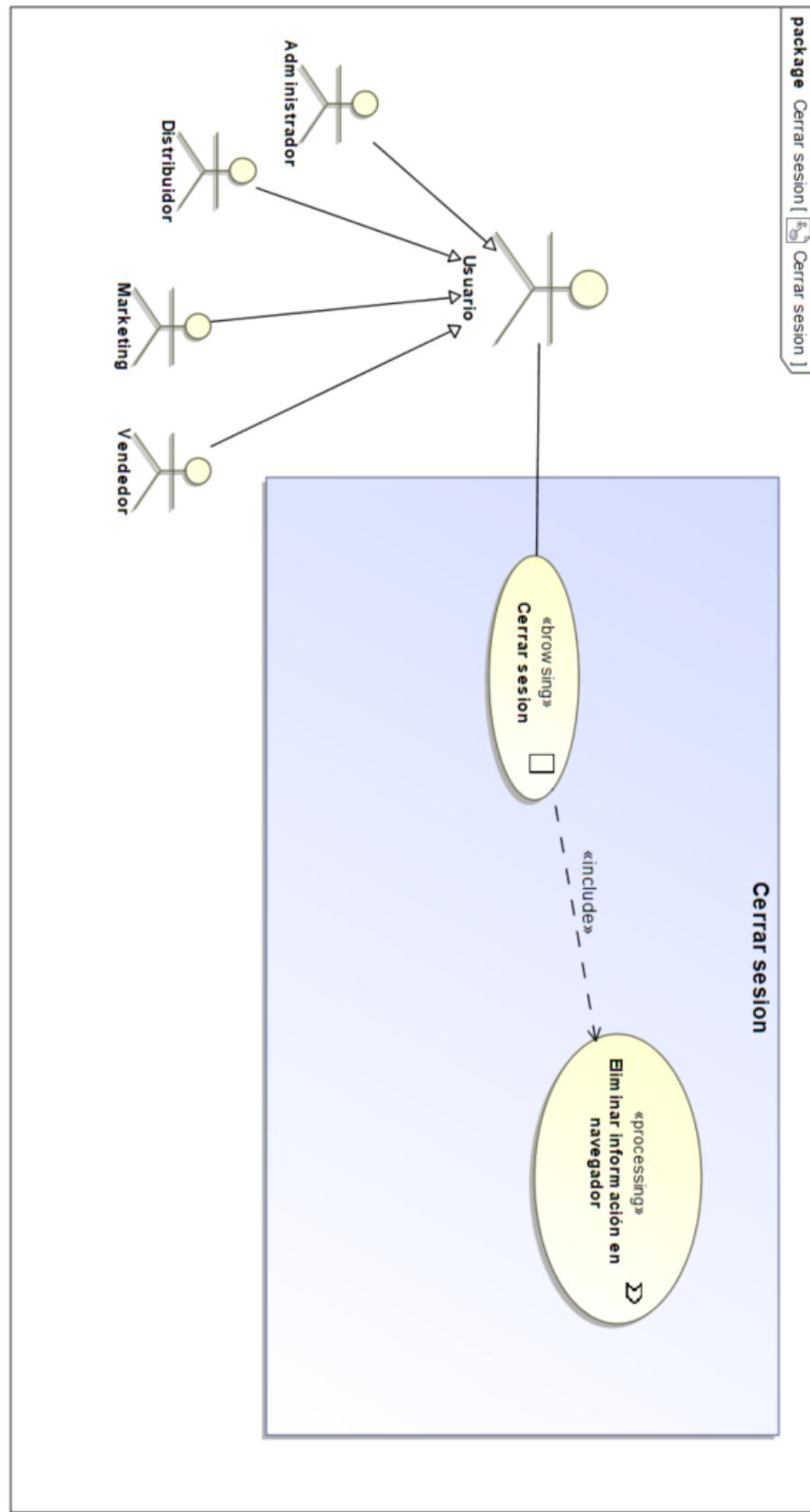
Fuente: Elaboración propia

Ilustración 54: RF-03



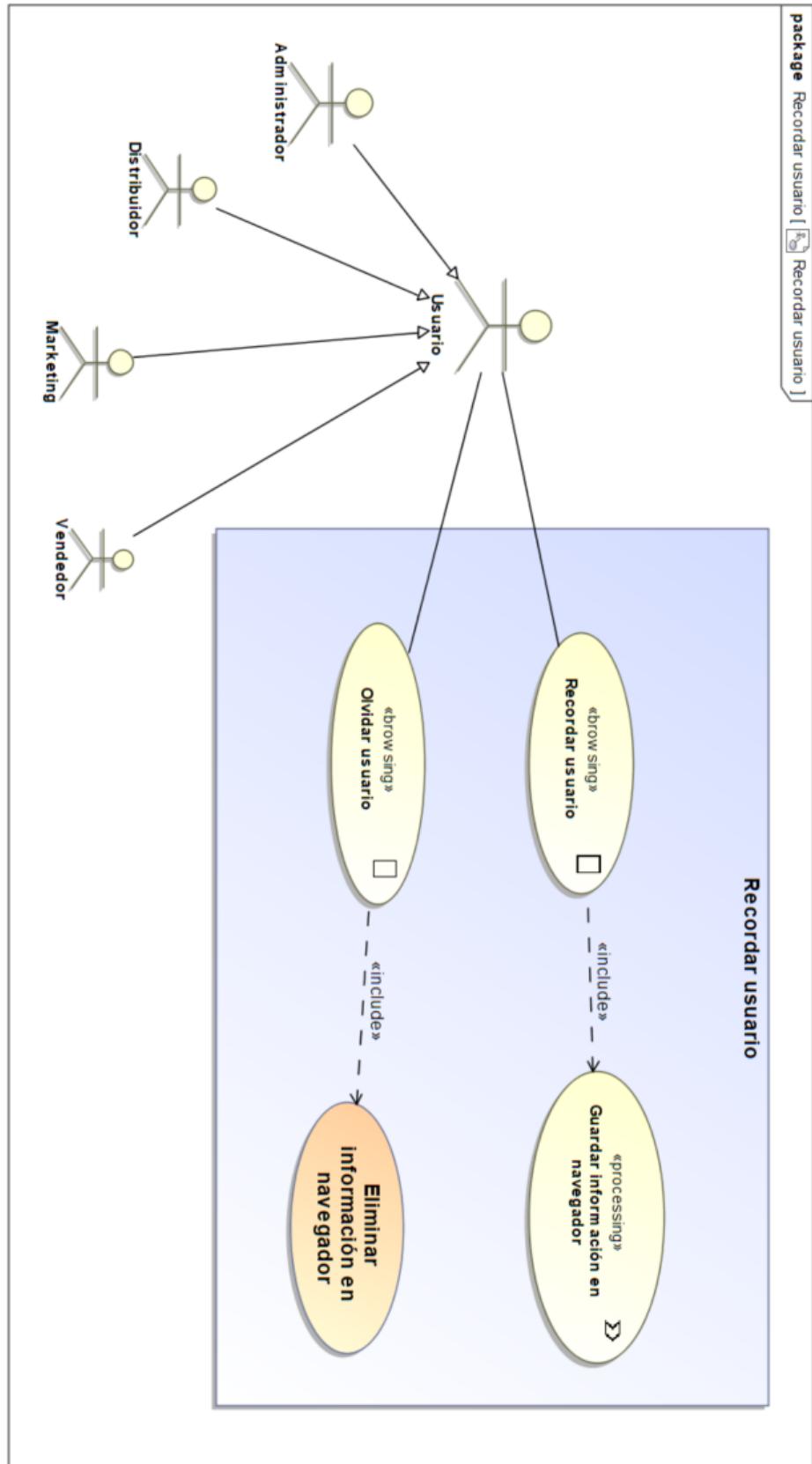
Fuente: Elaboración propia

Ilustración 55: RF-09



Fuente: Elaboración propia

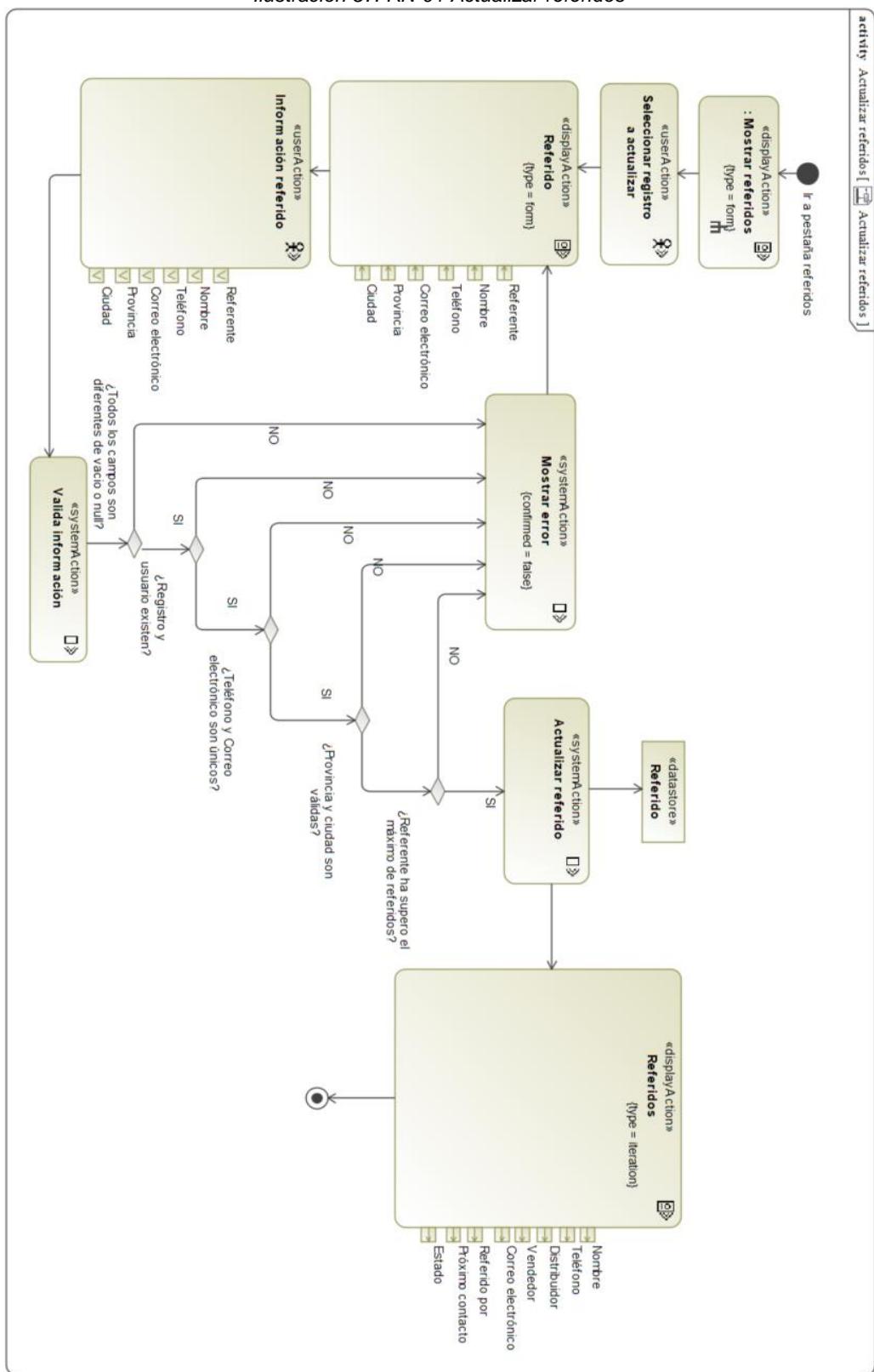
Ilustración 56: RF-12



Fuente: Elaboración propia

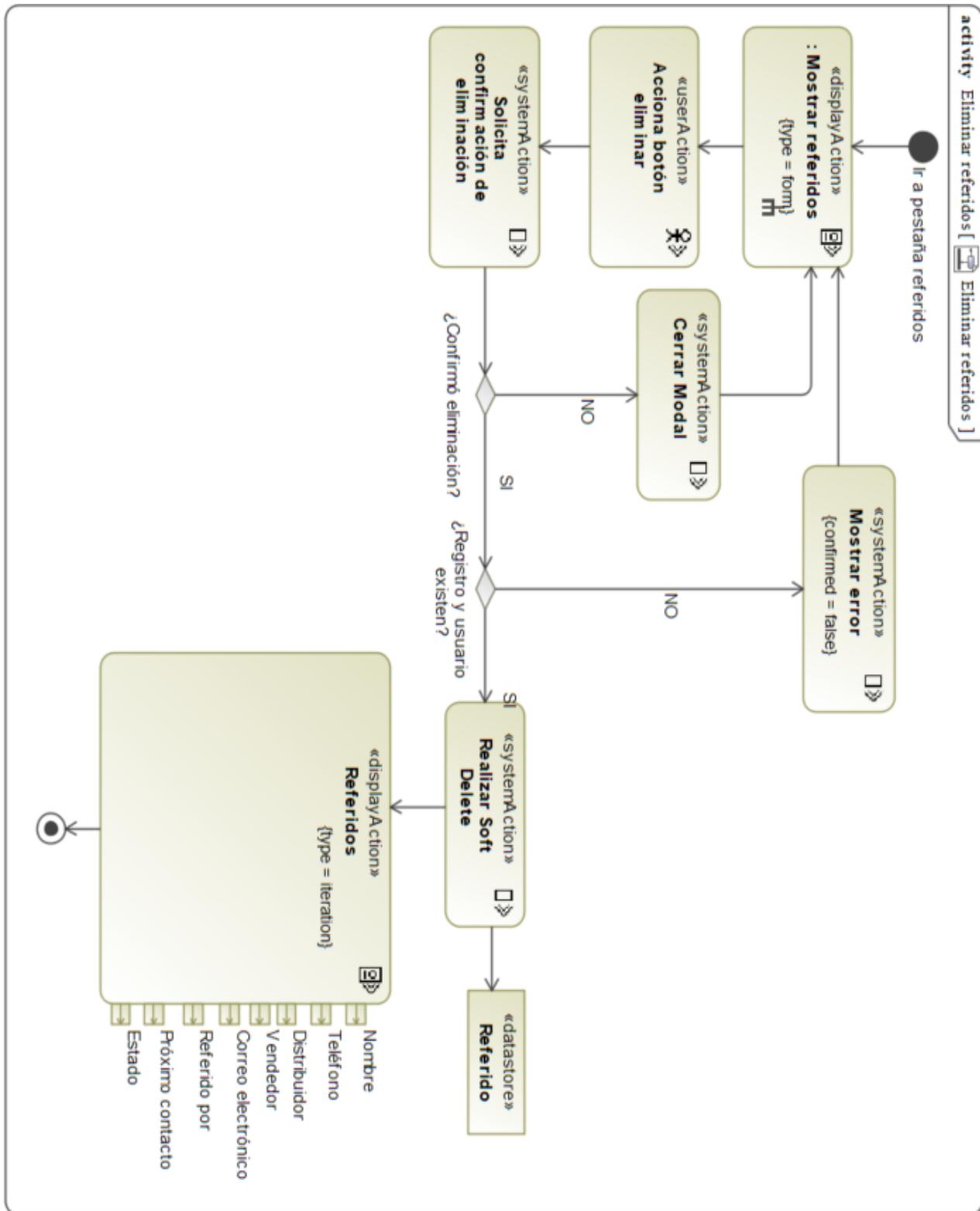
Anexollustración2: Modelo de requerimientos – Diagramas de actividades

Ilustración 57: RN-01 Actualizar referidos



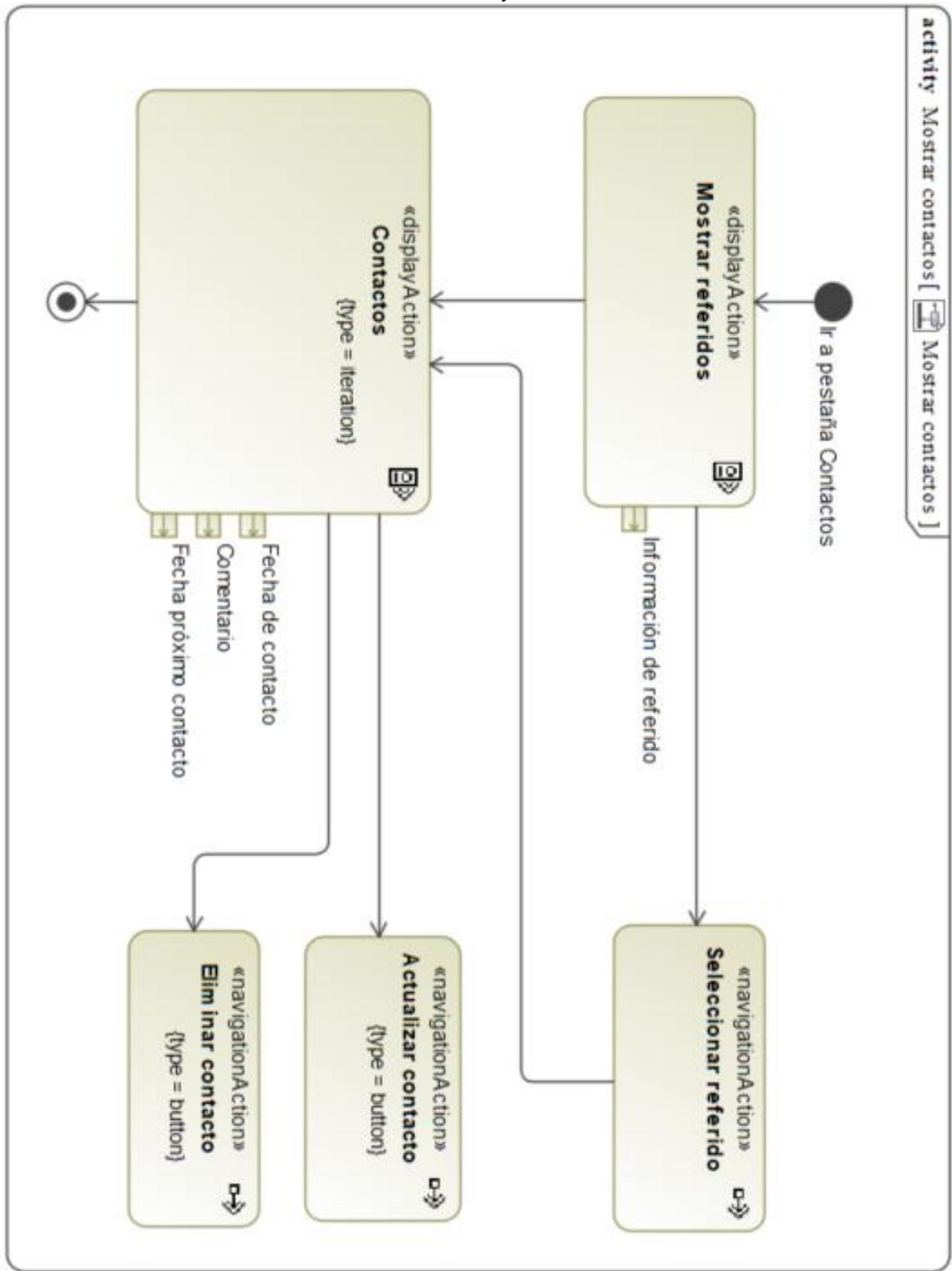
Fuente: Elaboración propia

Ilustración 58: RN-01 Eliminar referido



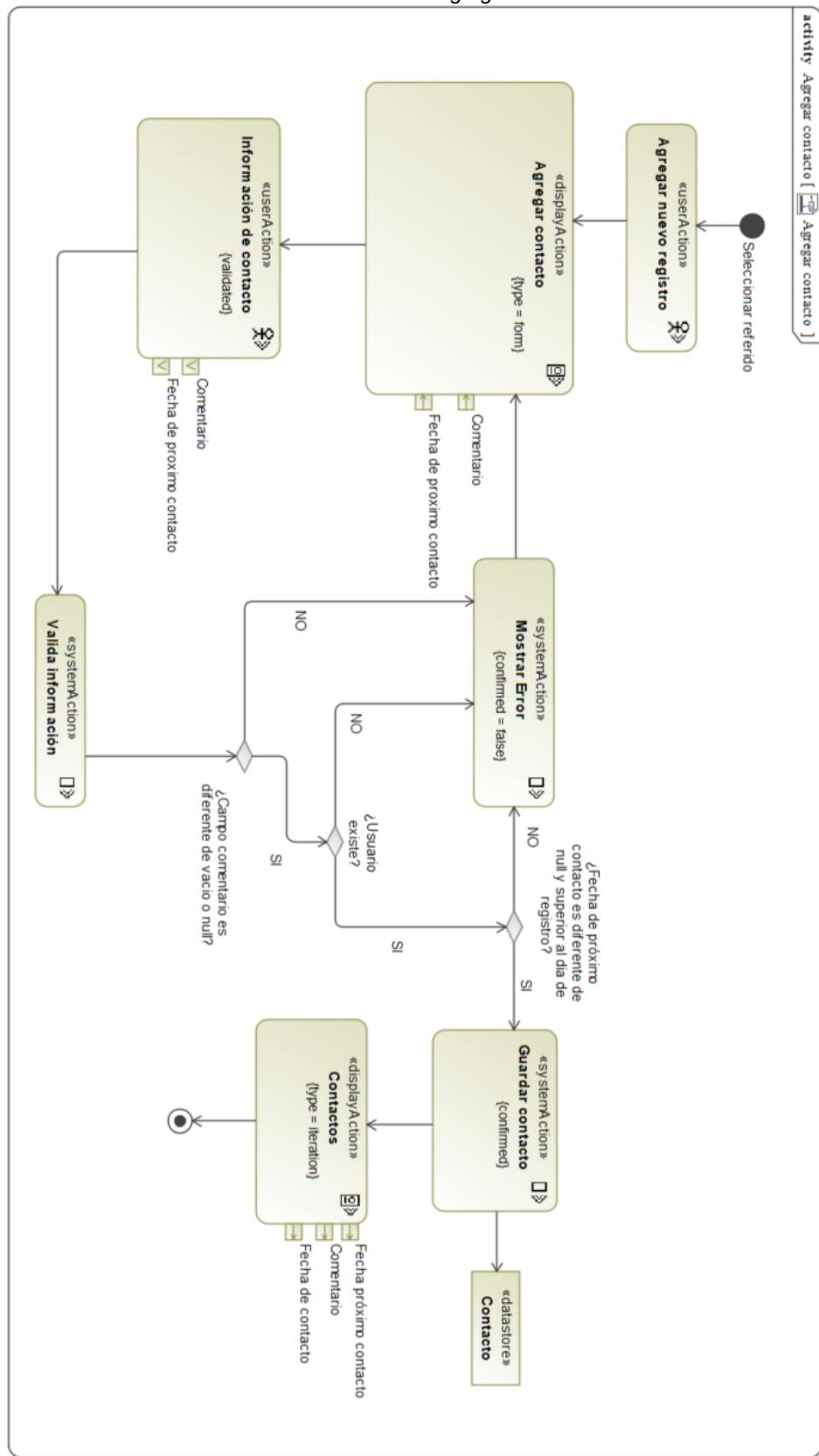
Fuente: Elaboración propia

Ilustración 59: RN-02 Mostrar y filtrar contactos



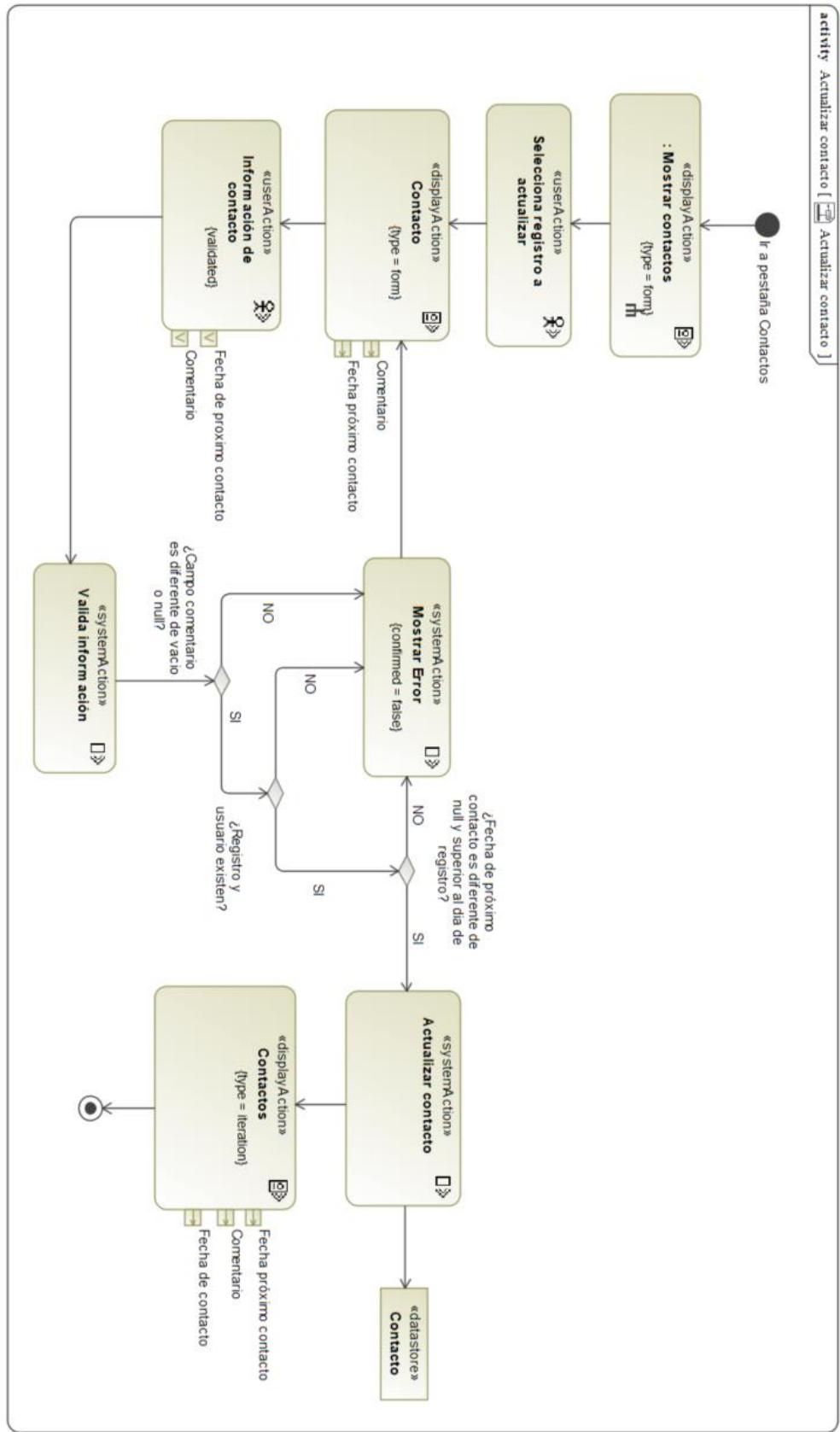
Fuente: Elaboración propia

Ilustración 60: RN-02 Agregar contacto



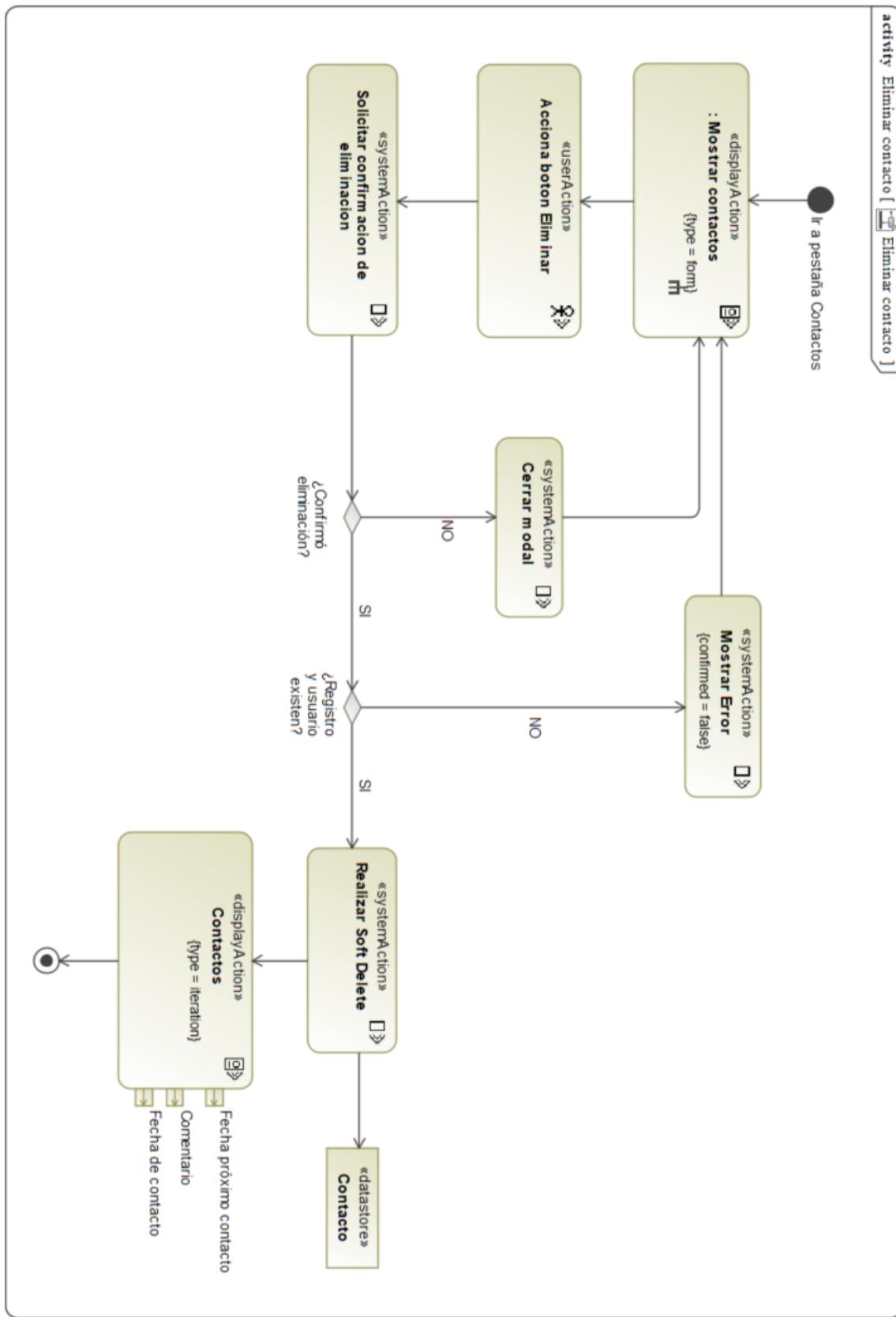
Fuente: Elaboración propia

Ilustración 61: RN-02 Actualizar contacto



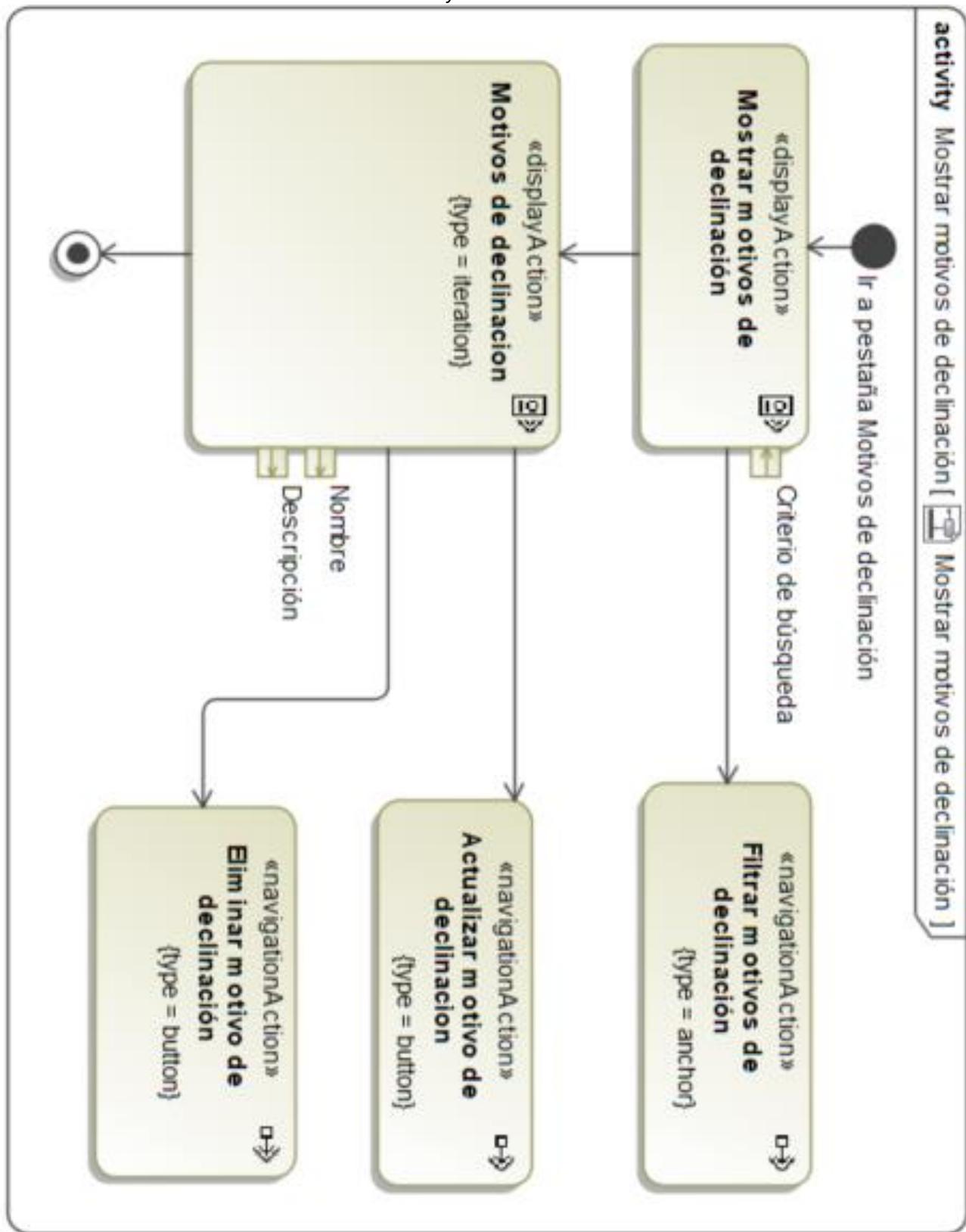
Fuente: Elaboración propia

Ilustración 62: RN-02 Eliminar contacto



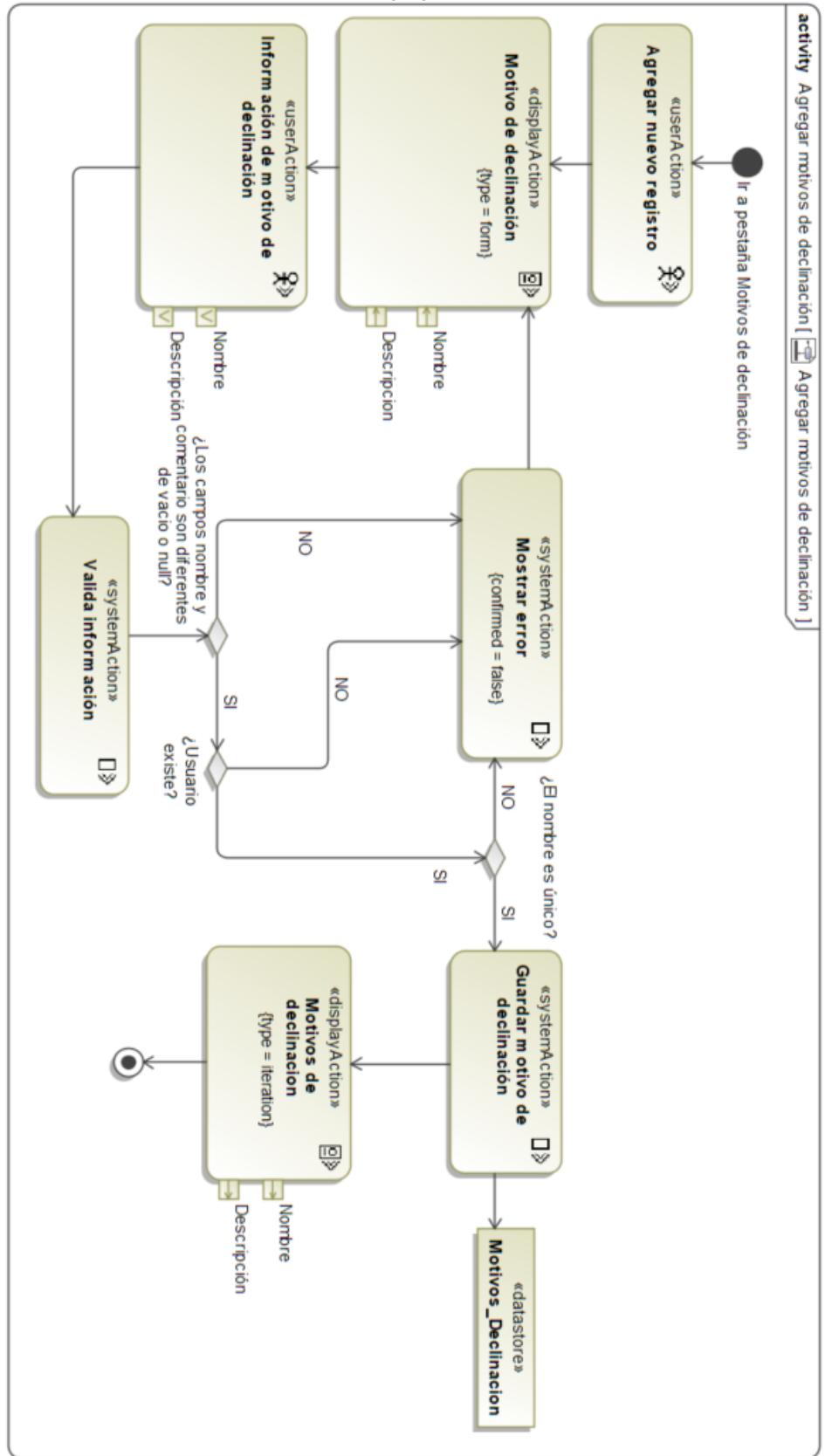
Fuente: Elaboración propia

Ilustración 63: RN-03 Mostrar y filtrar motivos de declinación



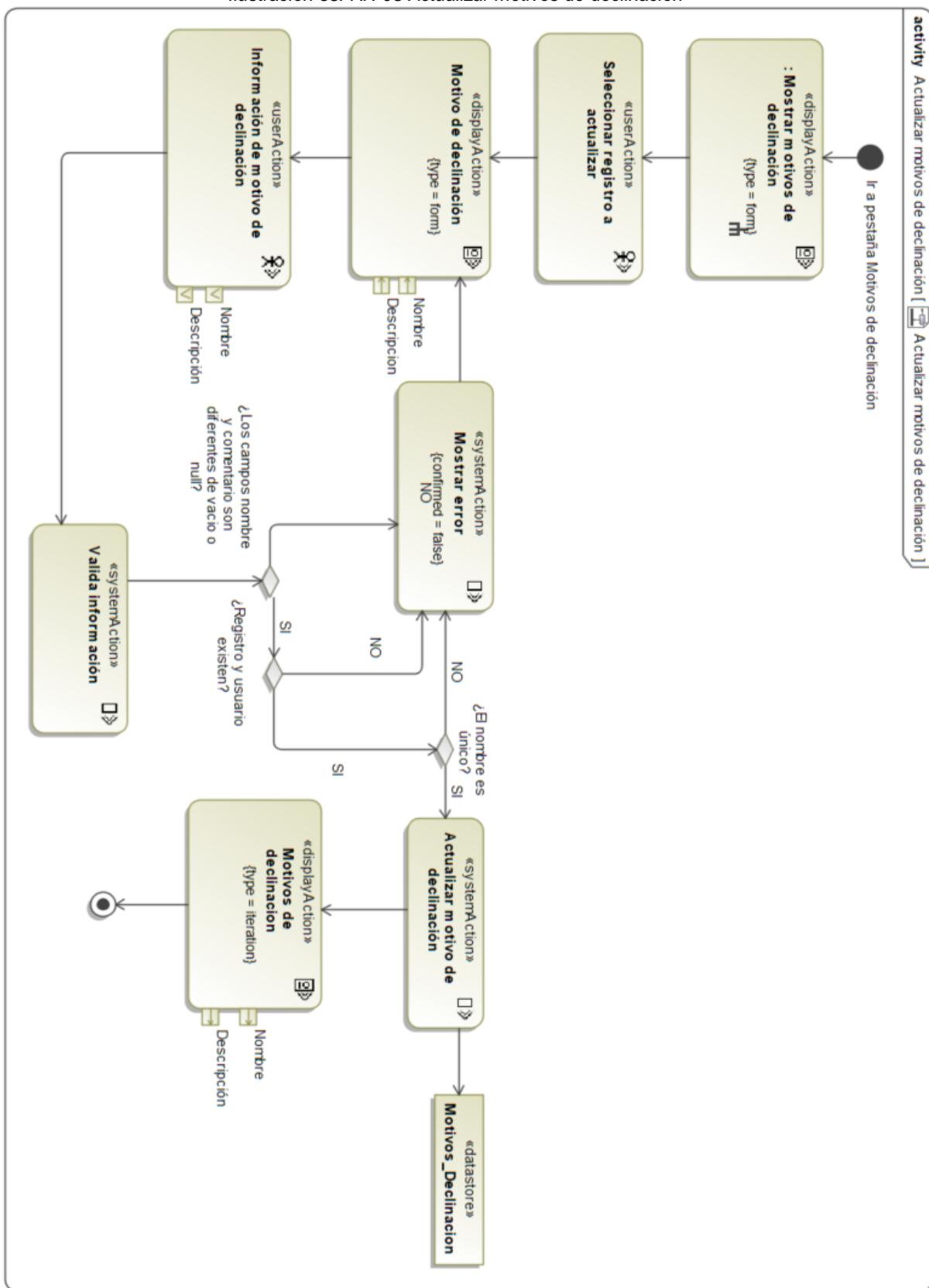
Fuente: Elaboración propia

Ilustración 64: RN-03 Agregar motivo de declinación



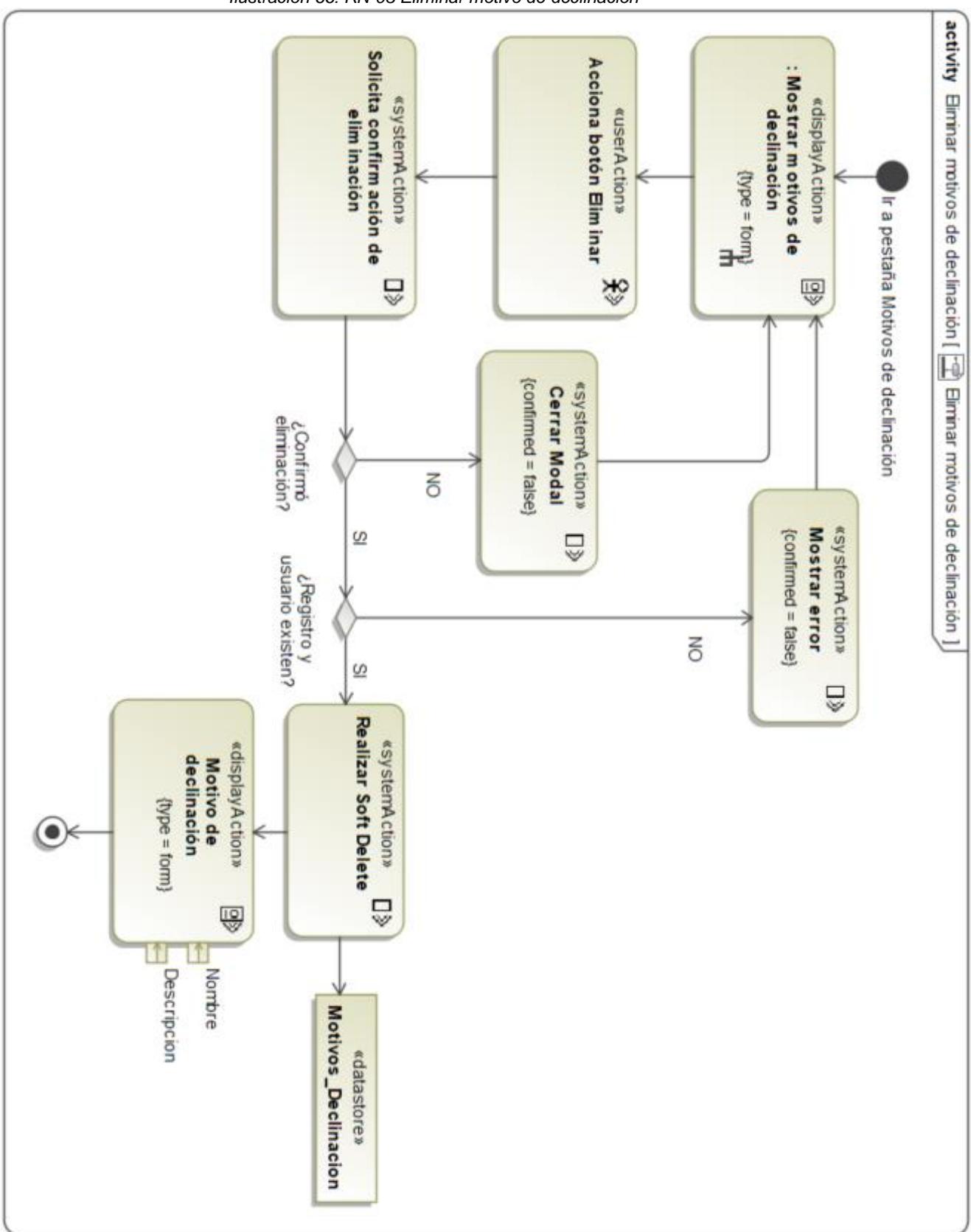
Fuente: Elaboración propia

Ilustración 65: RN-03 Actualizar motivos de declinación



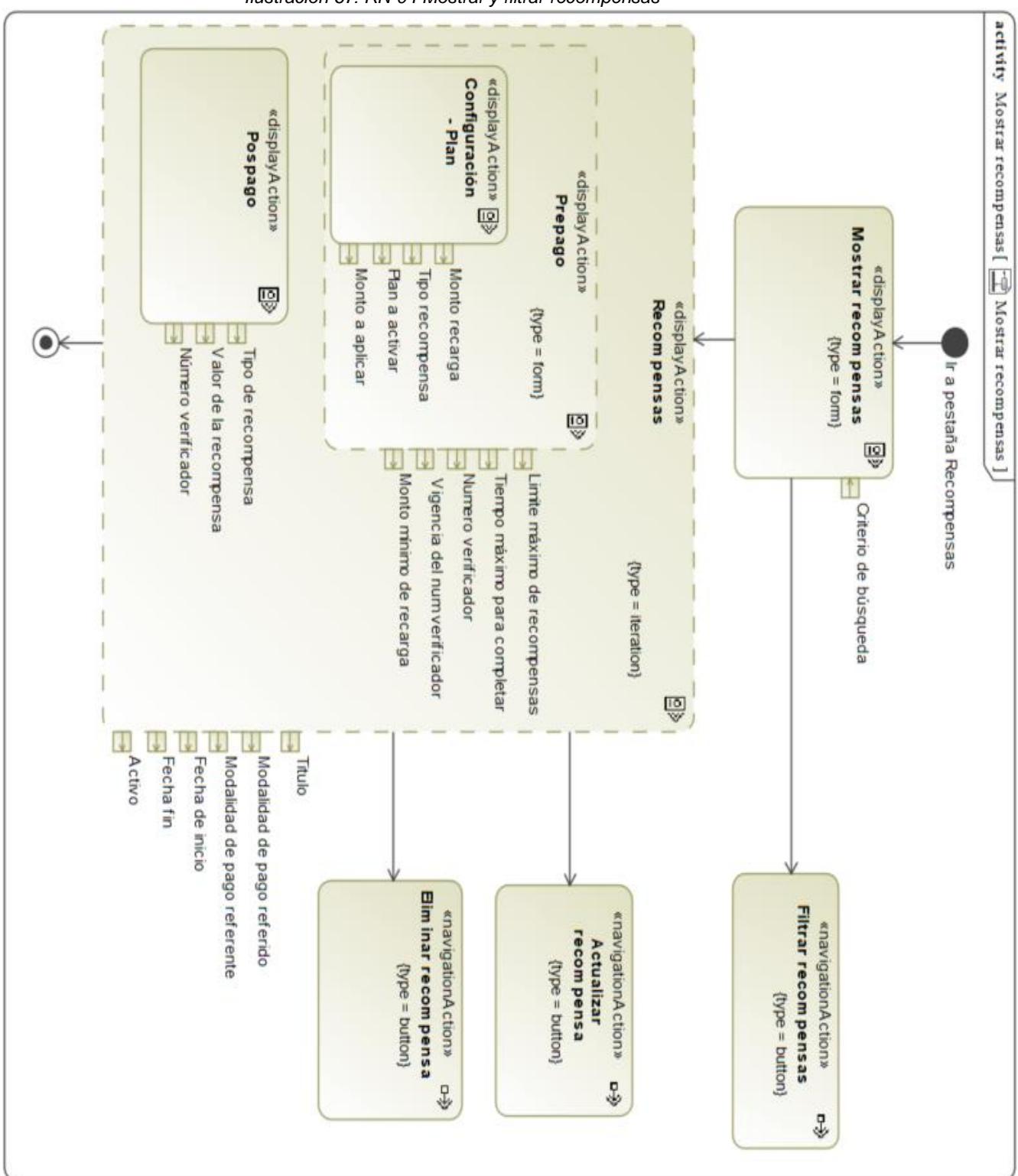
Fuente: Elaboración propia

activity Eliminar motivos de declinación [Eliminar motivos de declinación]



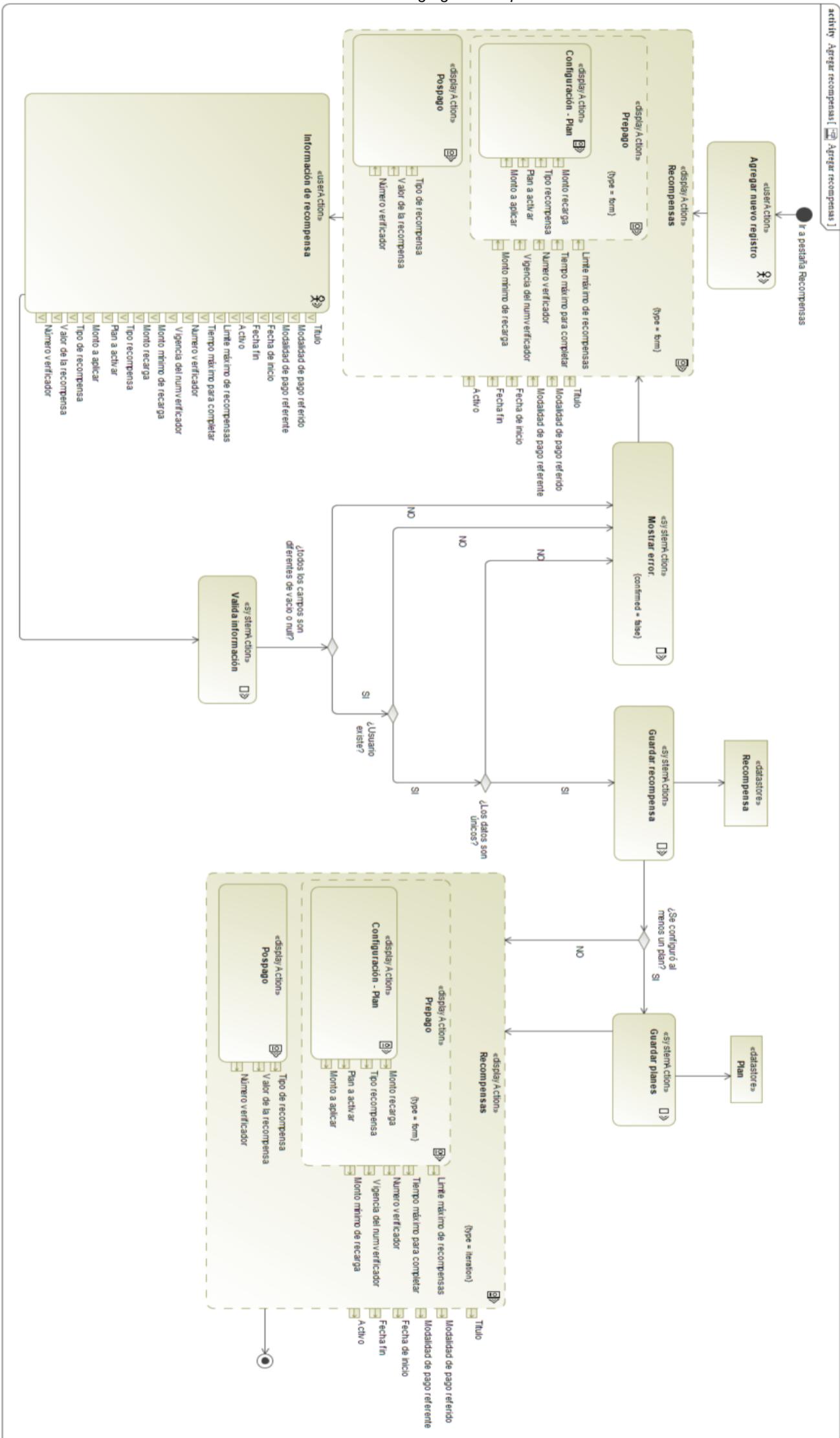
Fuente: Elaboración propia

activity Mostrar recompensas [ Mostrar recompensas]



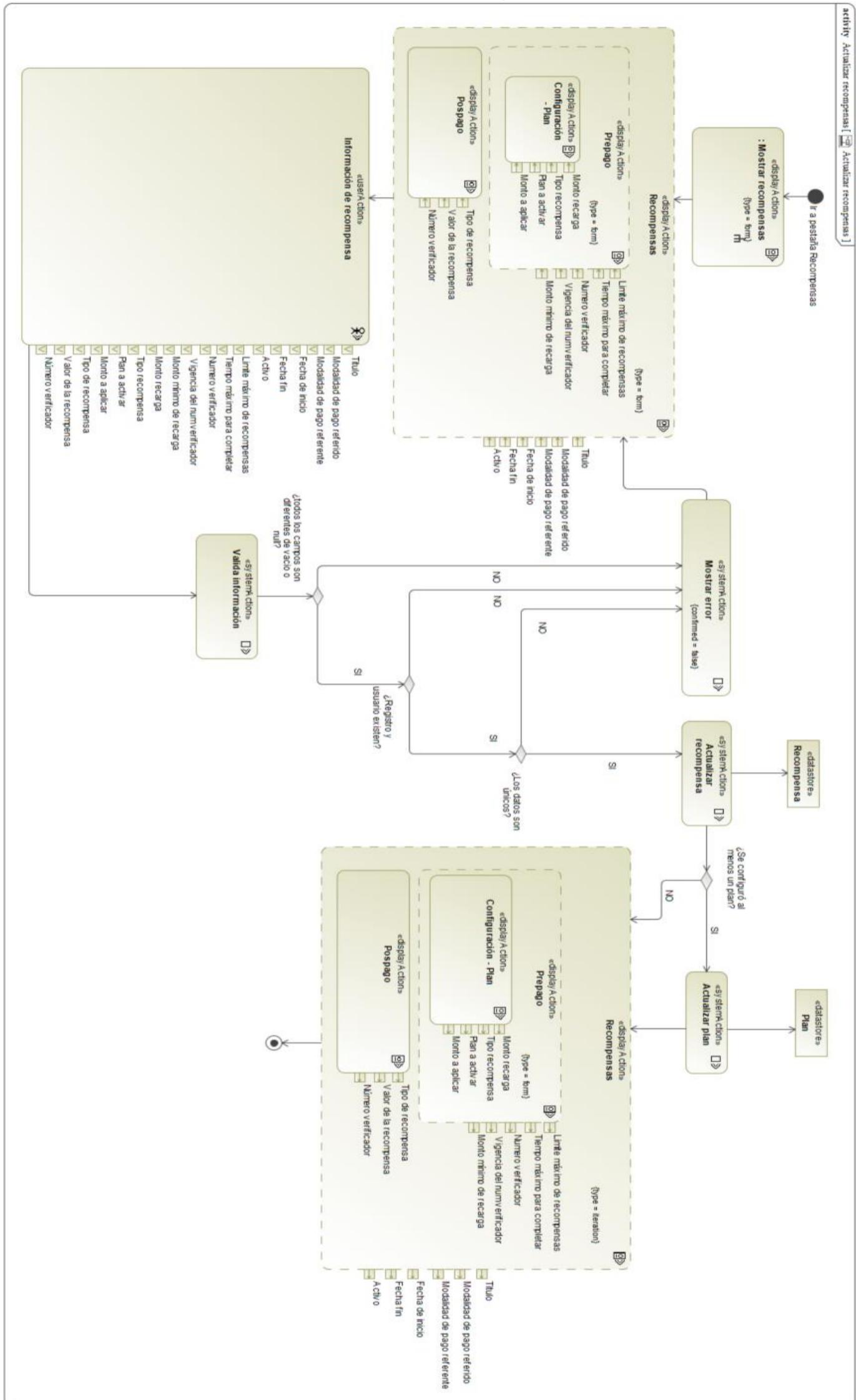
Fuente: Elaboración propia

Ilustración 68: RN-04 Agregar recompensa



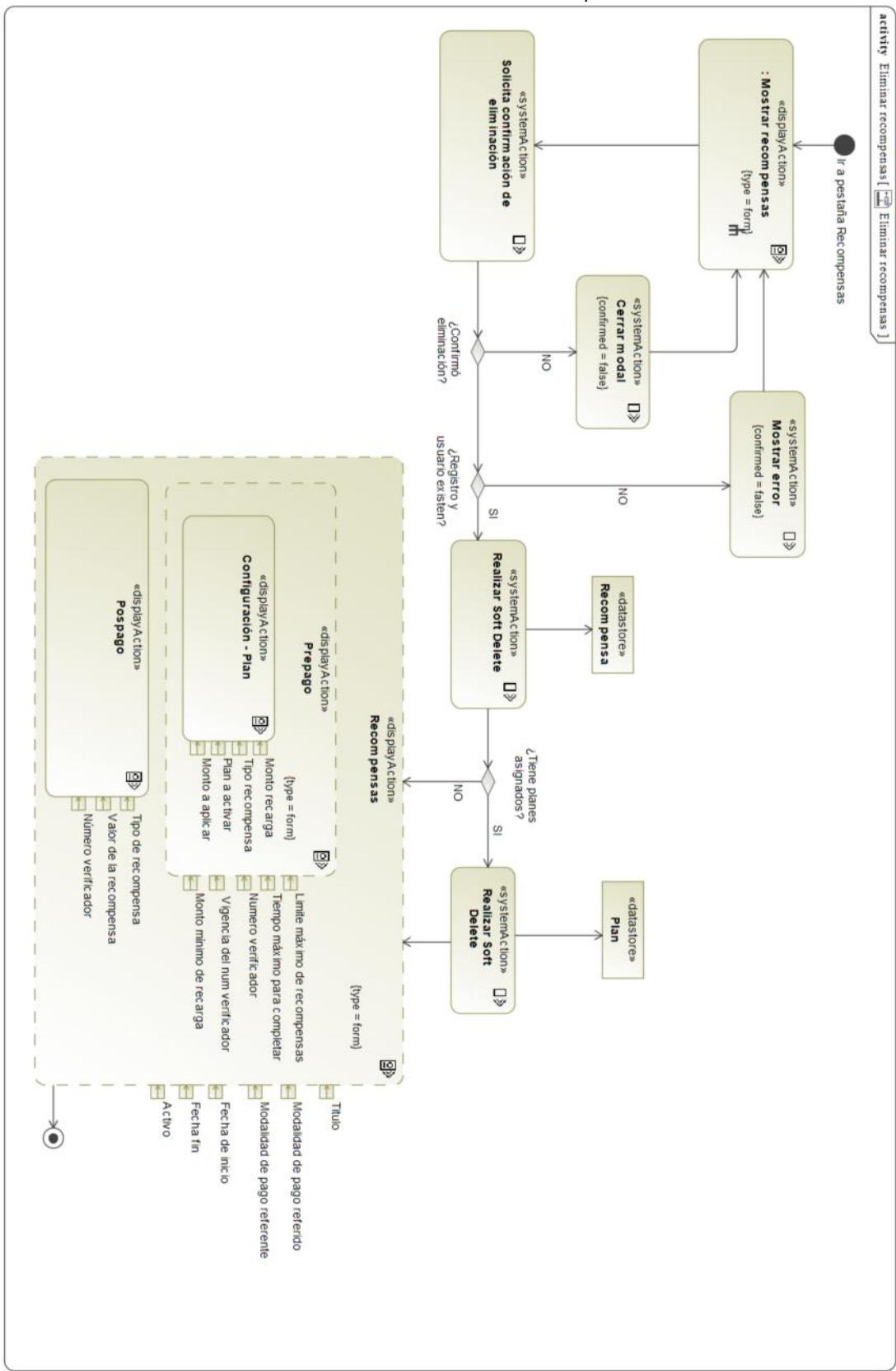
Fuente: Elaboración propia

Ilustración 69: RN-04 Actualizar recompensa



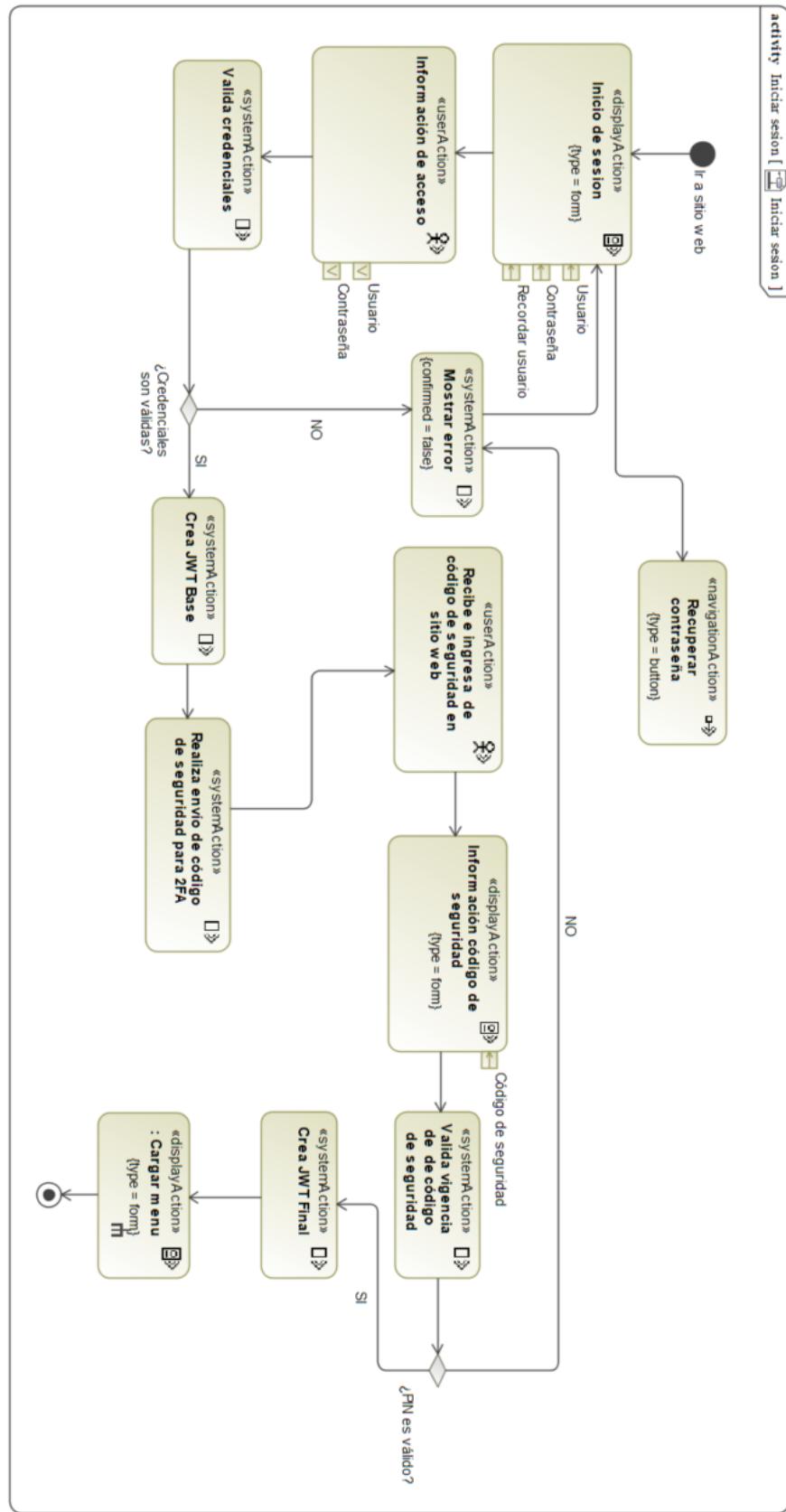
Fuente: Elaboración propia

Ilustración 70: RN-04 Eliminar recompensa



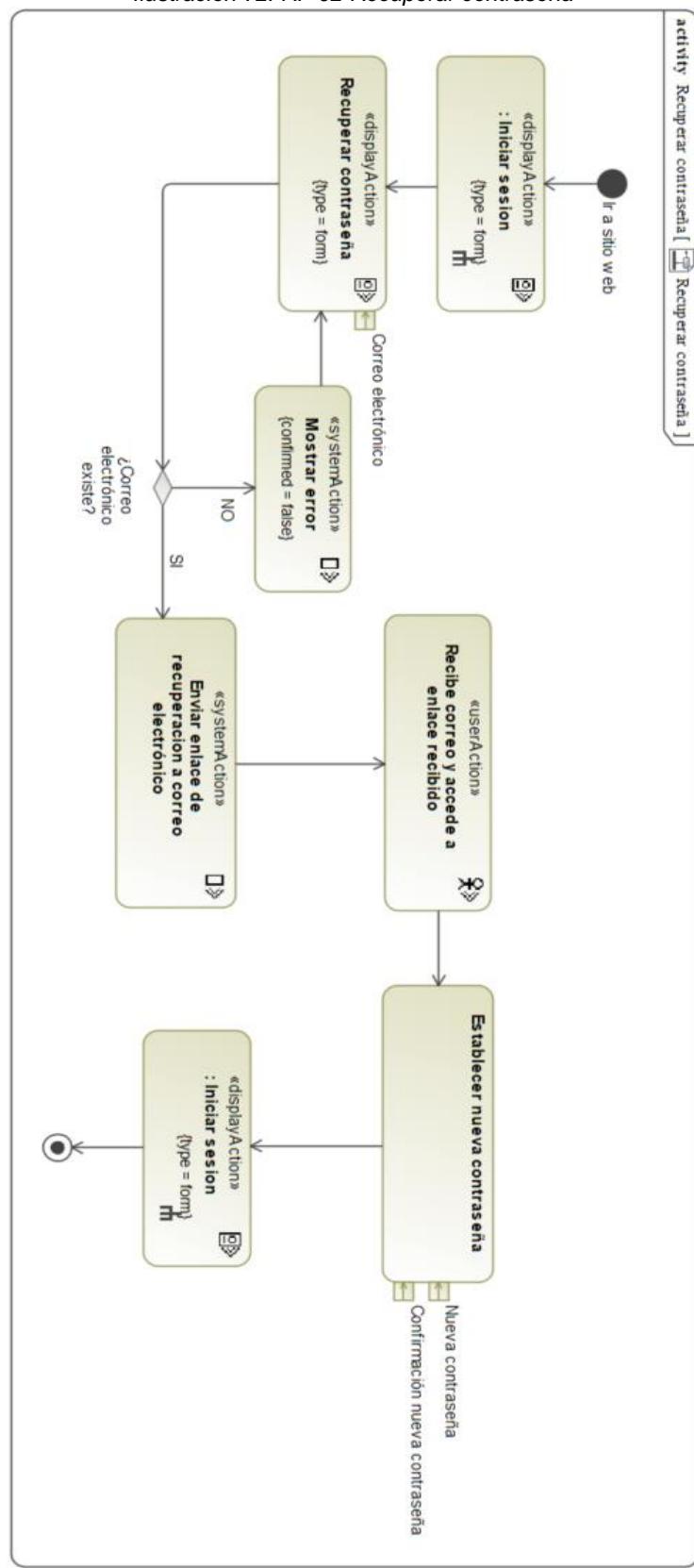
Fuente: Elaboración propia

Ilustración 71: RF-01 Iniciar sesión



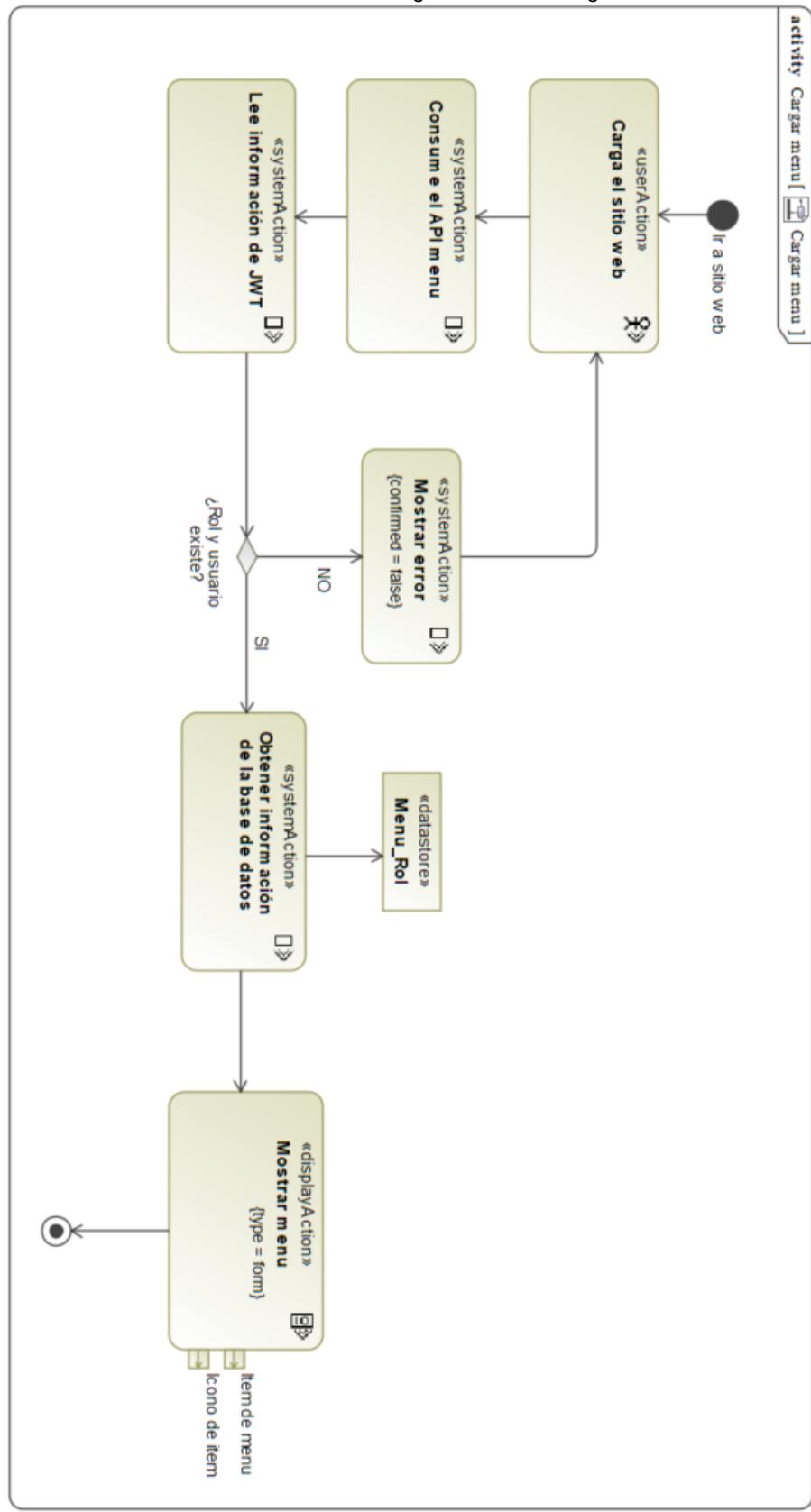
Fuente: Elaboración propia

Ilustración 72: RF-02 Recuperar contraseña



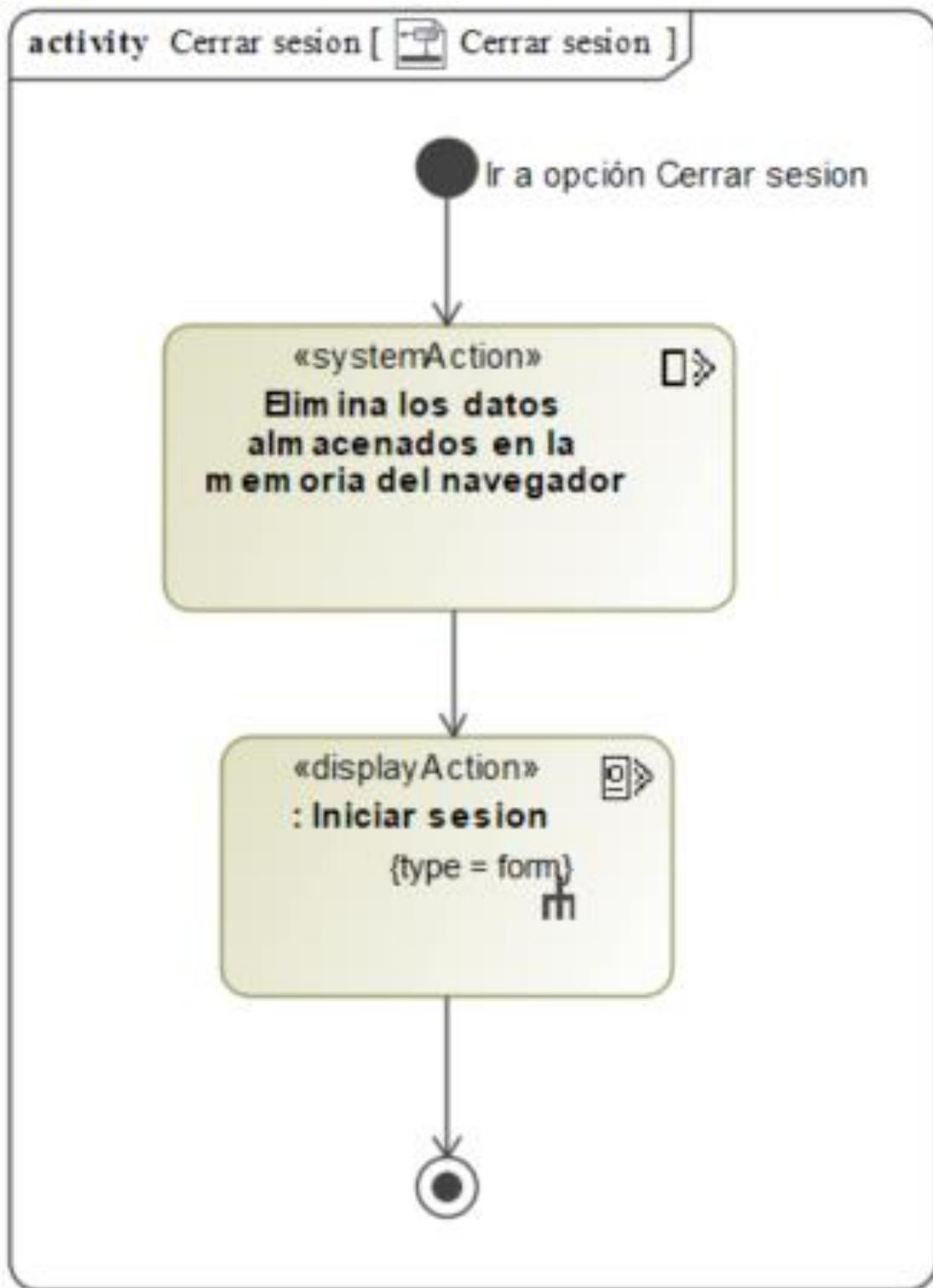
Fuente: Elaboración propia

Ilustración 73: RF-03 Cargar menu de navegación



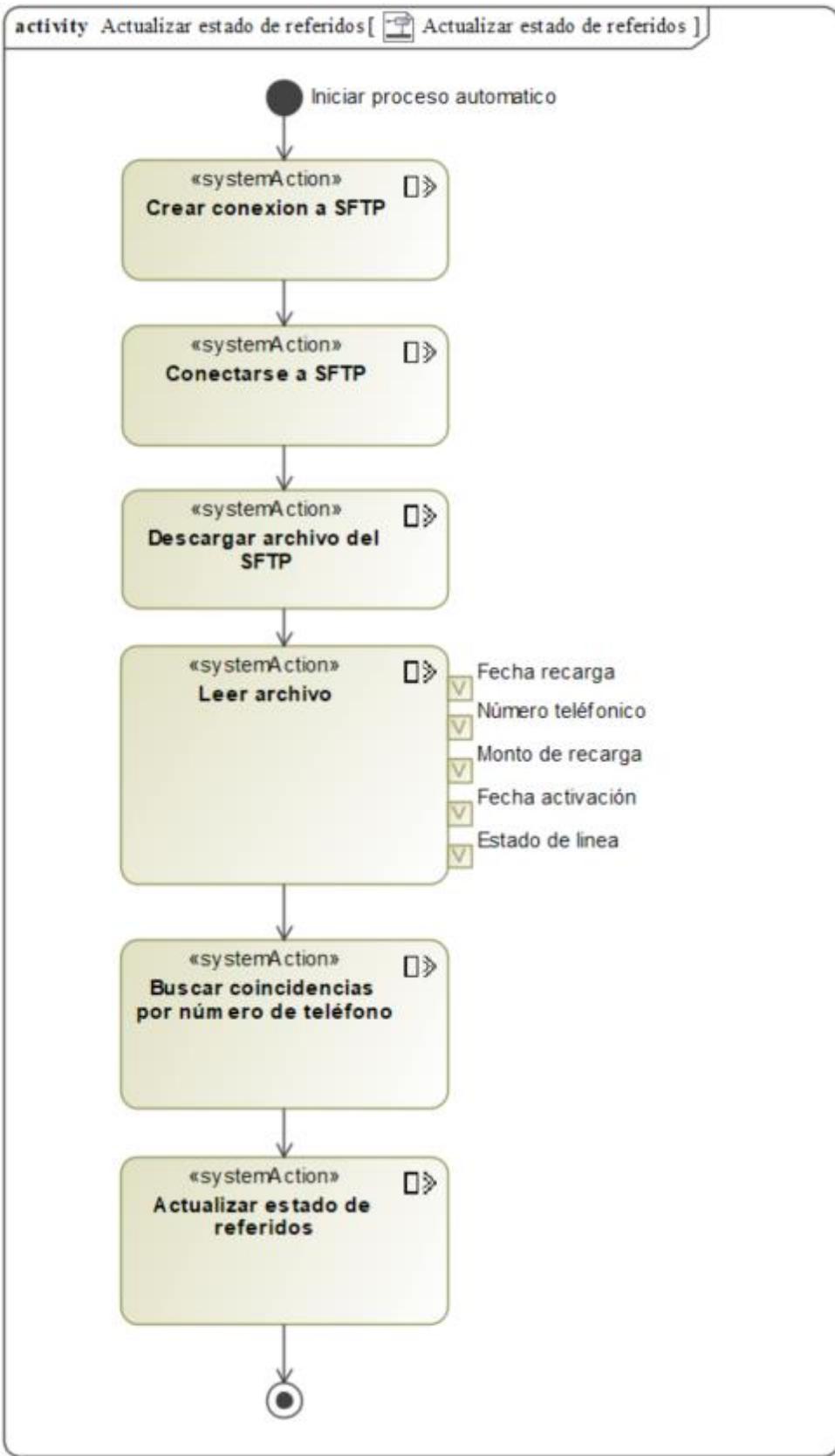
Fuente: Elaboración propia

Ilustración 74: RF-09 Cerrar sesión



Fuente: Elaboración propia

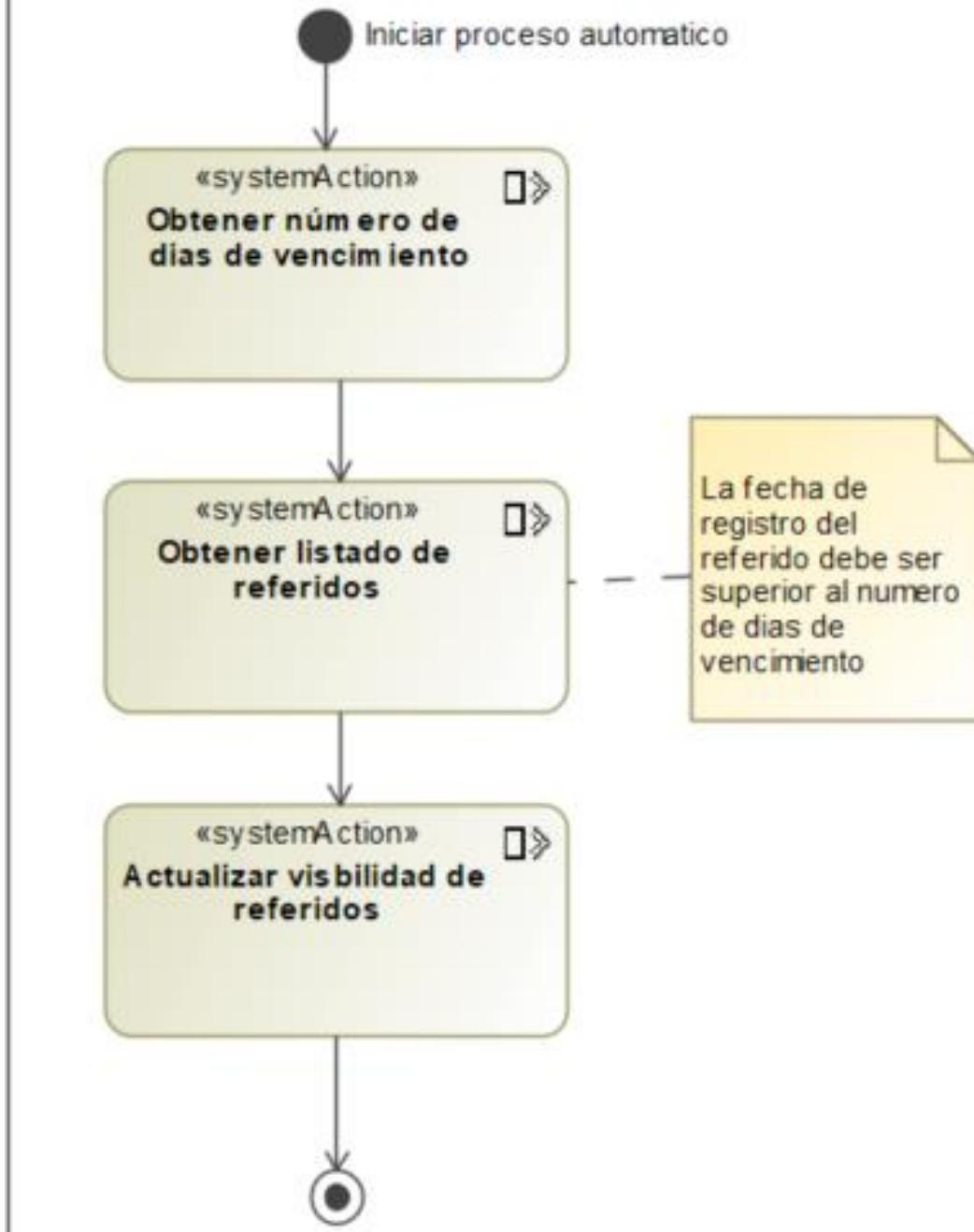
Ilustración 75: RN-07 Actualizar estado de referidos



Fuente: Elaboración propia

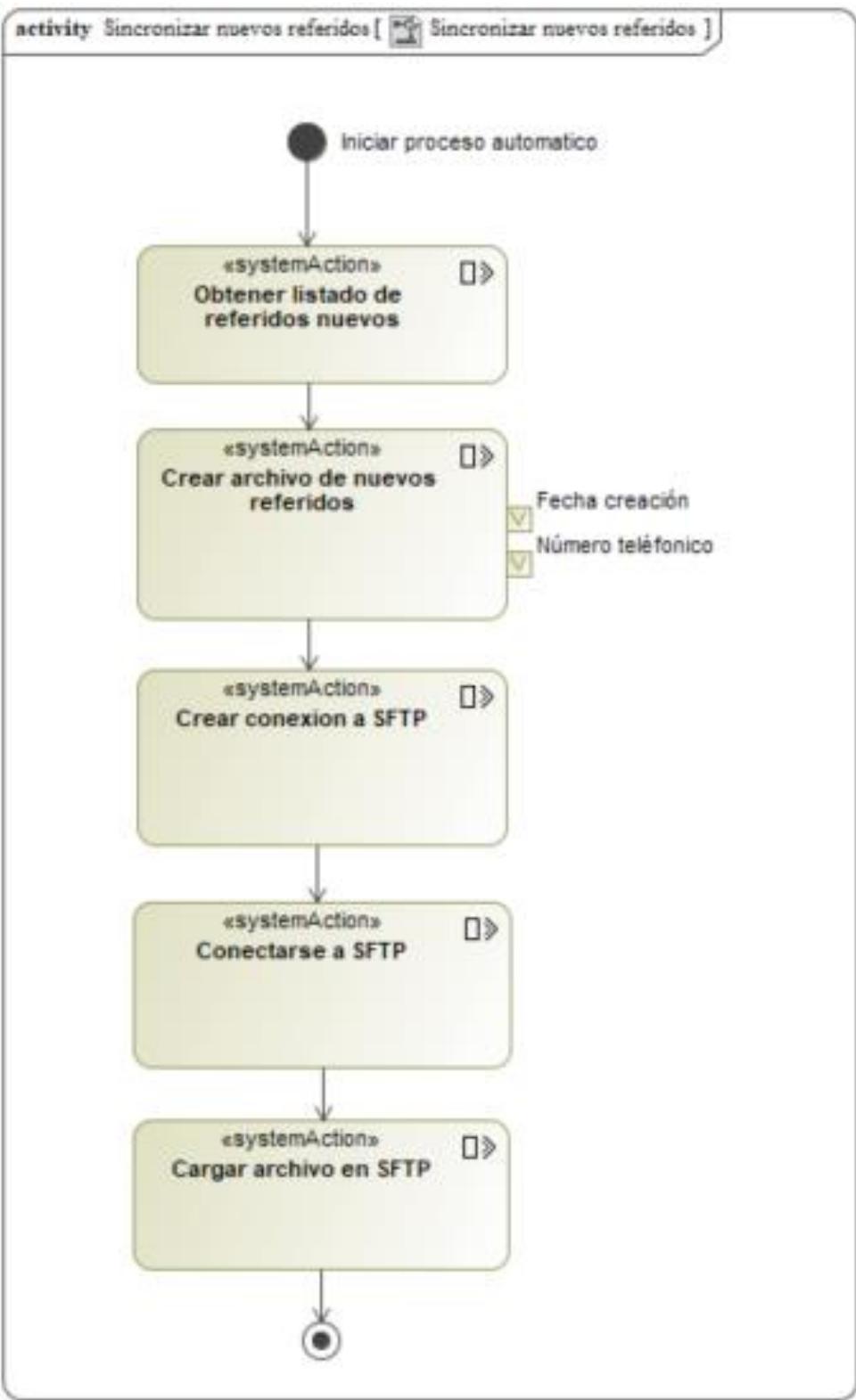
Ilustración 76: RN-06 Cambiar visibilidad de referidos

activity Cambiar visibilidad de referidos [ Cambiar visibilidad de referidos]



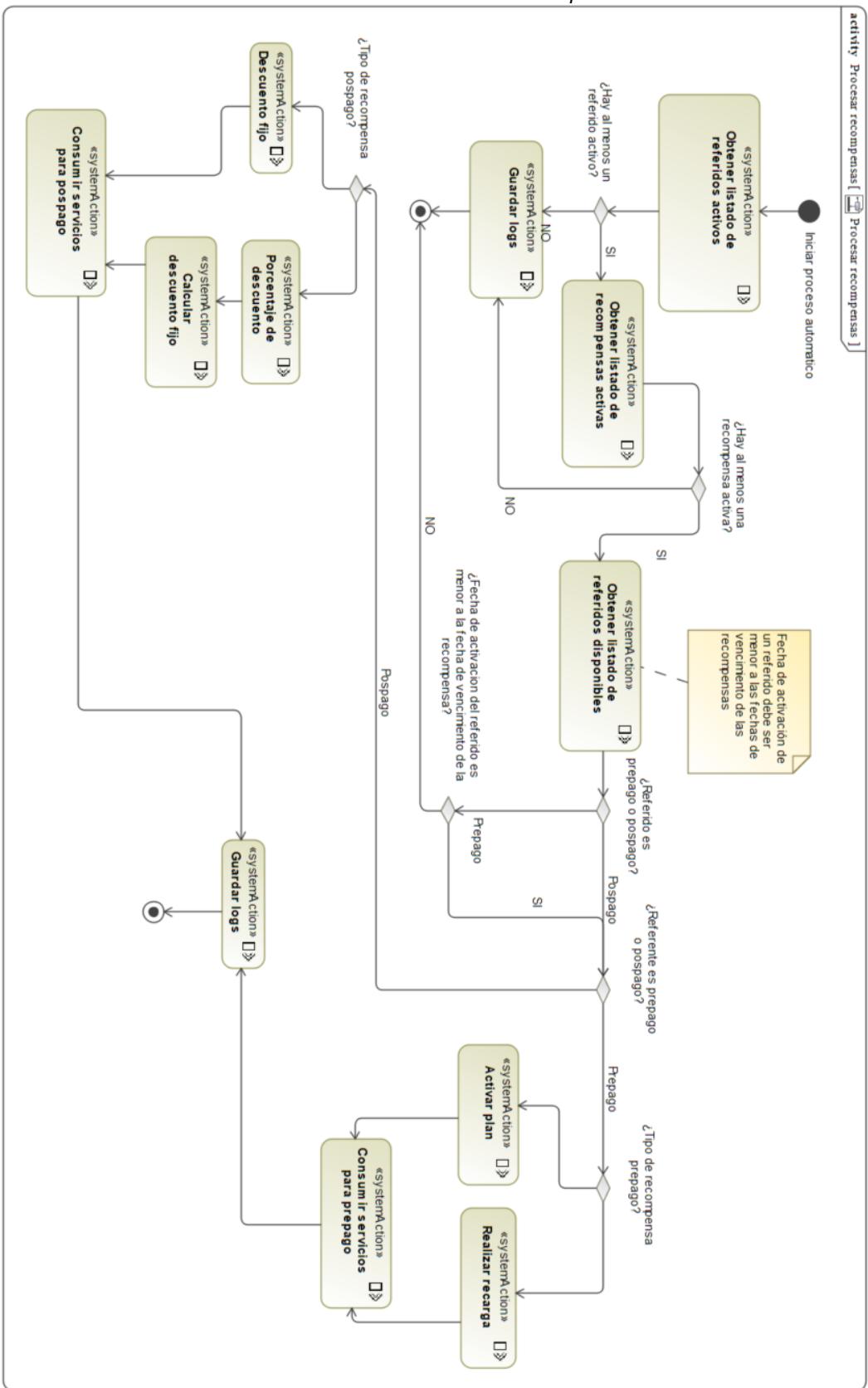
Fuente: Elaboración propia

Ilustración 77: RN-08 Sincronizar referidos



Fuente: Elaboración propia

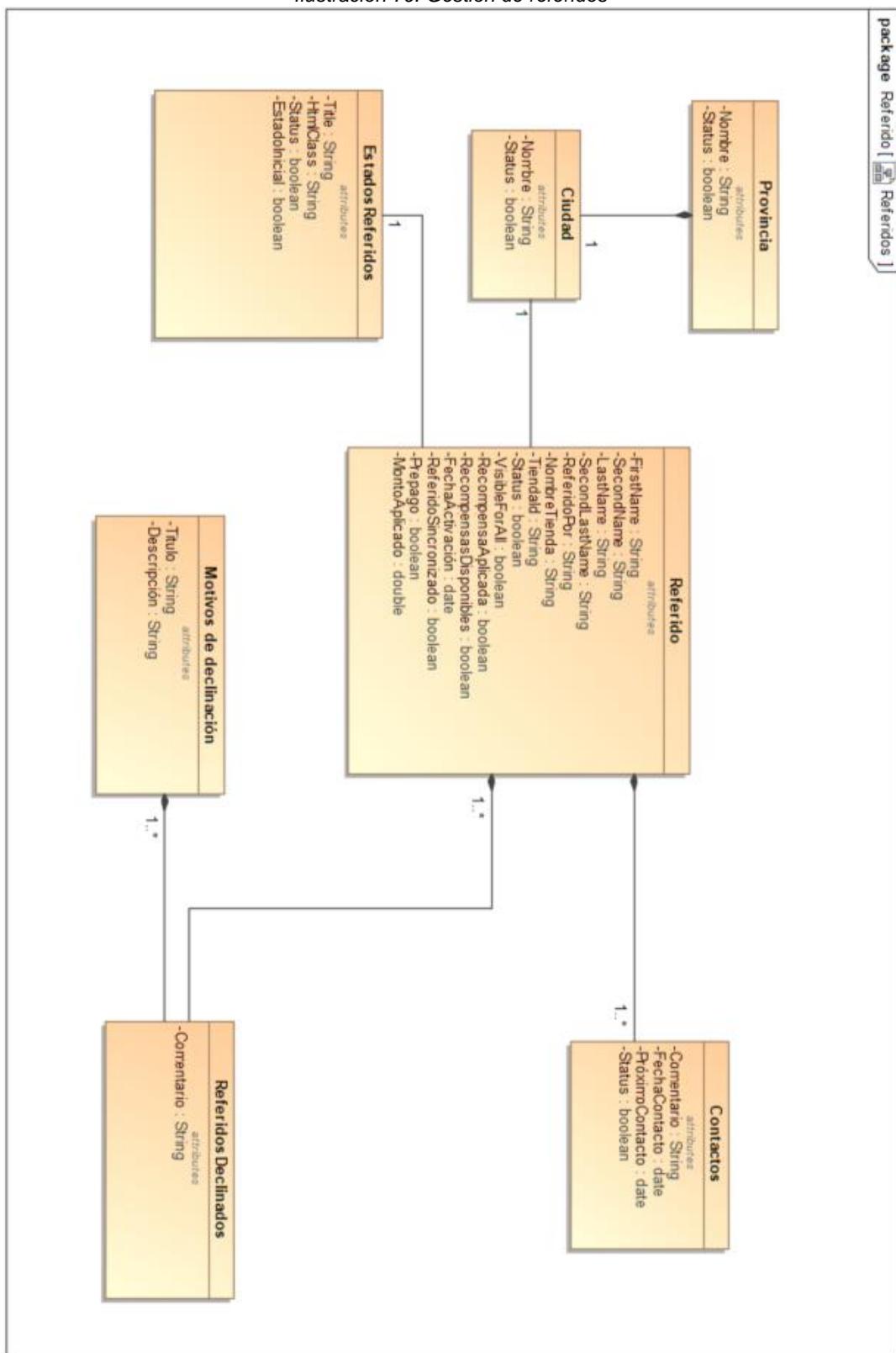
Ilustración 78: RN-10 Procesar recompensas



Fuente: Elaboración propia

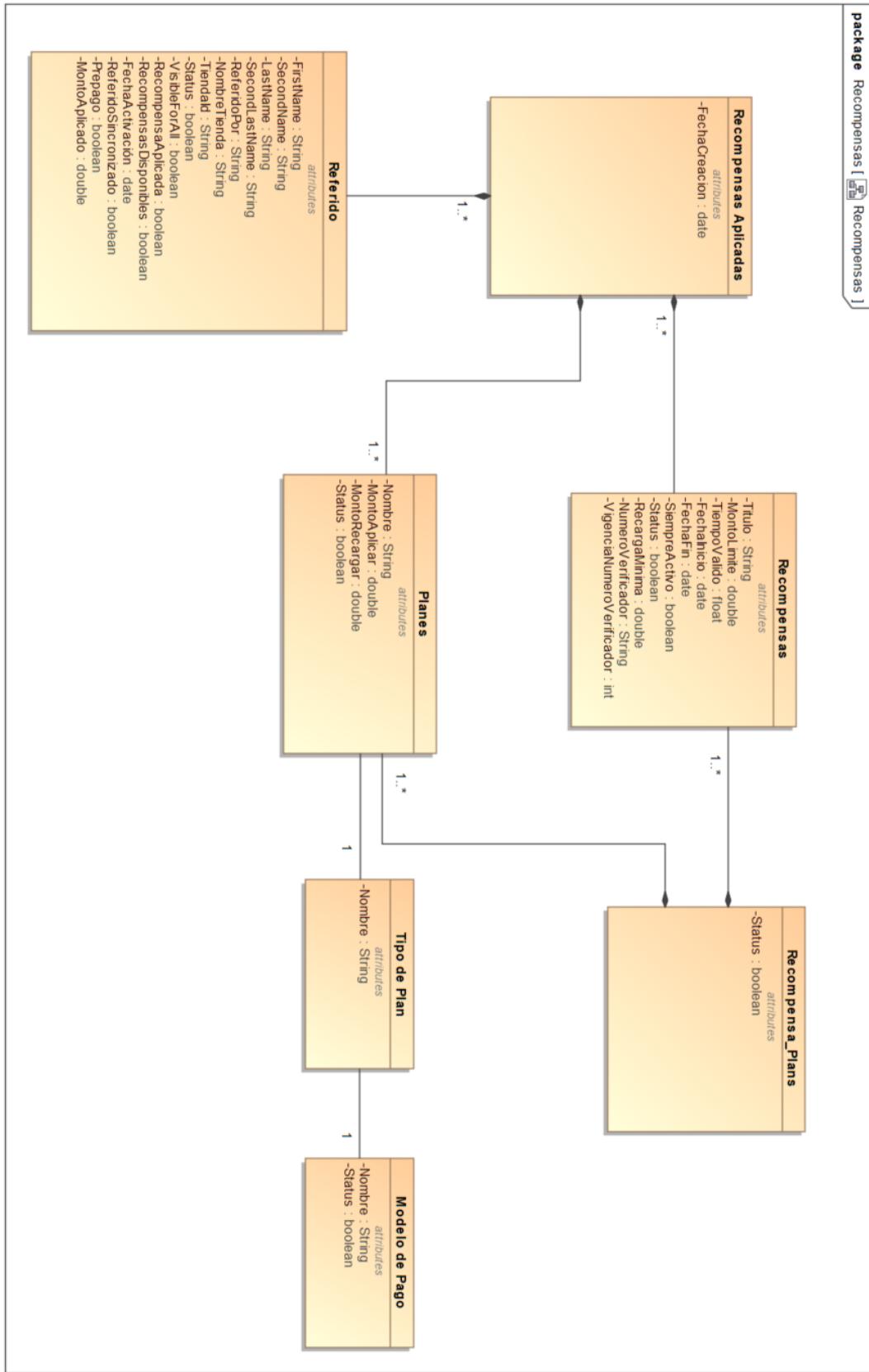
Anexo Ilustración 3: Modelo de contenido

Ilustración 79: Gestión de referidos



Fuente: Elaboración propia

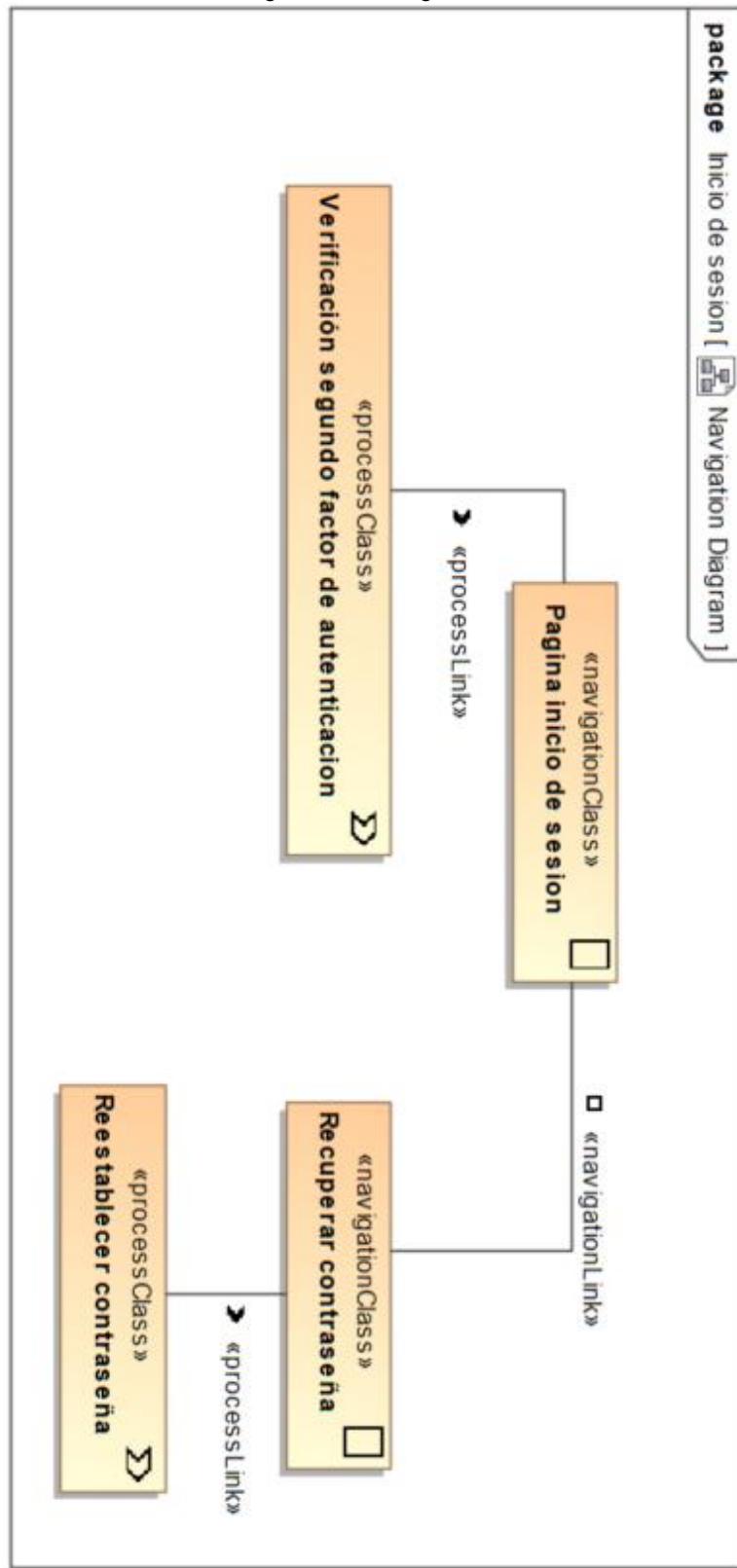
Ilustración 80: Gestión de recompensas



Fuente: Elaboración propia

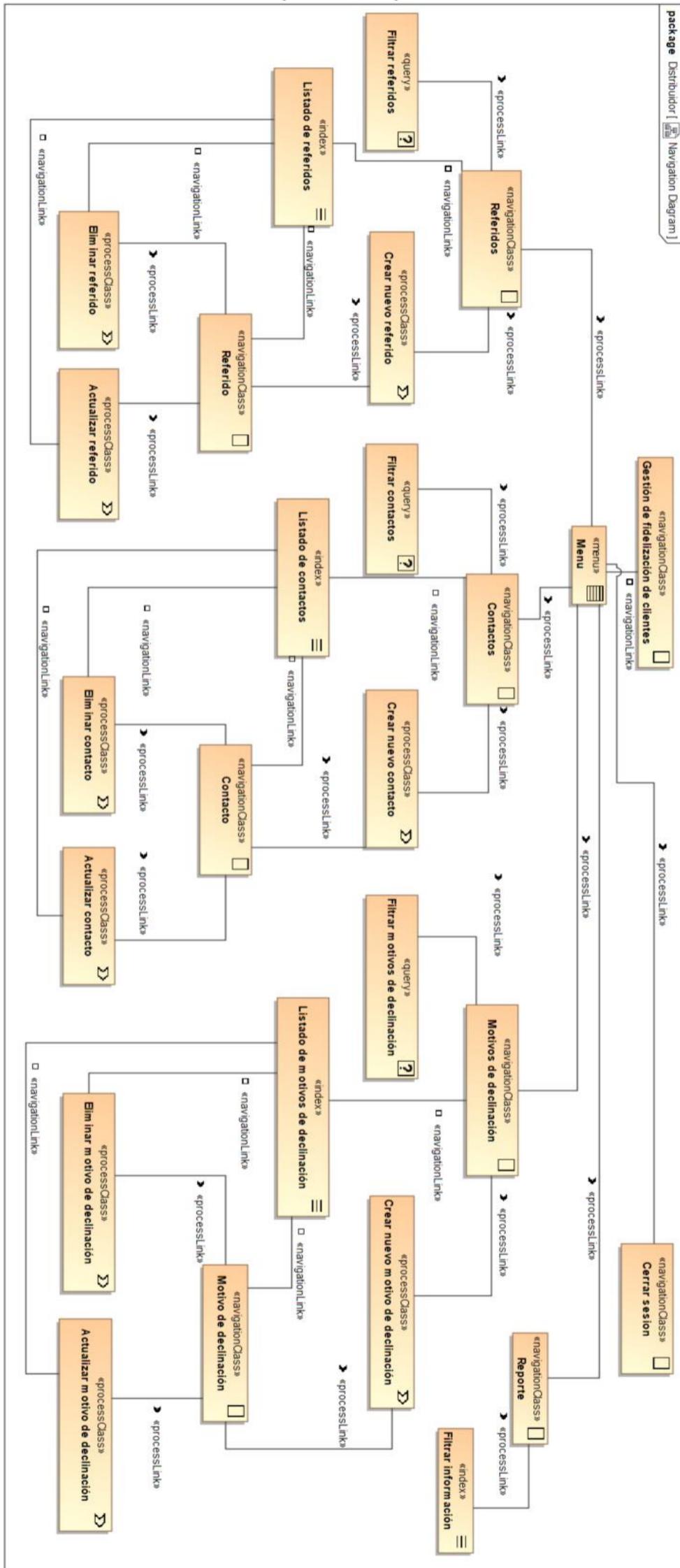
Anexo Ilustración 4: Modelo de navegación

Ilustración 81: Diagrama de navegación - Inicio de sesión



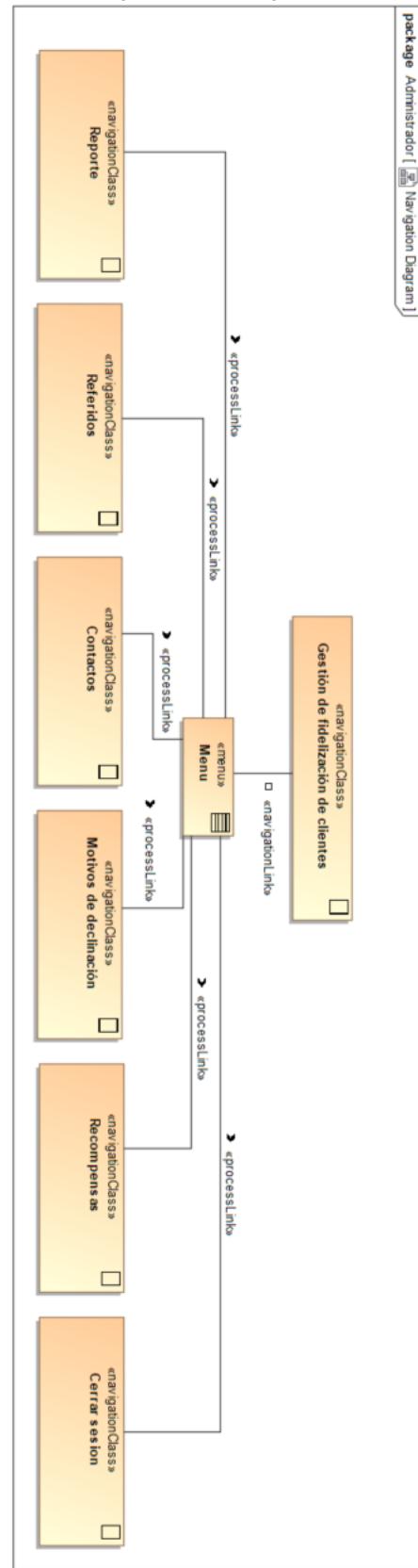
Fuente: Elaboración propia

Ilustración 82: Diagrama de navegación - Distribuidor



Fuente: Elaboración propia

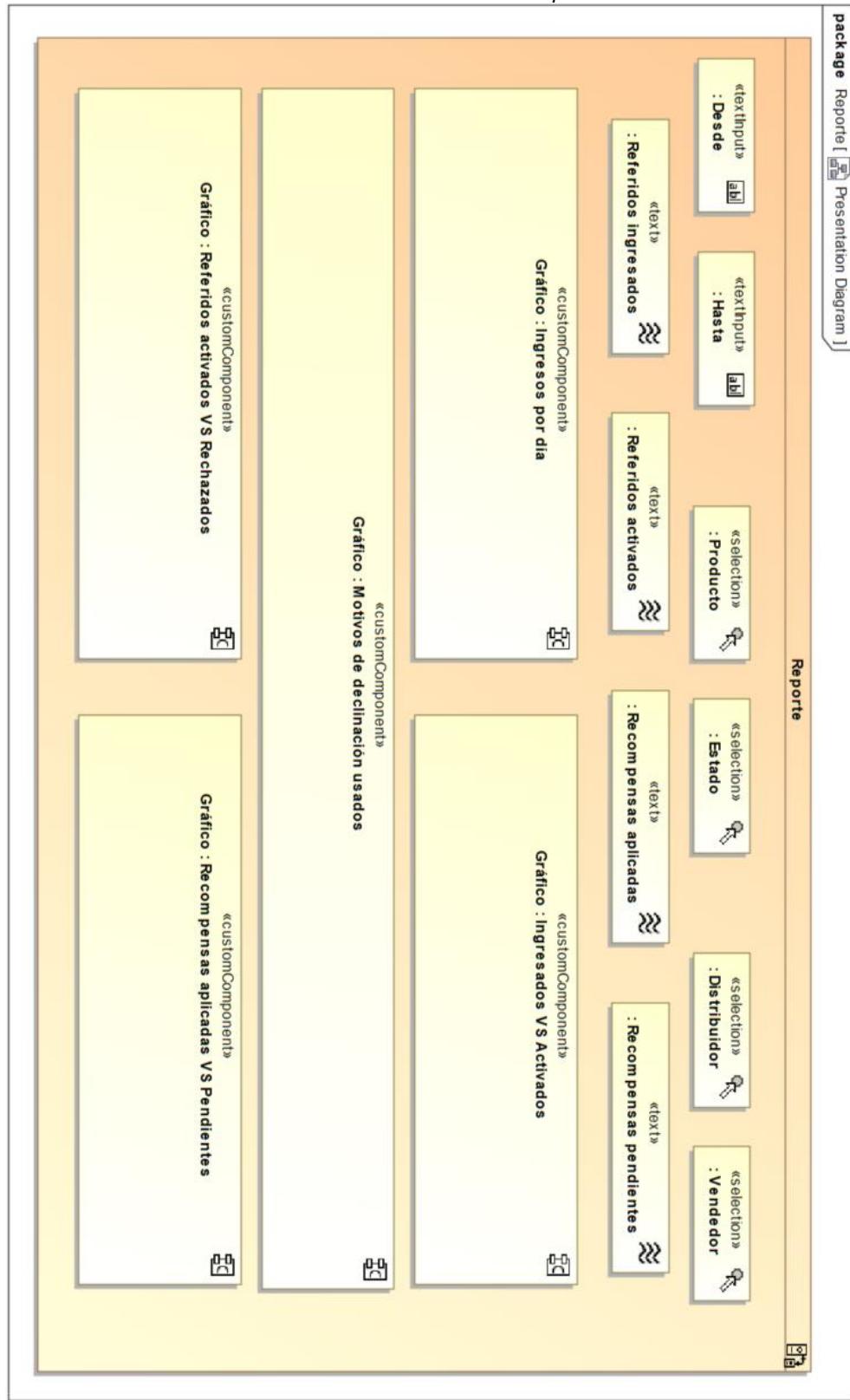
Ilustración 83: Diagrama de navegación - Administrador



Fuente: Elaboración propia

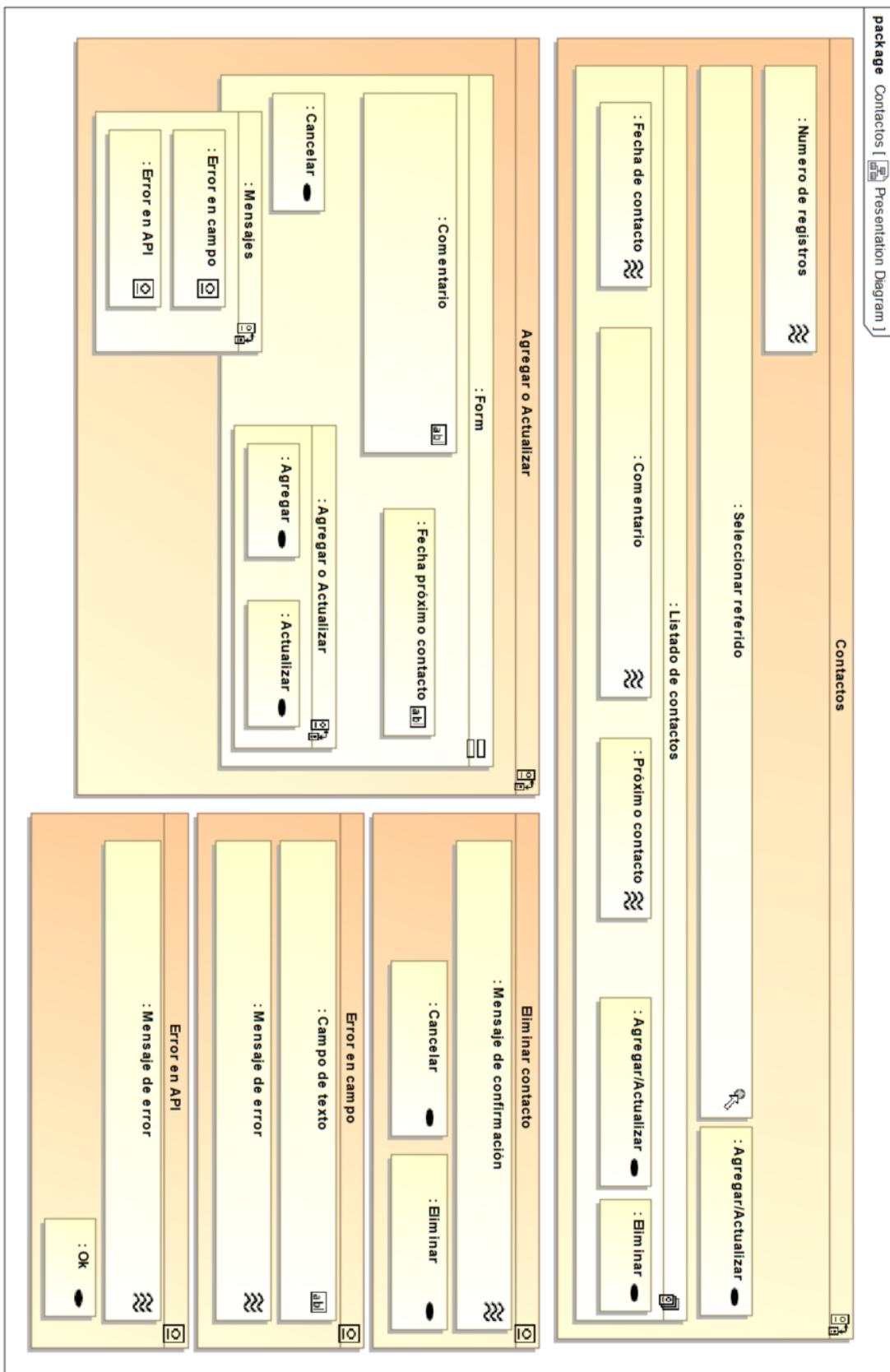
Anexollustración5: Modelo de presentación

Ilustración 84: Presentación - Reportes



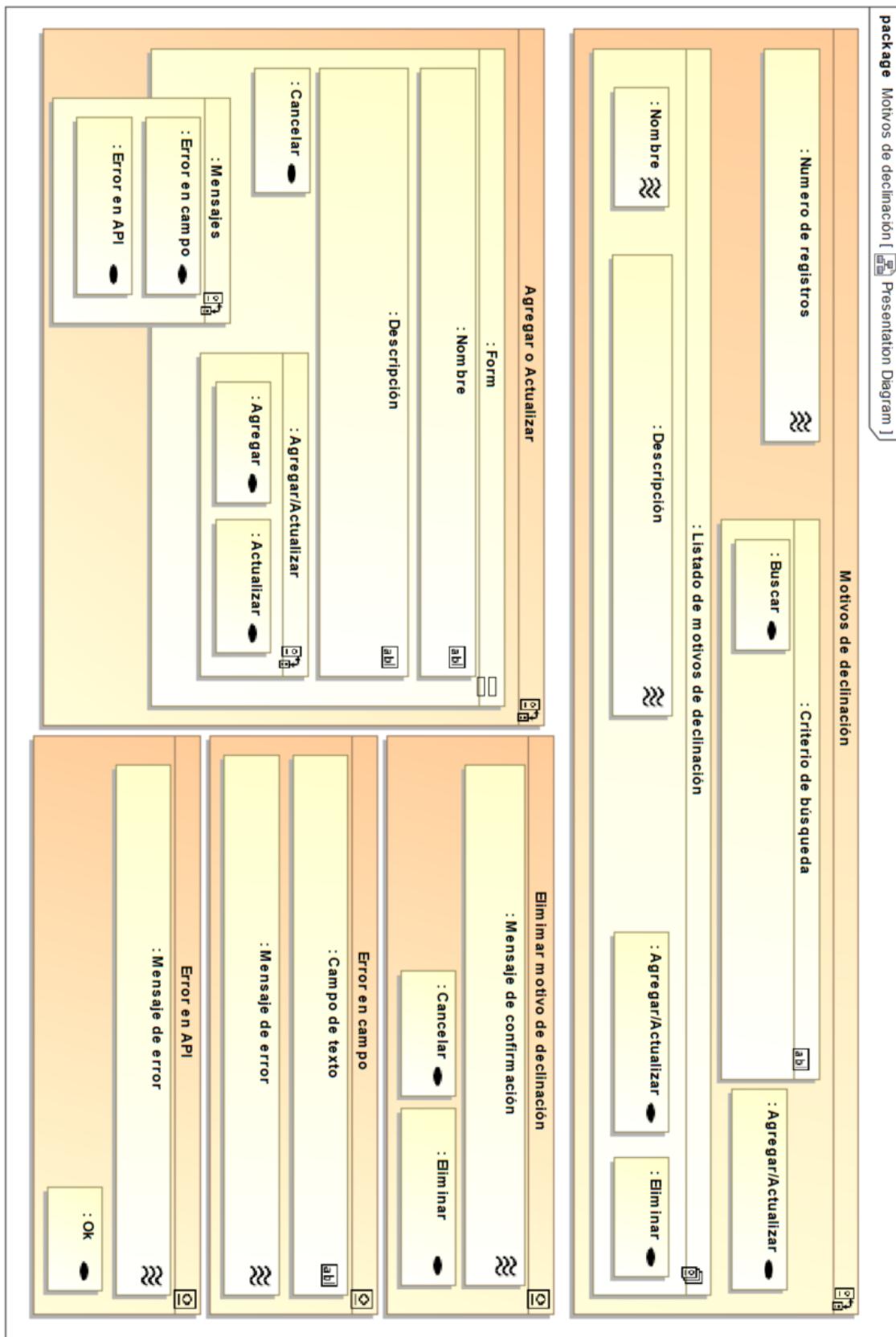
Fuente: Elaboración propia

Ilustración 85: Presentación - Contactos



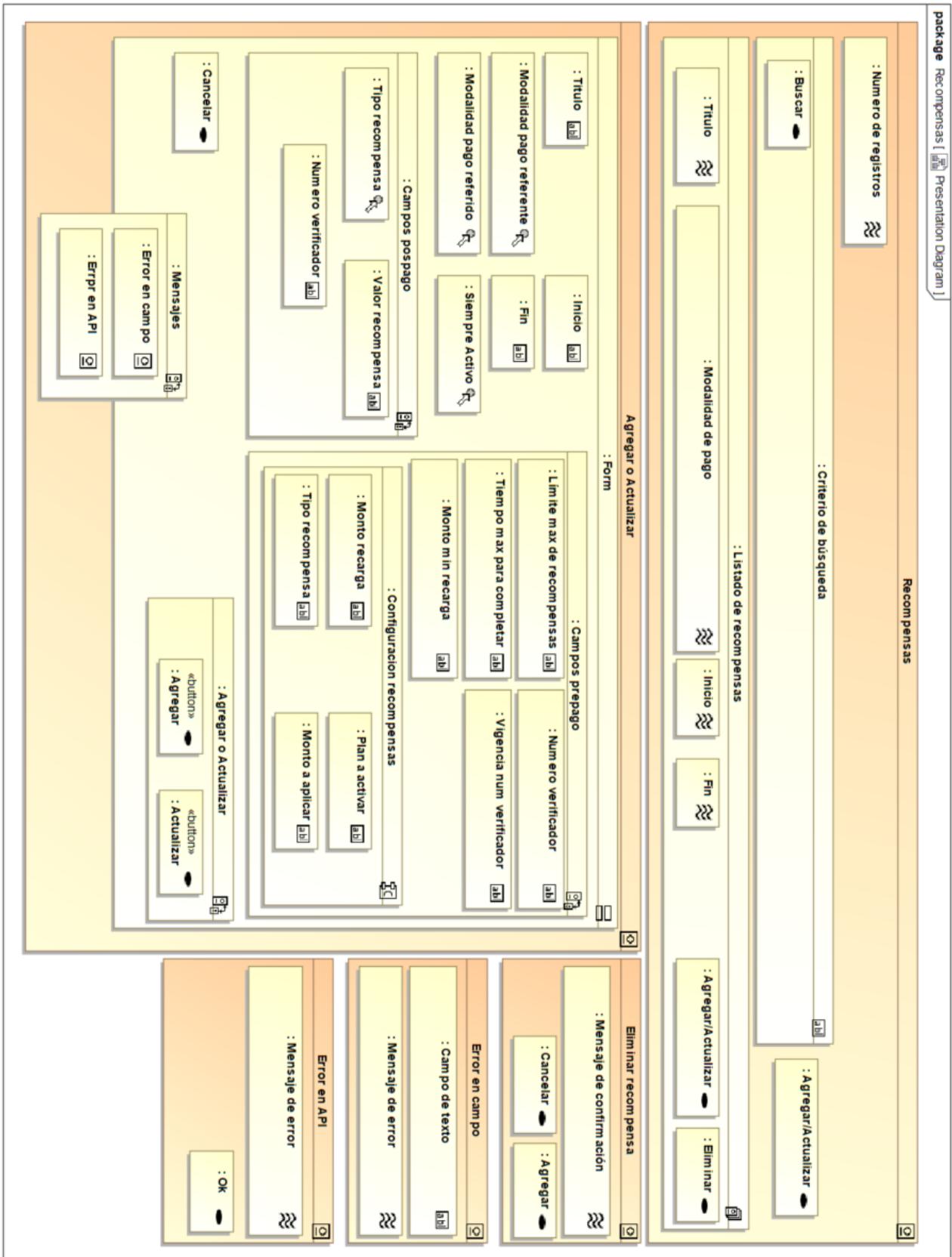
Fuente: Elaboración propia

Ilustración 86: Presentación - Motivos de declinación



Fuente: Elaboración propia

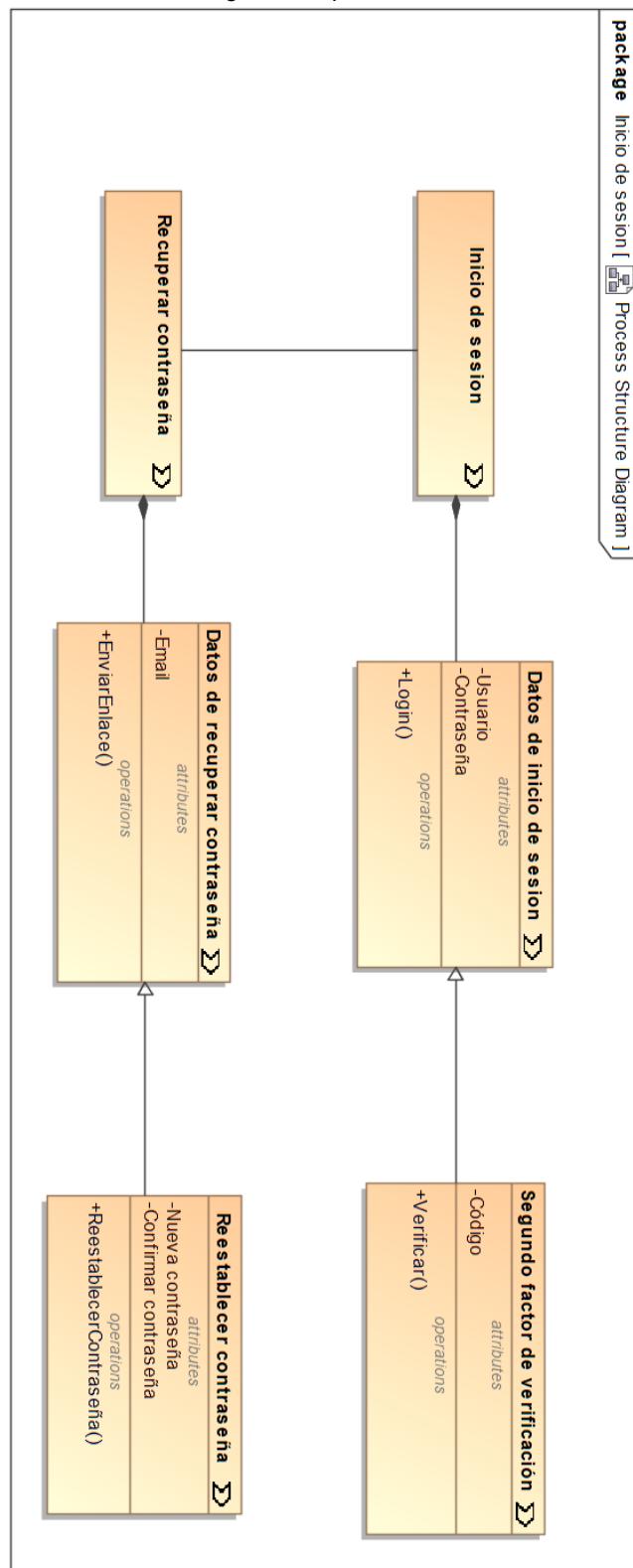
Ilustración 87: Presentación - Recompensas



Fuente: Elaboración propia

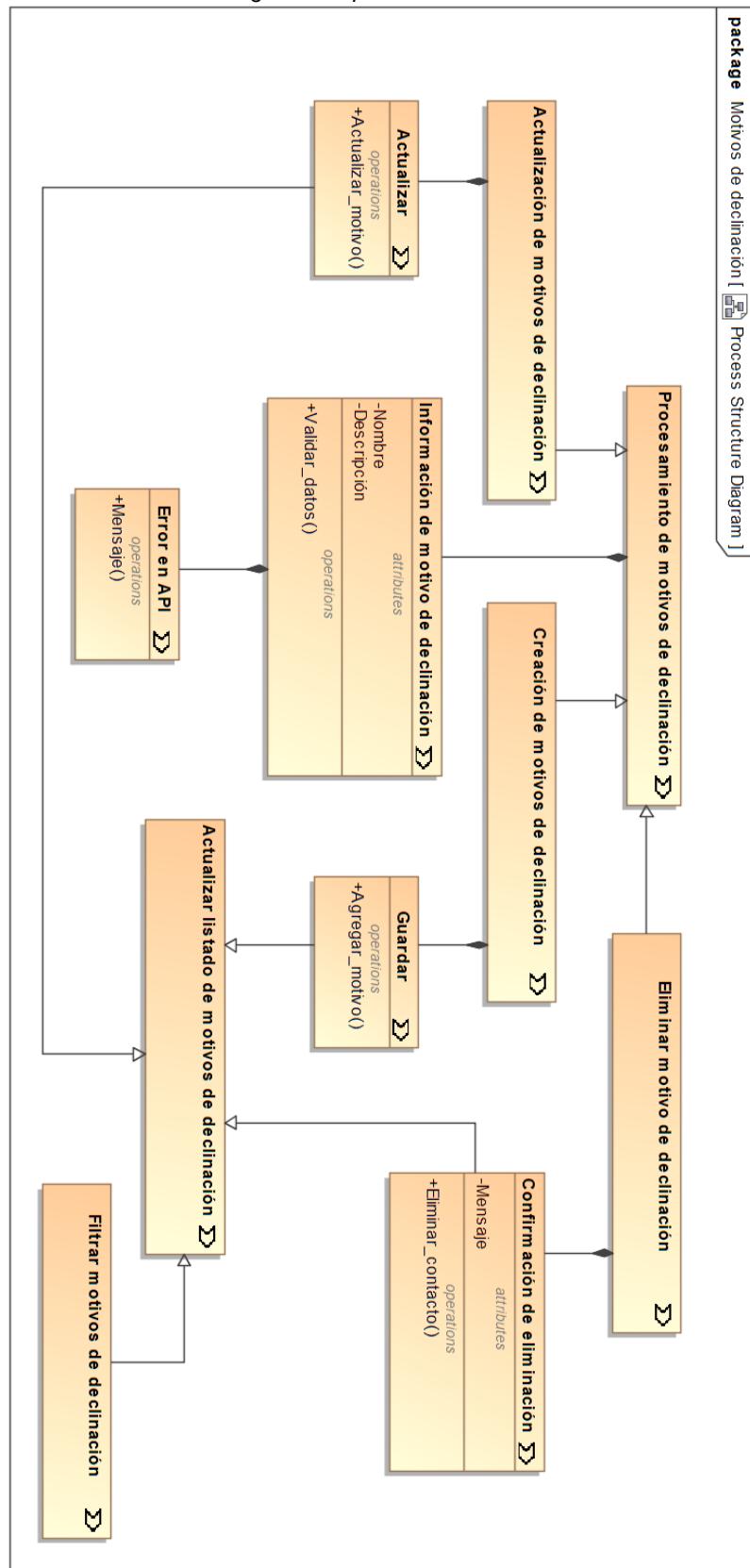
Anexo Ilustración 6: Modelo de procesos

Ilustración 88: Diagrama de procesos - Inicio de sesión



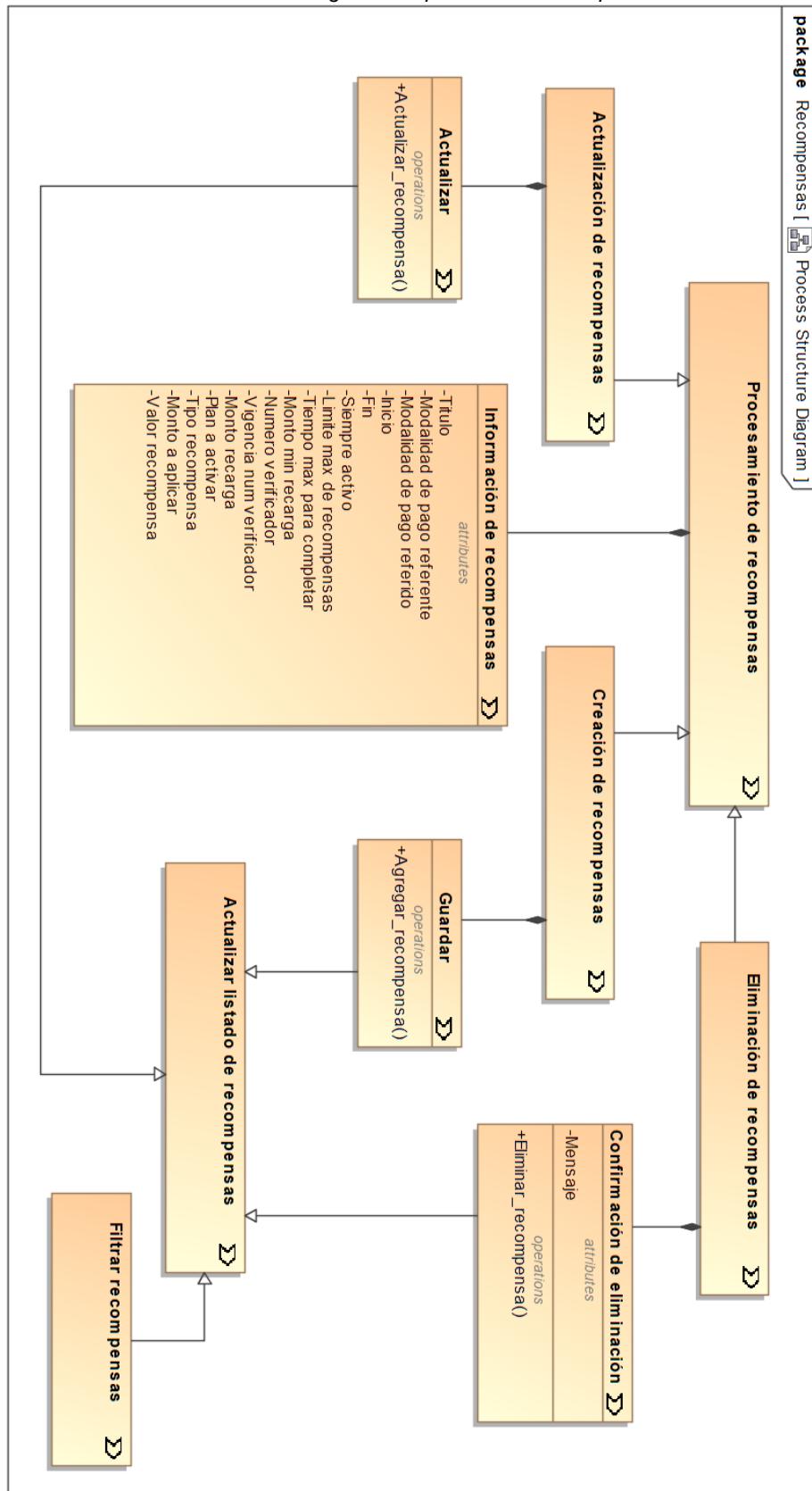
Fuente: Elaboración propia

Ilustración 89: Diagrama de procesos - Motivos de declinación



Fuente: Elaboración propia

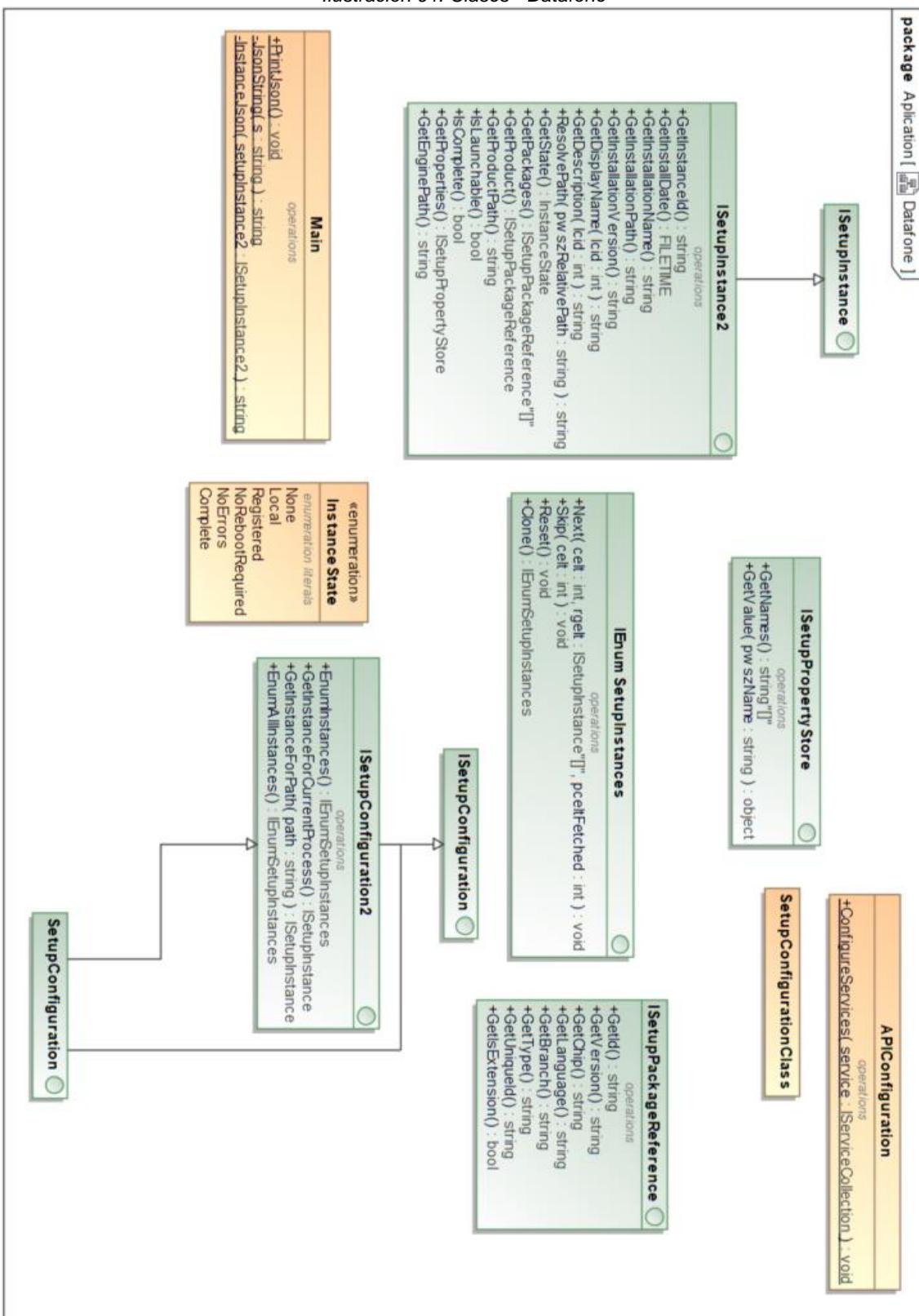
Ilustración 90: Diagrama de procesos - Recompensas



Fuente: Elaboración propia

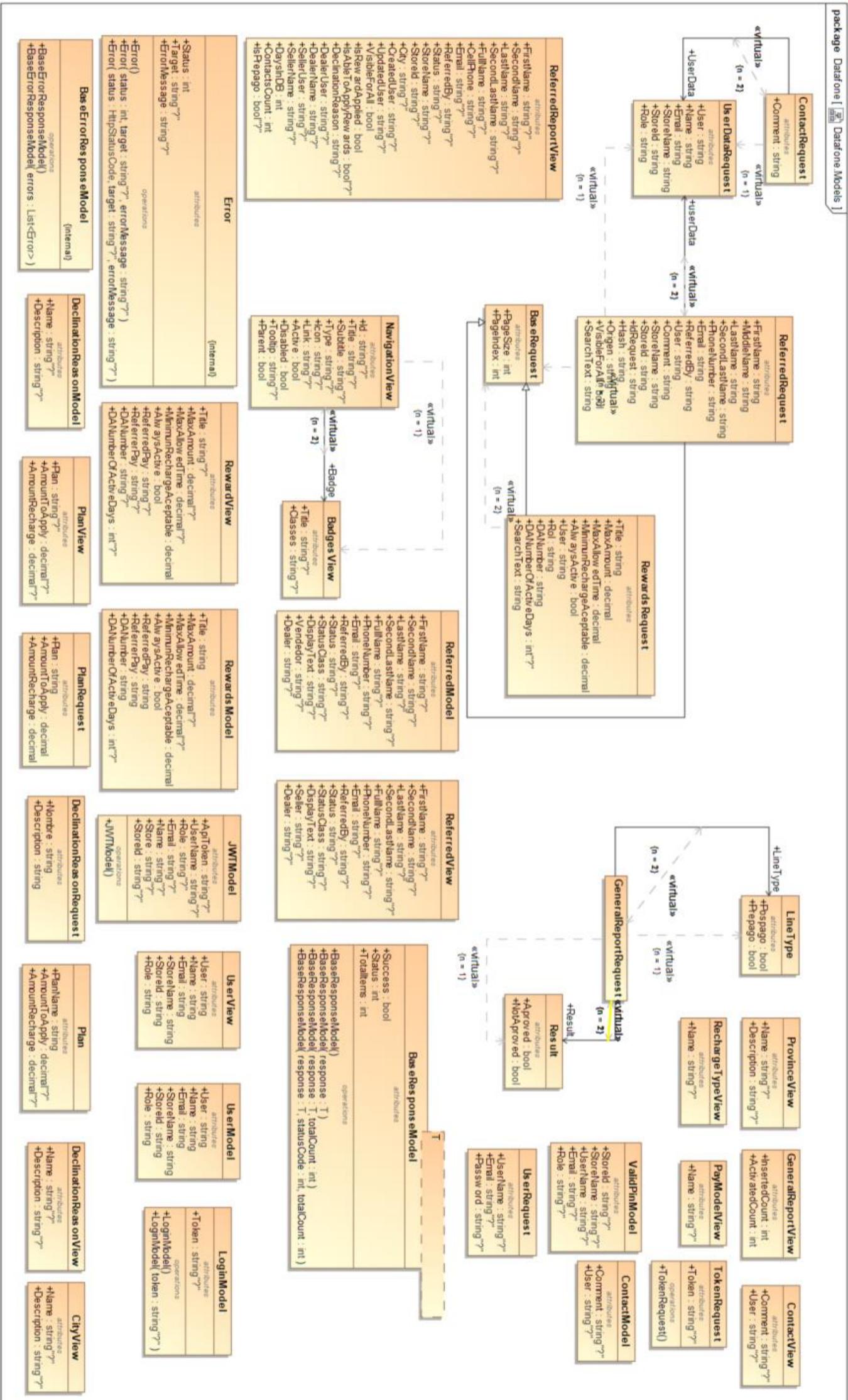
Anexo Ilustración 7: Especificación de las clases del sistema

Ilustración 91: Clases - Datafone



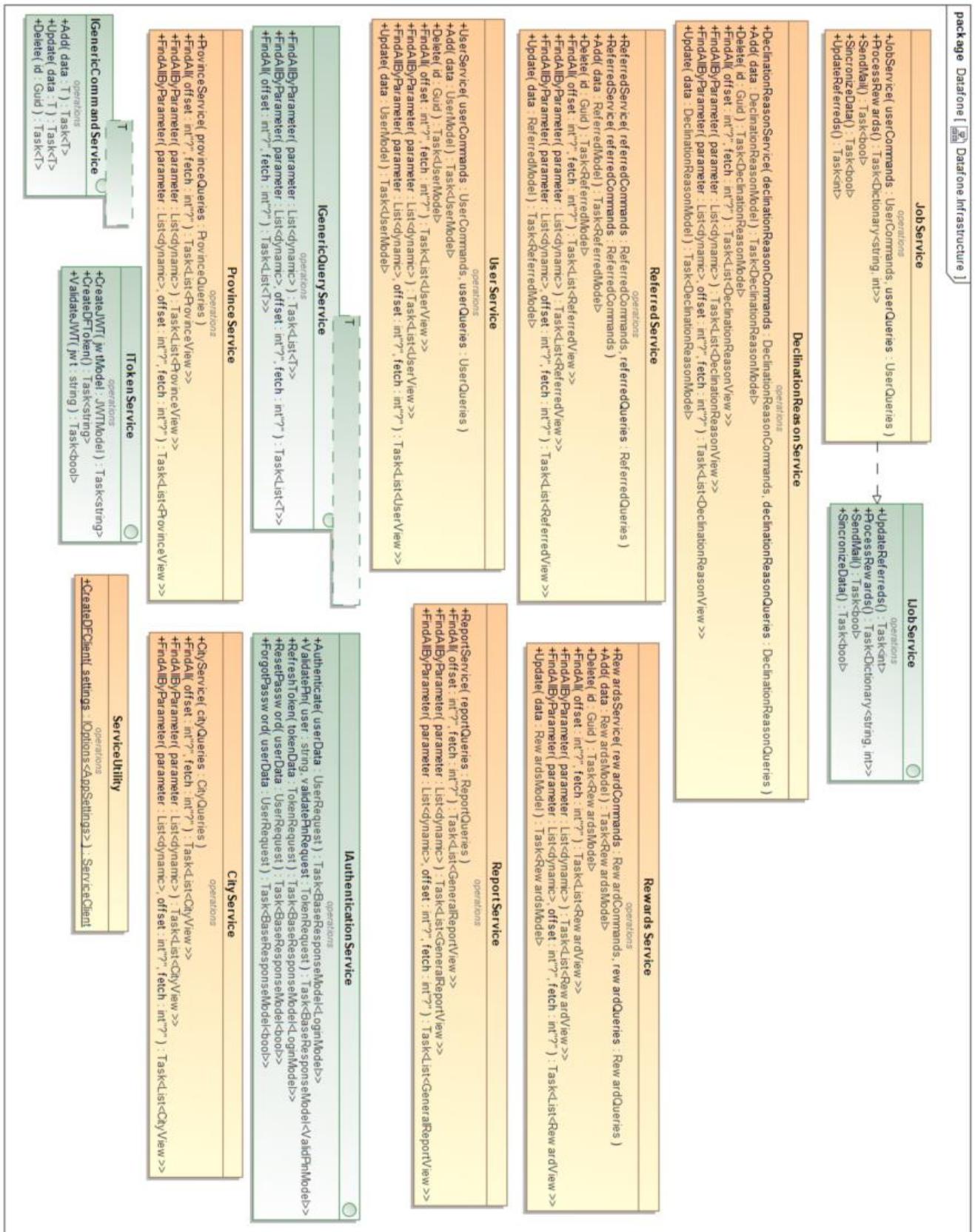
Fuente: Elaboración propia

Ilustración 92: Clases - Datafone.Models

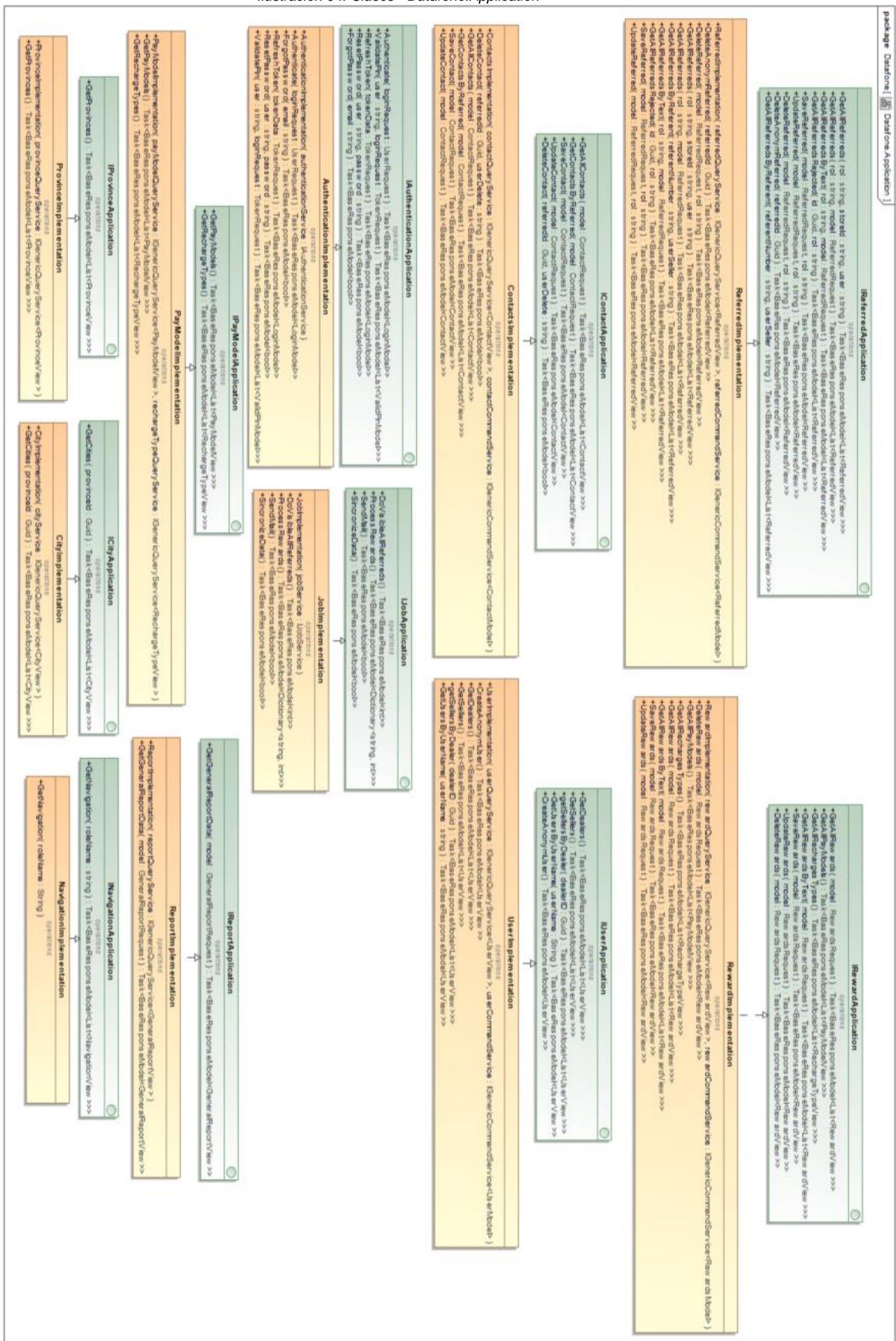


Fuente: Elaboración propia

Ilustración 93: Clases - Datafone.Infrastructure



Fuente: Elaboración propia



Fuente: Elaboración propia

Anexo 8: Diseño y uso de APIs

Ilustración 95: Swagger - Contactos

Contact	
POST	/api/Contact/all
POST	/api/Contact/byId
POST	/api/Contact/create
POST	/api/Contact
PUT	/api/Contact
DELETE	/api/Contact

Fuente: Elaboración propia

Ilustración 96: Swagger - Motivos de declinación

DeclinationReason	
GET	/api/DeclinationReason
POST	/api/DeclinationReason
PUT	/api/DeclinationReason
DELETE	/api/DeclinationReason
GET	/api/DeclinationReason/{title}/{description}
GET	/api/DeclinationReason/Details/{id}

Fuente: Elaboración propia

Ilustración 97: Swagger - Trabajos calendarizados

Jobs	
POST	/api/Jobs/updateReferrals
POST	/api/Jobs/processRewards
POST	/api/Jobs/sendMail
POST	/api/Jobs/sincronizeData

Fuente: Elaboración propia

Ilustración 98: Swagger - Navegación

Navigation	
GET	/api/Navigation

Fuente: Elaboración propia

Ilustración 99: Swagger - Tipos de modelo de pagos y recargas

PayModel	
GET	/api/PayModel
GET	/api/PayModel/rechargetypes

Fuente: Elaboración propia

Ilustración 100: Swagger - Provincias

The screenshot shows the Swagger UI for the 'Provinces' endpoint. It includes a header with a lock icon and a dropdown menu. Below the header, there is a table with one row containing a blue 'GET' button and the URL '/api/Provinces'. A note below the table states 'Fuente: Elaboración propia'.

GET	/api/Provinces
------------	----------------

Fuente: Elaboración propia

Ilustración 101: Swagger - Referidos

The screenshot shows the Swagger UI for the 'Referidos' endpoint. It includes a header with a lock icon and a dropdown menu. Below the header, there is a table with seven rows, each representing a different API operation: GET /api/Referred/all, POST /api/Referred/allWithPagination, POST /api/Referred/bysearchtext, POST /api/Referred/create, POST /api/Referred, PUT /api/Referred (highlighted in orange), and POST /api/Referred/delete. A note below the table states 'Fuente: Elaboración propia'.

GET	/api/Referred/all
POST	/api/Referred/allWithPagination
POST	/api/Referred/bysearchtext
POST	/api/Referred/create
POST	/api/Referred
PUT	/api/Referred
POST	/api/Referred/delete

Fuente: Elaboración propia

Ilustración 102: Swagger - Reportes

The screenshot shows the Swagger UI for the 'Report' endpoint. It includes a header with a lock icon and a dropdown menu. Below the header, there is a table with one row containing a green 'POST' button and the URL '/api/Report/general'. A note below the table states 'Fuente: Elaboración propia'.

POST	/api/Report/general
-------------	---------------------

Fuente: Elaboración propia

Ilustración 103: Swagger - Recompensas

The screenshot shows the Swagger UI for the 'Reward' endpoint. It includes a header with a lock icon and a dropdown menu. Below the header, there is a table with six rows, each representing a different API operation: POST /api/Reward/all, POST /api/Reward/bysearchtext, POST /api/Reward/create, POST /api/Reward, PUT /api/Reward (highlighted in orange), and POST /api/Reward/delete. A note below the table states 'Fuente: Elaboración propia'.

POST	/api/Reward/all
POST	/api/Reward/bysearchtext
POST	/api/Reward/create
POST	/api/Reward
PUT	/api/Reward
POST	/api/Reward/delete

Fuente: Elaboración propia

Ilustración 104: Swagger - Usuarios

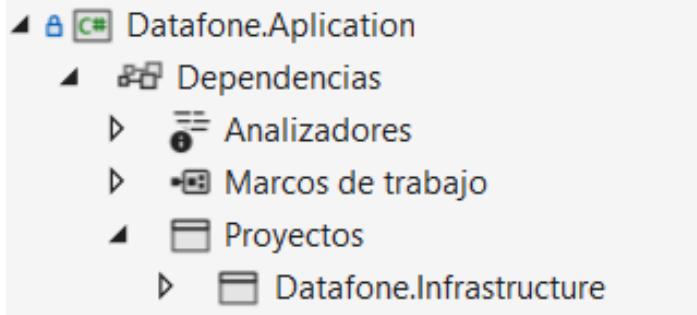
The screenshot shows the Swagger UI for the 'User' endpoint. It includes a header with a lock icon and a dropdown menu. Below the header, there is a table with four rows, each representing a different API operation: GET /api/User/info, GET /api/User/dealers, GET /api/User/sellers, and GET /api/User/sellersByDealerUser. A note below the table states 'Fuente: Elaboración propia'.

GET	/api/User/info
GET	/api/User/dealers
GET	/api/User/sellers
GET	/api/User/sellersByDealerUser

Fuente: Elaboración propia

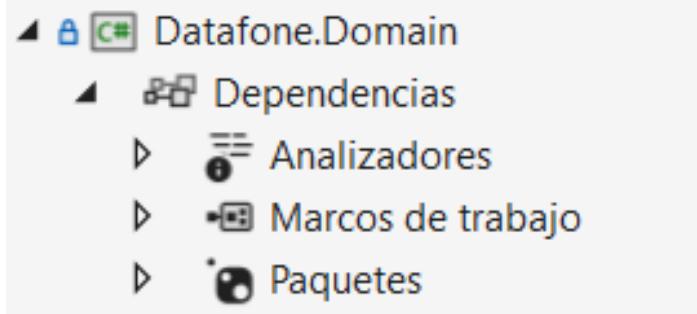
Anexo Ilustración 9: Implementación patrones de diseño – Clean Architecture

Ilustración 105: Dependencia de proyectos - Datafone.Application



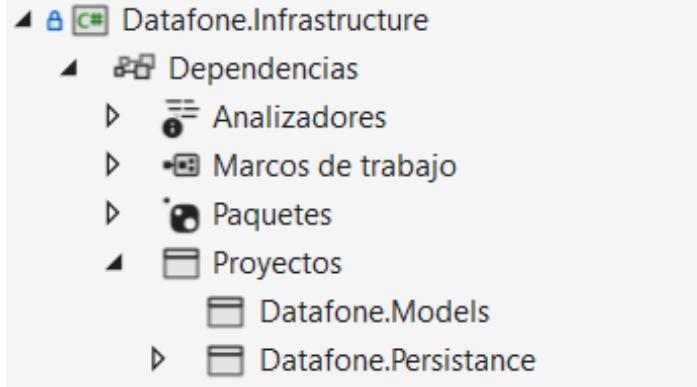
Fuente: Elaboración propia

Ilustración 106: Dependencia de proyectos - Datafone.Domain



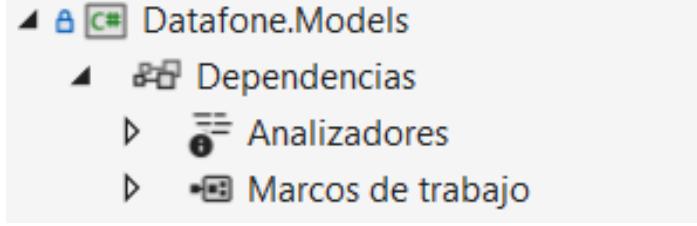
Fuente: Elaboración propia

Ilustración 107: Dependencia de proyectos - Datafone.Infrastructure



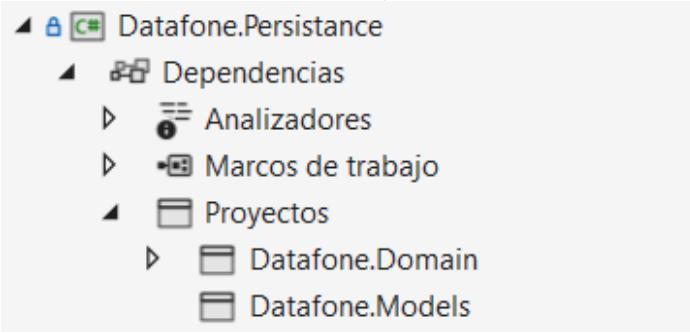
Fuente: Elaboración propia

Ilustración 108: Dependencia de proyectos - Datafone.Models



Fuente: Elaboración propia

Ilustración 109: Dependencia de proyectos - Datafone.Persistence



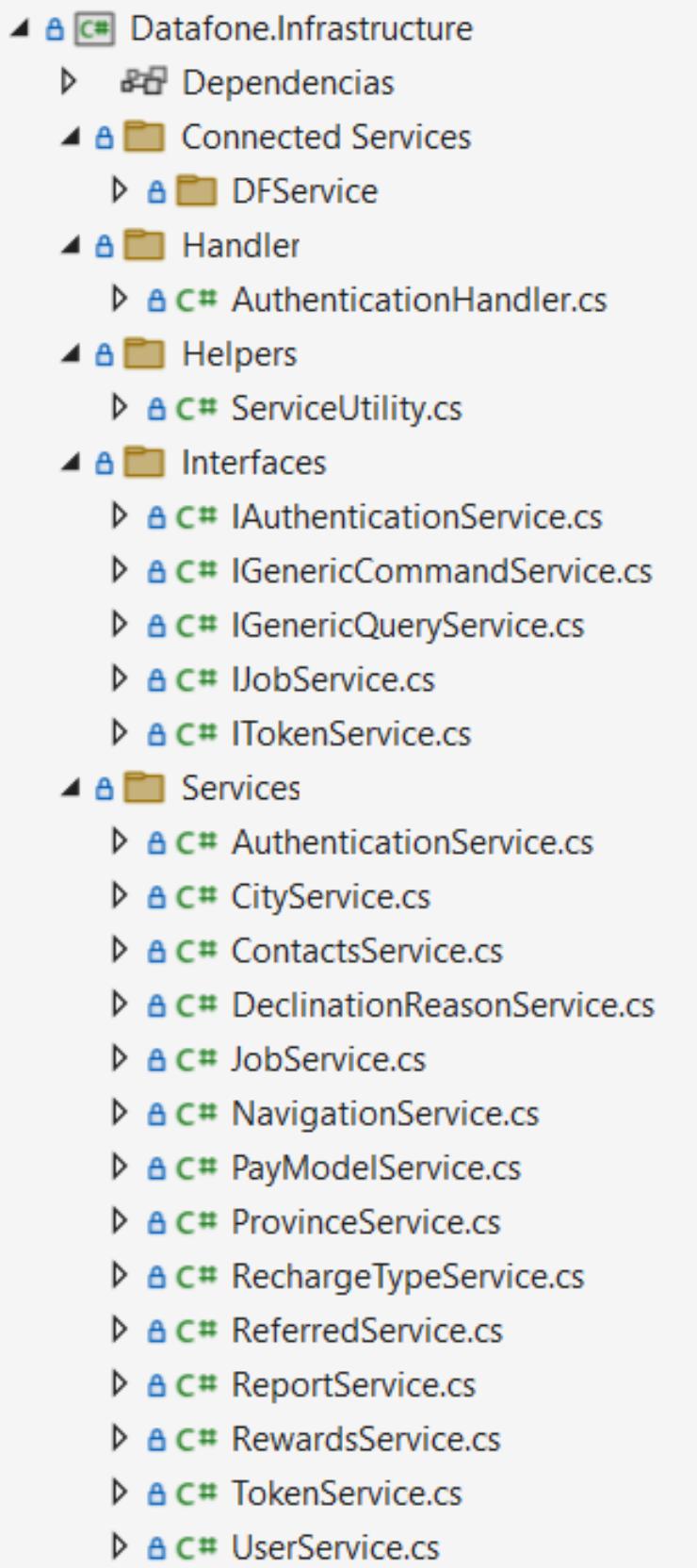
Fuente: Elaboración propia

Anexo Ilustración 10: Organización de los archivos del proyecto – Back-End

Ilustración 110: Estructura de carpetas y archivos - Datafone.Domain

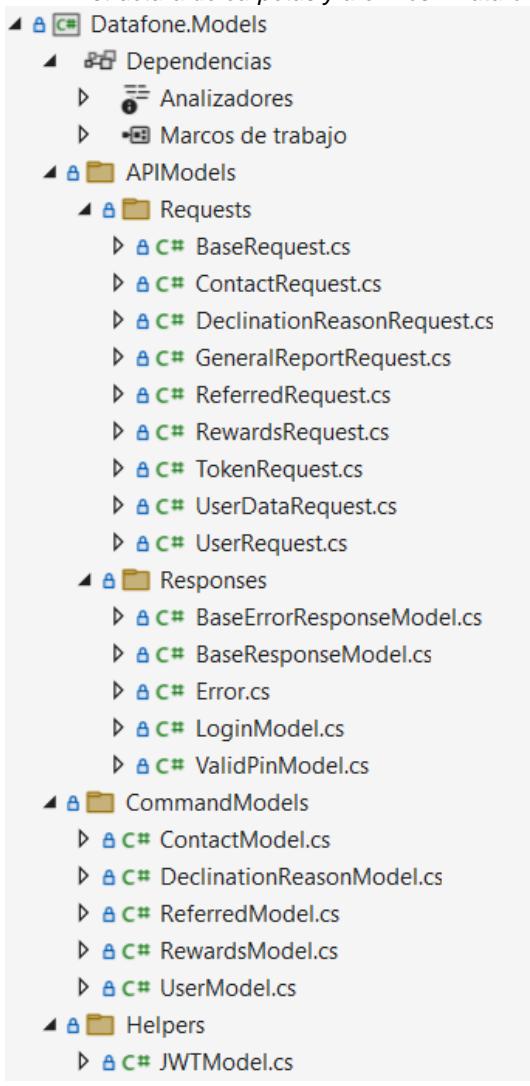


Fuente: Elaboración propia
Ilustración 111: Estructura de carpetas y archivos - Datafone.Infrastructure



Fuente: Elaboración propia

Ilustración 112: Estructura de carpetas y archivos - Datafone.Models I



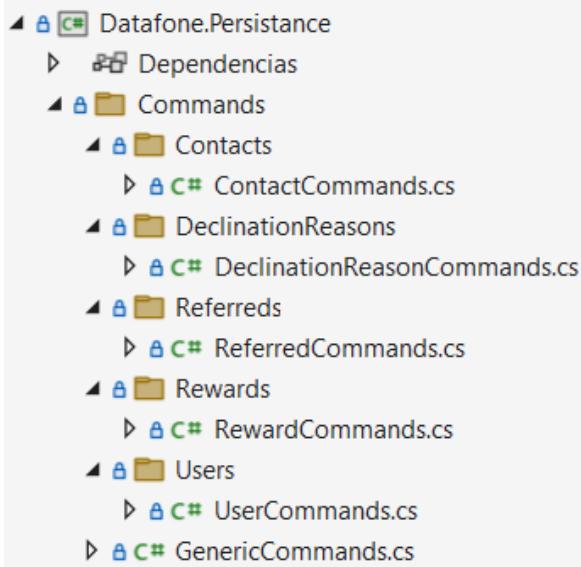
Fuente: Elaboración propia

Ilustración 113: Estructura de carpetas y archivos - Datafone.Models II



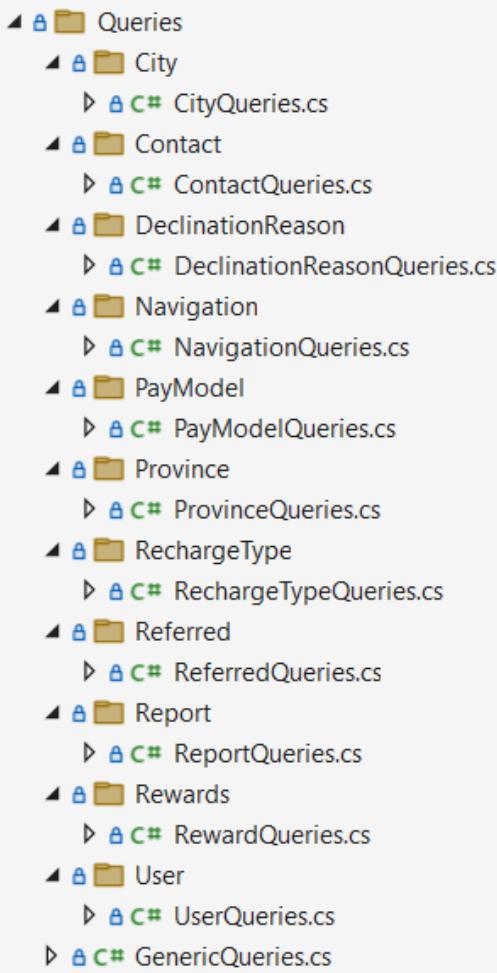
Fuente: Elaboración propia

Ilustración 114: Estructura de carpetas y archivos - Datafone.Persistence I



Fuente: Elaboración propia

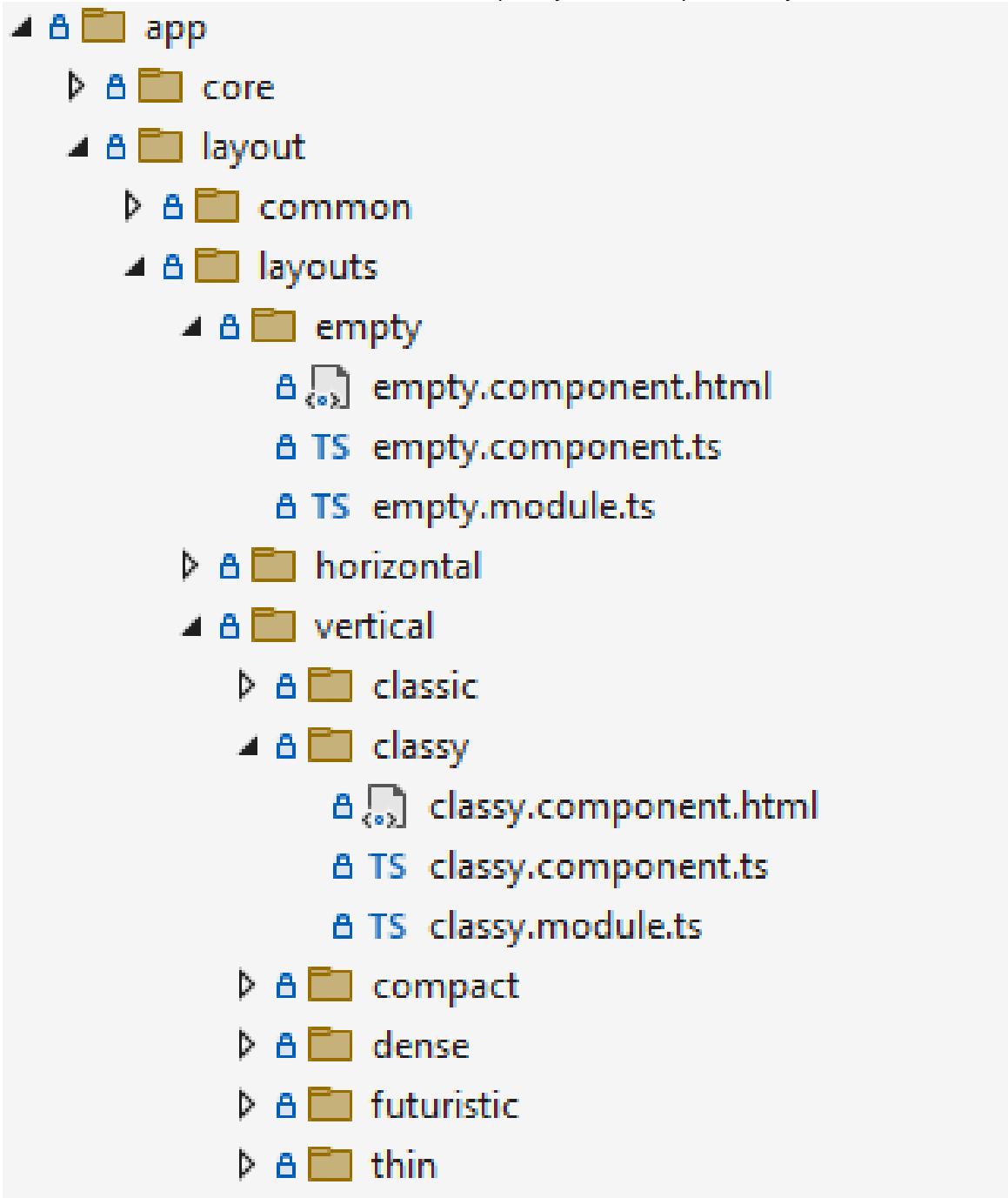
Ilustración 115: Estructura de carpetas y archivos - Datafone.Persistence II



Fuente: Elaboración propia

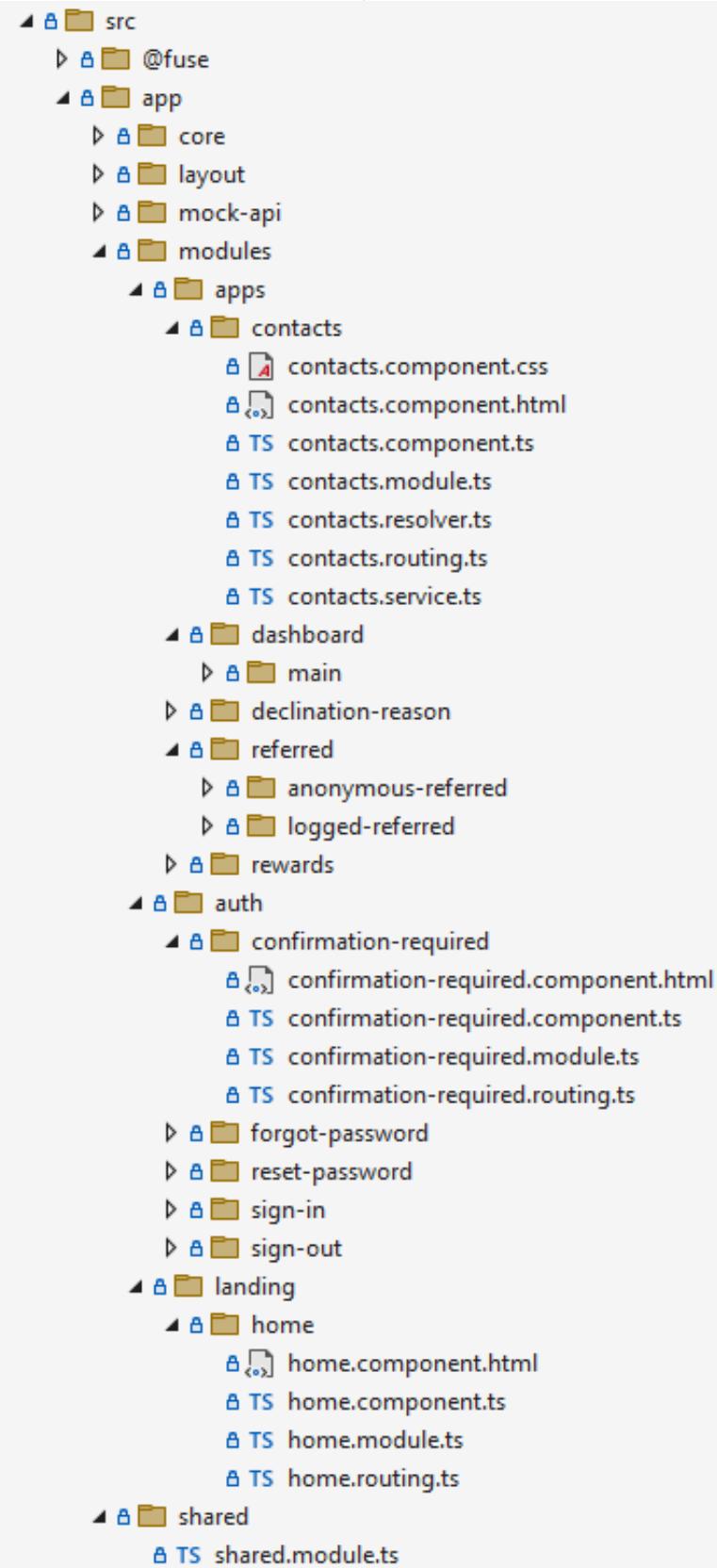
Anexo Ilustración 11: Organización de los archivos del proyecto – Front-End

Ilustración 116: Estructura de carpetas y archivos - Aplicacion.layout



Fuente: Elaboración propia

Ilustración 117: Estructura de carpetas y archivos – Aplicación.Modules&Shared



Fuente: Elaboración propia

Anexos Ilustración 12: Diseños de las interfaces del sistema

Interfaces: Menú principal

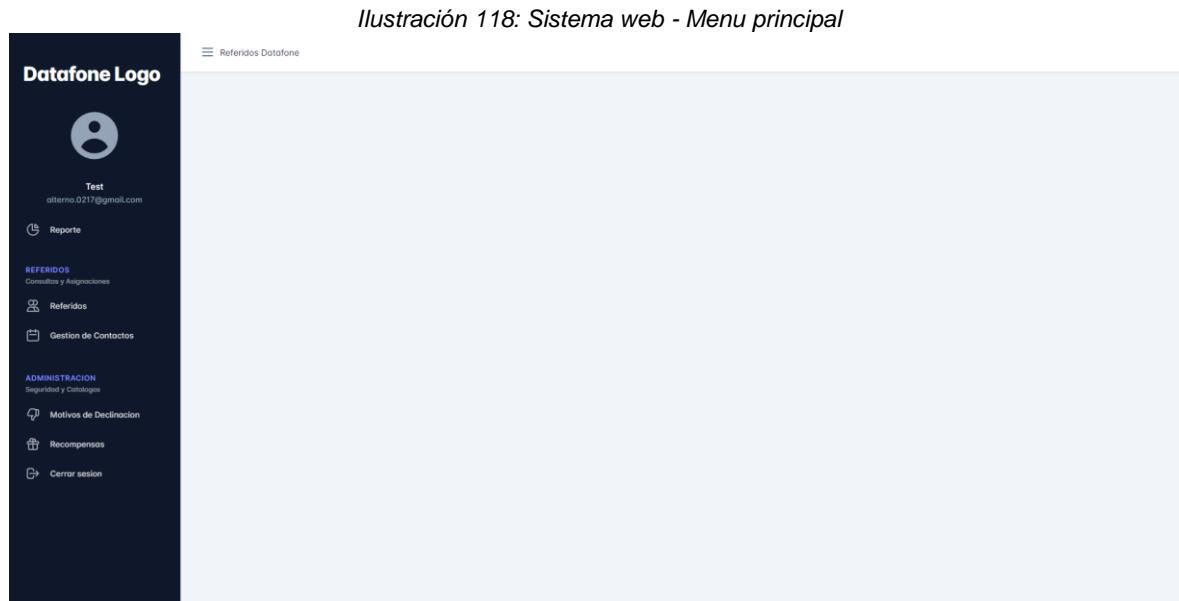
Propósito: Presentar las opciones disponibles para un determinado usuario

Acciones o eventos: Mostrar el subcomponente.

Elementos gráficos que la componen

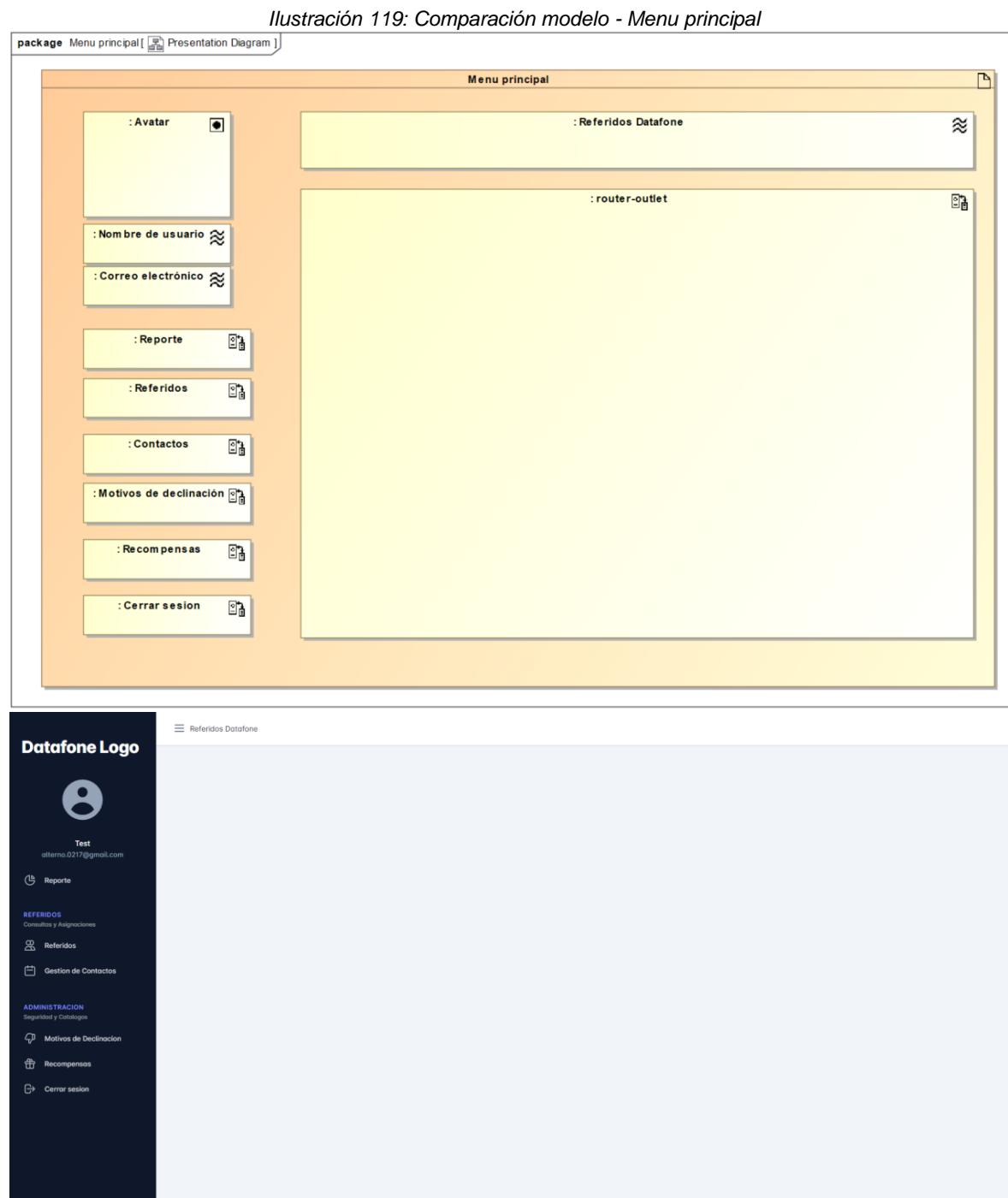
- Enlaces: Son hipervínculos que redireccionan a otro componente del sitio web
- Router-outlet: Componente de angular que permite renderizar otro componente

Capturas de pantalla (Ver ilustración #116):



Fuente: Elaboración propia

Comparación con modelo de presentación (Ver ilustración #117):



Fuente: Elaboración propia

Interfaces: Referidos

Propósito: Presentar la información correspondiente a los referidos.

Acciones o eventos: Mostrar, filtrar, agregar, actualizar y eliminar referidos, así como también es posible acceder a los contactos de un referido.

Elementos gráficos que la componen

- Enlaces: Son hipervínculos que redireccionan a otro componente del sitio web
- Cajas de texto: Permite mostrar o editar texto sin formato
- Lista de selección: Permite seleccionar un elemento de una lista de opciones
- Botones: Permite accionar los eventos en un sitio web
- Formularios emergentes: Permite mostrar pequeños formularios o solicitar confirmación.
- Tabla: Permite mostrar la información

Capturas de pantalla (Ver ilustraciones #118 - #121):

Ilustración 120: Sistema web - Referidos

The screenshot shows a web application for managing referrals. The header includes the Datafone logo and a user profile for 'Test' (correo: altermo.0217@gmail.com). A sidebar on the left contains links for 'Reporte', 'REFERIDOS' (selected), 'Contactos', and 'ADMINISTRACION'. The main content area is titled 'Referidos' and displays a table with two rows of data:

Nombre	Teléfono	Dealer	Vendedor	Correo	Referido por	Proximo contacto
Testing numero 4	83835358	Tienda 1	Vendedor 1	test@gmail.com	89251197	28/12/2021
Gloria Mena	6589123	Tienda 2	Vendedor 2	mayorga@mayorga.com	7856932	Ingresado

At the bottom, there are navigation buttons for 'Elementos por página' (10), page numbers (1-2 de 2), and arrows for navigating through the results.

Fuente: Elaboración propia

Ilustración 121: Sistema web - Agregar referidos

Nombre	Telefono	Dealer	Vendedor	Correo	Referido por	Proximo contacto
Testing numero 4	83835358	Tienda 1	Vendedor 1	test@gmail.com	89251197	28/12/2021
Gloria Mena	6589123	Tienda 2	Vendedor 2	moyorga@moyorga.com	7856932	Ingresado

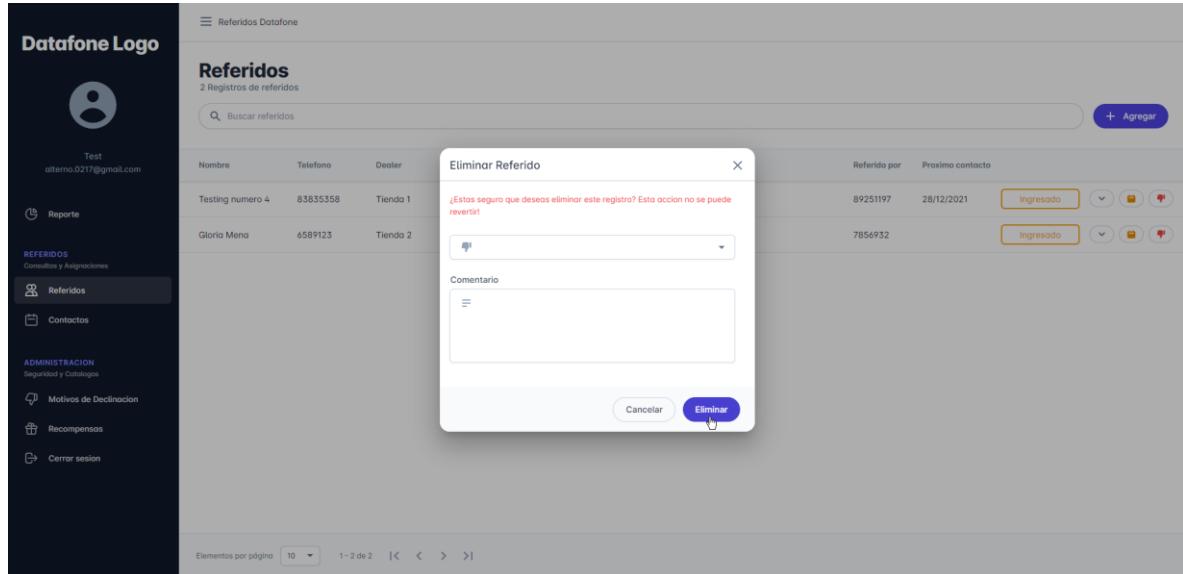
Fuente: Elaboración propia

Ilustración 122: Sistema web - Actualizar referido

Nombre	Telefono	Dealer	Vendedor	Correo	Referido por	Proximo contacto
Testing numero 4	83835358	Tienda 1	Vendedor 1	test@gmail.com	89251197	28/12/2021
Gloria Mena	6589123	Tienda 2	Vendedor 2	moyorga@moyorga.com	7856932	Ingresado

Fuente: Elaboración propia

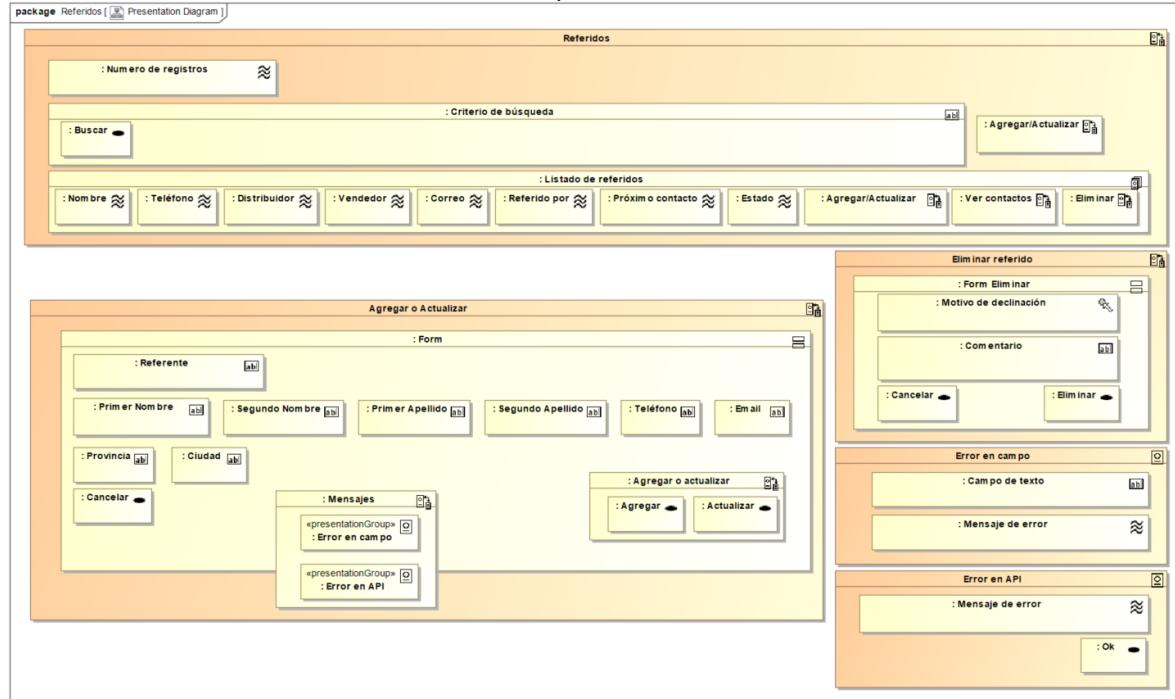
Ilustración 123: Sistema web - Eliminar referido



Fuente: Elaboración propia

Comparación con modelo de presentación (Ver ilustraciones #122 - #127):

Ilustración 124: Comparación modelo - Referidos



Fuente: Elaboración propia

Ilustración 125: Comparación modelo - Mostrar referidos

Referidos
2 Registros de referidos

Nombre	Teléfono	Dealer	Vendedor	Correo	Referido por	Proximo contacto	
Testing numero 4	83835358	Tienda 1	Vendedor 1	test@gmail.com	89251197	28/12/2021	Ingresado
Gloria Mena	6589123	Tienda 2	Vendedor 2	mayorga@mayorga.com	7856932		Ingresado

Fuente: Elaboración propia

Ilustración 126: Comparación modelo - Agregar referidos

Referidos
3 Registros de referidos

Referente

Referente es requerido

Primer Nombre	Segundo Nombre	Primer Apellido	Segundo Apellido	Teléfono	Email
<input type="text"/>					

Primer Nombre es requerido
Primer Apellido es requerido
Teléfono es requerido
Email es requerido

Prvincia

Selecione una provincia

Ciudad

Selecione una ciudad

Provincia es requerido
Ciudad es requerido

Cancelar Guardar

Fuente: Elaboración propia

Ilustración 127: Comparación modelo - Actualizar referido

Referidos
2 Registros de referidos

Referente

Referente es requerido

Primer Nombre	Segundo Nombre	Primer Apellido	Segundo Apellido	Teléfono	Email
<input type="text"/> Testing numero 4	<input type="text"/>	<input type="text"/> Testing numero 4	<input type="text"/>	<input type="text"/> 83835358	<input type="text"/> test@gmail.com

Prvincia

Selecione una provincia

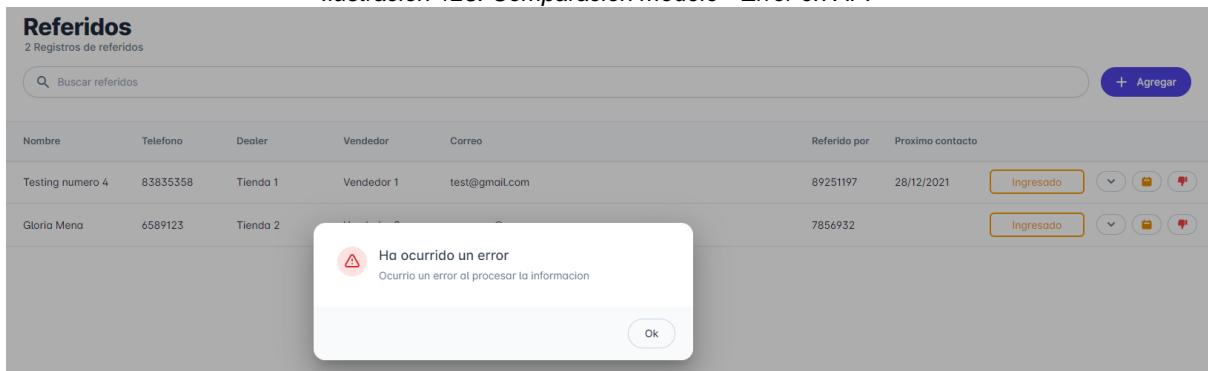
Ciudad

Selecione una ciudad

Cancelar Actualizar

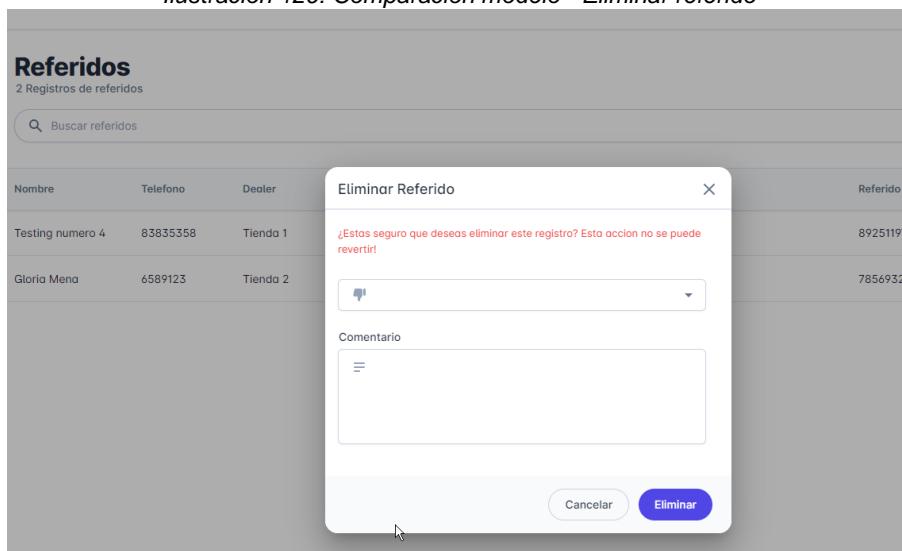
Fuente: Elaboración propia

Ilustración 128: Comparación modelo - Error en API



Fuente: Elaboración propia

Ilustración 129: Comparación modelo - Eliminar referido



Fuente: Elaboración propia

Interfaces: Contactos

Propósito: Presentar la información correspondiente a los contactos realizados a los referidos.

Acciones o eventos: Mostrar, filtrar, agregar, actualizar y eliminar contactos.

Elementos gráficos que la componen

- Cajas de texto: Permite mostrar o editar texto sin formato
- Lista de selección: Permite seleccionar un elemento de una lista de opciones
- Botones: Permite accionar los eventos en un sitio web
- Formularios emergentes: Permite mostrar pequeños formularios o solicitar confirmación.

- Tabla: Permite mostrar la información

Capturas de pantalla (Ver ilustraciones #128 - #131):

Ilustración 130: Sistema web - Mostrar contactos

Fecha Contacto	Comentario	Próximo Contacto
21/12/2021	Prueba	28/12/2021
17/12/2021	Prueba	25/12/2021

Fuente: Elaboración propia

Ilustración 131: Sistema web - Agregar contactos

Fecha Contacto	Comentario	Próximo Contacto
20/04/2023	Comentario	Fecha de Próximo Contacto

Fuente: Elaboración propia

Ilustración 132: Sistema web - Actualizar contactos

The screenshot shows the 'Gestión de contacto' (Contact Management) section of the Datafone web application. It lists two entries for 'Testing numero 4'. The first entry is dated '21/12/2021' with the comment 'Prueba'. The second entry is dated '17/12/2021' with the comment 'Prueba'. At the bottom right, a modal dialog box is open with the title 'Actualizar' (Update) and the text 'Actualizar' highlighted in blue.

Fuente: Elaboración propia

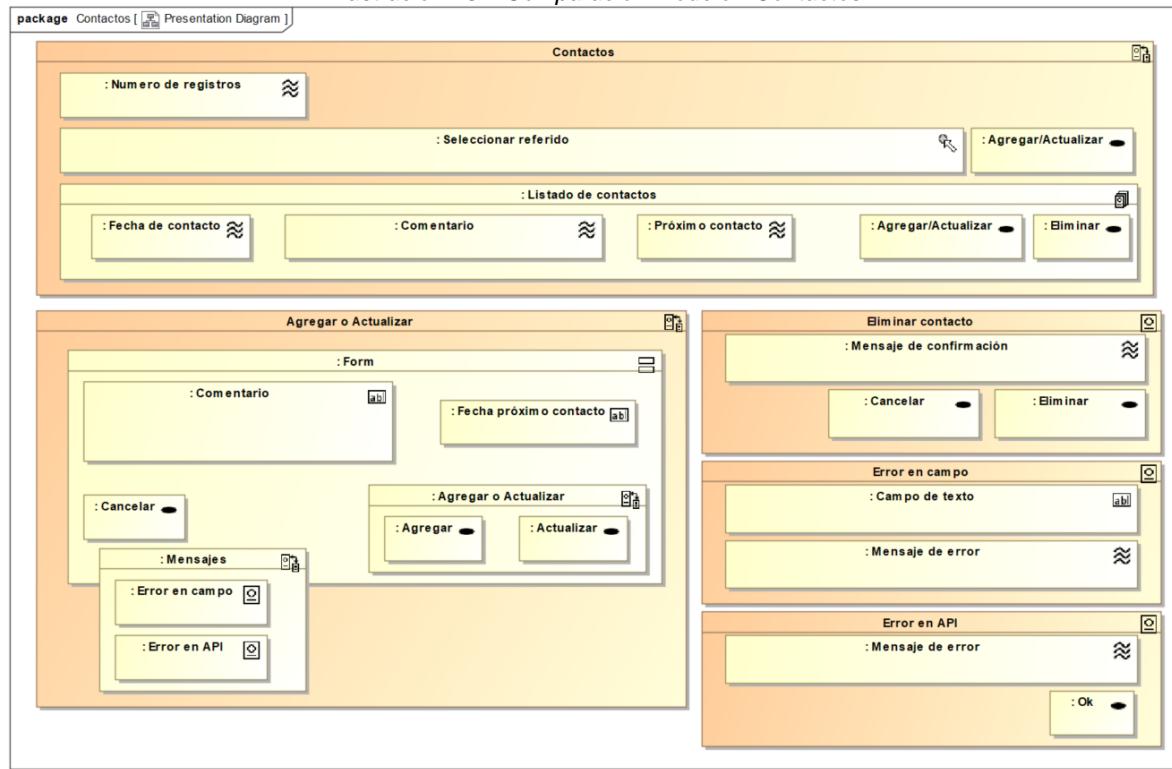
Ilustración 133: Sistema web - Eliminar contacto

The screenshot shows the 'Gestión de contacto' (Contact Management) section of the Datafone web application. It lists two entries for 'Testing numero 4'. A modal dialog box is open in the center, titled 'Eliminar contacto' (Delete contact), with the text '¿Estás seguro que deseas eliminar este registro? Esta acción no se puede revertir!' (Are you sure you want to delete this record? This action cannot be reversed!). The 'Eliminar' (Delete) button is highlighted in blue.

Fuente: Elaboración propia

Comparación con modelo de presentación (Ver ilustraciones #132 - #136):

Ilustración 134: Comparación modelo - Contactos



Fuente: Elaboración propia

Ilustración 135: Comparación modelo - Mostrar contactos

Gestión de contacto

2 Gestiones de contacto al referido Testing numero 4

Testing numero 4	83835358	test@gmail.com	+ Agregar
Fecha Contacto	Comentario	Próximo Contacto	
21/12/2021	Prueba	28/12/2021	

Fuente: Elaboración propia

Ilustración 136: Comparación modelo - Agregar contacto y Error en campo

Gestión de contacto

3 Gestiones de contacto al referido Testing numero 4

Testing numero 4	83835358	test@gmail.com	+ Agregar
Fecha Contacto	Comentario	Próximo Contacto	
20/04/2023			
Comentario	Comentario	Fecha de Próximo Contacto	
	Comentario es requerido		
	<input type="button" value="Cancelar"/>		<input type="button" value="Guardar"/>

Fuente: Elaboración propia

Ilustración 137: Comparación modelo - Eliminar contacto

The screenshot shows a contact management interface titled "Gestión de contacto". It displays a list of contacts with columns for Name, Phone number, and Email. A modal dialog box is overlaid on the screen, asking for confirmation to delete a contact. The dialog includes a warning icon, the text "Eliminar contacto", and a message about the不可 reversibility of the action. It features "Cancelar" and "Eliminar" buttons.

Gestión de contacto		
2 Gestiones de contacto al referido Testing numero 4		
	Nombre	Comentario
	Testing numero 4	83835358 test@gmail.com
Fecha Contacto	Comentario	
21/12/2021	Prueba	
17/12/2021	Prueba	

Fuente: Elaboración propia

Ilustración 138: Comparación modelo - Error en API

The screenshot shows a contact management interface titled "Gestión de contacto". It displays a list of contacts with columns for Name, Phone number, and Email. A modal dialog box is overlaid on the screen, indicating an error with a warning icon, the text "Ha ocurrido un error", and a message about processing errors. It features an "Ok" button.

Gestión de contacto		
2 Gestiones de contacto al referido Testing numero 4		
	Nombre	Comentario
	Testing numero 4	83835358 test@gmail.com
Fecha Contacto	Comentario	
21/12/2021	Prueba	
17/12/2021	Prueba	

Fuente: Elaboración propia

Interfaces: Motivos de declinación

Propósito: Presentar la información correspondiente a los motivos de declinación.

Acciones o eventos: Mostrar, filtrar, agregar, actualizar y eliminar contactos.

Elementos gráficos que la componen

- Cajas de texto: Permite mostrar o editar texto sin formato
- Botones: Permite accionar los eventos en un sitio web
- Formularios emergentes: Permite mostrar pequeños formularios o solicitar confirmación.
- Tabla: Permite mostrar la información

Capturas de pantalla (Ver ilustraciones #137 - #140):

Ilustración 139: Sistema web - Mostrar motivos de declinación

Nombre	Descripción
Título 2	Descripción 10
título prueba 1	descripción prueba 1
título prueba 3	descripción prueba 3

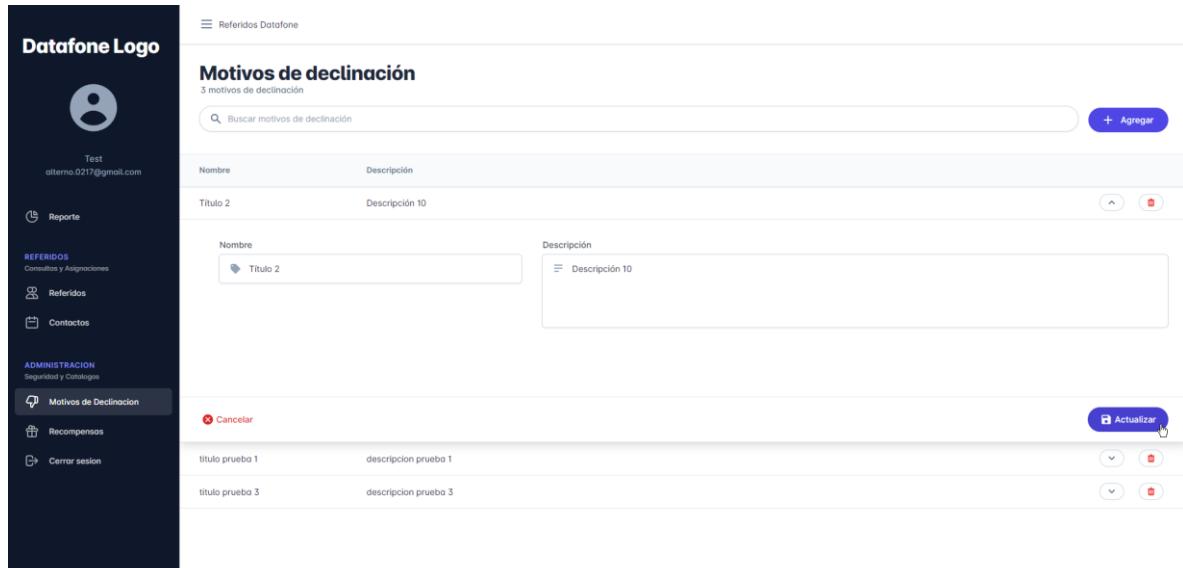
Fuente: Elaboración propia

Ilustración 140: Sistema web - Agregar motivo de declinación

Nombre	Descripción
Título 2	Descripción 10
título prueba 1	descripción prueba 1
título prueba 3	descripción prueba 3

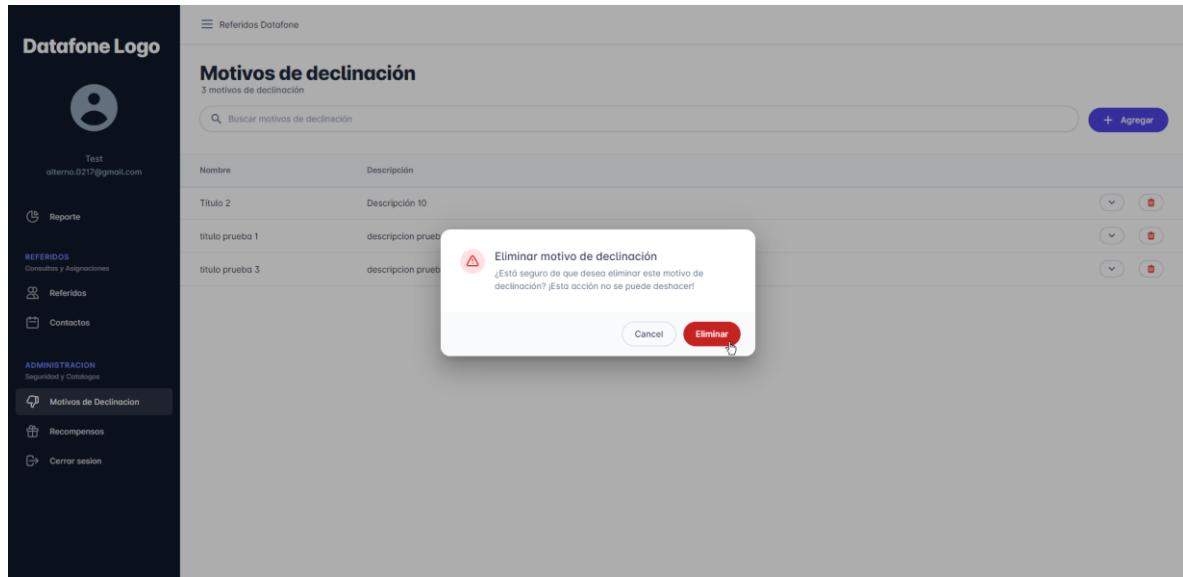
Fuente: Elaboración propia

Ilustración 141: Sistema web - Actualizar motivo de declinación



Fuente: Elaboración propia

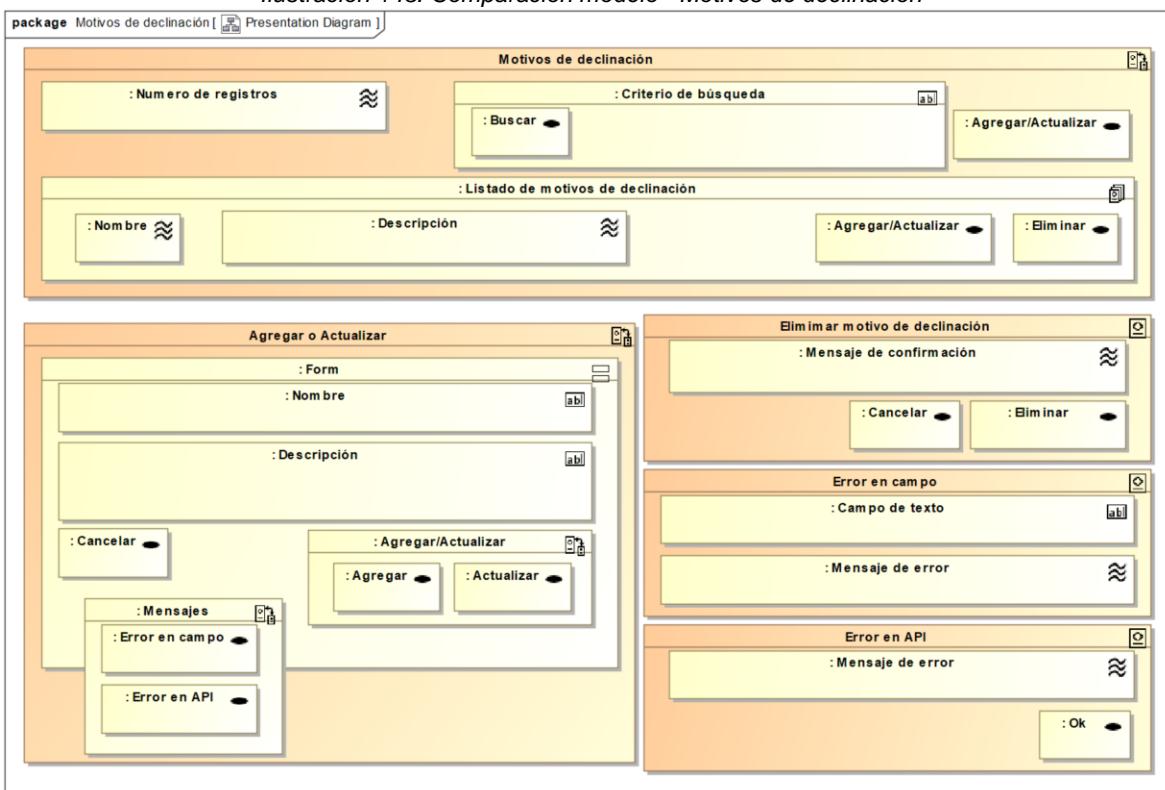
Ilustración 142: Sistema web - Eliminar motivo de declinación



Fuente: Elaboración propia

Comparación con modelo de presentación (Ver ilustraciones #141 - #146):

Ilustración 143: Comparación modelo - Motivos de declinación



Fuente: Elaboración propia

Ilustración 144: Comparación modelo - Mostrar motivos de declinación

Motivos de declinación

Motivos de declinación

Buscar motivos de declinación		+ Agregar
Nombre	Descripción	
Título 2	Descripción 10	 

Fuente: Elaboración propia

Ilustración 145: Comparación modelo - Agregar motivo de declinación y Error en campo

Motivos de declinación

4 motivos de declinación

Buscar motivo de declinación

+ Agregar

Nombre	Descripción
Nombre	Descripción
Nombre	Descripción

Nombre

Nombre es requerido

Descripción

Descripción es requerido

Cancelar

Guardar

Fuente: Elaboración propia

Ilustración 146: Comparación modelo - Actualizar motivo de declinación

Motivos de declinación
3 motivos de declinación

Buscar motivos de declinación + Agregar

Nombre	Descripción
Título 2	Descripción 10

Nombre
 Descripción

Cancelar Actualizar

Fuente: Elaboración propia

Ilustración 147: Comparación modelo - Eliminar motivo de declinación

Motivos de declinación
3 motivos de declinación

Buscar motivos de declinación

Nombre	Descripción
Título 2	Descripción 10
titulo prueba 1	descripcion prueba
titulo prueba 3	descripcion prueba

⚠ Eliminar motivo de declinación
¿Está seguro de que desea eliminar este motivo de declinación? ¡Esta acción no se puede deshacer!

Cancel Eliminar

Fuente: Elaboración propia

Ilustración 148: Comparación modelo - Error en API

The screenshot shows a web application titled "Motivos de declinación" (Reasons for Declination) with a subtitle "3 motivos de declinación". Below the title is a search bar labeled "Buscar motivos de declinación". A table lists three items:

Nombre	Descripción
Título 2	Descripción 10
titulo prueba 1	descripcion prueba 1
titulo prueba 3	descripcion prueba 3

An error dialog box is overlaid on the page, containing the message "Ha ocurrido un error" (An error has occurred) and "Ocurrió un error al procesar la información" (An error occurred while processing the information). An "Ok" button is at the bottom right of the dialog.

Fuente: Elaboración propia

Interfaces: Recompensas

Propósito: Presentar la información correspondiente a la configuración de recompensas.

Acciones o eventos: Mostrar, filtrar, agregar, actualizar y eliminar recompensas.

Elementos gráficos que la componen

- Cajas de texto: Permite mostrar o editar texto sin formato
- Botones: Permite accionar los eventos en un sitio web
- Formularios emergentes: Permite mostrar pequeños formularios o solicitar confirmación.
- Tabla: Permite mostrar la información
- Lista de selección: Permite seleccionar un elemento de una lista de opciones
- Casillas de verificación: Las casillas de verificación permiten que el usuario seleccione una o más opciones de un conjunto

Capturas de pantalla (Ver ilustraciones #147 - #150):

Ilustración 149: Sistema web - Mostrar recompensas

Título	Modalidad de Pago	Inicio	Fin
Recompensa Prepago	Prepago	31/05/2023	30/04/2023
Recompensa Postpago	Postpago	31/03/2023	30/04/2023

Fuente: Elaboración propia

Ilustración 150: Sistema web - Agregar recompensa

Elementos por página: 10 | 1-3 de 3

Fuente: Elaboración propia

Ilustración 151: Sistema web - Actualizar recompensa

The screenshot shows the 'Actualizar recompensa' (Update Reward) page. The reward is titled 'Recompensa Prepago'. Configuration includes:

- Modalidad de Pago: Prepago
- Límite máx. Reward por referido: \$ 10
- Número DA: 60
- Tiempo máx. para completar Reward (días): 30
- Vigencia DA (Días): 30
- Monto mínimo de recarga: \$ 5
- Inicio: 31/3/2023
- Fin: 30/4/2023
- Siempre Activo: checked

The activation plan table shows two entries:

Monto Recarga	Tipo Reward	Plan a Activar	Monto a Aplicar
10	Hacer Recarga	10	\$ 10
5	Hacer Recarga	5	\$ 5

Buttons at the bottom include 'Cancelar' (Cancel) and 'Actualizar' (Update).

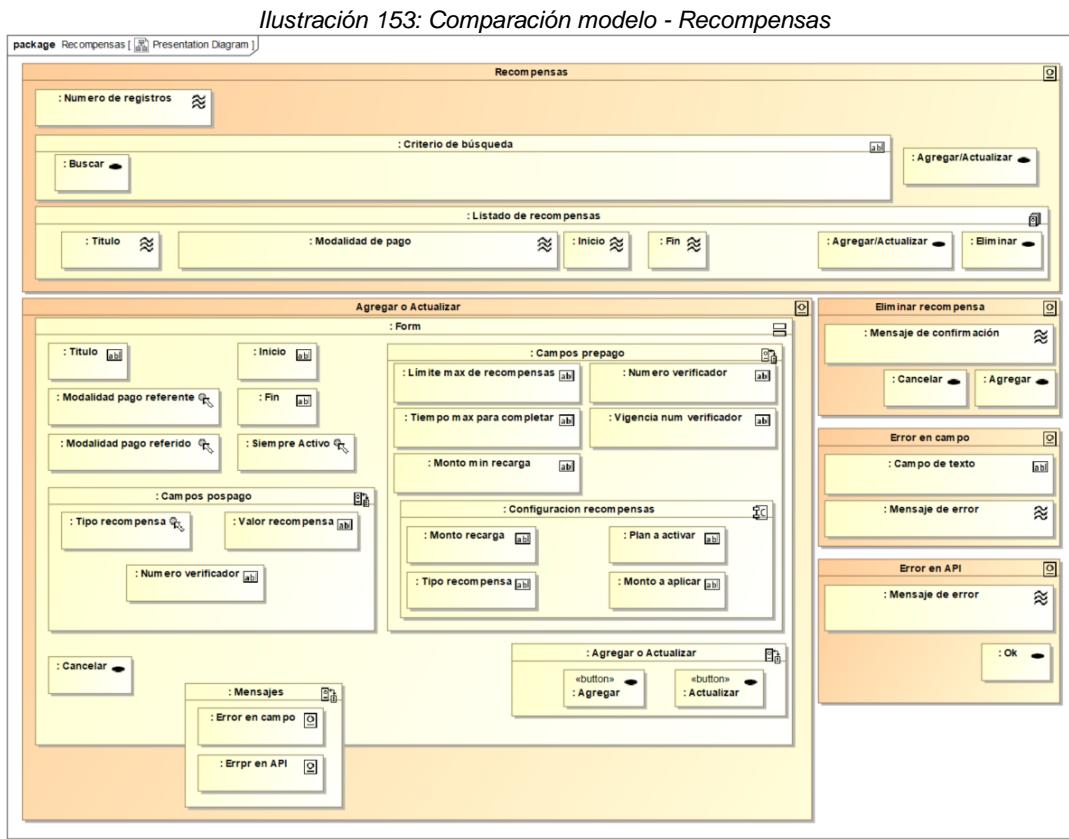
Fuente: Elaboración propia

Ilustración 152: Sistema web - Eliminar recompensa

The screenshot shows the 'Eliminar Reward' (Delete Reward) confirmation dialog. It contains the message: '¿Estás seguro que deseas eliminar este registro? Esta acción no se puede revertir!' (Are you sure you want to delete this record? This action cannot be reversed!). Buttons are 'Cancelar' (Cancel) and 'Eliminar' (Delete).

Fuente: Elaboración propia

Comparación con modelo de presentación (Ver ilustraciones #151 - #157):



Fuente: Elaboración propia

Ilustración 154: Comparación modelo - Mostrar y filtrar recompensas

Recompensas
2 Registros de rewards

Título	Modalidad de Pago	Inicio	Fin
Recompensa Prepago	Prepago	31/03/2023	30/04/2023

+ Agregar

Fuente: Elaboración propia

Ilustración 155: Comparación modelo - Agregar recompensas prepago

Recompensas
3 Registros de rewards

+ Agregar

Título	Modalidad de Pago	Inicio	Fin
	Límite max. Reward por referente \$ 0	Iniciar DA	
Título es requerido	Tiempo máx. para completar Reward (días) 0	Vigencia DA (Días)	
Modalidad de Pago del Referente <input type="radio"/> Postpago <input checked="" type="radio"/> Prepago	Monto mínimo de recarga \$ 0	Inicio	
Modalidad de Pago del Referido <input type="radio"/> Postpago <input checked="" type="radio"/> Prepago	Plan a Activar	Fin	
	Monto a Aplicar	20/4/2023	20/9/2023
+		<input type="checkbox"/> Siempre Activo	
Monto Recarga Tipo Reward Plan a Activar Monto a Aplicar		No data	

Cancelar Guardar

Fuente: Elaboración propia

Ilustración 156: Comparación modelo - Agregar recompensa pospago

Recompensas
3 Registros de rewards

+ Agregar

Título	Modalidad de Pago	Inicio	Fin
	Inicio 20/4/2023	Fin 20/9/2023	
Título es requerido			
Modalidad de Pago del Referente <input checked="" type="radio"/> Postpago <input type="radio"/> Prepago			
Modalidad de Pago del Referido <input type="radio"/> Postpago <input checked="" type="radio"/> Prepago			
Tipo de Reward 	Valor del Reward \$		
Código tipo ajuste			
Cancelar		Guardar	

Fuente: Elaboración propia

Ilustración 157: Comparación modelo - Actualizar recompensa

Recompensas
2 Registros de rewards

Buscar Reward + Agregar

Título	Modalidad de Pago	Límite máx. Reward por referido	Número DA
Recompensa Prepago	Prepago	\$ 10	58
Modalidad de Pago del Referente	Postpago	30	30
Modalidad de Pago del Referido	Postpago	5	31/3/2023
			Fin
			30/4/2023
			<input type="checkbox"/> Siempre Activo

+

Monto Recarga	Tipo Reward	Plan a Activar	Monto a Aplicar
10	Hacer Recarga	10	5
5	Hacer Recarga	5	5

Cancelar Actualizar

Fuente: Elaboración propia

Ilustración 158: Comparación modelo - Eliminar recompensa

Recompensas
2 Registros de rewards

Buscar Reward

Título	Modalidad de Pago
Recompensa Prepago	Prepago
Recompensa Postpago	Postpago

⚠ Eliminar Reward
¿Estás seguro que deseas eliminar este registro? Esta acción no se puede revertir!

Cancelar Eliminar

Fuente: Elaboración propia

Ilustración 159: Comparación modelo - Error en API

Recompensas
2 Registros de rewards

Buscar Reward

Título	Modalidad de Pago
Recompensa Prepago	Prepago
Recompensa Postpago	Postpago

⚠ Ha ocurrido un error
Ocurrió un error al procesar la información

Ok

Fuente: Elaboración propia

Interfaces: Reportes

Propósito: Presentar reportes.

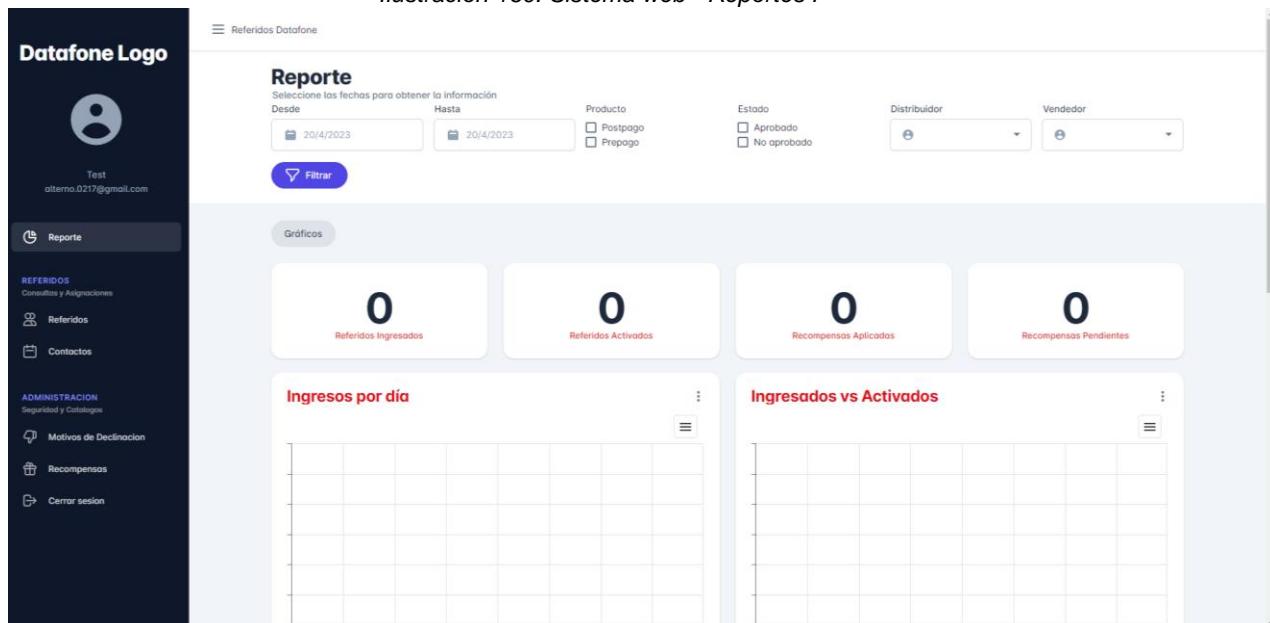
Acciones o eventos: Mostrar información complementaria.

Elementos gráficos que la componen

- Cajas de texto: Permite mostrar o editar texto sin formato
- Botones: Permite accionar los eventos en un sitio web
- Lista de selección: Permite seleccionar un elemento de una lista de opciones
- Casillas de verificación: Las casillas de verificación permiten que el usuario seleccione una o más opciones de un conjunto

Capturas de pantalla (Ver ilustraciones #158 - #159):

Ilustración 160: Sistema web - Reportes I



Fuente: Elaboración propia

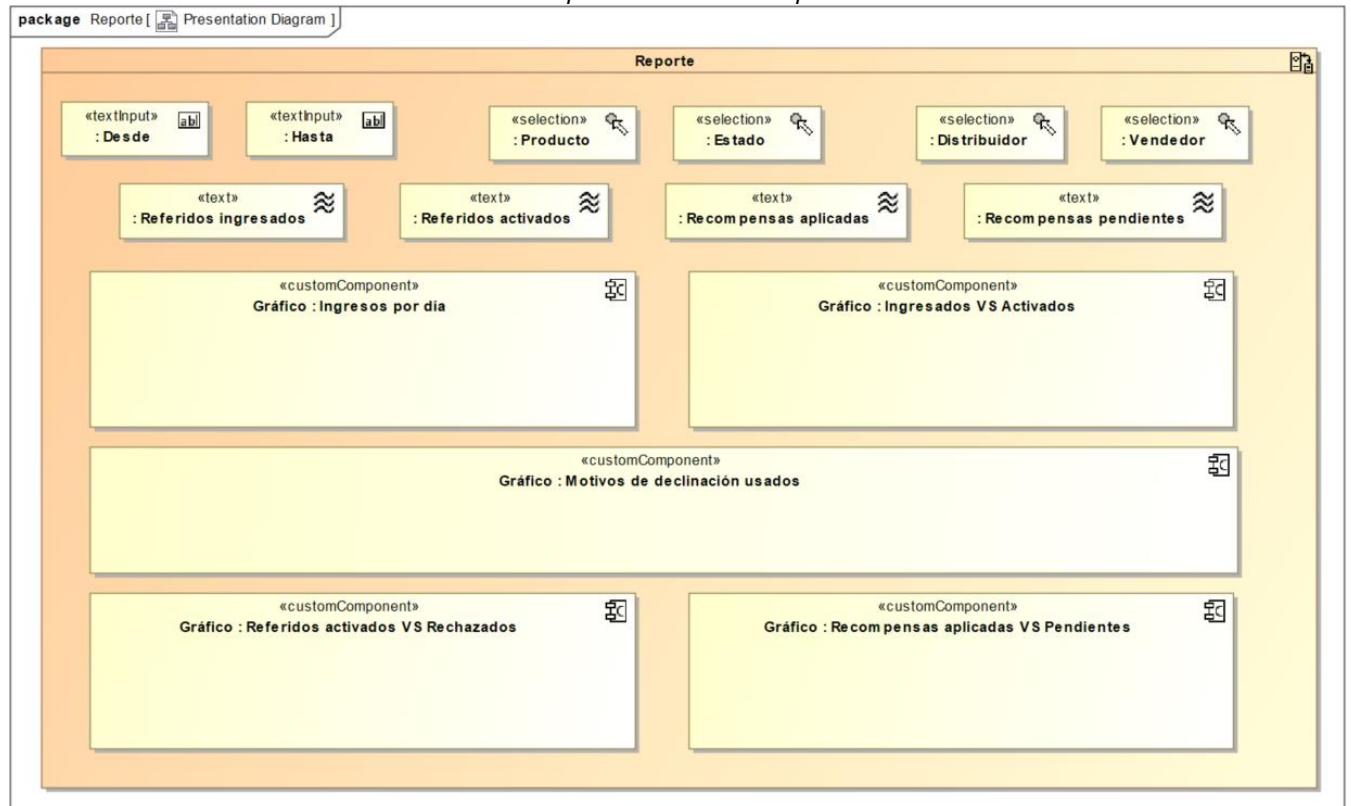
Ilustración 161: Sistema web - Reportes II



Fuente: Elaboración propia

Comparación con modelo de presentación (Ver ilustraciones #160 - #164):

Ilustración 162: Comparación modelo - Reporte



Fuente: Elaboración propia

Ilustración 163: Comparación modelo - Reporte I

Reporte
Seleccione las fechas para obtener la información

Desde	Hasta	Producto	Estado	Distribuidor	Vendedor
<input type="text" value="20/4/2023"/>	<input type="text" value="20/4/2023"/>	<input type="checkbox"/> Postpago <input type="checkbox"/> Prepago	<input type="checkbox"/> Aprobado <input type="checkbox"/> No aprobado	<input type="button" value=""/>	<input type="button" value=""/>

Filtrar

Gráficos

0
Referidos Ingresados

0
Referidos Activados

0
Recompensas Aplicadas

0
Recompensas Pendientes

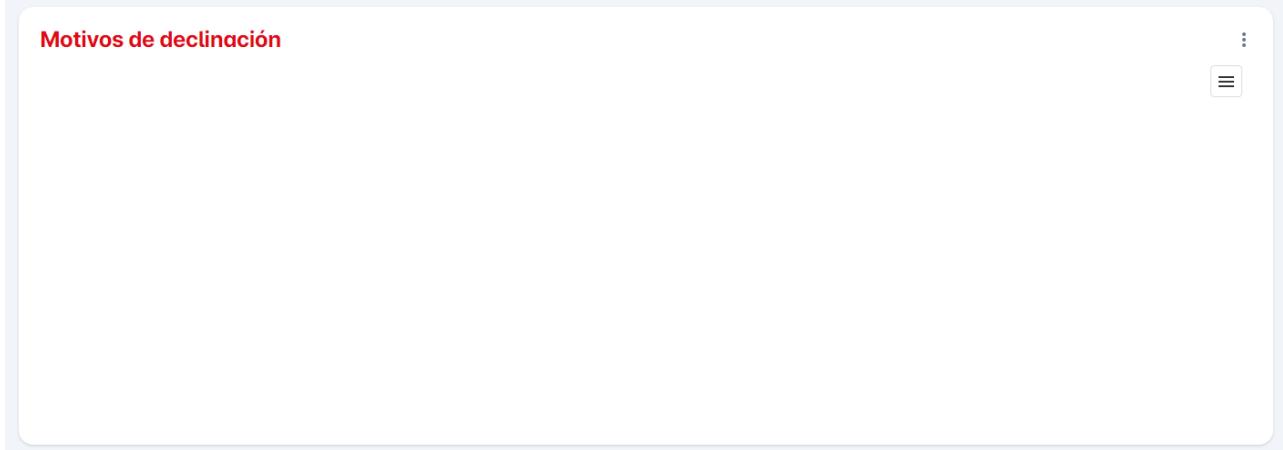
Fuente: Elaboración propia

Ilustración 164: Comparación modelo - Reporte II



Fuente: Elaboración propia

Ilustración 165: Comparación modelo - Reporte III



Fuente: Elaboración propia

Ilustración 166: Comparación modelo - Reporte IV



Fuente: Elaboración propia