

UNIVERSIDAD CENTRAL DE VENEZUELA
FACULTAD DE CIENCIAS
ESCUELA DE COMPUTACIÓN



Plataforma de Enrutamiento Transaccional para múltiples canales de pago en un contexto Multibanco: Caso Botón de Pago

**Trabajo Especial de Grado presentado ante la
Ilustre Universidad Central de Venezuela por la
Br. Emily Rebeca Corro Escobar (C.I. 20.191.602)**

Para optar por el título de Licenciado en Computación.

Tutor: Prof. Aparicio Peña

Ciudad Universitaria de Caracas, 03/06/2016

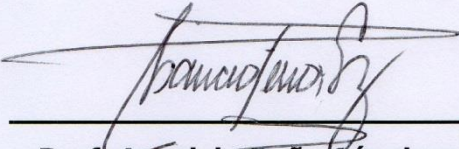
**UNIVERSIDAD CENTRAL DE VENEZUELA
FACULTAD DE CIENCIAS
ESCUELA DE COMPUTACIÓN**

ACTA

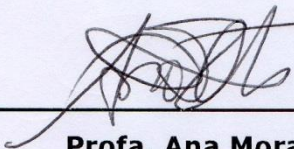
Quienes suscriben, miembros del jurado designado por el Consejo de la Escuela de Computación, para examinar el Trabajo Especial de Grado titulado "**Plataforma de Enrutamiento Transaccional para múltiples canales de pago en un contexto Multibanco: Caso Botón de Pago**" y presentado por la Br. **Emily Rebeca Corro Escobar C.I.V-20.191.602**, a los fines de optar al título de **Licenciado en Computación**, dejamos constancia de lo siguiente:

Leído como fue dicho trabajo, por cada uno de los miembros del jurado, se fijó el día 06 de Junio de 2016, a las 10:00am horas, para que el (los) autor(es) lo defendiera(n) en forma pública, lo que este (esta/estos) hizo (hicieron) en Sala de postgrado de la Escuela de Computación, mediante una presentación oral de su contenido, luego de lo cual respondieron a las preguntas formuladas. Finalizada la defensa pública del Trabajo Especial de Grado, el jurado decidió aprobar con la nota de 20 puntos.

En fe de lo cual se levanta la presente Acta, en Caracas el día 06 de Junio de 2016.



Prof. Aparicio Peña Sánchez
(Tutor)



Profa. Ana Morales
(Jurado)



Prof. Antonio Leal
(Jurado)

Agradecimientos

A mi madre, Rebeca, cuya fuerza de voluntad y determinación son ejemplos para mí.

A mi papá, José, quien a mis cuatro años me inculcó la lectura, abriendo las puertas a mi curiosidad nata.

A mi hermano José Antonio, mi fuente inagotable de risas y ánimo, eres quien me motiva a ser un buen ejemplo.

A mi abuela Vicenta y mi (casi) abuela Aida, pilares fundamentales de mi formación y a quienes quiero y admiro infinitamente.

A mi mejor amiga Laura, a quien admiro un montón y cuyo apoyo invaluable hizo más llevadero este camino.

A Rafa, Vicente y Pascu, por su amistad, apoyo constante y consejos de vez en cuando.

A mi tutor, Prof Aparicio Peña, un guía con experiencia y conocimiento invaluable.

A mi jefe, Víctor Vera, por darme la oportunidad de mi primer empleo en la familia de Solutions Systems y su apoyo en el desarrollo de este proyecto.

A mis compañeros de trabajo Denidson Álvarez y Mónica Mujica, siempre dispuestos a colaborar y aportar opiniones sobre mis ideas.

A Héctor Rangel por sus apreciados consejos y ayuda prestada.

Y a todos los que hicieron este proyecto posible.

Gracias.



Universidad Central de Venezuela
Facultad de Ciencias
Escuela de Computación

Plataforma de Enrutamiento Transaccional para múltiples canales de pago en un contexto Multibanco: Caso Botón de Pago

Autor: Emily Rebeca Corro Escobar
Correo-e: emily.corro@ciens.ucv.ve

Tutor: Prof. Aparicio Peña
Correo-e: aparicio.pena@ciens.ucv.ve

Fecha: 03 de junio de 2016

RESUMEN

Los medios y canales de pago son los puntos de acceso a las operaciones asociadas a la adquisición de bienes y servicios, constituyen las herramientas que les proporcionan las instituciones financieras a sus clientes para facilitar y automatizar el desarrollo del comercio. Con el pasar de los años, estos elementos han ido a la par de la evolución de la electrónica y las tecnologías de la información obligando a las instituciones involucradas (industria bancaria, de adquirencia, procesadores, franquicias, entre otros) a innovar constantemente, no solo en los medios y canales de pago en sí, sino también en todos los aspectos involucrados en el funcionamiento de los mismos como seguridad, redes, dispositivos, entre otros. Producto de este constante desarrollo en las tecnologías relacionadas con los pagos electrónicos es necesario que los sistemas de los entes e instituciones financieras sean capaces de adaptarse a nuevos esquemas que involucren medios y canales de pago con tecnologías emergentes.

Palabras Clave: instrumentos, canales, pago, transacción, interoperabilidad, seguridad, estándar, evolución.

Índice General

INTRODUCCIÓN.....	I
CAPÍTULO I.....	1
PROBLEMA DE INVESTIGACIÓN.....	1
1. <i>Planteamiento del problema.....</i>	<i>1</i>
2. <i>Objetivo General del Trabajo Especial de Grado.....</i>	<i>3</i>
3. <i>Objetivos Específicos del Trabajo Especial de Grado.....</i>	<i>3</i>
4. <i>Alcance del Trabajo Especial de Grado.....</i>	<i>4</i>
CAPÍTULO II.....	5
MARCO TEÓRICO CONCEPTUAL.....	5
1. <i>Pagos y sistemas de pago.....</i>	<i>5</i>
2. <i>La industria de las tarjetas de pago.....</i>	<i>8</i>
3. <i>Normas de Seguridad de Datos de la Industria de Tarjetas de Pago (PCI).....</i>	<i>14</i>
4. <i>Estándar ISO 8583.....</i>	<i>16</i>
5. <i>Regulaciones a nivel nacional para el sistema financiero.....</i>	<i>23</i>
CAPÍTULO III.....	26
MARCO METODOLOGÓGICO.....	26
1. <i>Metodología Scrum.....</i>	<i>26</i>
2. <i>Implantación de Scrum dentro de la empresa.....</i>	<i>30</i>
CAPÍTULO IV.....	32
DESARROLLO DE LA SOLUCIÓN.....	32
1. <i>Análisis y requerimientos.....</i>	<i>32</i>
1.1. <i>Análisis de plataforma actual.....</i>	<i>32</i>
1.2. <i>Requerimientos de la aplicación.....</i>	<i>33</i>
1.3. <i>Usuarios a los que está dirigida.....</i>	<i>34</i>
1.4. <i>Aplicaciones similares.....</i>	<i>35</i>
2. <i>Diseño de la plataforma.....</i>	<i>36</i>
2.1. <i>Arquitectura del sistema.....</i>	<i>36</i>
2.2. <i>Principales funcionalidades o herramientas provistas por la aplicación.....</i>	<i>38</i>
2.3. <i>Casos de uso del sistema.....</i>	<i>44</i>
2.4. <i>Procesos principales.....</i>	<i>57</i>
2.5. <i>Modelado de Datos.....</i>	<i>64</i>
3. <i>Implementación.....</i>	<i>76</i>
3.1. <i>Etapas seguidas para la creación de la aplicación.....</i>	<i>76</i>
3.2. <i>Plataforma utilizada para su desarrollo.....</i>	<i>77</i>
3.3. <i>Patrones de interfaz, interacción o diseño utilizados.....</i>	<i>80</i>
3.4. <i>Principales interfaces.....</i>	<i>80</i>
3.5. <i>Módulos que comprenden el Motor Transaccional V-Financial.....</i>	<i>85</i>
3.5.1. <i>Núcleo Operacional.....</i>	<i>85</i>
3.5.1.1. <i>Paquete: vpago.core.db.....</i>	<i>86</i>
3.5.1.2. <i>Paquete: vpago.core.transactions.....</i>	<i>87</i>
3.5.1.3. <i>Paquete: vpago.core.transactions.classes.....</i>	<i>87</i>
3.5.1.4. <i>Paquete: vpago.core.transactions.classes.types.....</i>	<i>90</i>
3.5.1.5. <i>Paquete: vpago.core.transactions.exceptions.....</i>	<i>90</i>
3.5.1.6. <i>Paquete: vpago.core.transactions.service.....</i>	<i>92</i>

3.5.1.7.	Paquete: <i>vpago.core.transactions.util</i>	92
3.5.2.	Interfaz de Pago	92
3.5.2.1.	Paquete: <i>vpago.interfaceapi.classes</i>	93
3.5.2.2.	Paquete: <i>vpago.interfaceapi.service</i>	93
3.5.2.3.	Paquete: <i>vpago.interfaceapi.types</i>	94
3.5.3.	API de Acceso a Datos	94
3.5.3.1.	Paquete: <i>vpago.dataapi.classes</i>	95
3.5.3.2.	Paquete: <i>vpago.dataapi.minified</i>	97
3.5.3.3.	Paquete: <i>vpago.dataapi.wrappers</i>	98
3.5.3.4.	Paquete: <i>vpago.dataapi.db</i>	99
3.5.3.5.	Paquete: <i>vpago.dataapi.migrate</i>	99
3.5.3.6.	Paquete: <i>vpago.dataapi.service</i>	100
3.5.3.7.	Paquete: <i>vpago.dataapi.types</i>	101
3.5.3.8.	Paquete: <i>vpago.dataapi.util</i>	101
4.	Despliegue y pruebas	102
4.1.	Plataforma necesaria para utilizar la aplicación.....	102
4.2.	Diagrama de despliegue	105
4.3.	Pruebas	105
4.3.1.	Conjunto de pruebas unitarias de aplicación C-Pago.....	106
4.3.2.	Resumen de resultados de pruebas unitarias aplicación C-Pago	107
4.3.3.	Conjunto de pruebas unitarias Motor V-Financial.....	112
4.3.3.1.	Resultados de pruebas unitarias en módulo API de Acceso a Datos	123
4.3.3.2.	Resultados de pruebas unitarias en módulo Interfaz de Pago.....	127
4.3.4.	Pruebas de integración.....	131
4.3.5.	Resultados pruebas de integración	131
	RESULTADOS DEL TRABAJO ESPECIAL DE GRADO	133
	CONCLUSIONES	134
	TRABAJOS FUTUROS	136
	REFERENCIAS BIBLIOGRÁFICAS Y DIGITALES	137

Índice de Ilustraciones

Figura 1. Múltiples canales como centros de pago de minoristas. (Capgemini's Strategic Analysis Group, 2012)	3
Figura 2. Ciclo de vida simplificado de un pago no en efectivo. (European Central Bank, 2010).....	5
Figura 3. Volumen global de transacciones con tarjetas (billones) por región, 2007-11. (Gupta & Roy, 2014).	9
Figura 4. Estructura típica de un flujo de transacción con tarjeta. (Gupta & Roy, 2014).	10
Figura 5. Comerciantes que no utilizan herramientas de detección de fraude en línea. (Capgemini's Strategic Analysis Group, 2012).	11
Figura 6. Ciclo de autorización de la tarjeta con los grupos de interés. (Capgemini's Strategic Analysis Group, 2012).	12
Figura 7. Elementos fundamentales del proceso Scrum. (Navarro, Fernández, & Morales, 2013).....	30
Figura 8. Esquema en producción de punto de venta virtual VPOS. (Elaboración propia).	33
Figura 9. Página principal de PayU Latam. (Payu Latam).....	35
Figura 10. Página principal de Authorize.Net. (Authorize.Net)	36
Figura 11. Componentes de la plataforma V-Financial. (Elaboración propia).	37
Figura 12. Interacción de componentes de la plataforma V-Financial. (Elaboración propia).	38
Figura 13. Motor Transaccional V-Financial – Interfaz de Pago: Diagrama de casos de uso nivel 0. (Elaboración propia).....	44
Figura 14. Transaccional V-Financial – Interfaz de Pago: Diagrama de casos de uso nivel 1. (Elaboración propia). .	45
Figura 15. Transaccional V-Financial – Interfaz de Pago: Diagrama de casos de uso nivel 2. (Elaboración propia). .	45
Figura 16. Motor Transaccional V-Financial – Núcleo Operacional: Diagrama de casos de uso nivel 0. (Elaboración propia).....	46
Figura 17. Motor Transaccional V-Financial – Núcleo Operacional: Diagrama de casos de uso nivel 1. (Elaboración propia).....	46
Figura 18. Motor Transaccional V-Financial – Núcleo Operacional: Diagrama de casos de uso nivel 2. (Elaboración propia).....	47
Figura 19. Motor Transaccional V-Financial – API de Acceso a Datos: Diagrama de casos de uso nivel 0. (Elaboración propia).	47
Figura 20. Motor Transaccional V-Financial – API de Acceso a Datos: Diagrama de casos de uso nivel 1. (Elaboración propia).	48
Figura 21. Motor Transaccional V-Financial – API de Acceso a Datos: Diagrama de casos de uso nivel 2 - Gestionar Comercios. (Elaboración propia).	50
Figura 22. Motor Transaccional V-Financial – API de Acceso a Datos: Diagrama de casos de uso nivel 2 - Gestionar Corporaciones. (Elaboración propia).	51
Figura 23. Motor Transaccional V-Financial – API de Acceso a Datos: Diagrama de casos de uso nivel 2 - Gestionar Terminales. (Elaboración propia).	52
Figura 24. Motor Transaccional V-Financial – API de Acceso a Datos: Diagrama de casos de uso nivel 2 - Gestionar Bancos. (Elaboración propia).	52
Figura 25. Motor Transaccional V-Financial – API de Acceso a Datos: Diagrama de casos de uso nivel 2 - Gestionar Afiliaciones. (Elaboración propia).....	53
Figura 26. Motor Transaccional V-Financial – API de Acceso a Datos: Diagrama de casos de uso nivel 2 - Gestionar Adquiriencia. (Elaboración propia).	53
Figura 27. Motor Transaccional V-Financial – API de Acceso a Datos: Diagrama de casos de uso nivel 2 - Gestionar Reglas. (Elaboración propia).	54
Figura 28. Motor Transaccional V-Financial – API de Acceso a Datos: Diagrama de casos de uso nivel 2 - Gestionar Dispositivos. (Elaboración propia).	55
Figura 29. Motor Transaccional V-Financial – API de Acceso a Datos: Diagrama de casos de uso nivel 2 - Gestionar Clasificación de Comercios. (Elaboración propia).	56
Figura 30. Motor Transaccional V-Financial – API de Acceso a Datos: Diagrama de casos de uso nivel 2 - Gestionar Canales de Pago. (Elaboración propia).	56
Figura 31. Aplicación C-Pago: Diagrama de casos de uso nivel 0. (Elaboración propia).	57
Figura 32. Aplicación C-Pago: Diagrama de casos de uso nivel 1. (Elaboración propia).	57

Figura 33. Proceso de compra con tarjeta de crédito a través del canal C-Pago. (Elaboración propia).....	58
Figura 34. Actividades llevadas a cabo por el módulo de Interfaz de Pago durante la ejecución de una transacción.	59
Figura 35. Actividades llevadas a cabo por el módulo Núcleo Operacional durante la ejecución de una transacción.	61
Figura 36. Actividades generales llevadas a cabo por el módulo API de Acceso a Datos para la ejecución de una función.	64
Figura 37. Diagrama de base de datos Motor V-Financial. (Elaboración propia).	65
Figura 38. Herramientas utilizadas para el desarrollo de la aplicación. (Elaboración propia).	79
Figura 39. Aplicación C-Pago: Formulario de datos del tarjetahabiente. (Elaboración propia).	81
Figura 40. Aplicación C-Pago: Pantalla de confirmación de la transacción. (Elaboración propia).	82
Figura 41. Aplicación C-Pago: Pantalla de error de la aplicación. (Elaboración propia).	83
Figura 42. Aplicación C-Pago: Resultado con transacción aprobada. (Elaboración propia).	84
Figura 43. Aplicación C-Pago: Resultado con transacción rechazada. (Elaboración propia).	85
Figura 44. Paquetes que comprenden el módulo Núcleo Operacional (Elaboración propia).....	86
Figura 45. Diagrama de clases del paquete vpago.core.db (Elaboración propia).	86
Figura 46. Diagrama de clases del paquete vpago.core.transactions (Elaboración propia).	87
Figura 47. Diagrama de clases del paquete vpago.core.transactions.classes (Elaboración propia).	88
Figura 48. Diagrama de clases del paquete vpago.core.transactions.classes.types (Elaboración propia).	90
Figura 49. Diagrama de clases del paquete vpago.core.transactions.exceptions (Elaboración propia).	91
Figura 50. Diagrama de clases del paquete vpago.core.transactions.service (Elaboración propia).	92
Figura 51. Diagrama de clases del paquete vpago.core.transaction.util (Elaboración propia).	92
Figura 52. Paquetes que comprenden el módulo Interfaz de Pago (Elaboración propia).....	92
Figura 53. Diagrama de clases del paquete vpago.interfaceapi.classes (Elaboración propia).	93
Figura 54. Diagrama de clases del paquete vpago.interfaceapi.service (Elaboración propia).	94
Figura 55. Diagrama de clases del paquete vpago.interfaceapi.types (Elaboración propia).	94
Figura 56. Paquetes que comprenden el módulo API de Acceso a Datos (Elaboración propia).	95
Figura 57. Diagrama de clases del paquete vpago.dataapi.classes (Elaboración propia).	96
Figura 58. Diagrama de clases del paquete vpago.dataapi.classes.minified (Elaboración propia).	97
Figura 59. Diagrama de clases del paquete vpago.dataapi.classes.wrappers (Elaboración propia).	98
Figura 60. Diagrama de clases del paquete vpago.dataapi.db (Elaboración propia).	99
Figura 61. Diagrama de clases del paquete vpago.dataapi.migrate (Elaboración propia).	99
Figura 62. Diagrama de clases del paquete vpago.dataapi.service (Elaboración propia).	100
Figura 63. Diagrama de clases del paquete vpago.dataapi.types (Elaboración propia).	101
Figura 64. Diagrama de clases del paquete vpago.dataapi.util (Elaboración propia).	101
Figura 65. Arquitectura de hardware propuesta para el despliegue de la aplicación. (Elaboración propia).	102
Figura 66. Arquitectura de alta disponibilidad propuesta. (Elaboración propia).	103
Figura 67. Diagrama de despliegue de los componentes de la Plataforma V-Financial (Elaboración propia).	105
Figura 68. Resultados de ejecución de casos de la API de Acceso a Datos en herramienta de pruebas SoapUI. (Elaboración propia).	124
Figura 69. Resultados de ejecución de casos de la API de Acceso a Datos en herramienta de pruebas SoapUI (cont.). (Elaboración propia).	125
Figura 70. Resultados de ejecución de casos de la API de Acceso a Datos en herramienta de pruebas SoapUI (cont.). (Elaboración propia).	126
Figura 71. Resumen de casos para la Interfaz de Pago en herramienta de pruebas SoapUI. (Elaboración propia).	127
Figura 72. Resumen de casos para la Interfaz de Pago en herramienta de pruebas SoapUI (cont.). (Elaboración propia).....	128
Figura 73. Resultados de ejecución de casos de la Interfaz de Pago en herramienta de pruebas SoapUI. (Elaboración propia).....	129
Figura 74. Resultados de ejecución de casos de la Interfaz de Pago en herramienta de pruebas SoapUI (cont.). (Elaboración propia).	130

Índice de Tablas

Tabla 1. Componentes principales de un mensaje en formato ISO 8583.	17
Tabla 2. Subcampos del Message Type Identifier de un mensaje con formato ISO 8583.	17
Tabla 3. Significado de los valores de los subcampos del Message Type Identifier de un mensaje con formato ISO 8583.	18
Tabla 4. Valores estándar del campo Message Type Indicator en un mensaje con formato ISO 8583.	18
Tabla 5. Tipos de datos de un Data Element en un mensaje con formato ISO 8583.	19
Tabla 6. Formatos de longitud de un Data Element en un mensaje con formato ISO 8583.	20
Tabla 7. Tipos de dato y longitud de los Data Element definidos por el estándar ISO.	20
Tabla 8. Factores de autenticación requeridos por la norma Sudeban 641-10 para las operaciones de banca electrónica.	24
Tabla 9. Cuadro resumen de las funciones de la API de Acceso a Datos.	39
Tabla 10. Construcción de una trama ISO 8583. (Elaboración propia).	61
Tabla 11. Análisis de una trama ISO 8583. (Elaboración propia).	62
Tabla 12. Descripción de los paquetes que comprenden el módulo Núcleo Operacional (Elaboración propia).	86
Tabla 13. Descripción de las clases del paquete vpago.core.db (Elaboración propia).	87
Tabla 14. Descripción de las clases del paquete vpago.core.transactions (Elaboración propia).	87
Tabla 15. Descripción de las clases del paquete vpago.core.transaction.classes (Elaboración propia).	89
Tabla 16. Descripción de las clases del paquete vpago.core.transactions.classes.types (Elaboración propia).	90
Tabla 17. Descripción de las clases del paquete vpago.core.transactions.exceptions (Elaboración propia).	91
Tabla 18. Descripción de las clases del paquete vpago.core.transactions.service (Elaboración propia).	92
Tabla 19. Descripción de las clases del paquete vpago.core.transaction.util (Elaboración propia).	92
Tabla 20. Descripción de los paquetes que comprenden el módulo Interfaz de Pago (Elaboración propia).	93
Tabla 21. Descripción de las clases del paquete vpago.interfaceapi.classes (Elaboración propia).	93
Tabla 22. Descripción de las clases del paquete vpago.interfaceapi.service (Elaboración propia).	94
Tabla 23. Descripción de las clases del paquete vpago.interfaceapi.types (Elaboración propia).	94
Tabla 24. Descripción de los paquetes que comprenden el módulo API de Acceso a Datos (Elaboración propia).	95
Tabla 25. Descripción de las clases del paquete vpago.dataapi.classes (Elaboración propia).	96
Tabla 26. Descripción de las clases del paquete vpago.dataapi.classes.minified (Elaboración propia).	97
Tabla 27. Descripción de las clases del paquete vpago.dataapi.classes.wrappers (Elaboración propia).	98
Tabla 28. Descripción de las clases del paquete vpago.dataapi.db (Elaboración propia).	99
Tabla 29. Descripción de las clases del paquete vpago.dataapi.migrate (Elaboración propia).	100
Tabla 30. Descripción de las clases del paquete vpago.dataapi.service (Elaboración propia).	100
Tabla 31. Descripción de las clases del paquete vpago.dataapi.types (Elaboración propia).	101
Tabla 32. Descripción de las clases del paquete vpago.dataapi.util (Elaboración propia).	101
Tabla 33. Componentes de software necesarios para el despliegue de la aplicación.	104
Tabla 34. Tabla resumen de las instancias de casos de prueba de aplicación C-Pago. (Elaboración propia).	106
Tabla 35. Tabla resumen de las instancias de casos de prueba del Motor V-Financial. (Elaboración propia).	112
Tabla 36. Tabla resumen de las instancias de casos de prueba de aplicación C-Pago y Motor Transaccional. (Elaboración propia).	131

Introducción

Los medios y canales de pago son los puntos de acceso a las operaciones asociadas a la adquisición de bienes y servicios, constituyen las herramientas que les proporcionan las instituciones financieras a sus clientes para facilitar y automatizar el desarrollo del comercio. Los medios de pago constituyen los instrumentos que utilizan dichos clientes para realizar operaciones comerciales a través de los canales de pago.

Con el pasar de los años, estos elementos han ido a la par de la evolución de la electrónica y las tecnologías de la información obligando a las instituciones involucradas (industria bancaria, de adquirencia, procesadores, franquicias, entre otros) a innovar constantemente, no solo en los medios y canales de pago en sí, sino también en todos los aspectos involucrados en el funcionamiento de los mismos como seguridad, redes, dispositivos, entre otros. Este constante avance en todos los aspectos mencionados ha traído como consecuencia el surgimiento de nuevos medios y canales de pago, y esta innovación no parece detenerse en un futuro próximo.

Producto de este constante desarrollo en las tecnologías relacionadas con los pagos electrónicos es necesario que los sistemas de los entes e instituciones financieras sean capaces de adaptarse a nuevos esquemas que involucren medios y canales de pago con tecnologías emergentes y sean capaces de ofrecer servicios y soluciones multicanal que puedan crecer en el tiempo y acomodarse a las nuevas tendencias.

El objetivo de este trabajo consiste en analizar la implementación en producción de un canal de pago con la finalidad de implantar una nueva plataforma que soporte enrutamiento transaccional, enfocado a enlazar el punto de generación de la transacción con los actores financieros involucrados y se adapte a las necesidades cambiantes, así como el surgimiento de nuevos canales y medios de pago.

En este trabajo se presentan los conceptos esenciales de canales y medios de pago, tecnologías, protocolos y estándares de la industria bancaria necesarios para llevar a cabo los objetivos planteados. Así mismo se describe el proceso de desarrollo llevado a cabo para la implantación de la plataforma mencionada.

La estructura del trabajo se describe a continuación:

Capítulo I: Planteamiento del problema de investigación, donde se especifica el objetivo general y los objetivos específicos del Trabajo Especial de Grado, junto con el alcance, las preguntas de investigación y los resultados esperados.

Capítulo II: Marco teórico, el cual incluye la investigación de los conceptos relacionados a los medios y canales de pago, funcionamiento de la industria de tarjetas de pago, normas y regulaciones de seguridad, estándares para el formato de las transacciones.

Capítulo III: Marco metodológico, el cual aborda la técnica *Scrum*, describiendo sus roles, eventos y artefactos involucrados.

Capítulo IV: Desarrollo del Trabajo Especial de Grado, planteando el diseño conceptual, el desarrollo de la solución y los resultados obtenidos.

Finalmente, se ofrecen las conclusiones del Trabajo Especial de Grado y las referencias bibliográficas y digitales consultadas para su elaboración.

CAPÍTULO I

PROBLEMA DE INVESTIGACIÓN

1. Planteamiento del problema

Como punto de partida de este trabajo, tomamos el análisis de una implementación de canal de pago que se encuentra en producción, dicha solución consiste en una aplicación de punto de venta virtual (con tarjetas de pago), que envía transacciones e información del comercio a una institución de adquirencia. Entre la problemática que presenta dicha aplicación encontramos:

- Comprende un solo canal de pago (POS virtual)
- Poca modularidad en su arquitectura.
- Difícil de mantener, y al hacerlo se invierten grandes cantidades de tiempo de trabajo.
- Dificultad para incorporar nuevas operaciones y funciones.
- Adaptado al esquema de datos y negocio de una sola institución de adquirencia. En otras palabras se limita a una institución financiera.

Como consecuencia su implantación con otra institución de adquirencia o tercerización así como su adaptación a nuevos esquemas y avances tecnológicos se convierte en una tarea ardua y compleja por el alto componente ad-hoc.

La solución de POS Virtual está limitada a un solo canal de pago: El POS virtual con su dispositivo lector (*PinPad*). Por lo tanto, cada nuevo canal de pago que surja en el mercado se debe incorporar en forma independiente y sin reutilizar las reglas de negocio que ya están integradas en el VPOS.

Tanto la lógica de negocio como el formato de la mensajería de transacciones se encuentran embebidos dentro de la programación del VPOS y, en consecuencia, no se puede reutilizar para desarrollar en forma más rápida otros canales de pago que utilicen esa misma lógica y de esta manera acortar los tiempos de desarrollo y puesta en operación.

Actualmente la utilización de tecnologías emergentes en la telefonía celular, con sus teléfonos inteligentes (*smatphones*) ha cambiado la forma de pensar sobre los canales y los medios de pagos. En consecuencia las soluciones tecnológicas deben ser lo suficientemente modulares para soportar que las reglas de negocio puedan estar separadas de los canales utilizados para aplicarlas.

Es por lo anterior, que se vuelve necesario utilizar la experiencia adquirida con esta implementación de canal de pago para desarrollar una solución que sea lo suficientemente extensible, esto para permitir adaptar nuevos canales de pago haciendo uso de los conceptos de reutilización de código y modularidad.

La **problemática actual** de la empresa dueña de la solución VPOS se puede resumir en lo limitado de su solución para adaptarse a múltiples bancos adquirentes y a la imposibilidad de evolucionar hacia nuevos canales de pagos con tecnologías emergentes:

- Pago en línea haciendo uso de un botón de pago
- Soluciones de POS virtuales en dispositivos móviles
- Aplicaciones de pago con teléfonos u otros dispositivos inteligentes
- Cualquier canal de pago que surja de una tecnología futura...

La visión actual del mercado de medios pagos es poseer la habilidad de implantar múltiples canales de pagos y tener la capacidad de ser multibanco adquirentes. Las instituciones financieras buscan focalizar sus esfuerzos en aprovechar las tecnologías o tendencias emergentes para ofrecer productos y servicios de valor agregado para los clientes.

Sin embargo, actualmente en Venezuela existe una gran brecha en el mercado, tanto en lo referente a la masificación de los canales de pago que ya se encuentran en producción, como también en cuanto a la introducción de nuevas implementaciones. Una solución que posea la capacidad de recibir transacciones de múltiples canales, y a su vez, permitir la integración de los que surjan en el mercado sin la necesidad realizar una implementación desde cero, resulta en una ventaja competitiva. No obstante, de esta adaptabilidad, nace también la necesidad de que puedan asociarse múltiples adquirentes para consolidar en una sola plataforma todas las operaciones de los comercios. Es necesario destacar que en el contexto venezolano, no se encontró evidencia documentada de la existencia de una implementación como la descrita.

Desde la perspectiva del comercio, la posibilidad de conectarse a cualquier institución bancaria con múltiples canales de pago es altamente conveniente. Poder ofrecer mayores opciones de pago a sus clientes y obtener los beneficios de diversas instituciones bancarias son cualidades deseables.

De lo anterior se deriva la necesidad de adaptar las plataformas y sistemas involucrados en la cadena de pagos a esta heterogeneidad que existe en los medios y canales para poder incorporar elementos de innovación.

La unificación de varios métodos de pago en una misma plataforma se ha convertido en una opción ideal para los clientes que comercializan productos y servicios, esto les permite aprovechar al máximo la capacidad de compra del consumidor final. Esta demanda obliga a las instituciones financieras a ofrecer soluciones multicanal a los comerciantes, como se muestra en la figura 1, para que estos, a su vez, puedan ofrecer variedad de opciones de pago a sus clientes.

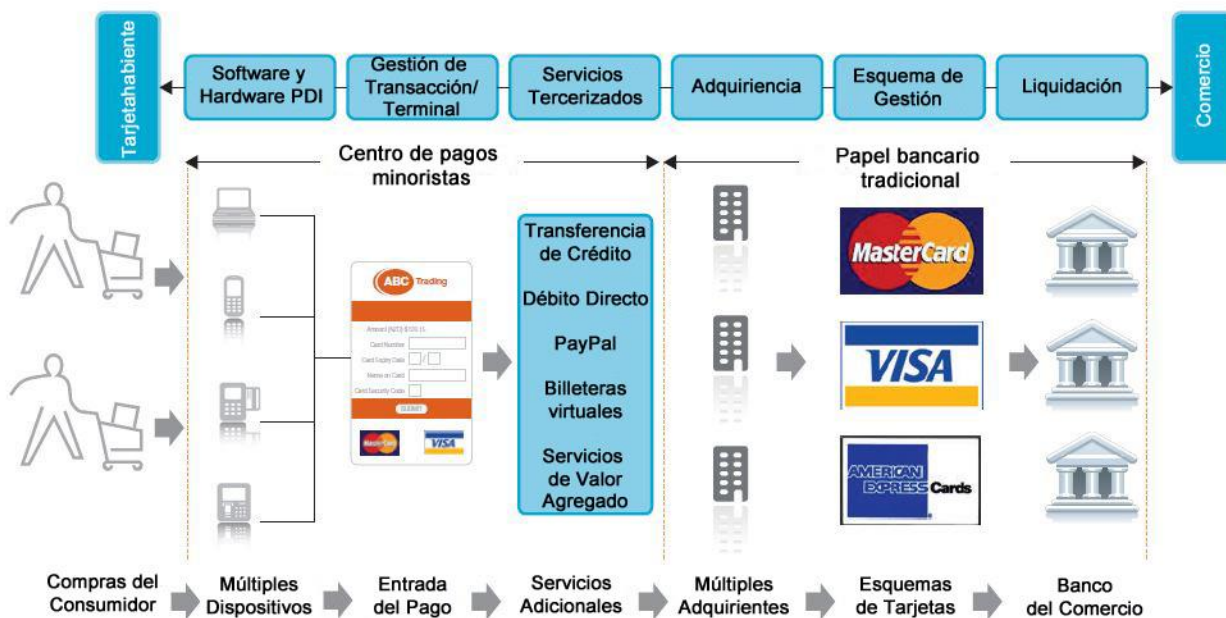


Figura 1. Múltiples canales como centros de pago de minoristas. (Capgemini's Strategic Analysis Group, 2012)

2. Objetivo General del Trabajo Especial de Grado

Diseñar y desarrollar una plataforma transaccional de medios de pagos con la capacidad de soportar diversos canales de pagos que puedan ser enrutados hacia múltiples instituciones financieras.

3. Objetivos Específicos del Trabajo Especial de Grado

- Estudiar todo el contexto teórico y conceptual necesario para llevar a cabo el diseño de una plataforma que reciba solicitudes de transacciones de múltiples canales de pago.
- Construir una plataforma transaccional que reciba solicitudes de transacciones con origen en diversos canales de pago.
- Desarrollar una plataforma transaccional que reciba solicitudes de transacciones con servicios web y los transforme a los formatos de la industria bancaria.
- Desarrollar una plataforma transaccional que reciba mensajes en los formatos de la industria bancaria y sea capaz de analizarlos para dar respuesta a solicitudes de transacciones mediante servicios web.

- Desarrollar una plataforma transaccional que permita establecer comunicación con los entes procesadores para el envío de las transacciones.
- Desarrollar una plataforma transaccional que proporcione funciones para la gestión de los datos de los componentes del sistema de pagos.
- Desarrollar una plataforma transaccional que se adapte a múltiples sistemas manejadores de bases de datos.
- Desarrollar una aplicación de botón de pago que funcione como el canal de pago con el cual se comprobarán las funciones de la plataforma.
- Diseñar una propuesta de arquitectura de alta disponibilidad para la futura implantación de los componentes de la plataforma.
- Realizar pruebas que permitan la verificación de la funcionalidad de la plataforma transaccional.
- Realizar pruebas que corroboren la integración entre la aplicación que fungirá como canal de pago y la plataforma de enrutamiento transaccional.

4. Alcance del Trabajo Especial de Grado

Este trabajo comprende el diseño e implantación de una solución que sustituya a la actual en términos de escalabilidad y buen funcionamiento.

Dicha solución abarca la construcción del software de dos puntos clave de la plataforma, que constan del servidor con soporte para múltiples canales de pago, que recibirá los datos transaccionales para enviarlos a la entidad bancaria utilizando el formato correspondiente y una implementación de canal de pago en la figura de botón de pago que recabe dichos datos.

El desarrollo consta de las funciones básicas de un proceso de pago llevado a cabo utilizando el servidor transaccional y la aplicación de botón de pago.

En futuros desarrollos se dará pie a la migración de la solución VPOS al nuevo esquema de funcionamiento con el servidor transaccional.

CAPÍTULO II

MARCO TEÓRICO CONCEPTUAL

1. Pagos y sistemas de pago

1.1. Sistemas de pago

En todas las economías, un gran número de transacciones tienen lugar cada día, motivadas por una amplia gama de actores económicos. Todas las transacciones, ya se trate de la adquisición de bienes, activos financieros o servicios (y siempre que no impliquen trueque), tienen dos componentes de liquidación: (i) la entrega del bien o servicio; y (ii) la transferencia de fondos - es decir, pago con dinero en efectivo (billetes y monedas) o depósitos mantenidos en bancos (fondos en cuentas abiertas en los bancos). Por tanto, un pago es una transferencia de fondos que descarga una obligación por parte de un pagador a un beneficiario. Un pagador es la parte en la operación de pago que emite la orden de pago o se compromete a la transferencia de fondos al beneficiario. Un beneficiario es el destinatario final de los fondos.

1.2. Ciclo de vida de un pago

Un ciclo de vida esquematizado de un pago no en efectivo (por ejemplo, una transferencia de fondos) podría ser el siguiente:

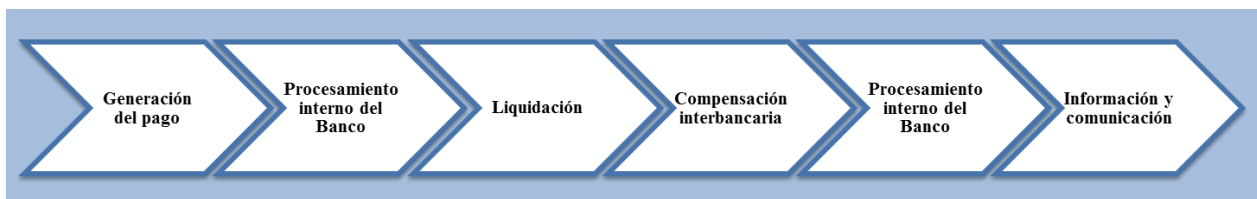


Figura 2. Ciclo de vida simplificado de un pago no en efectivo. (European Central Bank, 2010)

1.3. Tipos de pago

Los pagos se pueden clasificar en función de los diferentes tipos de pagador/beneficiario involucrados.

1. Los pagos al por mayor son los pagos entre las instituciones financieras. Ellos tienden a tener un valor alto.
2. Los pagos entre las instituciones no financieras (por ejemplo, los hogares particulares, las sociedades no financieras o agencias gubernamentales) son normalmente clasificados como pagos al por menor.

Los pagos también se pueden agrupar en función del número de contribuyentes y beneficiarios involucrados en una transacción en particular.

1. En una transacción de uno-a-uno, un ordenante transfiere fondos a un beneficiario.
2. En transacciones de uno-a-muchos, un ordenante realiza transferencias de fondos a varios beneficiarios con una sola instrucción.
3. En transacciones de muchos-a-uno, varios pagadores transfieren fondos a un solo beneficiario, usualmente a iniciativa del beneficiario.

1.4. Medios de pago

Un medio de pago es una herramienta o un conjunto de procedimientos que permiten la transferencia de fondos del ordenante al beneficiario.

Los pagos en efectivo se asocian generalmente con transacciones cara a cara de bajo valor entre individuos o entre un individuo y un comerciante.

En contraste, los pagos no en efectivo implican la transferencia de fondos entre cuentas.

Los instrumentos de pago distintos al efectivo pueden clasificarse sobre la base de lo siguiente.

- Forma física (en papel o instrumentos electrónicos)
- La parte que presenta el instrumento de pago para su procesamiento (instrumentos de crédito o débito)

El dinero electrónico es un valor monetario representado como un crédito sobre el emisor, se almacena en un dispositivo electrónico y es aceptado como medio de pago por empresas distintas al emisor (en contraste con los instrumentos prepago simples, donde el emisor y el aceptante son el mismo).

Los instrumentos de pago distintos de efectivo más comunes son:

A. Instrumentos de propósito general

Transferencias de crédito:

Las transferencias de crédito, también llamados "créditos directos", son instrucciones enviadas por un ordenante a su banco solicitando que una cantidad definida de los fondos se transferirá a la cuenta de un beneficiario.

Los débitos directos:

Los débitos directos son los instrumentos de pago que autorizan el débito de la cuenta bancaria del pagador. Los débitos directos en general se solicitan y procesan de manera electrónica.

Tarjetas de pago:

Las tarjetas son dispositivos de acceso que pueden ser utilizados por sus titulares para pagar bienes y servicios - ya sea en un punto de venta (*POS*) o de forma remota (en las transacciones "de tarjeta no presente") - o para retirar dinero en cajeros automáticos (*ATMs*).

Las tarjetas de pago de uso general más comunes son las tarjetas de débito, tarjetas de crédito y las tarjetas de débito diferido.

Las tarjetas de débito están vinculadas a una cuenta bancaria y permiten a los tarjetahabientes cargar compras o retiros en cajeros automáticos directa e individualmente a esta cuenta.

Las tarjetas de crédito ofrecen a los titulares de tarjetas una línea de crédito y la posibilidad de retrasar el pago.

Las tarjetas de débito diferido permiten al titular de la tarjeta aplazar el pago, pero el importe pendiente debe ser liquidado por completo al final de un período determinado.

Cheques:

Un cheque es una orden escrita de una parte (librador) a otra (librado) que requiere que el librado pague una suma especificada por el librado a un tercero indicado por el mismo.

Como es un instrumento basado en el papel, los cheques son el instrumento de pago no en efectivo más costoso de tramitar y resolver.

B. Instrumentos para usos especiales

Un giro es un producto de pago basado en transferencias de crédito y se utiliza para transferir dinero de forma remota.

Los cheques de viajero son productos pre pagados basados en papel y emitidos en denominaciones específicas para uso general en los negocios y viajes personales.

Un giro bancario (también llamado un "cheque de caja" o un "cheque de cajero") es un cheque emitido por un banco en sí mismo.

Una carta de crédito es una promesa por un banco u otro emisor de hacer un pago a un tercero por cuenta de un cliente de acuerdo con las condiciones especificadas.

1.5. Canales de pago

Se refieren a todas las formas a través de las cuales se pueden tomar pagos. Entre los canales de pago podemos encontrar los siguientes

- Punto de venta (POS): los terminales de tarjetas son exclusivamente para transacciones con el titular presente. Estos terminales incluyen sistemas de entrada de datos manuales y para deslizar las tarjetas.
- Punto de venta virtual (POS virtual): similar al anterior, con la diferencia de que la lógica para la realización de la transacción y la comunicación con la entidad bancaria se encuentran en una aplicación instalada en una pc.
- Botón de pago online: canal de pago utilizado para el procesamiento de transacciones con tarjeta no presente a través de una aplicación web utilizando Internet.
- Punto de venta móvil: utilizar una aplicación para tomar pagos a través de dispositivos móviles.
- Otros canales de pago con nuevas tecnologías.

2. La industria de las tarjetas de pago

2.1. Estado del arte

La industria de tarjetas de pago ha mostrado un crecimiento constante en los últimos años, ya que las transacciones no monetarias aumentan continuamente y las tarjetas siguen siendo el método de pago líder en términos de mayor cuota de mercado. Con la creciente penetración del comercio en línea y móvil, las tarjetas están siendo utilizadas cada vez más para transacciones tanto de bajo como de alto valor.

La industria de los adquirentes de pago es cada vez más competitiva, lo que está afectando los márgenes de cada actor individual. Como resultado, los adquirentes están tratando de preservar estos márgenes, ofreciendo servicios de valor añadido. Además de la competencia de precios, lo que ejerce presión sobre los adquirentes es que la creciente preocupación por las violaciones de datos también está creando un impacto y ha dado lugar a mayores requisitos de cumplimiento para ambos compradores y comerciantes.

La industria también está siendo testigo de un aumento constante de los pagos móviles, a medida que más y más consumidores están utilizando sus dispositivos móviles para pagar sus compras. Esto ha llevado a exigir la creación de soluciones que permiten la aceptación de este tipo de pagos. Empresas como Square®, Isis Mobile Wallet™ y PayPal™ están ofreciendo soluciones nuevas e innovadoras para atender esta demanda.

2.2. Comportamiento global de la industria de tarjetas de pago

El uso de tarjetas ha estado creciendo desde la crisis financiera que golpeó al mundo en 2008-2009. Durante 2011, el volumen de transacciones de tarjetas creció un 13,5%, que es

considerablemente más que la tasa de crecimiento del 9,2% registrado en 2010, debido en gran parte al aumento de la utilización de tarjetas de débito en todo el mundo. El porcentaje de participación de las tarjetas en el conjunto de métodos de pagos distintos del efectivo ha ido creciendo constantemente y fue del 55,8% en 2010. En 2011, las tarjetas también se mantuvieron como el instrumento de pago no en efectivo preferido a nivel mundial, con el 58,8% de los pagos no monetarios globales realizados a través de tarjetas. La creciente participación de las tarjetas en el conjunto de métodos de pago distintos del efectivo indica una creciente preferencia por las tarjetas en comparación con otros instrumentos de pago, tales como débito directo, transferencias de crédito y cheques.¹

En los países desarrollados, el volumen de transacciones que involucran tarjetas es muy alto, ya que los consumidores prefieren usar tarjetas incluso para transacciones de bajo valor. Sin embargo, los países desarrollados no han sido capaces de igualar el crecimiento exhibido por el mundo emergente. El crecimiento en el volumen de transacciones de tarjetas en Europa fue de un solo dígito en el 8,6% en 2011 y en los EE.UU. fue de 11,7%. Este crecimiento es apenas comparable a las tasas de crecimiento exhibidas por América Latina, que vio un crecimiento del 21,0% y Asia-Pacífico, con una tasa de crecimiento del 20% en 2011. Tanto América Latina y Asia-Pacífico han mostrado un enorme crecimiento durante los últimos años.

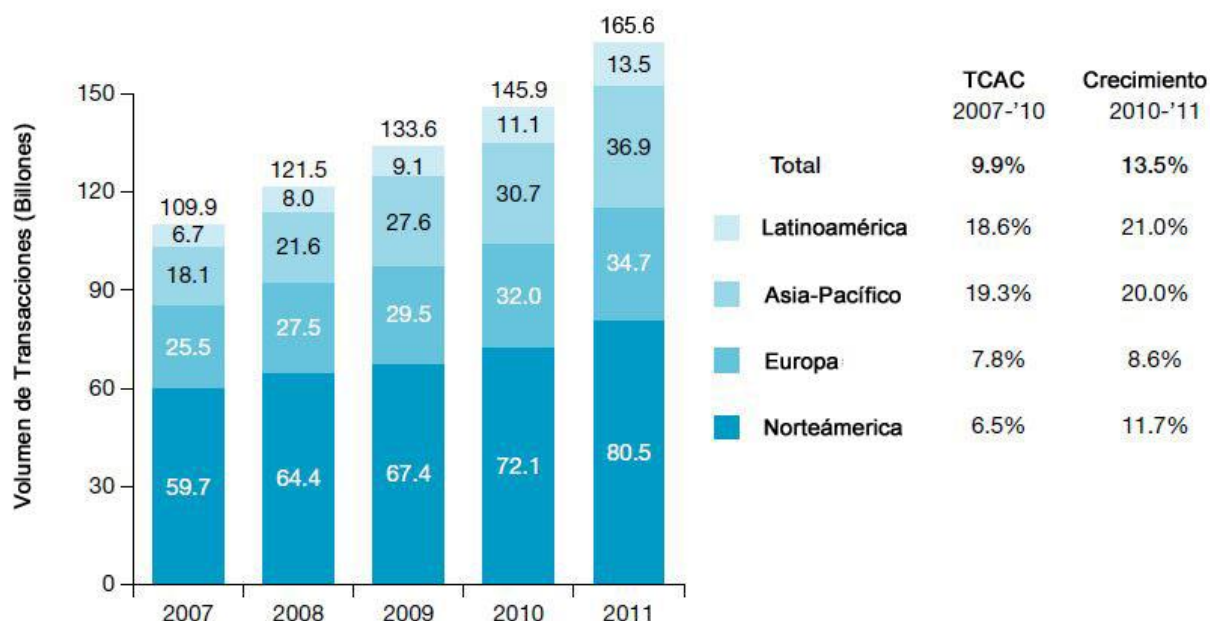


Figura 3. Volumen global de transacciones con tarjetas (billones) por región, 2007-11. (Gupta & Roy, 2014).

Los países emergentes tienen un alto potencial de crecimiento y el uso de tarjetas es probable que aumente en los próximos años como la preferencia del consumidor a cambiar de efectivo al plástico.

¹ Cifras dadas a conocer en 2012 por Statistical Data Warehouse

2.3. Participantes claves en la industria de las tarjetas de pago

Una transacción de tarjeta simple entre un titular y un comerciante implica varios partícipes que forman parte de redes complejas:

- **Adquirentes de tarjetas:** los bancos de los comerciantes.
- **Procesadores de tarjetas:** las organizaciones de terceros que realizan los procesos de autorización de tarjetas y liquidación.
- **Los emisores de tarjetas:** el banco del titular que emite la tarjeta y mantiene la cuenta del cliente.
- **Proveedores de la red de asociación de tarjetas:** normalmente las franquicias como MasterCard o Visa juegan un papel esencial en la realización del ciclo de autorización de la tarjeta y la liquidación, como se ilustra a continuación.

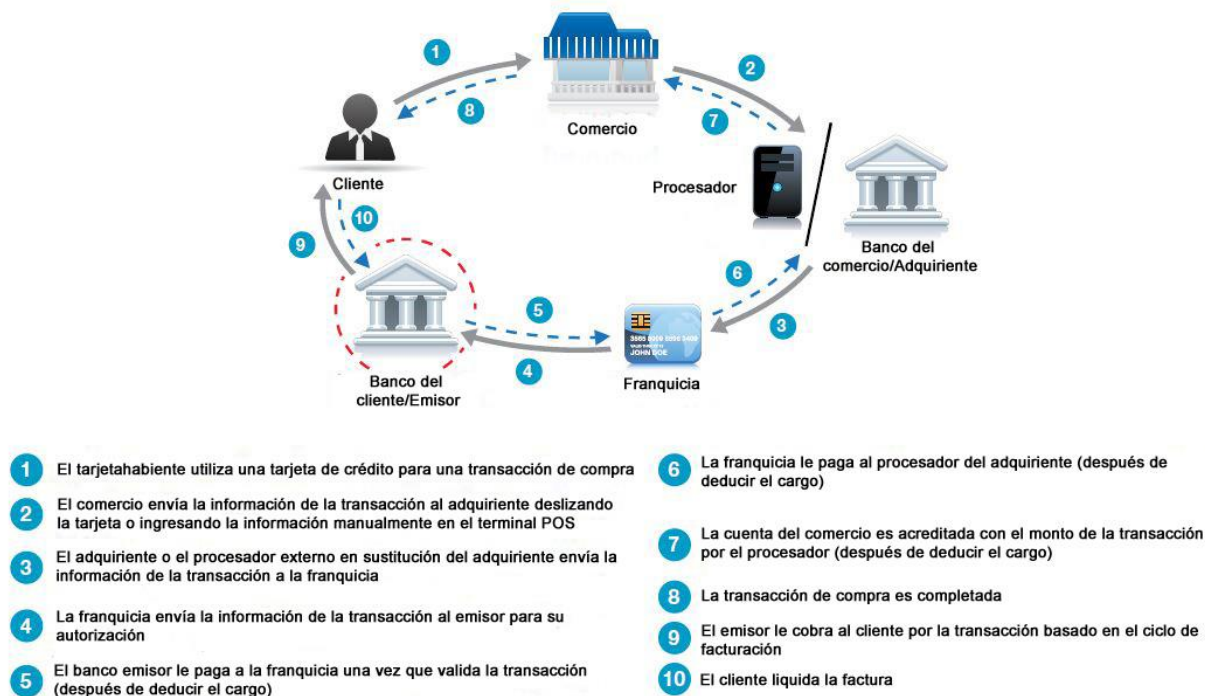


Figura 4. Estructura típica de un flujo de transacción con tarjeta. (Gupta & Roy, 2014).

2.4. Tendencias en la industria de las tarjetas de pago desde la perspectiva del procesador

2.4.1. Mayor enfoque en tecnologías contra el fraude en el comercio electrónico

En una transacción en línea, es responsabilidad del comerciante de comercio electrónico confirmar la identidad del comprador y para hacer frente a los conflictos asociados a compras no autorizadas. Como resultado, en la mayoría de los casos, los comerciantes tienen que soportar las pérdidas asociadas con el fraude.

Muchos minoristas en línea siguen utilizando métodos básicos como la verificación de la dirección, código de verificación de la tarjeta, y la autorización de la franquicia de la tarjeta para comprobar la legitimidad de la compra. Estos métodos de verificación básicos pueden no ser suficientes para gestionar con éxito el fraude ya que los ladrones cibernéticos están utilizando técnicas cada vez más sofisticadas. De acuerdo con la investigación llevada a cabo por LexisNexis en 2011, los comerciantes no han estado usando todas las últimas herramientas de prevención del fraude.

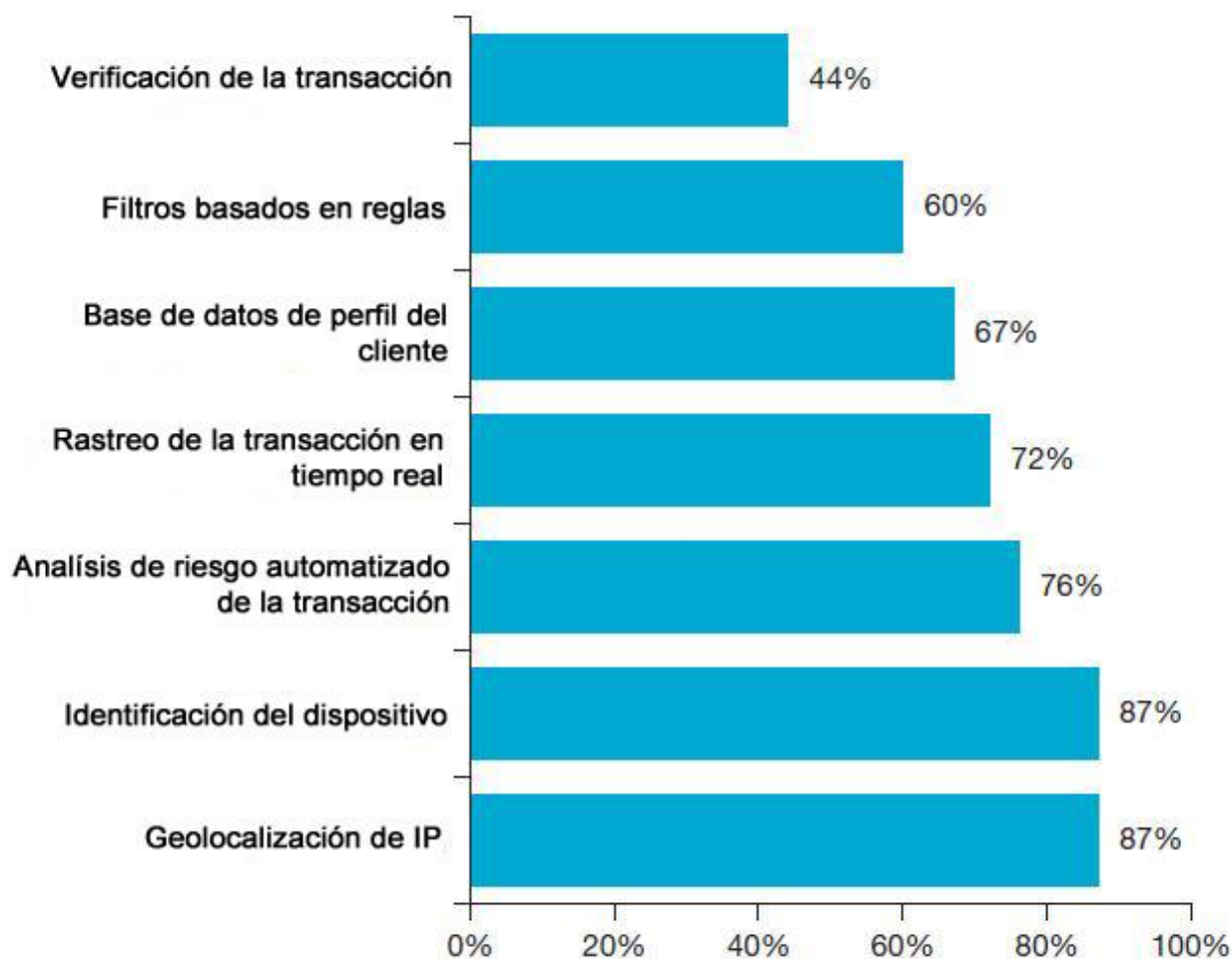


Figura 5. Comerciantes que no utilizan herramientas de detección de fraude en línea. (Capgemini's Strategic Analysis Group, 2012).

Como resultado de ello, el papel de un procesador tarjetas como un socio calificado de comerciantes en el desarrollo e implementación de soluciones sofisticadas de prevención y detección de fraude automatizado se ha convertido en algo crucial. Los procesadores también deben proporcionar capacitación a la comunidad de comerciantes de comercio electrónico en las últimas técnicas de fraude y las mejores prácticas para comprobar los riesgos.

2.4.2. Desarrollo de herramientas para ofrecer servicios especializados

Los procesadores de tarjetas desempeñan un papel importante en la industria de tarjetas de pago y tienen la capacidad de afectar todas las partes interesadas incluyendo clientes, comerciantes, instituciones de servicios financieros, emisores y adquirentes.

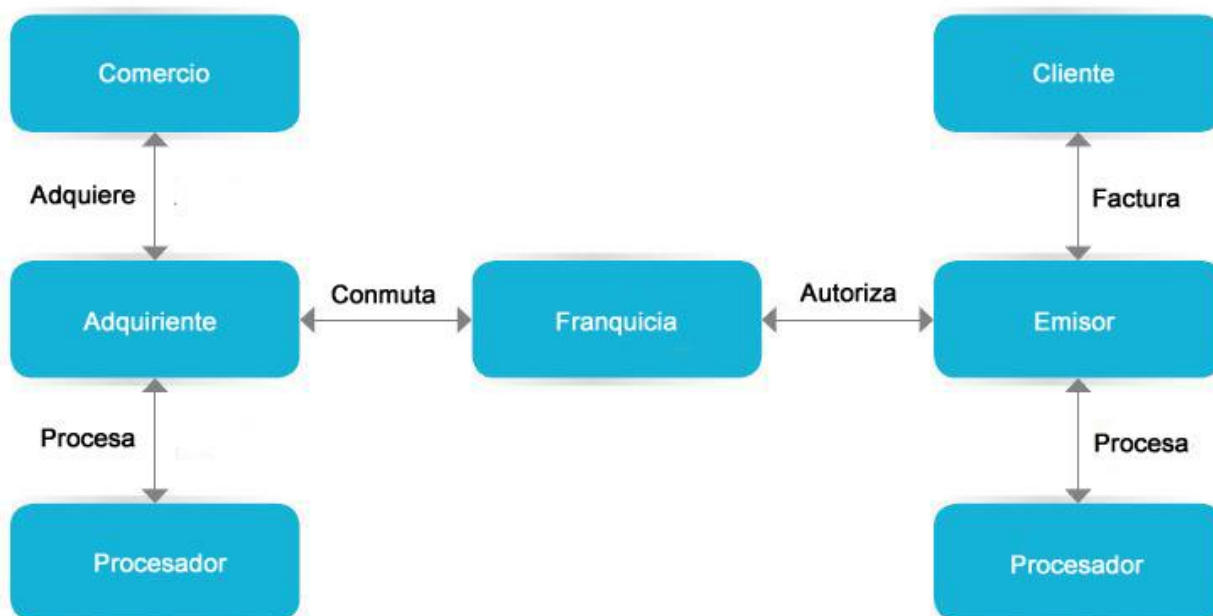


Figura 6. Ciclo de autorización de la tarjeta con los grupos de interés. (Capgemini's Strategic Analysis Group, 2012).

Los procesadores de tarjetas están ayudando a diferentes grupos de interés, centrándose en las siguientes áreas:

- **Fraude y gestión de riesgos:** con el alto crecimiento en el comercio en línea y móvil y el aumento de la sofisticación de los delincuentes cibernéticos, los procesadores están desarrollando y ofreciendo servicios de gestión de fraude a los comerciantes para combatir esta creciente amenaza.
- **Datos de business analytics:** los procesadores pueden aprovechar la enorme cantidad de información valiosa del cliente que tienen en su poder a partir de su negocio de procesamiento de pagos. Esta información puede ser utilizada por los emisores, adquirentes y comerciantes para mejorar sus productos y servicios.
- **Servicios de programa de fidelidad:** los procesadores están ayudando a los comerciantes y emisores a ofrecer programas de fidelidad más específicos. Estos tipos de programas pueden dar lugar a un mayor uso de la tarjeta, captación de clientes, mejora de la retención y el fortalecimiento de las relaciones entre emisores y consumidores y socios comerciales.
- **Experiencia de usuario fluida entre formas de pago:** los procesadores están buscando establecer vínculos con las partes interesadas para ofrecer una experiencia integrada y

personalizada de los consumidores en el comercio móvil, ir de compras de comercio electrónico, y en la tienda.

2.4.3. Innovaciones tecnológicas en procesamiento de pagos

Los procesadores de pago han construido la infraestructura para procesar pagos a través del uso de nuevas tecnologías para las operaciones, que muchos consumidores han estado utilizando para otros fines:

- NFC para intercambiar datos entre sí
- Los códigos QR para acceder rápidamente a determinados enlaces
- Detalles biométricos para fines de identificación, como en los controles de inmigración

Near Field Communication

A través de la tecnología NFC, dos dispositivos similares pueden establecer una comunicación por radio entre sí cuando se juntan en las proximidades. Esta tecnología ha sido adaptada para el mercado de pagos y se está aplicando cada vez más en una escala global. Con el fin de realizar una transacción, el consumidor sólo tiene que llevar un teléfono inteligente con NFC en estrecha proximidad con el dispositivo de pago NFC en el terminal punto de venta, y luego introducir el pasador de autenticación en el teléfono. Para realizar los pagos NFC, el usuario simplemente tiene que vincular la cuenta bancaria o tarjeta de crédito/débito con aplicaciones que soporten esta tecnología.

Aunque la infraestructura NFC está aumentando a un ritmo rápido, los pagos NFC sólo tienen una cuota pequeña en el volumen total de los pagos móviles. En 2013 se estimaba que 245,2 millones de personas habrían hecho pagos móviles por un total de 235.4 mil millones de dólares (USD) con un 2% de las transacciones por NFC. Para el 2017, se estima que 450 millones de usuarios a nivel mundial hagan pagos móviles por un monto estimado de 721 mil millones dólares (USD), con un 5% estimado de cuota para NFC.

Código de Respuesta Rápida

El código de respuesta rápida o código QR, para abreviar, se utilizó por primera vez por la industria del automóvil en Japón. Hoy se ha puesto en práctica con éxito en la industria global de pagos, también. Funciona así: el consumidor vincula su cuenta bancaria o tarjeta de débito/crédito con un procesador de pagos en línea equipado para llevar a cabo transacciones de códigos QR. Al escanear el código QR a través de un teléfono o tableta, la información bancaria del usuario es desbloqueada y la transacción se puede completar después de las autorizaciones apropiadas. Debido a que es fácil de generar un código QR, el sistema ofrece comodidad a las empresas y los consumidores, por igual. Se

puede imprimir en tarjetas de visita, puntos de venta, y etiquetas de los productos que los clientes pueden escanear a pagar por un producto o servicio.

Pagos biométricos

Dado que, idealmente, los pagos biométricos requieren un pulgar o huella digital para autorizar una transacción, una vez que se desliza una tarjeta, se percibe como muy segura. PayTouch, con sede en España, ha ido un paso más allá mediante la vinculación de las tarjetas de crédito o de débito a dos dedos del usuario en su servidor PayTouch. Para realizar un pago, el usuario simplemente presiona dos dedos en una almohadilla PayTouch. Cuando Saral Money se puso en marcha en la India, se comenzó a utilizar la base de datos Adhaar, que incluye información biométrica de la población de la India, junto con la tecnología de TSYS para autenticar al usuario mediante la confirmación de las huellas dactilares recogidas en el POS. Hay alcance en esta tecnología incluso para incorporar otras características biométricas, tales como escaneos de retina y reconocimiento de voz.

2.4.4. Enfoque en prestar servicios de pago multicanal a los comerciantes

Impulsado por un alto crecimiento en las economías en desarrollo, las ventas globales de comercio electrónico han tenido crecimiento constante en los últimos años. Los clientes que utilizan los teléfonos móviles y tabletas para pagos están aumentando a un ritmo rápido, obligando a los comerciantes a ofrecer opciones de pago multicanal.

Las necesidades de pago de los comerciantes son cada vez más sofisticadas. Por lo tanto los adquirentes se están centrando en la entrega de soluciones de pago que cumplan con los estándares PCI y les permitan mantenerse al día con el cambio de la dinámica del mercado.

3. Normas de Seguridad de Datos de la Industria de Tarjetas de Pago (PCI)

3.1. Visión general

Las Normas de Seguridad de Datos de la Industria de Tarjetas de Pago (PCI DSS) se desarrollaron para fomentar y mejorar la seguridad de los datos del titular de la tarjeta y facilitar la adopción de medidas de seguridad uniformes a nivel mundial. Las PCI DSS proporcionan una referencia de requisitos técnicos y operativos desarrollados para proteger los datos de los titulares de tarjetas. Las PCI DSS se aplican a todas las entidades que participan en el procesamiento de tarjetas de pago, entre las que se incluyen comerciantes, procesadores, adquirentes, entidades emisoras y proveedores de servicios, como también todas las demás entidades que almacenan, procesan o transmiten CHD (datos del titular de la tarjeta) o SAD (datos de autenticación confidenciales).

3.2. Alcance de los requerimientos PCI DSS

Los requisitos de seguridad de las PCI DSS se aplican a todos los componentes del sistema incluidos en el entorno de datos del titular de la tarjeta o conectados a este. El CDE (entorno de datos

del titular de la tarjeta) consta de personas, procesos y tecnologías que almacenan, procesan o transmiten datos de titulares de tarjetas o datos confidenciales de autenticación. El término "componentes del sistema" incluye dispositivos de red, servidores, dispositivos informáticos y aplicaciones. Los componentes del sistema incluyen, a modo de ejemplo:

- Sistemas que ofrecen servicios de seguridad (por ejemplo, servidores de autenticación), que facilitan la segmentación (por ejemplo, firewalls internos) o que pueden afectar la seguridad del CDE (por ejemplo, servidores de resolución de nombres o de redireccionamiento web).
- Componentes de virtualización, como máquinas virtuales, interruptores/routers virtuales, dispositivos virtuales, aplicaciones/escritorios virtuales e hipervisores.
- Los componentes de red incluyen, a modo de ejemplo, firewalls, interruptores, routers, puntos de acceso inalámbricos, aplicaciones de red y otras aplicaciones de seguridad.
- Los tipos de servidores incluyen, a modo de ejemplo: web, aplicación, bases de datos, autenticación, correo electrónico, proxy, NTP (protocolo de tiempo de red) y DNS (servidor de nombre de dominio).
- Aplicaciones, que abarcan todas las aplicaciones compradas y personalizadas, incluso las aplicaciones internas y externas (por ejemplo, Internet).
- Cualquier otro componente o dispositivo ubicado en el CDE o conectado a este.

3.3. Requerimientos y procedimientos PCI DSS

3.3.1. Desarrollar y mantener sistemas y redes seguros

- **Requisito 1:** Instalar y mantener una configuración de firewalls para proteger los datos de los titulares de las tarjetas
- **Requisito 2:** No utilizar contraseñas de sistemas y otros parámetros de seguridad provistos por los proveedores

3.3.2. Proteger los datos del titular de la tarjeta

- **Requisito 3:** Proteger los datos del titular de la tarjeta que fueron almacenados
- **Requisito 4:** Cifrar la transmisión de los datos del titular de la tarjeta en las redes públicas abiertas.

3.3.3. Mantener un programa de administración de vulnerabilidad

- **Requisito 5:** Proteger todos los sistemas contra malware y actualizar los programas o software antivirus regularmente.

- **Requisito 6:** Desarrollar y mantener sistemas y aplicaciones seguras

3.3.4. Implementar medidas sólidas de control de acceso

- **Requisito 7:** Restringir el acceso a los datos del titular de la tarjeta según la necesidad de saber que tenga la empresa.
- **Requisito 8:** Identificar y autenticar el acceso a los componentes del sistema.
- **Requisito 9:** Restringir el acceso físico a los datos del titular de la tarjeta

3.3.5. Supervisar y evaluar las redes con regularidad

- **Requisito 10:** Rastrear y supervisar todos los accesos a los recursos de red y a los datos de los titulares de las tarjetas
- **Requisito 11:** Probar con regularidad los sistemas y procesos de seguridad.

3.3.6. Mantener una política de seguridad de información

- **Requisito 12:** Mantener una política que aborde la seguridad de la información de todo el personal

4. Estándar ISO 8583

4.1. Descripción

Una transacción basada en una tarjeta usualmente sale desde un dispositivo de compra, tal como un POS o un cajero automático ATM, a través de una red (o redes) hacia un sistema del emisor de la tarjeta para obtener una autorización en función de la cuenta del titular de la tarjeta. La transacción contiene información que se obtiene de la tarjeta (ej. número de cuenta), la terminal (ej. nro. de comercio), la transacción (ej. importe), entre otros. El sistema emisor de la tarjeta podrá autorizar o rechazar la transacción, y genera un mensaje de respuesta que debe ser devuelto a la terminal en un tiempo breve.

ISO 8583 define un formato de mensaje y un flujo de comunicación para que diferentes sistemas puedan intercambiar estas transacciones. Todas las redes de tarjetas basan sus transacciones en el estándar ISO 8583. Las transacciones incluyen compras, extracciones, depósitos, reintegros, reversos, consultas de saldo, pagos y transferencias entre cuentas. ISO 8583 también define mensajes entre sistemas para intercambios seguros de claves, conciliación de totales y otros propósitos administrativos. Aunque el ISO 8583 define un estándar común, no se usa normalmente en forma directa por sistemas o redes. En lugar de eso cada red adapta el estándar para su propio uso con campos adaptados a sus necesidades particulares. Los acuerdos sobre especificaciones de aplicación son generalmente a nivel privado. Esta norma internacional está diseñada como una especificación de

interfaz que permite el intercambio de mensajes entre los sistemas adoptando la variedad de especificaciones entre las aplicaciones.

4.2. Formato del mensaje de intercambio

Un mensaje ISO 8583 consta de las siguientes partes:

- Message Type Indicator (MTI) - Indicador de Tipo de Mensaje
- Uno o más bitmaps, indicando que elementos están presentes en el mensaje
- Data elements, los campos del mensaje

Tabla 1. Componentes principales de un mensaje en formato ISO 8583.

Message Type Identifier (MTI)	Bitmap	Campos del Mensaje
Campo numérico de 8 dígitos, describe cada clase de mensaje y su función.	Indica que otros elementos (campos o subcampos) se encuentran en el mensaje. Un mensaje contendrá al menos un mapa de bits, llamado el Mapa de Bits Primario que indica que campos (Data Elements) del 1 al 64 están presentes.	Los Data Elements son los campos individuales que llevan la información sustancial acerca de la transacción. Hay 128 campos definidos en el estándar ISO8583:1987.

4.2.1. Indicador de Tipo de Mensaje (Message Type Indicator):

Este es un campo numérico de 4 dígitos que clasifica la función de alto nivel del mensaje. Un MTI incluye la versión ISO 8583, la clase (Message Class), la función (Message Function) y el origen del mensaje (Message Origin). Los sub campos son descritos a continuación:

Tabla 2. Subcampos del Message Type Identifier de un mensaje con formato ISO 8583.

Message Type Identifier(MTI)			
Versión ISO 8583	Clase de mensaje	Función del mensaje	Origen
Especifica la versión del standard ISO 8583 que se está usando en este mensaje	Especifica el propósito general del mensaje	Define como se procesará el mensaje dentro del sistema	Define la ubicación de la fuente del mensaje dentro de la cadena de pago

Tabla 3. Significado de los valores de los subcampos del Message Type Identifier de un mensaje con formato ISO 8583.

Message Type Identifier(MTI)							
Versión ISO 8583		Clase de mensaje		Función del mensaje		Origen	
Valor	Significado	Valor	Significado	Valor	Significado	Valor	Significado
0	Versión ISO 8583-1:1987	1	Autorización: Determina si existen fondos disponibles, obtiene una aprobación pero no se toma en cuenta para la conciliación, Dual Message System (DMS), espera el intercambio de archivos para imputar la cuenta.	0	Requerimiento (Request)	0	Comprador
1	Versión ISO 8583-2:1993	2	Financiero: Determina si existen fondos disponibles, obtiene una aprobación e imputa directamente a la cuenta, Single Message System (SMS), no se intercambian archivos.	1	Respuesta al Requerimiento (Request Response)	1	Comprador Repetición
2	Versión ISO 8583-1:2003	3	Manejo de Archivos: Usado por hot-card, TMS y otros cambios.	2	Aviso (Advice)	2	Emisor
9	Uso privado	4	Reverso: Reversa la acción de una autorización previa.	3	Respuesta al Aviso (Advice Response)	3	Emisor Repetición
		5	Conciliación: Transmite información de cierre.	4	Notificación	4	Otros
		6	Administrativo: Transmite información de falla en los mensajes.	8	Reconocimiento de respuesta (Response acknowledgment)	5	Otros Repetición
		7	Free Collection	9	No Reconocimiento (Negative acknowledgment)		
		8	Manejo de Red: Usado para intercambio seguro de claves, logon, echo test y otras funciones de red				
		9	Reservado por la ISO				

Algunos valores de MTI estándar son:

Tabla 4. Valores estándar del campo Message Type Indicator en un mensaje con formato ISO 8583.

MTI	Significado
0100	Requerimiento de autorización

MTI	Significado
0120	Aviso de Autorización
0121	Aviso de Autorización Repetición
0200	Requerimiento Financiero del Comprador
0220	Aviso Financiero del Comprador
0221	Aviso Financiero del Comprador Repetición
0400	Requerimiento de Reverso del Comprador
0420	Aviso de Reverso del Comprador
0421	Aviso de Reverso del Comprador Repetición
0800	Requerimiento de Manejo de Red
0820	Aviso de Manejo de Red

4.2.2. Mapa de Bits (Bitmap)

Dentro del ISO 8583, un mapa de bit es un campo o sub campo dentro de un mensaje que indica que otros elementos (campos o sub campos) se encuentran en el mensaje. Un mensaje contendrá al menos un mapa de bits, llamado el Mapa de Bits Primario que indica que campos (Data Elements) del 1 al 64 están presentes. Puede existir un mapa de bits secundario, generalmente como elemento 1 que indica que campos del 65 al 128 están presentes. De igual forma, un tercer bitmap puede usarse para indicar la presencia o ausencia de los campos del 129 al 192, aunque esos campos casi nunca se usan. El mapa de bits se puede transmitir como un dato binario de 8 bytes, o como un campo de 16 caracteres hexadecimales 0-9, A-F en el set de caracteres ASCII o EBCDIC.

Un campo está presente cuando el bit correspondiente está en true, ej. El byte 42x en binario es '0100 0010' lo que significa que los campos 2 y 7 están presentes en este mensaje.

4.2.3. Campos del mensaje (Data Elements)

Los Data Elements son los campos individuales que llevan la información sustancial acerca de la transacción. Hay 128 campos definidos en el estándar ISO8583:1987, y 192 en posteriores versiones. La revisión de 1993 agregó nuevas definiciones, eliminó algunas pero sin embargo dejó el formato del mensaje sin cambios. Mientras que cada Data Element tiene un significado y formato específico, el estándar también incluye algunos campos de propósito general y algunos especiales para sistemas o países, los cuales varían sustancialmente en su forma y uso de una implementación a otra. Cada campo se describe en un formato estándar que define el contenido permitido del campo (numérico, binario, etc.) y el largo del campo (variable o fijo), de acuerdo a la siguiente tabla:

Tabla 5. Tipos de datos de un Data Element en un mensaje con formato ISO 8583.

Abreviatura	Significado
a	Alfanumérico, incluyendo los espacios
n	Sólo valores numéricos
s	Sólo caracteres especiales
an	Alfanumérico
as	Sólo caracteres alfanuméricos y especiales
ns	Sólo caracteres numéricos y especiales

ans	Caracteres alfabéticos, numéricos y especiales
b	Información binaria
z	Tracks 2 y 3 code set como se define en la ISO 4909 y en ISO 7813.

Además, cada campo puede tener largo fijo o variable. Si es variable, el largo del campo será precedido por un indicador de largo.

Tabla 6. Formatos de longitud de un Data Element en un mensaje con formato ISO 8583.

Tipo	Significado
Fixed	Largo Fijo
LLVAR o (..xx)	Donde xx < 100, significa que los dos primeros digitos indican el largo del campo
LLLVAR o (...xxx)	Donde xxx < 1000, significa que los tres primeros digitos indican el largo del campo
Un campo LLVAR o LLLVAR puede ser comprimido o ASCII dependiendo del formato del mensaje que puede ser ASCII o Comprimido.	Por ejemplo un campo LLVAR puede tener 1 o 2 bytes, si está comprimido el hexa '23x significa que hay 23 elementos, si es ascii, bytes '32x, '31x significa que hay 21 elementos. Un elemento depende del tipo de dato, si es numérico este estará comprimido, ej. largo 87 se representará por un byte '87x, si es ASCII serán dos bytes '38x y '37x. Los campos LLLVAR usan 2 o 3 bytes (dependiendo del tipo de mensaje) con un '0' adelante si es comprimido.

Campos del mensaje definidos por el estándar ISO

Tabla 7. Tipos de dato y longitud de los Data Element definidos por el estándar ISO.

Data element	Tipo	Uso
1	b 64	Bit Map extendido
2	n ..19	Número de cuenta primario
3	n 6	Código de procesamiento
4	n 12	Monto, transacción
5	n 12	Monto, liquidación
6	n 12	Monto, facturación titular de la tarjeta
7	n 10	Hora y fecha de la transmisión
8	n 8	Monto, tarifa facturación titular de la tarjeta
9	n 8	Tasa de conversión, liquidación
10	n 8	Tasa de conversión, facturación titular de la tarjeta
11	n 6	Número de rastreo de sistemas de auditoría
12	n 6	Hora, transacción local
13	n 4	Fecha, transacción local
14	n 4	Fecha, expiración
15	n 4	Fecha, Liquidación
16	n 4	Fecha, conversión
17	n 4	Fecha, captura

Data element	Tipo	Uso
18	n 4	Tipo de comercio
19	n 3	Código de país de la institución adquirente
20	n 3	Número de cuenta primario
21	n 3	Institución de reenvío, código de país
22	n 3	Modo de entrada del POS
23	n 3	Número de aplicación PAN
24	n 3	Código de función
25	n 2	Código de condición del POS
26	n 2	Código de captura del POS
27	n 1	Longitud del identificador de respuesta de autorización
28	n 8	Monto, Tarifa de transacción
29	n 8	Monto, Tarifa de compensación
30	n 8	Monto, Tarifa de procesamiento de transacción
31	n 8	Monto, Tarifa de procesamiento de compensación
32	n ..11	Código de identificación de institución de adquisiciencia
33	n ..11	Código de identificación de institución de reenvío
34	n ..28	Número de cuenta primario, extendido
35	z ..37	Datos del Track 2
36	n ..104	Datos del Track 3
37	an 12	Número de referencia de recuperación
38	an 6	Respuesta identificación de autorización
39	an 2	Código de respuesta
40	an 3	Código de restricción del servicio
41	ans 8	Identificación del terminal aceptor de la tarjeta
42	ans 15	Identificación del aceptor de la tarjeta
43	ans 40	Nombre/locación del aceptor de la tarjeta
44	an ..25	Datos adicionales de respuesta
45	an ..76	Datos del Track 1
46	an ..999	Datos adicionales ISO
47	an ..999	Datos adicionales, Nacional
48	an ..999	Datos adicionales, Privado
49	a 3	Código de moneda, transacción
50	an 3	Código de moneda, compensación
51	a 3	Código de moneda, facturación del tarjetahabiente
52	b 16	Datos de número de identificación personal
53	n 18	Información de control relacionada a la seguridad
54	an 120	Montos adicionales
55	ans ..999	Reservado para uso de la ISO
56	ans ..999	Reservado para uso de la ISO
57	ans ..999	Reservado para uso nacional
58	ans ..999	Reservado para uso nacional
59	ans ..999	Reservado para uso nacional
60	an ..7	Código de aviso
61	ans ..999	Reservado para uso privado
62	ans ..999	Reservado para uso privado
63	ans ..999	Reservado para uso privado
64	b 16	Código de autenticación del mensaje (MAC)
65	b 16	Mapa de bits, terciario
66	n 1	Código de compensación
67	n 2	Código de pago extendido
68	n 3	Código de país de institución receptora
69	n 3	Código de país de institución de compensación

Data element	Tipo	Uso
70	n 3	Código de información de gestión de red
71	n 4	Número del mensaje
72	ans ..999	Registro de datos
73	n 6	Fecha, Acción
74	n 10	Créditos, número
75	n 10	Créditos, número de reversión
76	n 10	Débitos, número
77	n 10	Débitos, número de reversión
78	n 10	Transferencia, número
79	n 10	Transferencia, número de reversión
80	n 10	Número de consultas
81	n 10	Autorizaciones, número
82	n 12	Créditos, monto de la tarifa de procesamiento
83	n 12	Créditos, monto de la tarifa de transacción
84	n 12	Débitos, monto de la tarifa de procesamiento
85	n 12	Débitos, monto de la tarifa de transacción
86	n 15	Créditos, monto
87	n 15	Créditos, monto de reversión
88	n 15	Débitos, monto
89	n 15	Débitos, monto de reversión
90	n 42	Elementos de datos originales
91	an 1	Código de actualización de archivo
92	n 2	Código de seguridad de archivo
93	n 5	Indicador de respuesta
94	an 7	Indicador de servicio
95	an 42	Cantidades de reemplazo
96	an 8	Código de seguridad del mensaje
97	n 16	Monto, liquidación neta
98	ans 25	Beneficiario
99	n ..11	Código de identificación de la institución de compensación
100	n ..11	Código de identificación de la institución de recepción
101	ans 17	Nombre de archivo
102	ans ..28	Identificación de la Cuenta 1
103	ans ..28	Identificación de la Cuenta 2
104	ans ..100	Descripción de la transacción
105	ans ..999	Reservado para uso de la ISO
106	ans ..999	Reservado para uso de la ISO
107	ans ..999	Reservado para uso de la ISO
108	ans ..999	Reservado para uso de la ISO
109	ans ..999	Reservado para uso de la ISO
110	ans ..999	Reservado para uso de la ISO
111	ans ..999	Reservado para uso de la ISO
112	ans ..999	Reservado para uso nacional
113	n ..11	Código identificador de la institución de autorización
114	ans ..999	Reservado para uso privado
115	ans ..999	Reservado para uso privado
116	ans ..999	Reservado para uso privado
117	ans ..999	Reservado para uso privado
118	ans ..999	Reservado para uso privado
119	ans ..999	Reservado para uso privado
120	ans ..999	Reservado para uso privado
121	ans ..999	Reservado para uso privado

Data element	Tipo	Uso
122	ans ..999	Reservado para uso privado
123	ans ..999	Reservado para uso privado
124	ans ..255	Texto de información
125	ans ..50	Información de gestión de red
126	ans ..6	ID de rastreo del emisor
127	ans ..999	Reservado para uso privado
128	b 16	Código de autenticación del mensaje

5. Regulaciones a nivel nacional para el sistema financiero

Las regulaciones nacionales en materia de banca electrónica (y en general) en nuestro país son dictadas por la Superintendencia de las Instituciones del Sector Bancario.

5.1. Resolución 641-10: Normas que regulan el uso de los servicios de la banca electrónica

El 23 de diciembre de 2010, la Superintendencia de Instituciones del Sector Bancario (Sudeban) emitió la resolución 641-10, que regula el uso de los servicios de la banca electrónica en Venezuela. Esta resolución contiene los mecanismos de autenticación requeridos para incrementar la seguridad de los clientes de la banca electrónica ante el creciente número de fraudes electrónicos.

Esta norma contempla medidas de seguridad en las principales etapas del procesamiento de la transacción, desde su generación hasta la finalización, incluyendo el monitoreo de los sistemas. Podemos categorizar las normas en cuatro procesos principales

5.1.1. Normas relativas a la afiliación, identificación y autenticación del cliente en los servicios de banca electrónica.

Este subconjunto de las normas se refiere a la fase de generación de los datos transaccionales. Los clientes deben afiliarse a los servicios relacionados con banca electrónica y los proveedores de dichos servicios están obligados a indicar todas las instrucciones y términos de dichos servicios a los clientes potenciales.

Para realizar la afiliación, identificación y autenticación del cliente en sus servicios, las instituciones deben poseer información previa obtenida de la ficha del cliente además de implementar métodos de autenticación. Para esto se han establecido cinco tipos de factores, que deben ser incorporados en las operaciones electrónicas, dependiendo del nivel de seguridad requerido:

Factor de autenticación de categoría 1: Información obtenida de la ficha del cliente.

Factor de autenticación de categoría 2: Usuario y contraseña conocida por el cliente.

Factor de autenticación de categoría 3: Claves dinámicas OTP (tokens).

Factor de autenticación de categoría 4: Firmas electrónicas certificadas y emitidas a nombre del cliente.

Factor de autenticación de categoría 5: Información de tipo biométrico.

El factor mínimo exigido por Sudeban para iniciar sesión en la banca electrónica es el tipo 2, pero para realizar cualquier tipo de transacción que involucre movimientos de dinero o modificación de información de la cuenta, se exige un factor de tipo 3, 4 o 5. (OTPs, firmas electrónicas o información biométrica).

La siguiente tabla ilustra los factores de autenticación adicionales requeridos según el tipo de operación:

Tabla 8. Factores de autenticación requeridos por la norma Sudeban 641-10 para las operaciones de banca electrónica.

Tipo de Operación	Factor de autenticación base requerido	Factor de autenticación adicional requerido
Afiliación y desafiliación de productos y servicios	2	3,4 o 5
Mantenimiento de productos y servicios de pago programados	2	3,4 o 5
Pagos o transferencias electrónicas a terceros	2	3,4 o 5
Retiros o adelantos de efectivo	2	3,4 o 5
Apertura de segundas cuentas o productos financieros	2	4
Actualización de los datos de la ficha de cliente a través de banca por Internet	2	4
Mantenimiento de contraseñas, activación o desactivación de tarjetas	1,2	N/A
Consultas	2	N/A
Transacciones ofrecidas a través de dispositivos de autoservicio	2	N/A
Pago o transferencias electrónicas mismo titular y mismo banco	2	N/A

5.1.2. Normas relativas a la operación de los servicios de banca electrónica.

Las instituciones financieras deben informar a los clientes sobre los términos de los servicios y funcionamiento de sus canales electrónicos. También deben ofrecerles información oportuna sobre las operaciones que el cliente realice (estados de cuenta, comprobantes, entre otros). Respecto al inicio de sesión en los canales electrónicos la norma establece límites de tiempo de inactividad. En cuanto a los puntos de venta virtuales, estos serán dispuestos por las instituciones bancarias y no será permitido que el comercio coloque uno propio.

5.1.3. Normas relativas a la información transmitida, almacenada o procesada a través de los canales electrónicos.

En lo que concierne a la transmisión y almacenamiento de los datos transaccionales, las instituciones financieras deben implementar mecanismos de cifrado. Las contraseñas de categoría 2, números de identificación personal y de productos no pueden ser transmitidos a través de ningún medio de comunicación electrónica. El resto de los datos sensibles solo podrán ser almacenados por el tiempo necesario para la autorización de la transacción.

5.1.4. Normas relativas al monitoreo y control de las operaciones y servicios de banca electrónica.

Las instituciones financieras deben establecer mecanismos de monitoreo y control para garantizar el correcto funcionamiento de los canales electrónicos. Estos mecanismos deben encargarse de notificar al cliente las alertas asociadas a operaciones realizadas por estos canales así como bloqueos preventivos en caso de detección de comportamiento anormal en el perfil transaccional del cliente

CAPÍTULO III

MARCO METODOLOGÓICO

1. Metodología *Scrum*

1.1. Metodologías de desarrollo ágiles

Las metodologías ágiles son flexibles, pueden ser modificadas para que se ajusten a la realidad de cada equipo y proyecto. Los proyectos ágiles se subdividen en proyectos más pequeños mediante una lista ordenada de características. Cada proyecto es tratado de manera independiente y desarrolla un subconjunto de características durante un periodo de tiempo corto, de entre dos y seis semanas. La comunicación con el cliente es constante al punto de requerir un representante de él durante el desarrollo. Los proyectos son altamente colaborativos y se adaptan mejor a los cambios; de hecho, el cambio en los requerimientos es una característica esperada y deseada, al igual que las entregas constantes al cliente y la retroalimentación por parte de él. Tanto el producto como el proceso son mejorados frecuentemente.

1.2. Visión general de *Scrum*

Scrum es un marco de trabajo de procesos que ha sido usado para gestionar el desarrollo de productos complejos desde principios de los años 90. *Scrum* no es un proceso o una técnica para construir productos; en lugar de eso, es un marco de trabajo dentro del cual se pueden emplear varias técnicas y procesos. *Scrum* muestra la eficacia relativa de las prácticas de gestión de producto y las prácticas de desarrollo, de modo que podamos mejorar.

1.3. Roles

1.3.1. Dueño del Producto

El Dueño de Producto es el responsable de maximizar el valor del producto y del trabajo del Equipo de Desarrollo. El cómo se lleva a cabo esto podría variar ampliamente entre distintas organizaciones, Equipos *Scrum* e individuos.

El Dueño de Producto es la única persona responsable de gestionar la Lista del Producto (*Product Backlog*). La gestión de la Lista del Producto incluye:

- Expresar claramente los elementos de la Lista del Producto;
- Ordenar los elementos en la Lista del Producto para alcanzar los objetivos y misiones de la mejor manera posible;
- Optimizar el valor del trabajo desempeñado por el Equipo de Desarrollo;

- Asegurar que la Lista del Producto es visible, transparente y clara para todos, y que muestra aquello en lo que el equipo trabajará a continuación; y,
- Asegurar que el Equipo de Desarrollo entiende los elementos de la Lista del Producto al nivel necesario.

El Dueño de Producto podría hacer el trabajo anterior, o delegarlo en el Equipo de Desarrollo. Sin embargo, en ambos casos el Dueño de Producto sigue siendo el responsable de dicho trabajo.

1.3.2. Equipo de Desarrollo

El Equipo de Desarrollo consiste en los profesionales que desempeñan el trabajo de entregar un Incremento de producto "Terminado", que potencialmente se pueda poner en producción, al final de cada *Sprint*. Solo los miembros del Equipo de Desarrollo participan en la creación del Incremento.

Los Equipos de Desarrollo tienen las siguientes características:

- Son auto organizados. Nadie (ni siquiera el *Scrum Master*) indica al Equipo de Desarrollo cómo convertir elementos de la Lista del Producto en Incrementos de funcionalidad potencialmente desplegables;
- Los Equipos de Desarrollo son multifuncionales, contando como equipo con todas las habilidades necesarias para crear un Incremento de producto;
- *Scrum* no reconoce títulos para los miembros de un Equipo de Desarrollo, todos son Desarrolladores, independientemente del trabajo que realice cada persona; no hay excepciones a esta regla;
- *Scrum* no reconoce sub-equipos en los equipos de desarrollo, no importan los dominios particulares que requieran ser tenidos en cuenta, como pruebas o análisis de negocio; no hay excepciones a esta regla;

1.3.3. Scrum Master

El *Scrum Master* es el responsable de asegurar que *Scrum* es entendido y adoptado. Los *Scrum Masters* hacen esto asegurándose de que el Equipo *Scrum* trabaja ajustándose a la teoría, prácticas y reglas de *Scrum*.

1.4. Eventos

1.4.1. Sprint

Es un bloque de tiempo (*time-box*) de un mes o menos durante el cual se crea un incremento de producto "Terminado", utilizable y potencialmente desplegable. Es más conveniente si la duración de

los *Sprints* es consistente a lo largo del esfuerzo de desarrollo. Cada nuevo *Sprint* comienza inmediatamente después de la finalización del *Sprint* previo.

Los *Sprints* contienen y consisten de la Reunión de Planificación del *Sprint* (*Sprint Planning Meeting*), los *Scrums* Diarios (*Daily Scrums*), el trabajo de desarrollo, la Revisión del *Sprint* (*Sprint Review*), y la Retrospectiva del *Sprint* (*Sprint Retrospective*).

1.4.2. Reunión de Planificación de *Sprint*

El trabajo a realizar durante el *Sprint* se planifica en la Reunión de Planificación de *Sprint*. Este plan se crea mediante el trabajo colaborativo del Equipo *Scrum* completo. La Reunión de Planificación de *Sprint* tiene un máximo de duración de ocho horas para un *Sprint* de un mes.

La Reunión de Planificación de *Sprint* responde a las siguientes preguntas:

- ¿Qué puede entregarse en el Incremento resultante del *Sprint* que comienza?
- ¿Cómo se conseguirá hacer el trabajo necesario para entregar el Incremento?

1.4.3. *Scrum* Diario

El *Scrum* Diario es una reunión con un bloque de tiempo de 15 minutos para que el Equipo de Desarrollo sincronice sus actividades y cree un plan para las siguientes 24 horas. Esto se lleva a cabo inspeccionando el trabajo avanzado desde el último *Scrum* Diario y haciendo una proyección acerca del trabajo que podría completarse antes del siguiente.

1.4.4. Revisión de *Sprint*

Al final del *Sprint* se lleva a cabo una Revisión de *Sprint* para inspeccionar el Incremento y adaptar la Lista de Producto si fuese necesario. Durante la Revisión de *Sprint*, el Equipo *Scrum* y los interesados colaboran acerca de lo que se hizo durante el *Sprint*.

1.4.5. Retrospectiva de *Sprint*

La Retrospectiva de *Sprint* es una oportunidad para el Equipo *Scrum* de inspeccionarse a sí mismo y crear un plan de mejoras que sean abordadas durante el siguiente *Sprint*.

La Retrospectiva de *Sprint* tiene lugar después de la Revisión de *Sprint* y antes de la siguiente Reunión de Planificación de *Sprint*. Se trata de una reunión restringida a un bloque de tiempo de tres horas para *Sprints* de un mes.

Para el final de la Retrospectiva de *Sprint*, el Equipo *Scrum* debería haber identificado mejoras que implementará en el próximo *Sprint*.

1.5. Artefactos

1.5.1. Lista de Producto (*Product Backlog*)

La Lista de Producto enumera todas las características, funcionalidades, requisitos, mejoras y correcciones que constituyen cambios a ser hechos sobre el producto para entregas futuras. Los elementos de la Lista de Producto tienen como atributos la descripción, la ordenación, la estimación y el valor.

Una Lista de Producto nunca está completa. El desarrollo más temprano de la misma solo refleja los requisitos conocidos y mejor entendidos al principio. La Lista de Producto evoluciona a medida de que el producto y el entorno en el que se usará también lo hacen. La Lista de Producto es dinámica; cambia constantemente para identificar lo que el producto necesita para ser adecuado, competitivo y útil.

1.5.2. Lista de Pendientes del *Sprint* (*Sprint Backlog*)

La Lista de Pendientes del *Sprint* es el conjunto de elementos de la Lista de Producto seleccionados para el *Sprint*, más un plan para entregar el Incremento de producto y conseguir el Objetivo del *Sprint*. La Lista de Pendientes del *Sprint* es una predicción hecha por el Equipo de Desarrollo acerca de qué funcionalidad formará parte del próximo Incremento y del trabajo necesario para entregar esa funcionalidad en un Incremento "Terminado". El Equipo de Desarrollo modifica la Lista de Pendientes del *Sprint* durante el *Sprint*.

1.5.3. Incremento

El Incremento es la suma de todos los elementos de la Lista de Producto completados durante un *Sprint* y el valor de los incrementos de todos los *Sprints* anteriores. Al final de un *Sprint*, el nuevo Incremento debe estar "Terminado", lo cual significa que está en condiciones de ser utilizado y que cumple la Definición de "Terminado" del Equipo *Scrum*. El incremento debe estar en condiciones de utilizarse sin importar si el Dueño de Producto decide liberarlo o no.

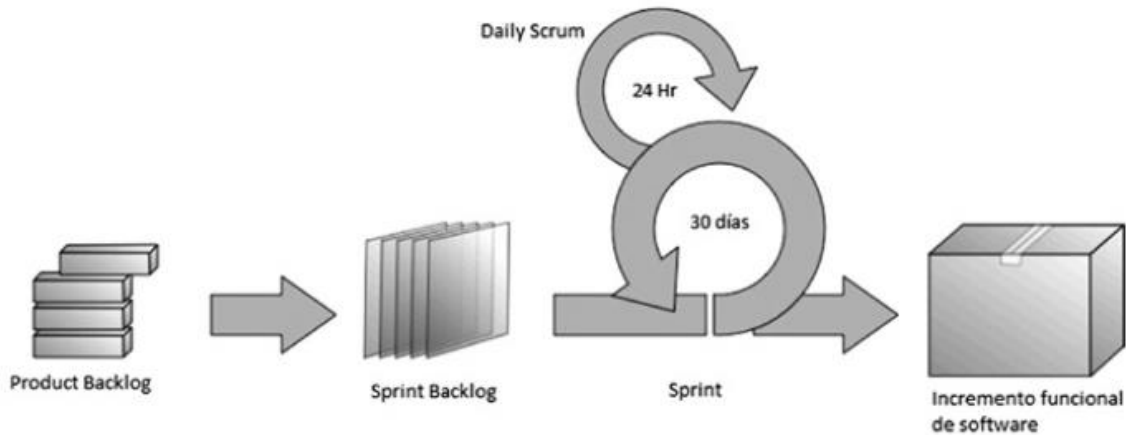


Figura 7. Elementos fundamentales del proceso Scrum. (Navarro, Fernández, & Morales, 2013)

2. Implantación de Scrum dentro de la empresa

En la empresa donde se desarrolló el trabajo se aplica la técnica *Scrum* de manera simplificada y adaptada al equipo involucrado en el proceso de construcción del sistema, desde la fase de diseño e implementación, hasta su puesta en producción y posterior mantenimiento.

2.1. Roles

Dueño del Producto: Persona a cargo de la supervisión del proyecto en la empresa

Equipo de Desarrollo: Desarrollador que plantea la presente propuesta Trabajo Especial de Grado.

2.2. Eventos

Todas las actividades del Trabajo Especial de Grado, incluyendo el diseño del sistema, desarrollo, documentación e implantación serán enmarcadas dentro de los eventos de la metodología *Scrum*:

- *Sprints* con una duración de dos semanas
- Reunión de Planificación de *Sprint*
- Revisión de *Sprint*

2.3. Artefactos

En primer lugar se listan los artefactos a utilizar propios de la metodología *Scrum*:

- Lista de Producto
- Lista de Pendientes del *Sprint*

En segundo lugar los artefactos elegidos para documentar el sistema desde cuatro perspectivas:

1. Diseño: consiste en la definición de clases, interfaces y colaboraciones.
 - Diagrama de Clases: representan la visión orientada a objetos de un sistema de manera estática.
 - Diagrama de Casos de Uso: Un caso de uso representa una funcionalidad particular de un sistema. Este diagrama describe las relaciones entre las funcionalidades y sus controladores internos/externos. Estos controladores son conocidos como actores.
2. Procesos: define el flujo del sistema.
 - Diagrama de Secuencia: se utiliza para visualizar la secuencia de llamadas en un sistema para realizar una funcionalidad específica.
 - Diagrama de Colaboración: el propósito específico de diagrama de colaboración es visualizar la organización de los objetos y su interacción.
3. Despliegue: representan los nodos físicos del sistema que forman el hardware.
 - Diagrama de despliegue: son un conjunto de nodos y sus relaciones. Estos nodos son entidades físicas donde se utilizan los componentes.

En tercer lugar los artefactos elegidos para modelar los datos en el sistema:

- Diagrama Entidad-Relación: describe los datos como entidades, relaciones y atributos.
- Esquema de Base de Datos: contiene las tablas de la base de datos y sus relaciones.

Finalmente, los artefactos que documentan las pruebas.

- Casos de prueba: contiene las especificaciones de los casos de prueba de la aplicación.

CAPÍTULO IV

DESARROLLO DE LA SOLUCIÓN

1. Análisis y requerimientos

1.1. Análisis de plataforma actual

El esquema de punto de venta virtual VPOS provee la funcionalidad de un punto de venta (POS) tradicional a través de una aplicación que puede ser instalada en un computador personal. Esta plataforma funciona mediante la sinergia de tres componentes principales:

- Una aplicación de punto de venta: es el componente de software que provee la interfaz para el ingreso de los datos del tarjetahabiente, la lógica de construcción del formato de la transacción y el establecimiento de la conexión con la entidad bancaria adquirente. Esta aplicación se instala en cada caja y funciona en conjunto con un dispositivo *pinpad* suministrado por la institución de adquirencia para la lectura segura de la tarjeta de pago y el ingreso del *pin* del tarjetahabiente.
- Un *switch* transaccional: es un dispositivo gestionado por la institución bancaria adquirente, este se encarga del procesamiento de las transacciones.
- Un servidor de datos transaccionales: donde se almacenan los datos de las cajas, comercios, estadísticas sobre transacciones y datos útiles para la gestión y generación de reportes. Esta aplicación provee un portal donde los representantes de los comercios pueden auto gestionar reglas de negocio para las transacciones, tales como configuración de las cuentas hacia donde estas serán dirigidas.

En la ilustración siguiente se muestra el esquema actual de punto de venta virtual VPOS:

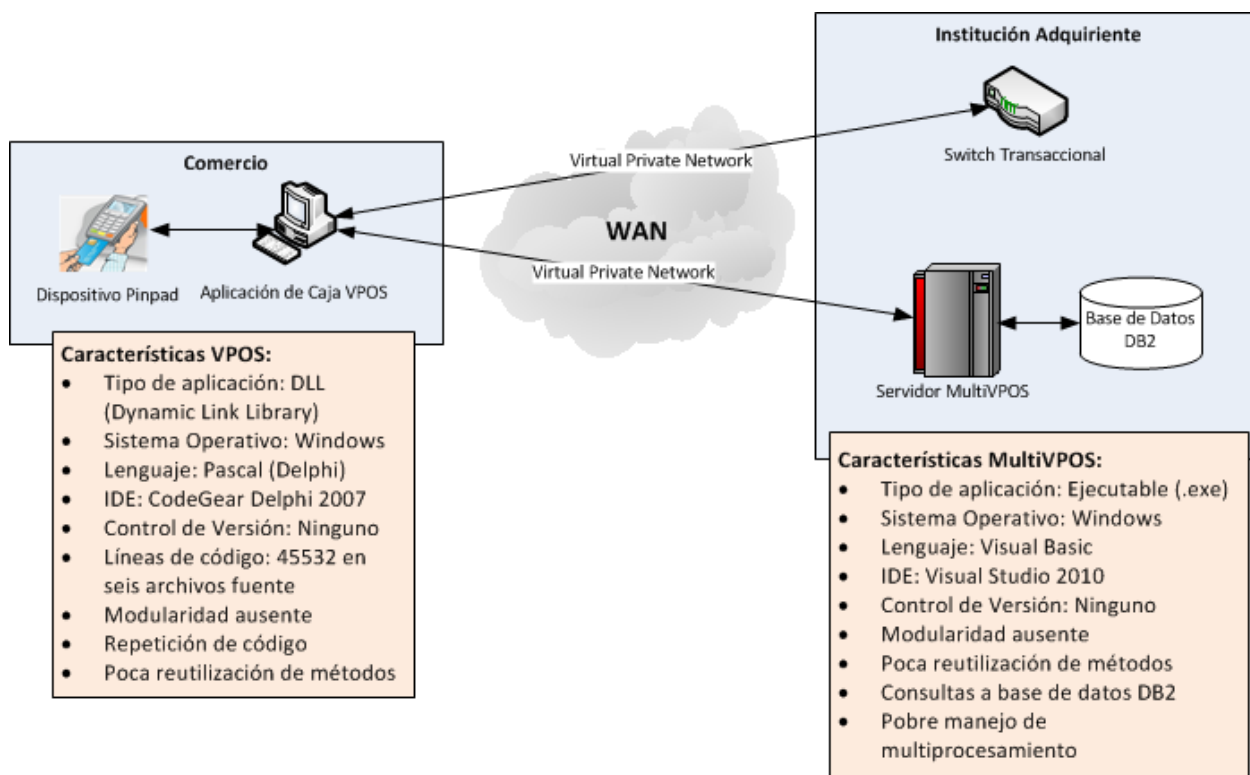


Figura 8. Esquema en producción de punto de venta virtual VPOS. (Elaboración propia).

Partiendo del análisis de las características de esta aplicación, se requiere una solución:

- Multiplataforma: que pueda utilizarse en diversos entornos y sistemas operativos.
- Extensible y reutilizable: que permita la incorporación del soporte a nuevos canales de pago con la máxima reutilización y el menor impacto.
- Interoperable: proveer una interfaz unificada y estándar para el uso de sus funciones.

Esta solución consiste en una aplicación que denominamos Motor Transaccional V-Financial, la cual posee entre sus principales características la adaptabilidad a múltiples canales de pago, y el soporte para que dichos canales invoquen las operaciones del motor mediante interfaces interoperables con tecnología Internet.

La comprensión de la plataforma actual constituye la base de conocimiento para la implementación de los estándares de transacciones y seguridad de la industria bancaria en el desarrollo la aplicación.

1.2. Requerimientos de la aplicación

Partiendo de la problemática que encontramos en la aplicación actualmente en producción, cuyo diseño y arquitectura monolítica dificulta la incorporación de nuevos canales de pago sin

una reingeniería de la misma, y, tomando en cuenta la necesidad de un nuevo esquema, escalable, interoperable y multiplataforma, se establecieron los siguientes requerimientos:

- Proporcionar una interfaz homogénea que provea escalabilidad para integrar diversos canales de pago de múltiples instituciones financieras.
- Proveer métodos unificados para las invocaciones de las operaciones transaccionales a través de servicios web estandarizados.
- Manejo de formatos utilizados por la industria bancaria: Norma ISO 8583.
- Diseño adaptable con la capacidad de soportar la personalización de los formatos de transacciones pertenecientes a diversas entidades bancarias.
- Enrutamiento de transacciones a las entidades aprobadoras correspondientes.
- Soporte para múltiples motores de bases de datos.
- Almacenar los datos transaccionales para su posterior consulta y gestión de conformidad con las normativas de seguridad vigentes.
- Proporcionar mecanismos de acceso a los datos a través de una interfaz utilizando de servicios web, y de esta manera, realizar operaciones de gestión y análisis de los mismos
- Establecer mecanismos de seguridad en la realización de las transacciones y el acceso a los datos.
- Cumplir con protocolos, estándares y normativas de seguridad de la industria bancaria: EMV, PCI DSS, SUDEBAN.

1.3. Usuarios a los que está dirigida

La aplicación de Motor Transaccional V-Financial está dirigida a instituciones bancarias o de adquisición que necesiten una plataforma para tercerizar la generación de las transacciones, el enrutamiento a sus sistemas de aprobación y la gestión de todos los datos relativos a esas transacciones.

La aplicación que implementa el canal de pago en forma de Botón de pago on-line, y a la que denominaremos C-Pago, está concebida para comercios con presencia en Internet que deseen aceptar este método de pago para la comercialización de sus productos y servicios.

1.4. Aplicaciones similares

1.4.1. Payu Latam

Plataforma para aceptar pagos online en siete países de Latinoamérica, proveen diversos métodos de integración a las empresas afiliadas y múltiples medios de pago a los compradores. Las empresas que deseen integrarse deben crear una cuenta en el sitio y recibirán todos los datos de autenticación y pautas para la integración.



Figura 9. Página principal de PayU Latam. (PayU Latam)

1.4.2. Authorize.Net

Ofrece una pasarela de pagos a la cual los comercios pueden conectarse fácilmente, proporcionando la infraestructura y seguridad necesarias para garantizar la transmisión rápida, segura y fiable de los datos al realizar transacciones. Esta pasarela se encarga del enrutamiento de las transacciones y tiene disponibilidad 24/7.

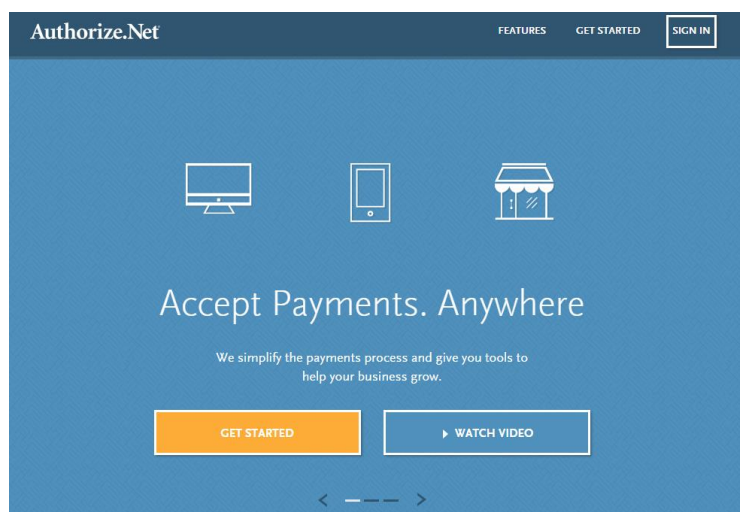


Figura 10. Página principal de Authorize.Net. (Authorize.Net)

2. Diseño de la plataforma

2.1. Arquitectura del sistema

La plataforma diseñada para cumplir con los requerimientos definidos, y a la cual llamaremos Plataforma V-Financial consta de:

- Una aplicación servidor que recibe las solicitudes de los diversos canales de pago a través de una interfaz de servicios web. Esta aplicación mantiene los datos necesarios para la parametrización de las transacciones, la construcción de la trama en formato ISO 8583 y su posterior envío a la entidad financiera aprobadora. Así mismo, provee un conjunto de servicios web para su uso por parte de portales destinados a la configuración y acceso a los datos.
- Múltiples canales de pago que serán los responsables de los datos del cliente y realizar la solicitud de la operación correspondiente a la aplicación servidor. Aunque la aplicación servidor está implementada para el soporte a múltiples canales de pago, a efectos de este trabajo se desarrolló el caso del canal de botón para procesamiento de transacciones en línea de tarjeta no presente (C-Pago).

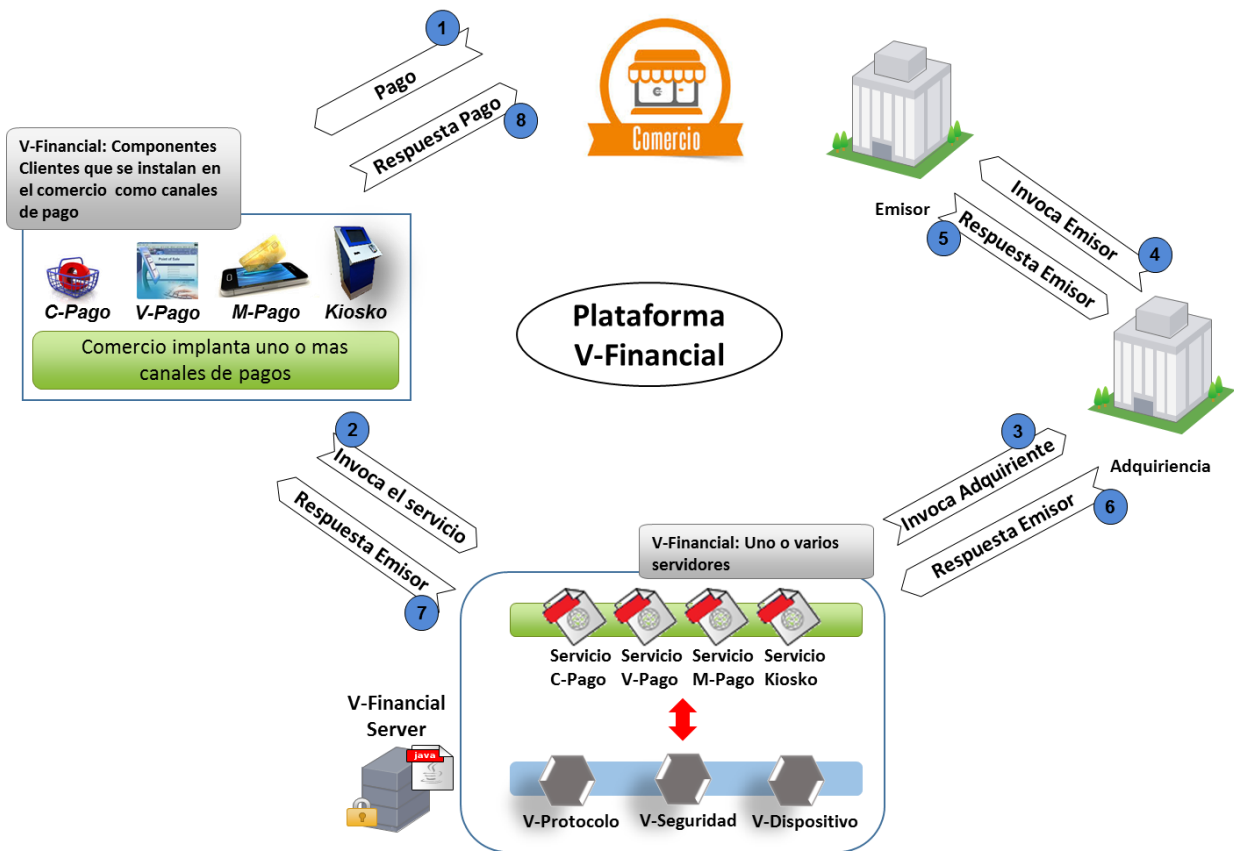


Figura 11. Componentes de la pataforma V-Financial. (Elaboración propia).

Como se muestra en la figura 11, la plataforma transaccional diseñada consta de una aplicación de tipo servidor que constituye el Motor Transaccional, el cual recibe las solicitudes de transacciones de diversos canales de pago a través de servicios web, las procesa y envía la trama en formato ISO 8583 a la institución que procesará la misma. La plataforma también se compone del conjunto de canales de pago que coexisten con el Motor y realizan las solicitudes al mismo. Para la comprobación de las funcionalidades del Motor Transaccional en el desarrollo de este trabajo, se muestra la implementación de un canal de pago en la figura de botón de pago en línea, al que llamaremos C-Pago. El proceso de interacción entre el canal de pago y el Motor se ilustra a continuación:

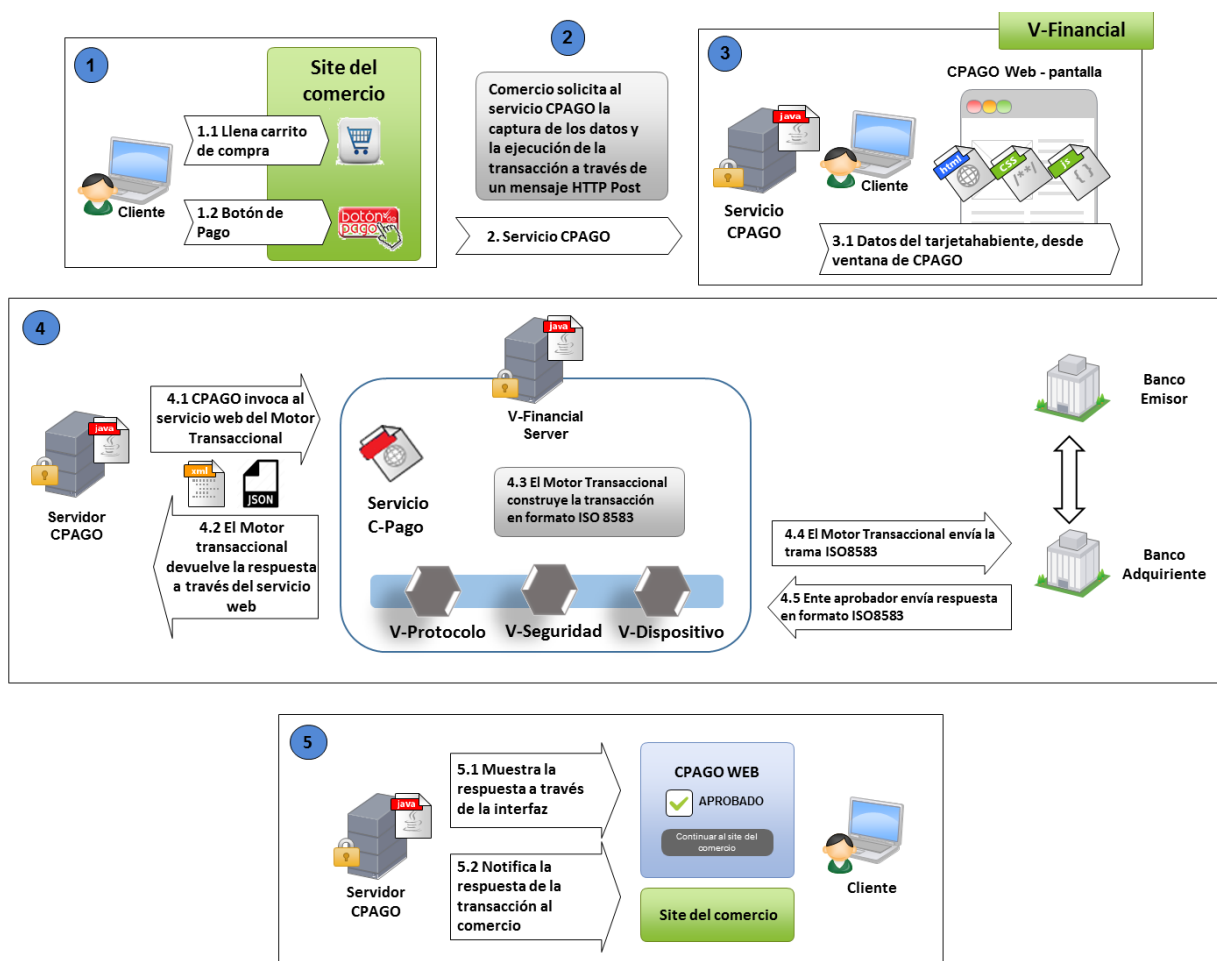


Figura 12. Interacción de componentes de la plataforma V-Financial. (Elaboración propia).

2.2. Principales funcionalidades o herramientas provistas por la aplicación

2.2.1. Motor Transaccional V-Financial

1. Proveer servicios web para procesar transacciones de diversos canales de pago

El Motor Transaccional V-Financial implementa servicios web REST en formato JSON y XML, dichos servicios son invocados por los diversos canales de pago para la realización de múltiples operaciones bancarias. Los parámetros y las operaciones disponibles son particulares al estándar ISO 8583 de cada institución financiera.

Las primitivas transaccionales básicas implementadas en el Motor transaccional consisten en:

- Venta
- Reverso automático

- Test de conexión

2. Construcción del Formato de la Transacción en formato ISO 8583:

Una vez recibida la petición a través de la invocación del servicio web, el Motor se encarga de construir la transacción en formato ISO 8583 utilizando los parámetros proporcionados por el terminal del canal de pago y los valores que tiene almacenados en la configuración local.

La configuración de los formatos transaccionales se establece mediante plantillas, cada canal de pago funciona mediante un conjunto de las mismas. Cada plantilla posee la información para la construcción de un tipo de transacción.

3. Enrutamiento de Transacciones a la plataforma procesadora de la entidad financiera correspondiente:

Dependiendo de la configuración para el canal de pago, se envía la transacción en formato ISO 8583 al ente aprobador correspondiente. Una vez recibida la respuesta, se interpreta y se envía al solicitante en una respuesta al servicio web.

4. Gestión y almacenamiento de Datos Transaccionales:

El motor implementa comunicación con diversas bases de datos a través de un módulo de ORM (Mapeo Objeto Relacional). Esto con la finalidad de almacenar los datos necesarios para la realización de todas las operaciones y recopilación de estadísticas para generación de informes.

5. Interfaz de programación de aplicaciones (API) para la configuración y acceso a los datos de manera interoperable:

Para realizar procesos afiliatorios, configuración de terminales, y gestión de la lógica de enrutamiento, se proporciona un API para facilitar el desarrollo de otras aplicaciones que implementarán la interfaz gráfica para estos procesos. La API sigue lineamientos de la arquitectura REST y puede ser invocada a través de métodos HTTP en conjunción con los formatos XML y/o JSON. Los métodos proporcionados por la API se listan a continuación:

Tabla 9. Cuadro resumen de las funciones de la API de Acceso a Datos.

Función	Gestión de									
	Comercios	Corporaciones	Terminales	Bancos	Afiliaciones	Adquiriencia	Reglas	Dispositivos	Clasificación	Canales de Pago
Listar todos los Comercios	✓									
Listar	✓									

Función	Gestión de									
	Comercios	Corporaciones	Terminales	Bancos	Afiliaciones	Adquiriencia	Reglas	Dispositivos	Clasificación	Canales de Pago
Comercios de Multicomercio										
Consultar Comercio	✓									
Afiliar Comercio	✓									
Modificar datos de Comercio	✓									
Consultar Comercio por Terminal	✓									
Crear Corporación		✓								
Consultar Corporación		✓								
Listar Corporaciones		✓								
Modificar datos de Corporación		✓								
Listar comercios asociados a Corporación		✓								
Consultar Terminales de Comercio			✓							
Consultar Terminal			✓							
Crear Terminal			✓							
Habilitar Terminal			✓							
Deshabilitar Terminal			✓							
Listar parámetros de terminal bancario de Terminal			✓							
Anadir parámetro de terminal bancario a Terminal			✓							
Listar Bancos				✓						

Función	Gestión de									
	Comercios	Corporaciones	Terminales	Bancos	Afiliaciones	Adquiriencia	Reglas	Dispositivos	Clasificación	Canales de Pago
Consultar Banco				✓						
Crear Afiliación					✓					
Listar Adquirientes de Comercio						✓				
Listar Adquirientes de Multicomercio						✓				
Consultar Adquiriente						✓				
Crear Adquiriente						✓				
Listar Conjuntos de Reglas de Comercio							✓			
Consultar Conjunto de Reglas							✓			
Crear Conjunto de Reglas							✓			
Modificar Conjunto de Reglas							✓			
Asignar Conjunto de Reglas a Afiliación de Comercio							✓			
Eliminar Conjunto de Reglas							✓			
Listar Reglas de Conjunto de Reglas							✓			
Listar Reglas por Afiliación							✓			
Crear Regla							✓			
Consultar Regla							✓			
Modificar Regla							✓			

Función	Gestión de									
	Comercios	Corporaciones	Terminales	Bancos	Afiliaciones	Adquiriencia	Reglas	Dispositivos	Clasificación	Canales de Pago
Eliminar Regla							✓			
Agregar Regla a Conjunto de Reglas							✓			
Remover Regla de Conjunto de Reglas							✓			
Consultar Tipo de Dispositivo								✓		
Crear Tipo de Dispositivo								✓		
Modificar Tipo de Dispositivo								✓		
Eliminar Tipo de Dispositivo								✓		
Listar Tipos de Dispositivos								✓		
Consultar Dispositivo								✓		
Crear Dispositivo								✓		
Modificar Dispositivo								✓		
Eliminar Dispositivo								✓		
Listar Dispositivos								✓		
Asignar Dispositivo a Terminal								✓		
Consultar Asignación de Dispositivo a Terminal								✓		
Revocar Dispositivo Asignar								✓		
Obtener lista de sectores económicos									✓	
Obtener tipos de industrias por sector									✓	

Función	Gestión de									
	Comercios	Corporaciones	Terminales	Bancos	Afiliaciones	Adquiriencia	Reglas	Dispositivos	Clasificación	Canales de Pago
económico										
Consultar datos de sector									✓	
Consultar datos de tipo de industria									✓	
Listar Canales de Pago										✓

2.2.2. Aplicación Web de Botón de Pago C-Pago

1. Ofrecer una interfaz web a través de la cual se tenga acceso a operaciones de punto de venta con tarjetas de pago

Esta interfaz será invocada por el sitio web del comercio, el cliente coloca los datos y una procesada la transacción es redirigido al sitio del comercio nuevamente.

2. Obtener la información necesaria para la realización de la transacción
3. Enviar la información necesaria para la realización de la transacción al Motor Transaccional Multibanco.

2.3. Casos de uso del sistema

2.3.1. Motor Transaccional V-Financial

2.3.1.1. Módulo Interfaz de Pago

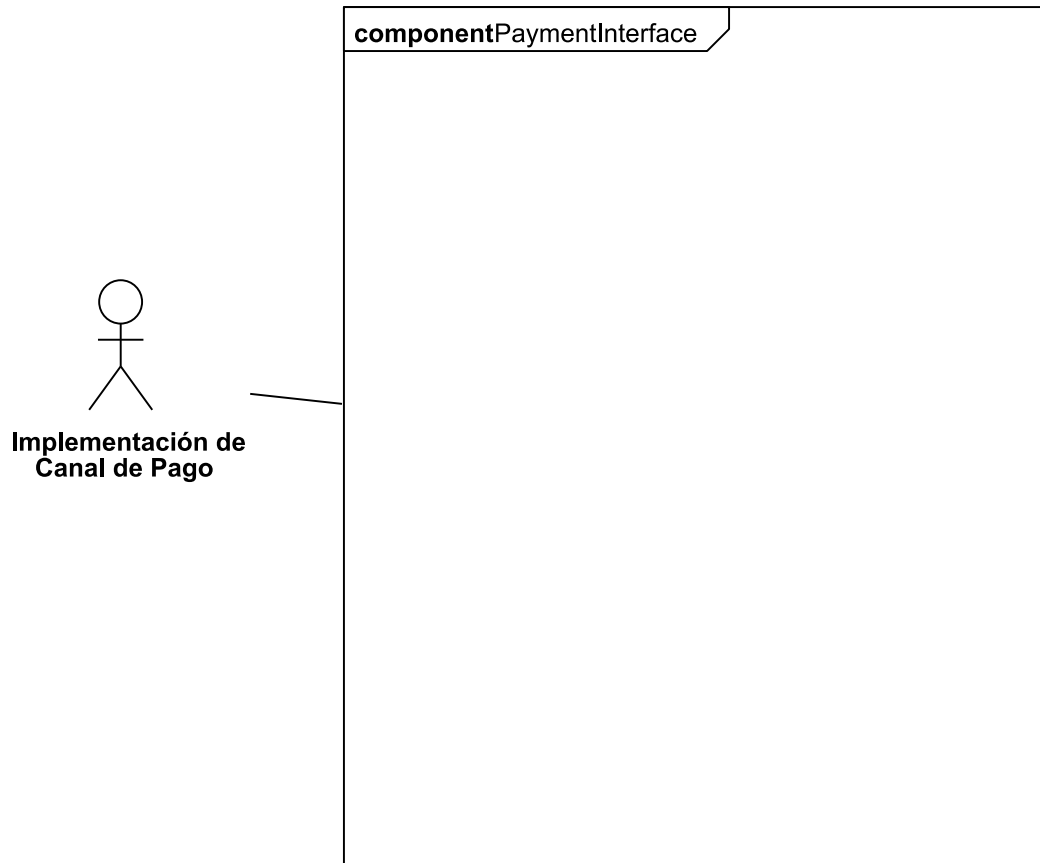


Figura 13. Motor Transaccional V-Financial – Interfaz de Pago: Diagrama de casos de uso nivel 0. (Elaboración propia).

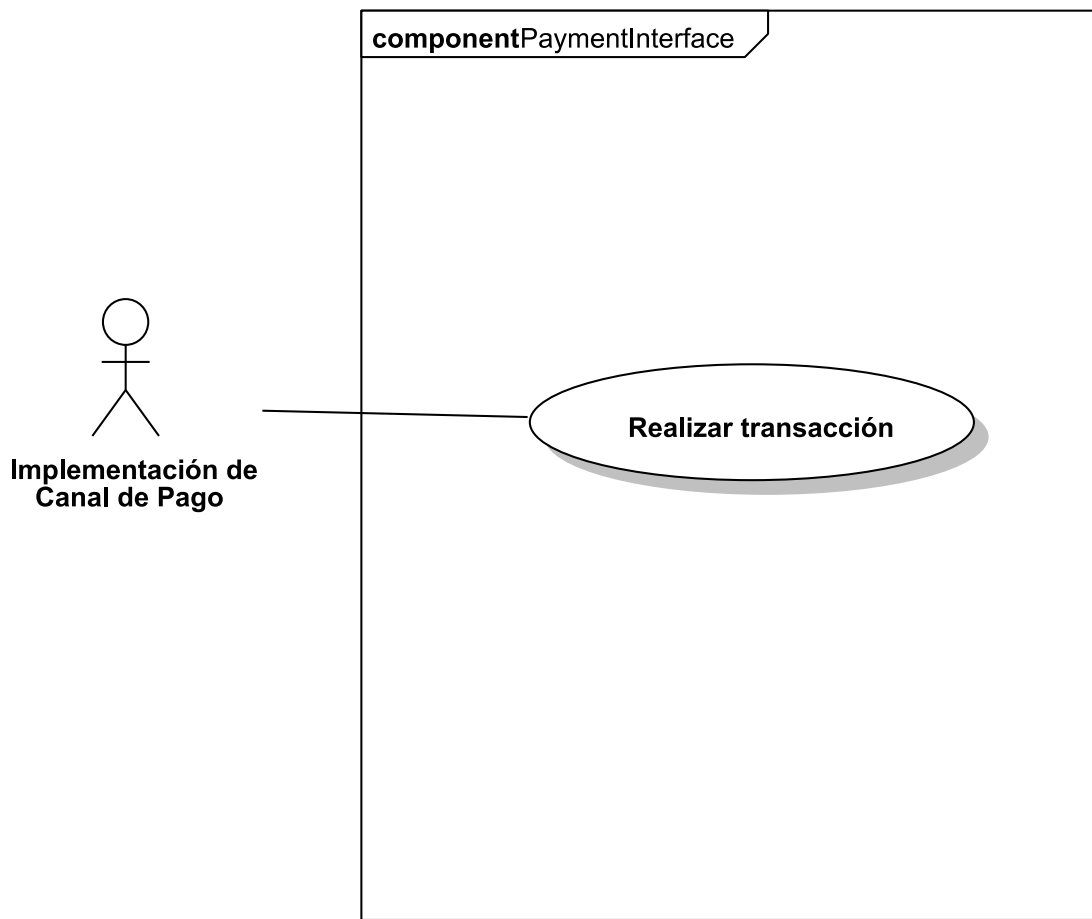


Figura 14. Transaccional V-Financial – Interfaz de Pago: Diagrama de casos de uso nivel 1. (Elaboración propia).

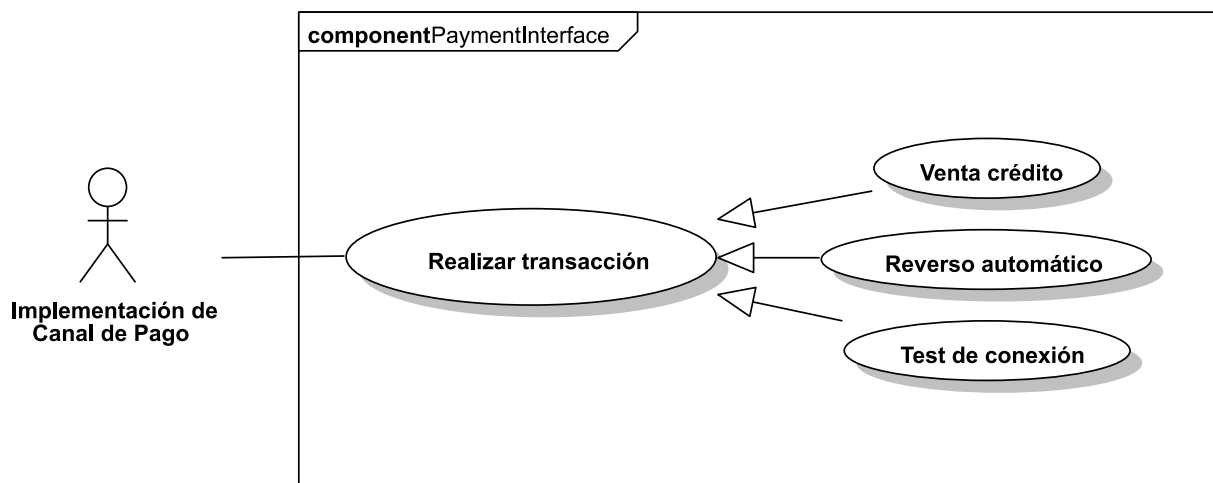


Figura 15. Transaccional V-Financial – Interfaz de Pago: Diagrama de casos de uso nivel 2. (Elaboración propia).

2.3.1.2. Módulo Núcleo Operacional

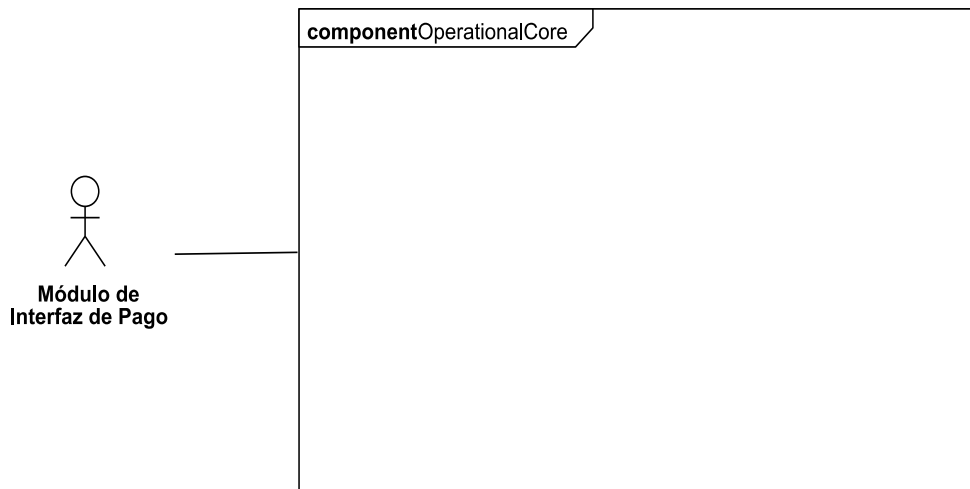


Figura 16. Motor Transaccional V-Financial – Núcleo Operacional: Diagrama de casos de uso nivel 0. (Elaboración propia).

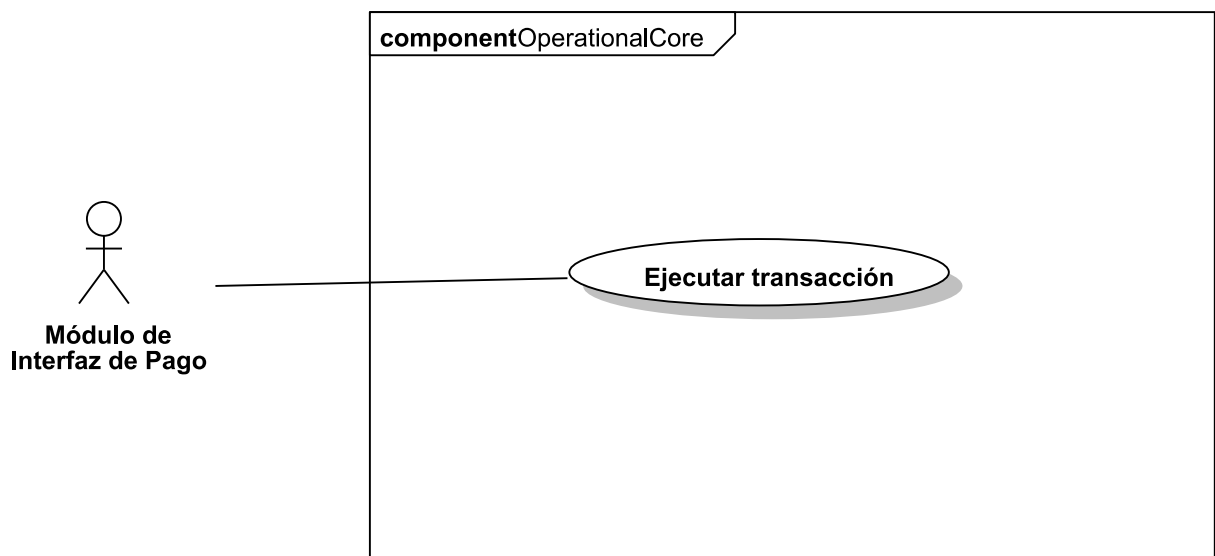


Figura 17. Motor Transaccional V-Financial – Núcleo Operacional: Diagrama de casos de uso nivel 1. (Elaboración propia).

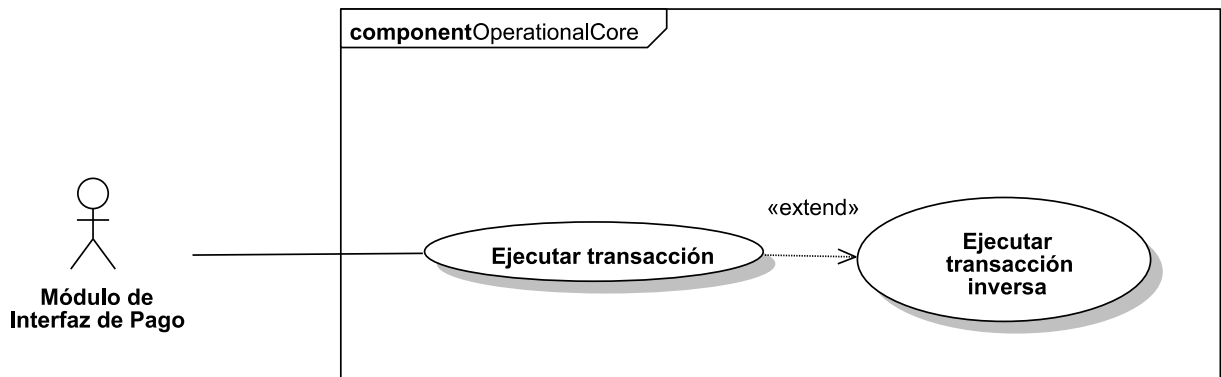


Figura 18. Motor Transaccional V-Financial – Núcleo Operacional: Diagrama de casos de uso nivel 2. (Elaboración propia).

2.3.1.3. Módulo API de Acceso a Datos

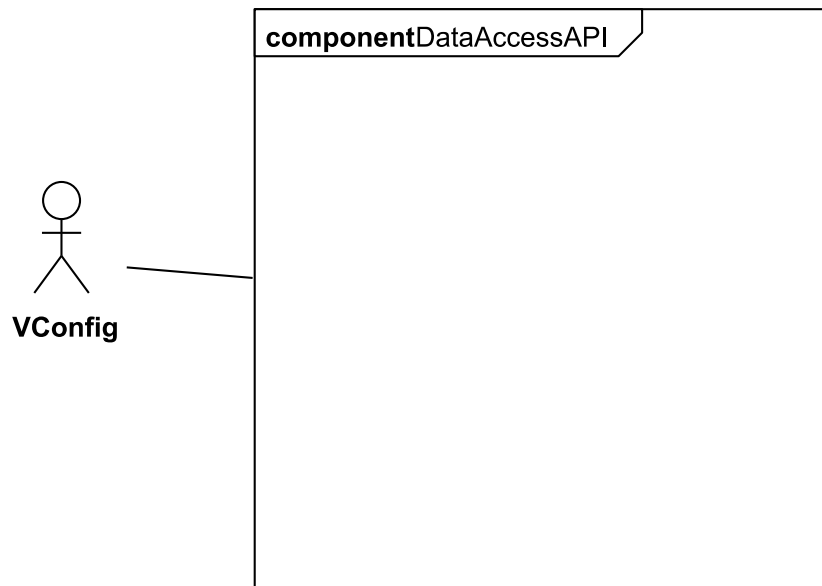


Figura 19. Motor Transaccional V-Financial – API de Acceso a Datos: Diagrama de casos de uso nivel 0. (Elaboración propia).

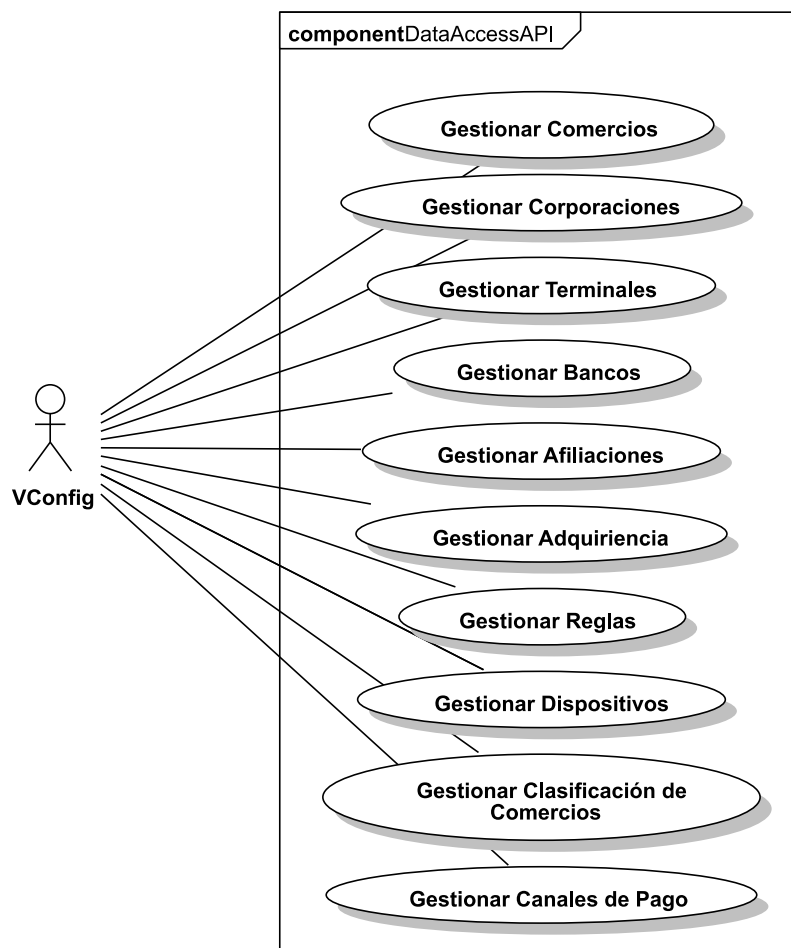


Figura 20. Motor Transaccional V-Financial – API de Acceso a Datos: Diagrama de casos de uso nivel 1. (Elaboración propia).

Las categorizaciones de la API de Acceso a Datos agrupan las funciones en base al tipo de objeto sobre el cual ejercen acción:

- **Gestionar Comercios:** Esta categorización comprende las operaciones que se aplican sobre los comercios. Un comercio representa una entidad jurídica que tiene facultad para realizar operaciones comerciales. Se pueden agrupar múltiples comercios bajo la figura del multicomercio.
- **Gestionar Corporaciones:** Una Corporación agrupa un conjunto de Comercios y/o multicomercios.
- **Gestionar Terminales:** Los Terminales representan los puntos donde se generan las transacciones para un Comercio determinado. Dependiendo de la naturaleza del Canal de Pago

al que este afiliado y otras características del negocio, un Comercio puede tener uno o múltiples terminales

- **Gestionar Bancos:** Operaciones relacionadas con las entidades financieras registradas en el sistema.
- **Gestionar Afiliaciones:** Representan las asociaciones entre los comercios y los canales de pago.
- **Gestionar Adquiriencia:** Gestión de la asociación entre los comercios y las entidades financieras con las cuales guarda una relación comercial.
- **Gestionar Reglas:** Reglas de negocio determinadas por los comercios, estas se usan como criterio decidir a qué afiliación bancaria del comercio se dirigen las transacciones.
- **Gestionar Dispositivos:** Los dispositivos electrónicos que utilizan los terminales para poder realizar transacciones, por ejemplo, un lector de tarjetas.
- **Gestionar Clasificación de Comercios:** Gestión de la clasificación de los comercios según los sectores indicados por el Banco Central de Venezuela.
- **Gestionar Canales de Pago:** Representan todos los métodos a través de las cuales se pueden tomar pagos.

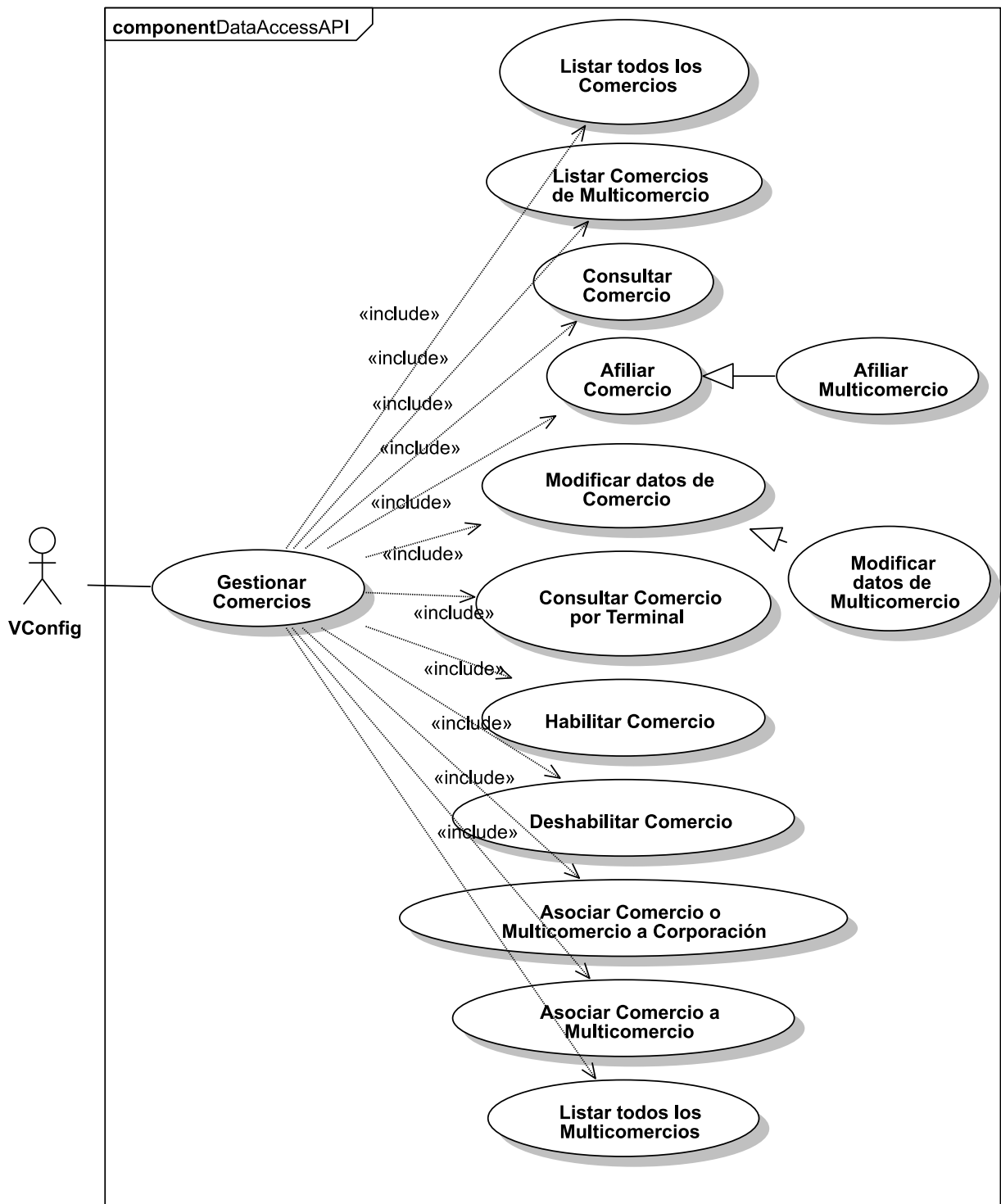


Figura 21. Motor Transaccional V-Financial – API de Acceso a Datos: Diagrama de casos de uso nivel 2 - Gestionar Comercios. (Elaboración propia).

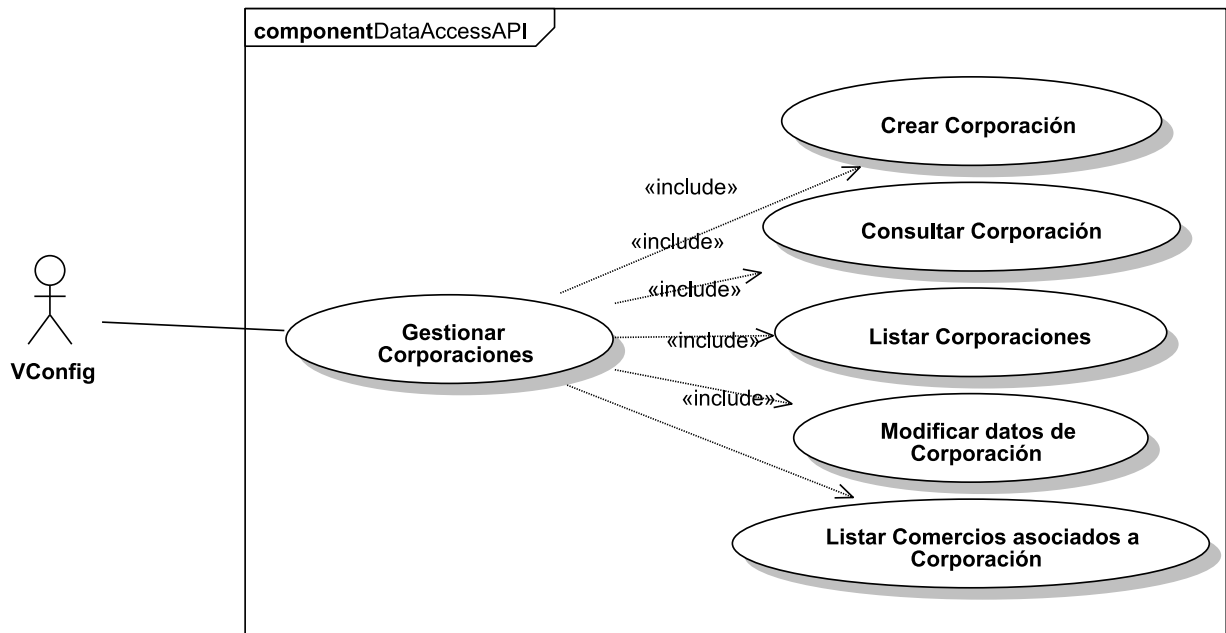


Figura 22. Motor Transaccional V-Financial – API de Acceso a Datos: Diagrama de casos de uso nivel 2 - Gestionar Corporaciones. (Elaboración propia).

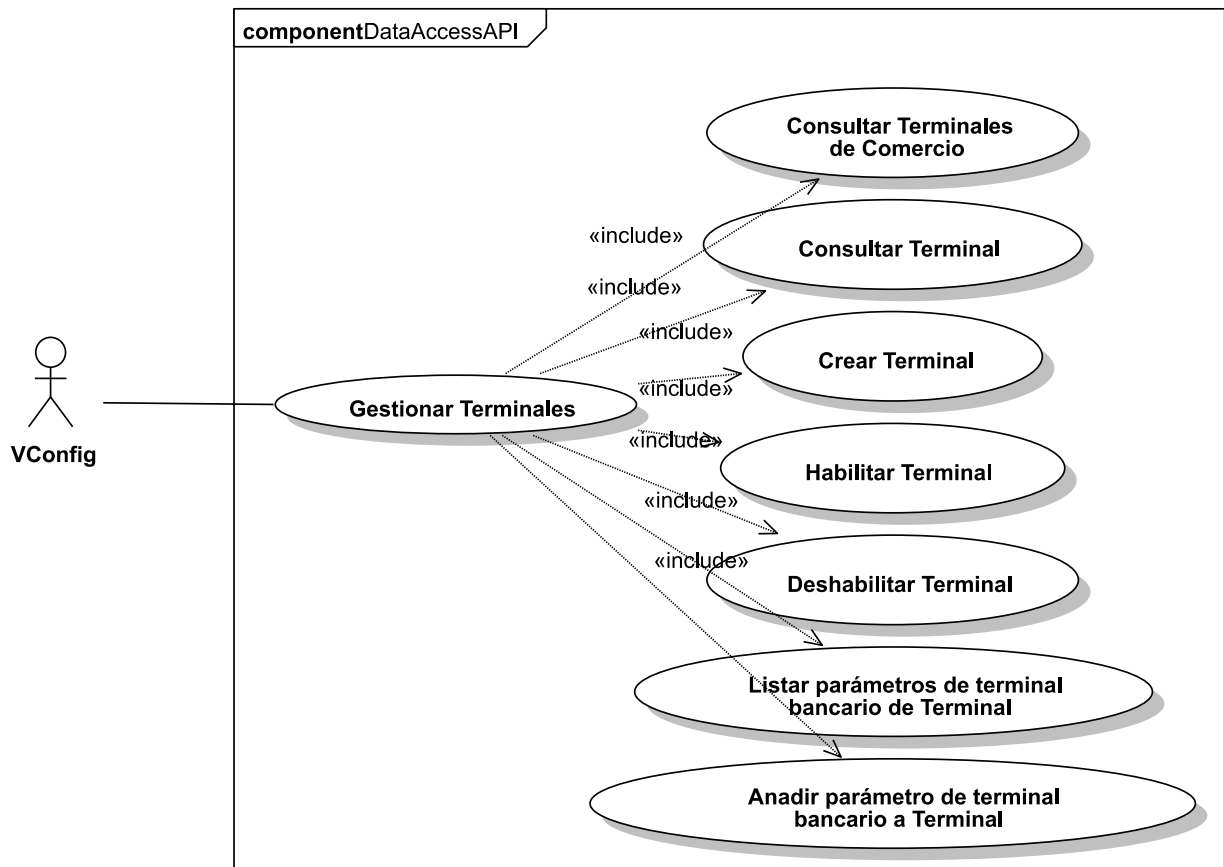


Figura 23. Motor Transaccional V-Financial – API de Acceso a Datos: Diagrama de casos de uso nivel 2 - Gestionar Terminales. (Elaboración propia).

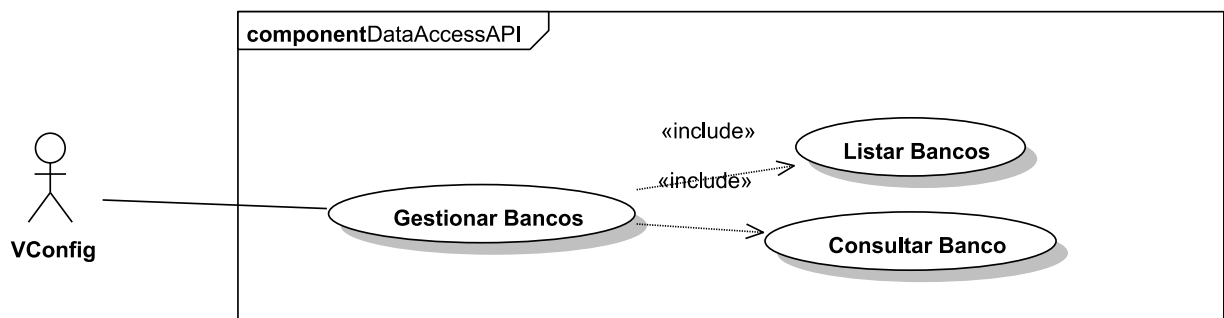


Figura 24. Motor Transaccional V-Financial – API de Acceso a Datos: Diagrama de casos de uso nivel 2 - Gestionar Bancos. (Elaboración propia).

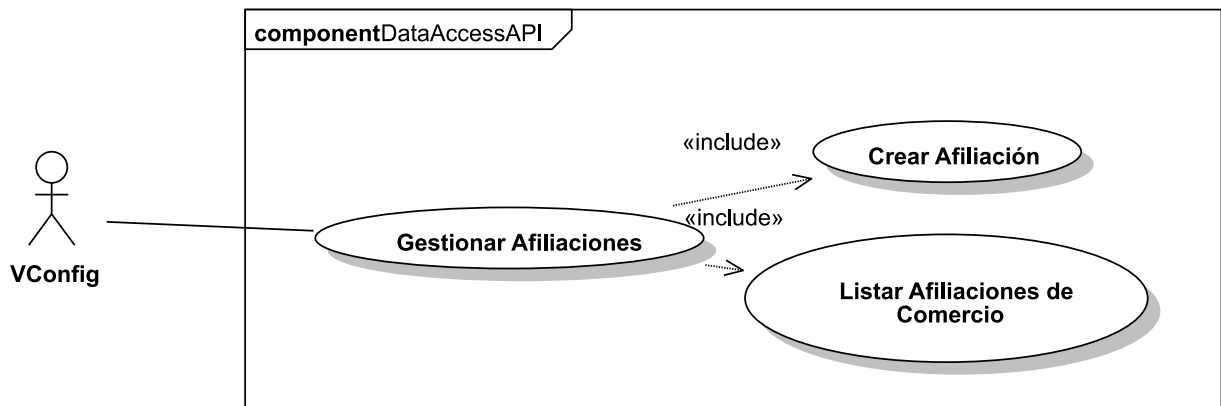


Figura 25. Motor Transaccional V-Financial – API de Acceso a Datos: Diagrama de casos de uso nivel 2 - Gestionar Afiliaciones. (Elaboración propia).

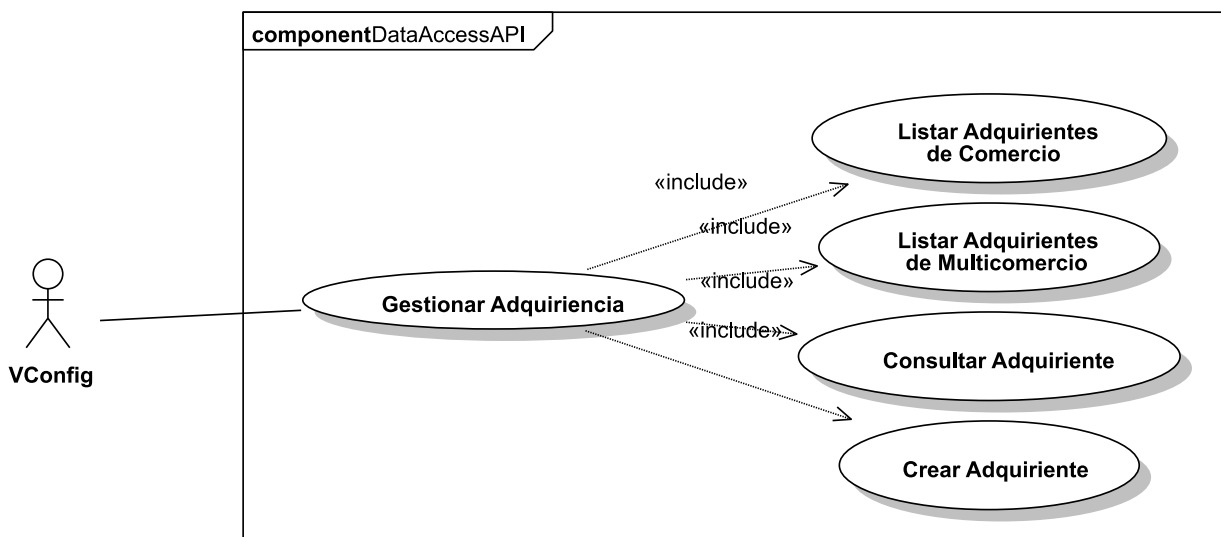


Figura 26. Motor Transaccional V-Financial – API de Acceso a Datos: Diagrama de casos de uso nivel 2 - Gestionar Adquiriencia. (Elaboración propia).

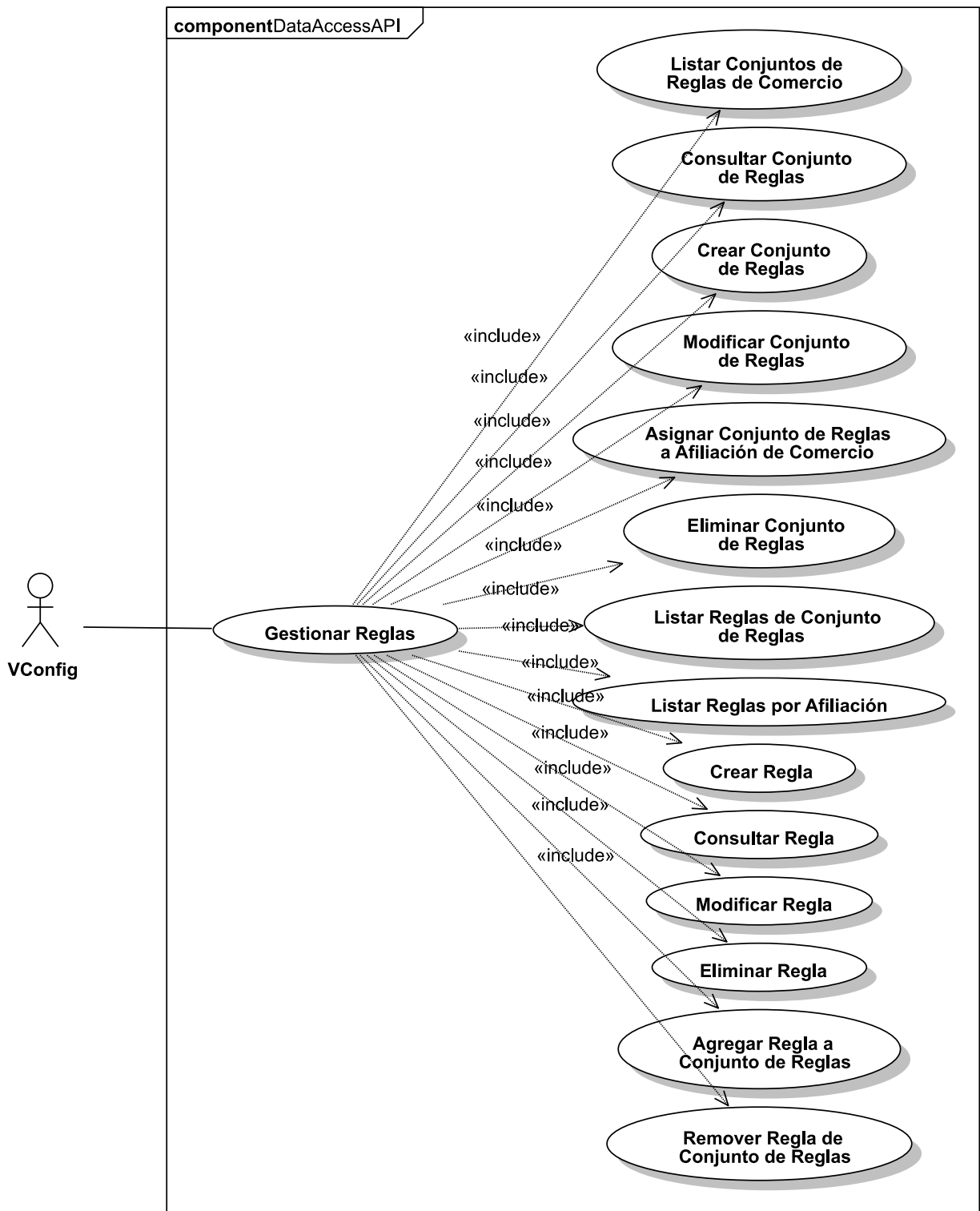


Figura 27. Motor Transaccional V-Financial – API de Acceso a Datos: Diagrama de casos de uso nivel 2 - Gestionar Reglas. (Elaboración propia).

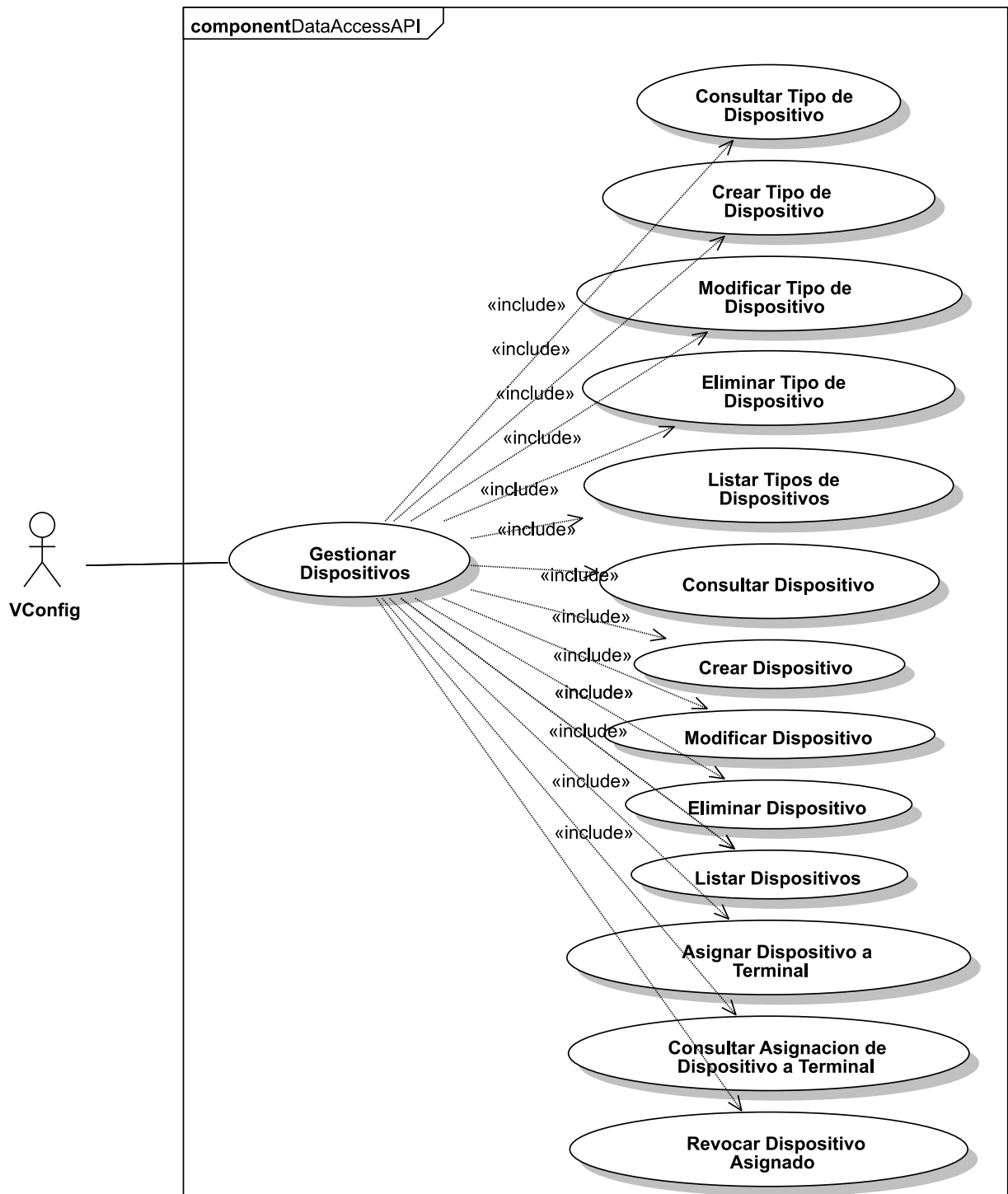


Figura 28. Motor Transaccional V-Financial – API de Acceso a Datos: Diagrama de casos de uso nivel 2 - Gestionar Dispositivos. (Elaboración propia).

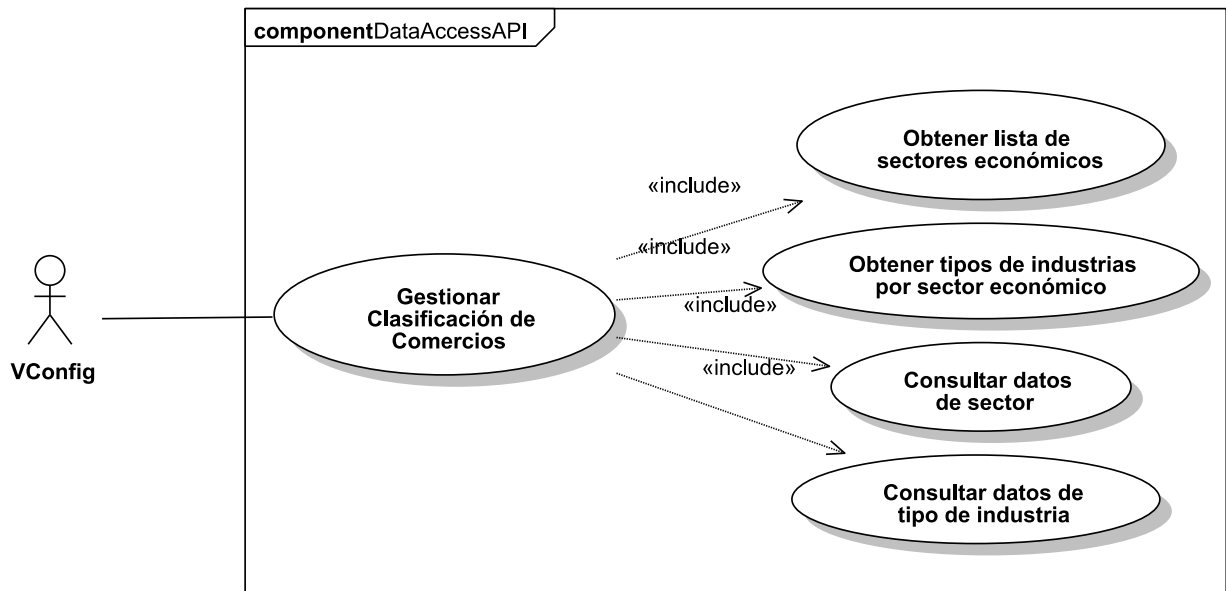


Figura 29. Motor Transaccional V-Financial – API de Acceso a Datos: Diagrama de casos de uso nivel 2 - Gestionar Clasificación de Comercios. (Elaboración propia).

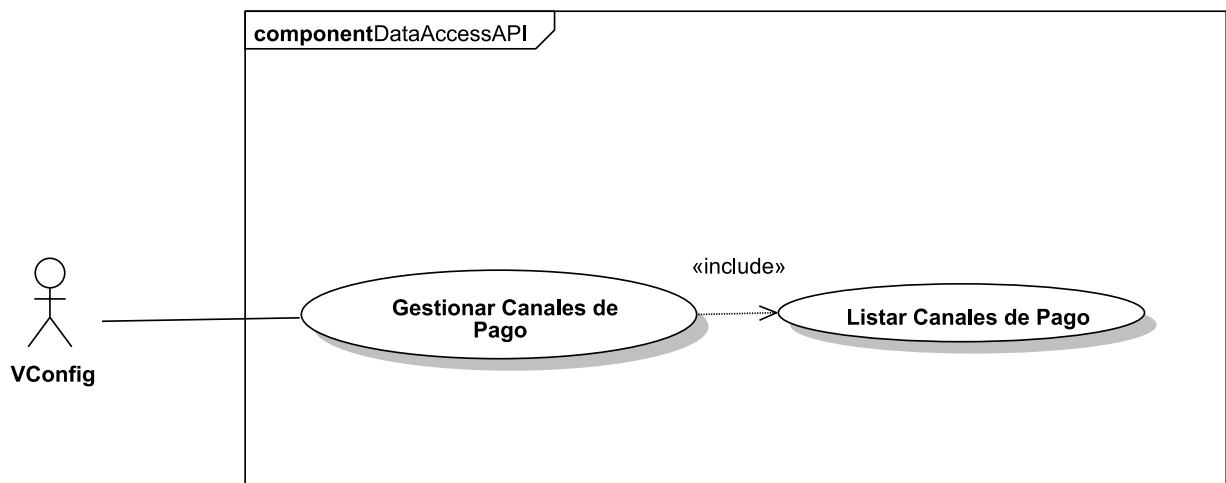


Figura 30. Motor Transaccional V-Financial – API de Acceso a Datos: Diagrama de casos de uso nivel 2 - Gestionar Canales de Pago. (Elaboración propia).

2.3.2. Aplicación Web de Botón de Pago C-Pago

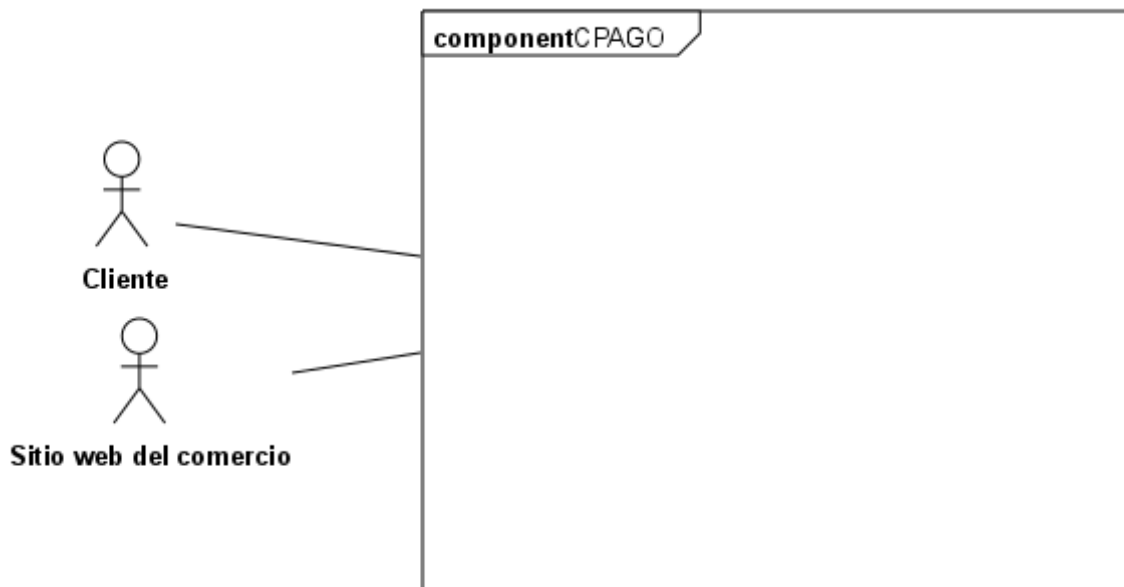


Figura 31. Aplicación C-Pago: Diagrama de casos de uso nivel 0. (Elaboración propia).

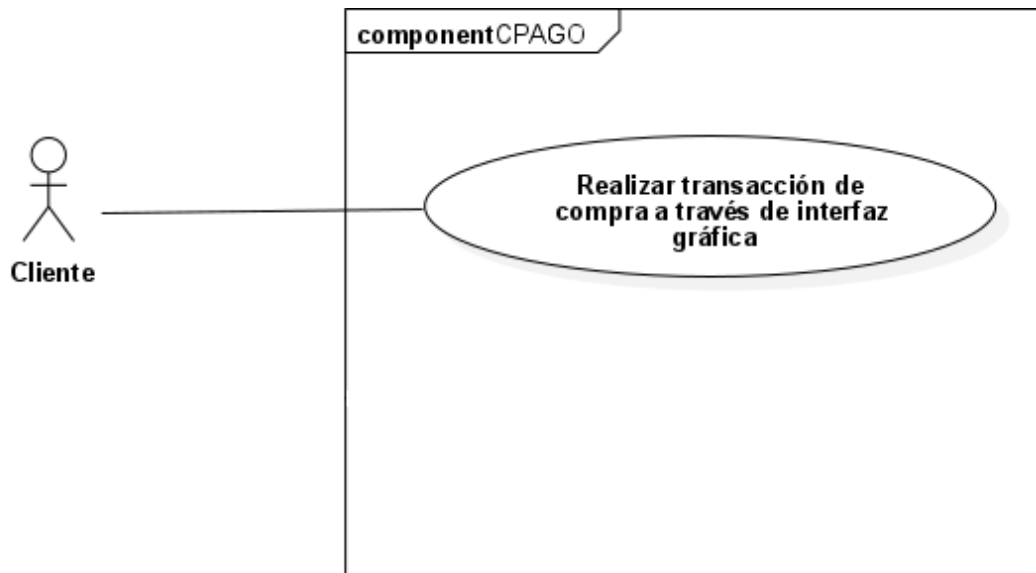


Figura 32. Aplicación C-Pago: Diagrama de casos de uso nivel 1. (Elaboración propia).

2.4. Procesos principales

2.4.1. Proceso general de venta utilizando el canal C-Pago

Para que el proceso de la ejecución de una transacción utilizando la aplicación C-Pago sea completado se siguen una serie de pasos que se ilustran en la siguiente imagen.

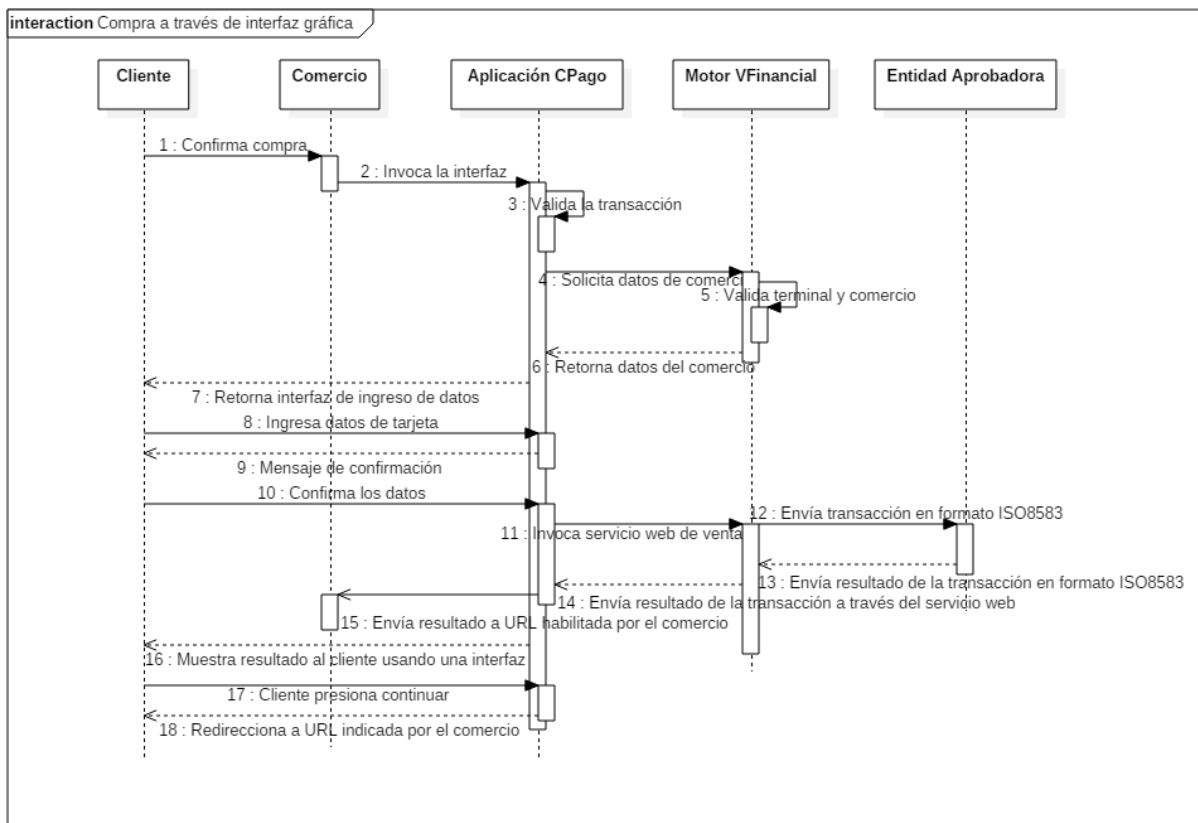


Figura 33. Proceso de compra con tarjeta de crédito a través del canal C-Pago. (Elaboración propia).

2.4.2. Curso de una transacción dentro del Motor Transaccional V-Financial

Tomando en cuenta que el Motor V-Financial realiza la tarea de la construcción de la trama que será enviada a la entidad aprobadora, este constituye el núcleo esencial de la plataforma. El proceso interno para elaborar dicho formato lo podemos dimensionar desde la perspectiva de los dos módulos involucrados:

1. Interfaz de Pago: encargada de recibir la solicitud del servicio web e invocar al módulo encargado de la construcción de la transacción. Los pasos que lleva a cabo para realizar esta tarea se muestran en el siguiente diagrama.

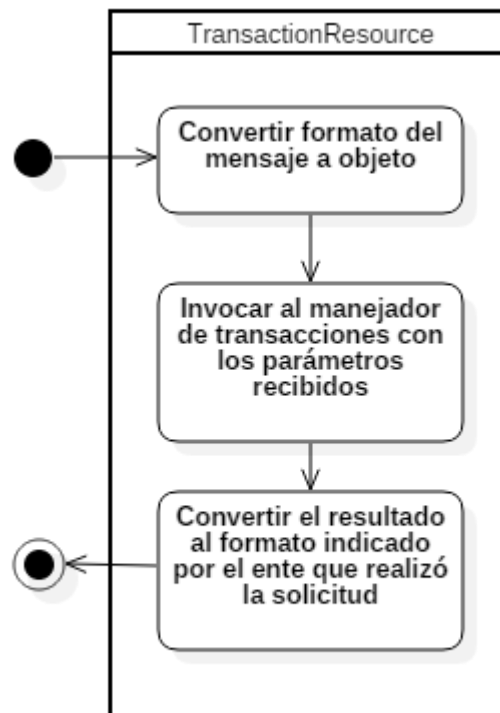


Figura 34. Actividades llevadas a cabo por el módulo de Interfaz de Pago durante la ejecución de una transacción.

2. Núcleo Operacional: Invocado por la Interfaz de Pago, este se encarga de obtener todos los parámetros requeridos por un tipo de transacción, para luego convertirlos e insertarlos en el campo del estándar ISO 8583 correspondiente. Los pasos que realiza se ilustran a continuación.

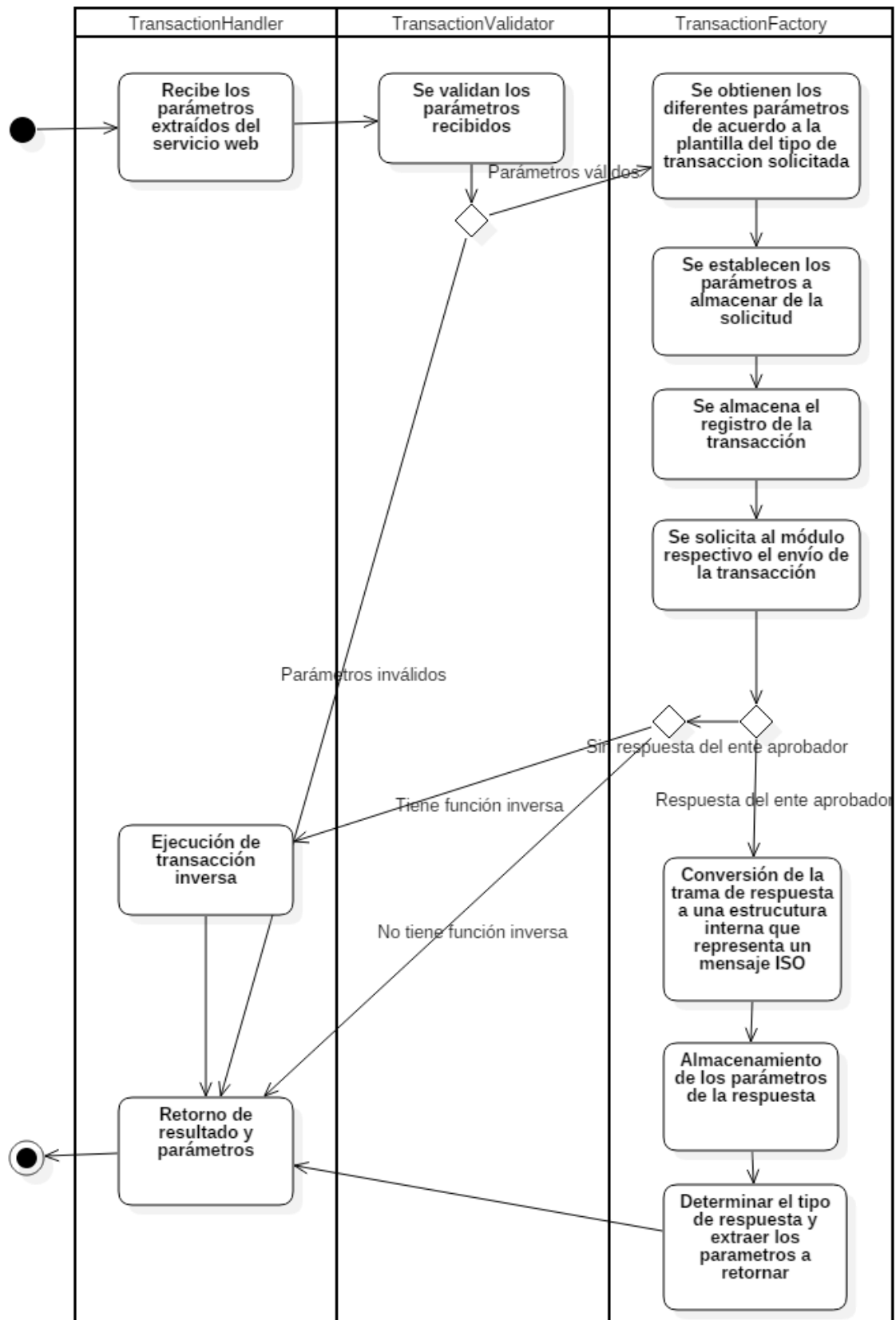
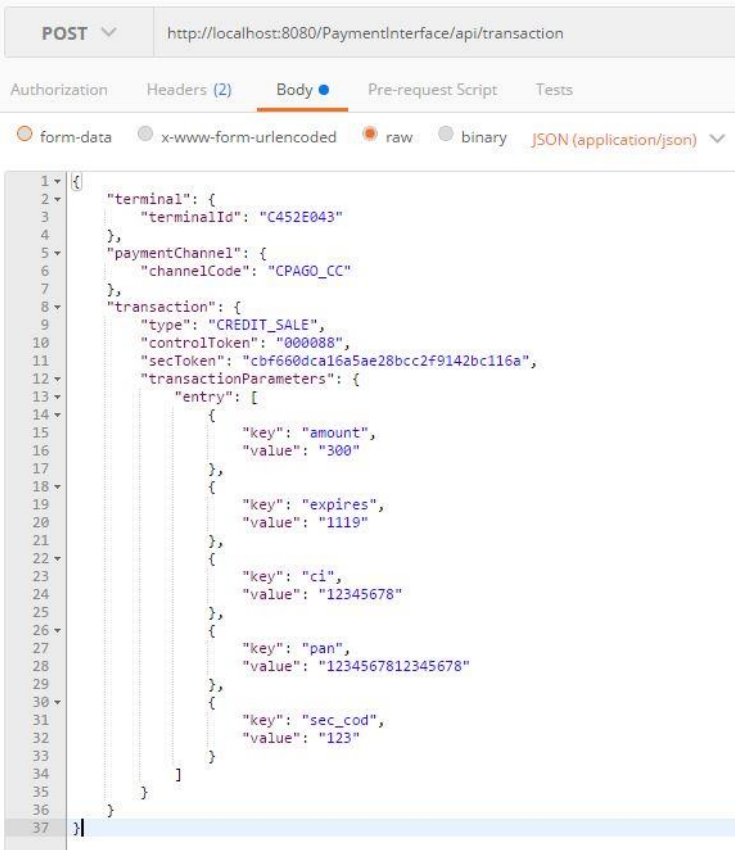


Figura 35. Actividades llevadas a cabo por el módulo Núcleo Operacional durante la ejecución de una transacción.

2.4.3. Construcción de una trama ISO 8583

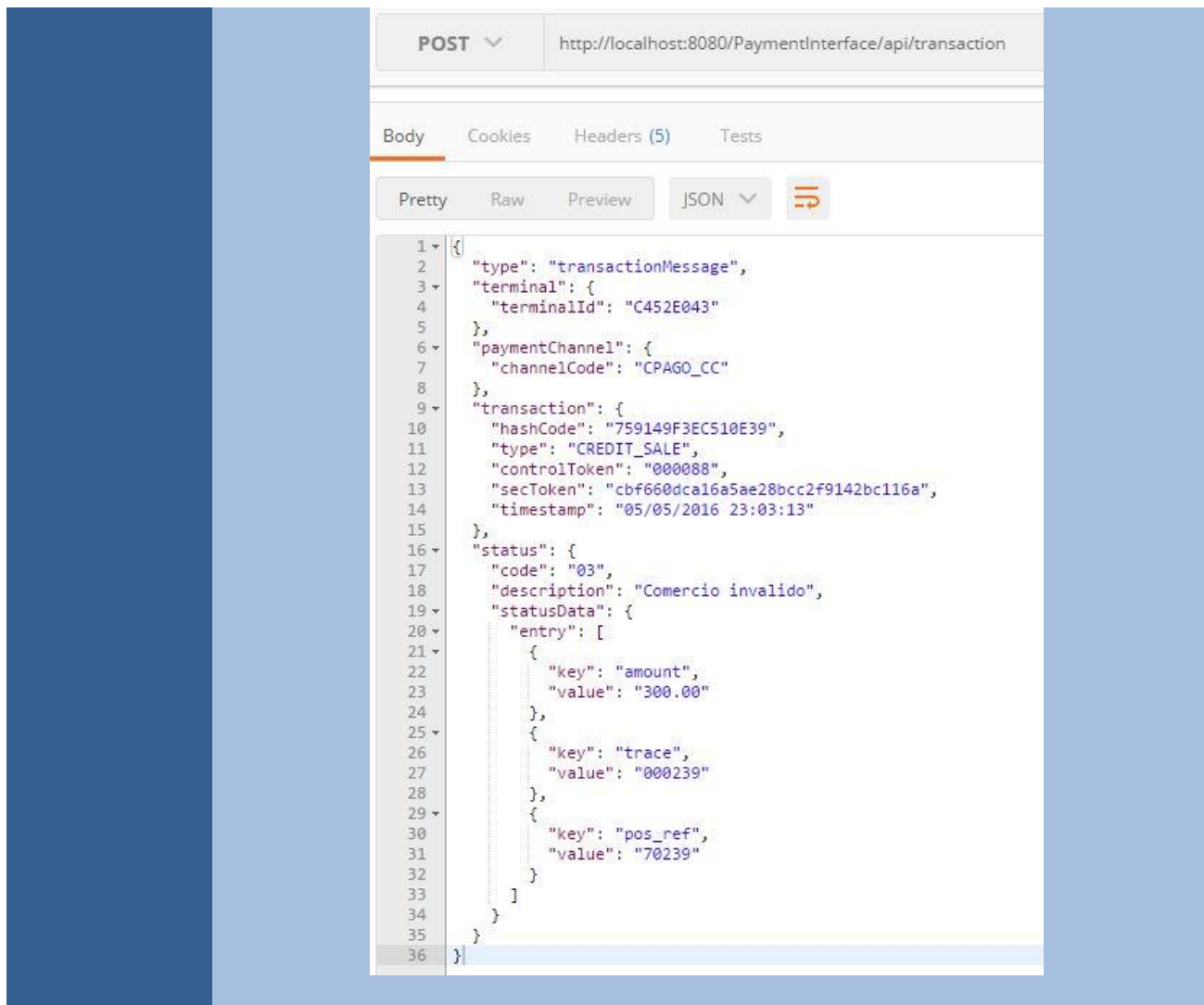
Tabla 10. Construcción de una trama ISO 8583. (Elaboración propia).

<p>Invocación del servicio web</p>	 <pre> 1 { 2 "terminal": { 3 "terminalId": "C452E043" 4 }, 5 "paymentChannel": { 6 "channelCode": "CPAGO_CC" 7 }, 8 "transaction": { 9 "type": "CREDIT_SALE", 10 "controlToken": "000088", 11 "secToken": "cbf660dca16a5ae28bcc2f9142bc116a", 12 "transactionParameters": { 13 "entry": [14 { 15 "key": "amount", 16 "value": "300" 17 }, 18 { 19 "key": "expires", 20 "value": "1119" 21 }, 22 { 23 "key": "ci", 24 "value": "12345678" 25 }, 26 { 27 "key": "pan", 28 "value": "1234567812345678" 29 }, 30 { 31 "key": "sec_cod", 32 "value": "123" 33 } 34] 35 } 36 } 37 } </pre>
<p>Análisis de la plantilla y los campos recibidos</p>	<pre> DEBUG v.c.transactions.TransactionFactory - TYPE: 0200 DEBUG v.c.transactions.TransactionFactory - F 2(LLVAR): 1234567812345678 -> '1234567812345678' DEBUG v.c.transactions.TransactionFactory - F 3(NUMERIC): 3000 -> '003000' DEBUG v.c.transactions.TransactionFactory - F 4(AMOUNT): 300 -> '000000030000' DEBUG v.c.transactions.TransactionFactory - F 11(NUMERIC): 239 -> '000239' DEBUG v.c.transactions.TransactionFactory - F 14(NUMERIC): 1119 -> '1119' DEBUG v.c.transactions.TransactionFactory - F 22(NUMERIC): 0812 -> '0812' DEBUG v.c.transactions.TransactionFactory - F 24(NUMERIC): 0003 -> '0003' DEBUG v.c.transactions.TransactionFactory - F 25(NUMERIC): 00 -> '00' DEBUG v.c.transactions.TransactionFactory - F 32(LLVAR): 441132 -> '441132' DEBUG v.c.transactions.TransactionFactory - F 41(NUMERIC): 3030303034303032 -> '3030303034303032' DEBUG v.c.transactions.TransactionFactory - F 42(NUMERIC): 30303130303030303037362020202020 -> '30303130303030303037362020202020' DEBUG v.c.transactions.TransactionFactory - F 48(NUMERIC): CompositeField[NUMERIC, NUMERIC, NUMERIC] -> '00143031323330303132333435363738' DEBUG v.c.transactions.TransactionFactory - F 62(NUMERIC): 0006313730323339 -> '0006313730323339' DEBUG v.c.transactions.TransactionFactory - F 63(ALPHA): 001000083337303030303031 -> '001000083337303030303031' </pre>
<p>Trama ISO 8583 a enviar</p>	<pre> 2016-05-05T23:03:13.293-0400 Información: 23:03:13.293 [http-listener-1(1)] TRACE v.c.transactions.TransactionFactory - 600007000002007024058100C10006161234567812345678003000000000030000000239111908120003000 644113230303030343030323030313030303030373620202020200014303132333030313233343536373800 06313730323339001000083337303030303031 </pre>

2.4.4. Análisis de una trama ISO 8583

Tabla 11. Análisis de una trama ISO 8583. (Elaboración propia).

<p>Trama ISO 8583 recibida</p>	<pre>2016-05-05T23:03:13.352-0400 Información: 23:03:13.352 [http-listener-1(1)] TRACE v.c.transactions.TransactionFactory - msgResponse: 60000000070210303801000A80000000300000000300000002392305580505000336313236303330303 2333930333030303034303032</pre>
<p>Análisis de la plantilla y los campos recibidos</p>	<pre>DEBUG v.c.transactions.TransactionFactory - TYPE: 0210 DEBUG v.c.transactions.TransactionFactory - F 3(NUMERIC): 003000 -> '003000' DEBUG v.c.transactions.TransactionFactory - F 4(AMOUNT): 300.00 -> '000000030000' DEBUG v.c.transactions.TransactionFactory - F 11(NUMERIC): 000239 -> '000239' DEBUG v.c.transactions.TransactionFactory - F 12(TIME): Thu May 05 23:05:58 PYT 2016 -> '230558' DEBUG v.c.transactions.TransactionFactory - F 13(DATE4): Thu May 05 12:00:00 PYT 2016 -> '0505' DEBUG v.c.transactions.TransactionFactory - F 24(NUMERIC): 0003 -> '0003' DEBUG v.c.transactions.TransactionFactory - F 37(ALPHA): 363132363033303030333339 -> '363132363033303030333339' DEBUG v.c.transactions.TransactionFactory - F 39(ALPHA): 3033 -> '3033' DEBUG v.c.transactions.TransactionFactory - F 41(NUMERIC): 3030303034303032 -> '3030303034303032'</pre>
<p>Respuesta del servicio web</p>	



```
POST http://localhost:8080/PaymentInterface/api/transaction

Body
Pretty Raw Preview JSON

1 {
2   "type": "transactionMessage",
3   "terminal": {
4     "terminalId": "C452E043"
5   },
6   "paymentChannel": {
7     "channelCode": "CPAGO_CC"
8   },
9   "transaction": {
10    "hashCode": "759149F3EC510E39",
11    "type": "CREDIT_SALE",
12    "controlToken": "000088",
13    "secToken": "cbf660dca16a5ae28bcc2f9142bc116a",
14    "timestamp": "05/05/2016 23:03:13"
15  },
16  "status": {
17    "code": "03",
18    "description": "Comercio invalido",
19    "statusData": {
20      "entry": [
21        {
22          "key": "amount",
23          "value": "300.00"
24        },
25        {
26          "key": "trace",
27          "value": "000239"
28        },
29        {
30          "key": "pos_ref",
31          "value": "70239"
32        }
33      ]
34    }
35  }
36 }
```

2.4.5. Invocación general de un método de la API de Acceso a Datos

El módulo que comprende la API de Acceso a Datos se encarga de proporcionar una interfaz REST (Representational State Transfer) para la gestión de los elementos que participan en el proceso transaccional (comercios, bancos, entre otros). Acepta los formatos JSON y XML para la representación de los objetos. Las funciones de la API se cimentan sobre las cuatro funciones fundamentales que se aplican a los objetos: consultar, crear, modificar y eliminar.

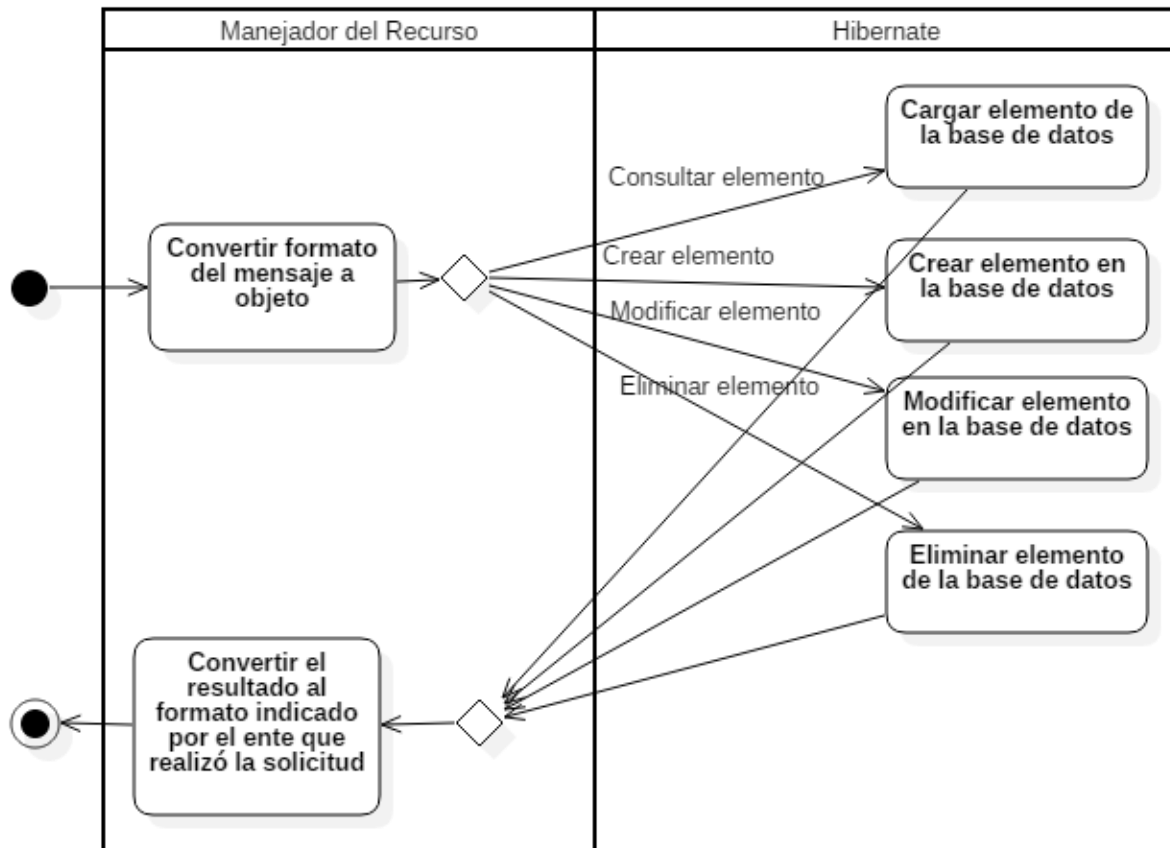


Figura 36. Actividades generales llevadas a cabo por el módulo API de Acceso a Datos para la ejecución de una función.

2.5. Modelado de Datos

2.5.1. Diagrama de base de datos Motor V-Financial

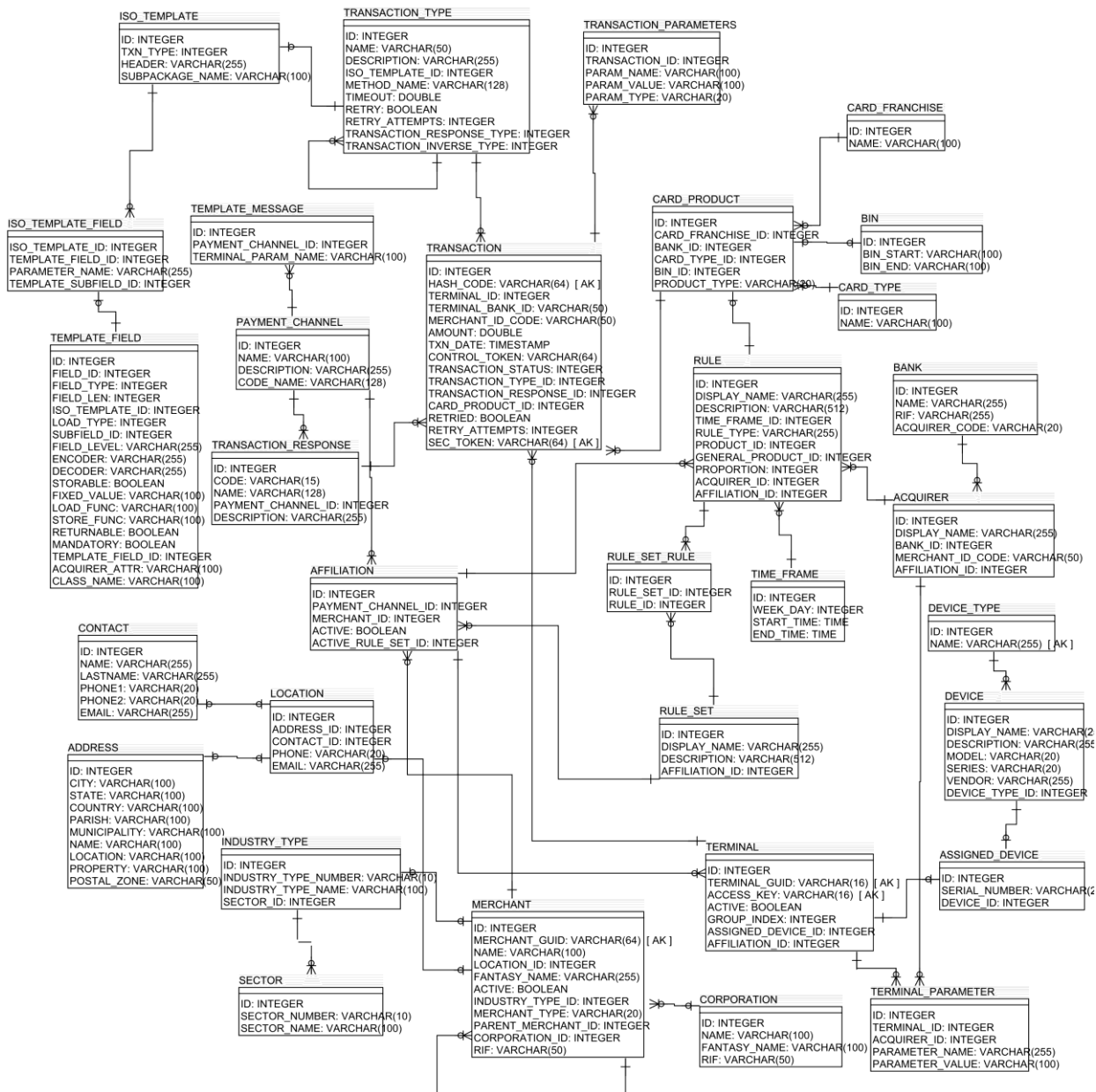


Figura 37. Diagrama de base de datos Motor V-Financial. (Elaboración propia).

2.5.2. Diccionario de Datos Motor V-Financial

Tabla	Descripción	Campo	Descripción	Tipo de dato	¿Null?
acquirer	Almacena los datos de la clase persistente para	id	Identificador incremental generado por el motor de base	integer	No

Tabla	Descripción	Campo	Descripción	Tipo de dato	¿Null?
	representar un Adquiriente		de datos		
		display_name	Nombre para mostrar	varchar	No
		bank_id	Campo identificador del terminal en el banco (41 ISO8583)	integer	No
		merchant_id_cod e	Comercio al que está asociada la Cuenta	varchar	No
		affiliation_id	Identificador del Comercio al que está asociado el Adquiriente	integer	No
address	Almacena los datos de la clase persistente para representar una Dirección	id	Identificador incremental generado por el motor de base de datos	integer	No
		city	Ciudad	varchar	Si
		state	Estado	varchar	Si
		country	País	varchar	Si
		line1	Línea 1 de dirección	varchar	Si
		line2	Línea 2 de dirección	varchar	Si
affiliation	Almacena los datos de la clase persistente para representar una Afiliación	id	Identificador incremental generado por el motor de base de datos	integer	No
		payment_channel_id	Identificador del Canal de Pago al cual está afiliado el Comercio	integer	No
		merchant_id	Comercio afiliado	integer	No
		active	Indica si el Comercio está activo para este Canal de Pago	boolean	No
		active_rule_set_id	Identificador del Conjunto de Reglas activo del Comercio para esta Afiliación	integer	Si
assigned_device	Almacena los datos de la clase persistente para representar un Dispositivo asignado a un Terminal	id	Identificador incremental generado por el motor de base de datos	integer	No
		serial_number	Serial del dispositivo asignado	varchar	Si
		device_id	Id de Dispositivo asignado	integer	No
bank	Almacena los datos de la clase persistente para	id	Identificador incremental generado por el motor de base	integer	No

Tabla	Descripción	Campo	Descripción	Tipo de dato	¿Null?
	representar un Banco		de datos		
		name	Nombre para mostrar	varchar	No
bin	Bines para identificar los productos	rif	Registro de Información Fiscal	varchar	No
		id	Identificador incremental generado por el motor de base de datos	integer	No
		bin_start	Comienzo del rango	varchar	No
card_franchise	Franquicias	bin_end	Fin del rango	varchar	No
		id	Identificador incremental generado por el motor de base de datos	integer	No
card_product	Marcas de tarjetas	name	Nombre de la franquicia	varchar	No
		id	Identificador incremental generado por el motor de base de datos	integer	No
card_type	Tipos de tarjetas	card_franchise_id	Franquicia	integer	No
		bank_id	Banco	integer	No
		card_type_id	Tipo de tarjeta	integer	No
		bin_id	Rango de bin para identificar el tipo de tarjeta	integer	No
		product_type	Campo para discriminar el tipo de producto (general, bank)	varchar	No
contact	Almacena los datos de la clase persistente para representar un Contacto	id	Identificador incremental generado por el motor de base de datos	integer	No
		name	Nombre del tipo de tarjeta	varchar	No
contact	Almacena los datos de la clase persistente para representar un Contacto	name	Nombre del contacto	varchar	Si
		lastname	Apellido del contacto	varchar	Si
		phone1	Número de teléfono del contacto	varchar	Si
		phone2	Número de teléfono	varchar	Si

Tabla	Descripción	Campo	Descripción	Tipo de dato	¿Null?
			alternativo del contacto		
		email	Dirección de correo electrónico del contacto	varchar	Si
corporation	Almacena los datos de la clase persistente para representar una Corporación	id	Identificador incremental generado por el motor de base de datos	integer	No
		name	Nombre o razón social	varchar	Si
		fantasy_name	Nombre fantasía	varchar	No
		rif	Registro de identificación fiscal	varchar	Si
device	Almacena los datos de la clase persistente para representar un Dispositivo	id	Identificador incremental generado por el motor de base de datos	integer	No
		display_name	Nombre para mostrar	varchar	Si
		description	Descripción del dispositivo	varchar	Si
		model	Modelo del dispositivo	varchar	Si
		series	Serie del dispositivo	varchar	Si
		vendor	Fabricante del dispositivo	varchar	Si
		device_type_id	Identificador de Tipo de Dispositivo	integer	No
device_type	Almacena los datos de la clase persistente para representar el tipo de Dispositivo	id	Identificador incremental generado por el motor de base de datos	integer	No
		name	Nombre del Tipo de Dispositivo	varchar	Si
industry_type	Almacena los datos de la clase persistente para representar un Tipo de Industria según categorización del Banco Central de Venezuela	id	Identificador incremental generado por el motor de base de datos	integer	No
		industry_type_number	Nombre del Tipo de Industria	varchar	Si
		industry_type_name	Numero representativo	varchar	No
		sector_id	Identificador del Sector al que está asociado el Tipo de Industria	integer	No
iso_template	Almacena los datos de la clase persistente para representar una	id	Identificador incremental generado por el motor de base de datos	integer	No

Tabla	Descripción	Campo	Descripción	Tipo de dato	¿Null?
	Plantilla ISO que almacena los parametros para mapear campos de una transacción a campos ISO según el esquema elegido	txn_type	Tipo de operación ISO a la cual está asociada la plantilla (MTI 8583)	integer	No
		header	Encabezado del formato ISO	varchar	Si
		subpackage_name	Nombre del paquete donde se encuentran las clases para el manejo de los campos	varchar	Si
iso_template_field	Asociación entre una Plantilla ISO y una plantilla de campo/sub-campo	iso_template_id	Identificador de la Plantilla ISO	integer	No
		template_field_id	Identificador de la Plantilla de campo ISO	integer	Si
		parameter_name	Nombre del parámetro al que está asociado el campo o subcampo	varchar	No
		template_subfield_id	Identificador de la Plantilla de subcampo ISO	integer	Si
location	Almacena los datos de la clase persistente para representar una Ubicación	id	Identificador incremental generado por el motor de base de datos	integer	No
		address_id	Dirección de la ubicación	integer	No
		contact_id	Contacto de la ubicación	integer	No
		phone	Número de teléfono de la ubicación	varchar	Si
		email	Dirección de correo electrónico de la ubicación	varchar	Si
merchant	Almacena los datos de la clase persistente para representar un Comercio	id	Identificador incremental generado por el motor de base de datos	integer	No
		merchant_guid	Identificador único del comercio en la base de datos	varchar	No
		name	Nombre del comercio	varchar	No
		rif	Registro de Identificación Fiscal del Comercio	varchar	No
		location_id	Ubicación del comercio	integer	No
		fantasy_name	Nombre fantasía	varchar	No
		active	Comercio activo	boolean	No

Tabla	Descripción	Campo	Descripción	Tipo de dato	¿Null?
		industry_type_id	Identificador de Tipo de Industria del Comercio	integer	No
		merchant_type	Campo para discriminar el tipo de comercio (simple, multi)	varchar	No
		parent_merchant_id	Identificador del Multicomercio al cual pertenece el Comercio	integer	Si
		corporation_id	Identificador de la Corporación a la cual pertenece el Comercio	integer	Si
payment_channel	Almacena los datos de la clase persistente para representar un Canal de Pago	id	Identificador incremental generado por el motor de base de datos	integer	No
		name	Nombre del canal de pago	varchar	No
		description	Descripción opcional del canal de pago	varchar	Si
		code_name	Código que será enviado por el Terminal	varchar	No
rule	Almacena los datos de la clase persistente para representar una Regla	id	Identificador incremental generado por el motor de base de datos	integer	No
		display_name	Nombre para mostrar	varchar	No
		description	Descripción	varchar	Si
		rule_type	Campo para discriminar el tipo de regla (proportion, product, general)	varchar	No
		acquirer_id	Identificador de Adquiriente	integer	No
		affiliation_id	Identificador de la afiliación a la que está asociada la Regla	integer	No
		product_id	Identificador del Producto	integer	Si
		general_product_id	Identificador del Producto General	integer	Si
		proportion	Proporción	integer	Si
rule_set	Almacena los datos de la clase persistente para representar un	id	Identificador incremental generado por el motor de base de datos	integer	No

Tabla	Descripción	Campo	Descripción	Tipo de dato	¿Null?
	Conjunto de Reglas	display_name	Nombre para mostrar	varchar	No
		description	Descripción	varchar	Si
		time_frame_id	Rango de tiempo	integer	No
		affiliation_id	Identificador de la Afiliación a la que está asociado el Conjunto de Reglas	integer	No
rule_set_rule	Asociación entre una Regla y un Conjunto de Reglas	id	Identificador incremental generado por el motor de base de datos	integer	No
		rule_set_id	Identificador de conjunto de reglas	integer	No
		rule_id	Identificador de regla	integer	No
sector	Almacena los datos de la clase persistente para representar un Sector económico que agrupa Tipos de Industrias según categorización del Banco Central de Venezuela	id	Identificador incremental generado por el motor de base de datos	integer	No
		sector_number	Nombre del Sector	varchar	Si
		sector_name	Número del Sector	varchar	No
template_field	Representa una Plantilla de Campo o Subcampo con la descripción de un campo o subcampo transaccional en la plantilla de un esquema almacenado en el motor	id	Identificador incremental generado por el motor de base de datos	integer	No
		field_id	Identificador del campo ISO	integer	No
		field_type	Tipo de campo ISO de acuerdo a la librería j8583	integer	No
		field_len	Longitud del campo ISO	integer	No
		iso_template_id	Identificador de la Plantilla ISO a la que está asociado	integer	No
		load_type	Tipo de Carga del parámetro	integer	No
		subfield_type	Identificador e índice del subcampo dentro del campo al que pertenece	integer	Si
		field_level	Campo para discriminar el tipo de campo (Field, Subfield)	varchar	No

Tabla	Descripción	Campo	Descripción	Tipo de dato	¿Null?
		encoder	Nombre de la función codificadora del subcampo	varchar	Si
		decoder	Nombre de la función decodificadora del camp	varchar	Si
		storable	Indica si el parámetro debe almacenarse en la base de datos	boolean	No
		fixed_value	Almacena el valor del parámetro en caso de que sea tipo FIXED	varchar	Si
		load_func	Nombre de la función a aplicar durante la carga del parámetro	varchar	Si
		store_func	Nombre de la función a aplicar durante el almacenamiento del parámetro	varchar	Si
		returnable	Indica si el parámetro debe retornarse en la respuesta de la solicitud	boolean	No
		mandatory	Indica si es obligatoria la presencia del parámetro en el mensaje	boolean	No
		template_field_id	Plantilla de campo ISO a la que pertenece el subcampo	integer	Si
		acquirer_attr	Nombre del atributo de la clase Acquirer al que está asociado en caso de ser de tipo RULE	varchar	Si
		class_name	Nombre de la clase que contiene los métodos para manejar el campo	varchar	Si
template_message	Almacena los datos de la clase persistente para representar una Plantilla de Mensajes	id	Identificador incremental generado por el motor de base de datos	integer	No
		payment_channel_id	Identificador del Canal de Pago al que está asociada la Plantilla de Mensajes	integer	No
		terminal_param_name	Nombre del parámetro para los terminales	varchar	Si

Tabla	Descripción	Campo	Descripción	Tipo de dato	¿Null?
			bancarios asociados a este canal de pago		
terminal	Almacena los datos de la clase persistente para representar un Terminal	id	Identificador incremental generado por el motor de base de datos	integer	No
		terminal_guid	Identificador único del terminal en la base de datos	varchar	No
		access_key	Llave para autenticar al terminal	varchar	No
		active	Terminal activo	boolean	No
		group_index	Número de Terminal dentro del Grupo de Terminales	integer	No
		assigned_device_id	Identificador de Dispositivo Asignado	integer	Si
		affiliation_id	Identificador de la Afiliación a la que está asociado el Terminal	integer	No
terminal_parameter	Almacena los datos de la clase persistente para representar un Parámetro de Terminal	id	Identificador incremental generado por el motor de base de datos	integer	No
		terminal_id	Identificador de Terminal al que está asociado el Parámetro de Terminal	integer	No
		acquirer_id	Identificador del Adquiriente al que está asociado el Parámetro de Terminal	integer	No
		parameter_name	Nombre del Parámetro de Terminal	varchar	No
		parameter_value	Valor del Parámetro de Terminal	varchar	No
time_frame	Almacena los datos de la clase persistente para representar un Rango de Tiempo para la aplicación de una Regla	id	Identificador incremental generado por el motor de base de datos	integer	No
		week_day	Día de la semana	integer	No
		start_time	Hora inicio	time	No
		end_time	Hora fin	time	No
transaction	Almacena los datos de la clase persistente para representar una	id	Identificador incremental generado por el motor de base de datos	integer	No

Tabla	Descripción	Campo	Descripción	Tipo de dato	¿Null?
	Transacción	hash_code	Código único generado para la transacción	varchar	No
		terminal_id	Identificador del terminal que generó la transacción	integer	No
		terminal_bank_id	Campo 41 ISO8583	varchar	Si
		merchant_id_code	Campo 42 ISO8583	varchar	Si
		amount	Monto de la transacción	double	No
		txn_date	Fecha en que se realizó la transacción	timestamp with time zone	No
		control_token	Código de control enviado por el terminal	varchar	No
		transaction_statuses	Estado de la transacción	integer	No
		transaction_type_id	Identificador del tipo de transacción	integer	No
		transaction_response_id	Respuesta de la transacción	integer	Si
		card_product_id	Marca de tarjeta	integer	Si
		retried (boolean)	Si se venció el timeout para recibir respuesta a la transacción	boolean	No
		retry_attempts	Número de veces que se reintentó realizar la transacción	integer	No
sec_token	Firma de seguridad para validación de integridad y evitar duplicación	varchar	No		
transaction_parameters	Almacena los datos de la clase persistente para almacenar parámetros asociados a las transacciones	id	Identificador incremental generado por el motor de base de datos	integer	No
		transaction_id	Transacción a la que está asociado el parámetro	integer	Si
		param_name	Nombre del parámetro	varchar	No
		param_value	Valor del parámetro	varchar	No
		param_type	Campo para discriminar el tipo de parámetro (request,	varchar	No

Tabla	Descripción	Campo	Descripción	Tipo de dato	¿Null?
			response)		
transaction_res ponse	Almacena los datos de la clase persistente para representar una Respuesta de Transacción	id	Identificador incremental generado por el motor de base de datos	integer	No
		code	Código de respuesta. Campo 39 ISO8583	varchar	No
		name	Nombre o breve descripción de la respuesta	varchar	No
		payment_channel_id	Canal de pago al que está asociado este tipo de respuesta	integer	No
		description	Breve descripción de la respuesta	varchar	Si
transaction_typ e	Almacena los datos de la clase persistente para representar un Tipo de Transacción	id	Identificador incremental generado por el motor de base de datos	integer	No
		name	Nombre de la transacción (como será indicado por el terminal)	varchar	No
		description	Descripción opcional del tipo de transacción	varchar	Si
		iso_template_id	Identificador de la plantilla para construir esta operación en formato ISO8583	integer	No
		method_name	Nombre de la función a utilizar para ejecutar la operación	varchar	Si
		timeout	Timeout en segundos configurado para la operación	double	No
		retry	Define si se debe reintentar la operación en caso de vencimiento del timeout	boolean	No
		retry_attempts	Número máximo de intentos si se vence el timeout	integer	No
		transaction_respo nse_type	Tipo de transacción que debe recibir como respuesta	varchar	Si
		template_messag e_id	Identificador de la Plantilla de Mensajes a la que está asociado	integer	No

Tabla	Descripción	Campo	Descripción	Tipo de dato	¿Null?
			el método		
		transaction_inver se_type	Identificador del Tipo de transacción que debe ejecutarse en caso de timeout para anular efectos	integer	Si

3. Implementación

3.1. Etapas seguidas para la creación de la aplicación

Utilizando la metodología *Scrum* se obtuvo un producto de manera incremental:

1. Construcción inicial del formato de la transacción en formato ISO 8583.
2. Implementación inicial de la interfaz y servicio web CPAGO.
3. Implementación del servicio web para la invocación de las transacciones.
4. Integración con la solución ORM.
5. Desarrollo de la API de Acceso a Datos para los procesos afiliatorios y de configuración.
6. Construcción de la transacción en formato ISO 8583 utilizando los parámetros locales y los suministrados por el servicio web.
7. Envío de transacción a la entidad aprobadora y procesamiento de la respuesta.
8. Refinamiento de los componentes de CPAGO.
9. Comunicación entre CPAGO y servicio web del Motor Transaccional.
10. Integración.
11. Pruebas.

3.2. Plataforma utilizada para su desarrollo

3.2.1. Estándar J2EE (Java 2 Platform, Enterprise Edition)

Java 2 Platform, Enterprise Edition (J2EE), es un estándar que define una plataforma de desarrollo empresarial basada en Java. Ofrece un modelo de aplicaciones distribuidas multicapa, con componentes reusables, un modelo de seguridad unificado, un control flexible de transacciones y soporte para Servicios Web. Proporciona soluciones al diseño, desarrollo, montaje y despliegue de aplicaciones empresariales, independientes del sistema operativo, en un entorno distribuido.

Una aplicación J2EE, puesto que se desarrolla de acuerdo a un estándar, es decir, sólo utiliza las APIS estándares, no depende de una implementación particular de la plataforma. Aunque existen múltiples implementaciones de distintos fabricantes, con todas ellas la aplicación será totalmente compatible.

3.2.2. Oracle Glassfish Server

GlassFish es un servidor de aplicaciones de código abierto, es la implementación de referencia de Java EE y como tal soporta Enterprise JavaBeans, JPA, JavaServer Faces, JMS, RMI, JavaServer Pages, servlets, etc. Esto permite a los desarrolladores crear aplicaciones empresariales que son portátiles y escalables. Componentes opcionales también se pueden instalar para servicios adicionales.

GlassFish se basa en el código fuente liberado por el sistema de persistencia TopLink de Sun y Oracle Corporation. Utiliza un derivado de Apache Tomcat como el contenedor servlet para servir contenido Web, con un componente añadido llamado Grizzly que utiliza Java New I/O (NIO) para la escalabilidad y la velocidad.

3.2.3. Jersey RESTful Framework

Es un framework open source, con calidad para ambientes de producción para desarrollar servicios web REST en Java.

3.2.4. Hibernate ORM

Hibernate es una solución de Mapeo Objeto/Relacional para entornos Java. El término Mapeo Objeto/Relacional se refiere a la técnica de mapeo de datos entre una representación del modelo de objetos a una representación del modelo de datos relacional.

Hibernate se encarga de la asignación de clases de Java a las tablas de bases de datos, y de tipos de datos Java con tipos de datos SQL. Además, proporciona métodos para consulta y obtención de datos.

3.2.5. Flyway

Es una herramienta de migración de base de datos de código abierto para lenguaje Java. Se favorece fuertemente la sencillez y la convención sobre configuración. Entre sus funciones podemos destacar:

- Recrear una base de datos desde cero
- Dejar claro en todo momento en qué estado se encuentra una base de datos.
- Migrar de una manera determinista de la versión actual de la base de datos a una más nueva.

3.2.6. HyperSQL

HSQLDB (base de datos HyperSQL) es un motor de base de datos relacional escrito en Java. HyperSQL se utiliza para el desarrollo, prueba y despliegue de aplicaciones de bases de datos.

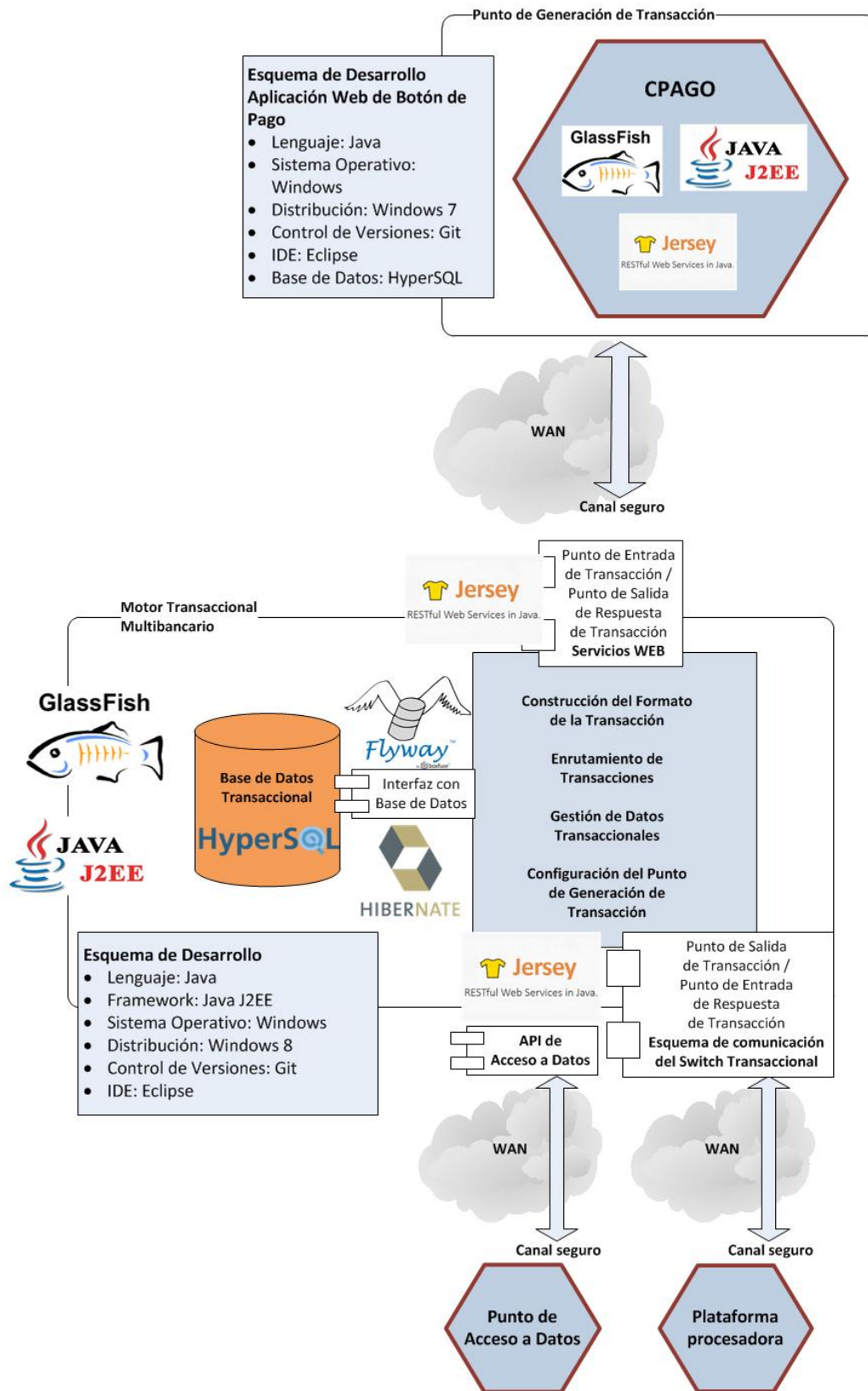


Figura 38. Herramientas utilizadas para el desarrollo de la aplicación. (Elaboración propia).

3.3. Patrones de interfaz, interacción o diseño utilizados

3.3.1. Arquitecturas Orientadas a Servicios (Service Oriented Architectures)

La Arquitectura SOA establece un marco de diseño para la integración de aplicaciones independientes de manera que desde la red pueda accederse a sus funcionalidades, las cuales se ofrecen como servicios. La forma más habitual de implementarla es mediante Servicios Web, una tecnología basada en estándares e independiente de la plataforma, con la que SOA puede descomponer aplicaciones monolíticas en un conjunto de servicios e implementar esta funcionalidad en forma modular.

La estrategia de orientación a servicios permite la creación de servicios y aplicaciones compuestas que pueden existir con independencia de las tecnologías subyacentes. En lugar de exigir que todos los datos y lógica de negocio residan en un mismo ordenador, el modelo de servicios facilita el acceso y consumo de los recursos de IT a través de la red. Puesto que los servicios están diseñados para ser independientes, autónomos y para interconectarse adecuadamente, pueden combinarse y recombinarse con suma facilidad en aplicaciones complejas que respondan a las necesidades de cada momento en el seno de una organización.

Los servicios Web son la forma más habitual de implementar SOA. Los servicios Web son aplicaciones que utilizan estándares para el transporte, codificación y protocolo de intercambio de información. Los servicios Web permiten la intercomunicación entre sistemas de cualquier plataforma y se utilizan en una gran variedad de escenarios de integración.

3.3.2. Representational State Transfer (REST)

El término REST, acrónimo de REpresentational State Transfer, fue introducido por primera vez por Roy Fielding (uno de los creadores de HTTP) en la lectura de su tesis para describir un tipo de arquitectura de los sistemas en red. Un Servicio Web REST es un SOA basado en el concepto de recurso. Un recurso es cualquier cosa que tiene una URI (Uniform Resource Identifier), pudiendo tener cero o más representaciones.


Los Servicios Web REST requieren poca infraestructura, aparte de las tecnologías HTTP estándar y XML, que actualmente son soportadas por la mayoría de los lenguajes de programación y plataformas. Los servicios web REST son simples y efectivos, ya que HTTP es el interfaz más extendido y es soportado por la mayoría de las aplicaciones.

3.4. Principales interfaces

Las principales interfaces gráficas de usuario de este desarrollo corresponden a aquellas provistas por la aplicación C-Pago, en este apartado se presentan las mismas.

3.4.1. Formulario de datos del tarjetahabiente

Formulario de pago Compra



Solutions Systems
2526 C.A. +31599417-8

DATOS DE LA TARJETA

*** Número de la tarjeta**



*** Fecha de vencimiento**

*** Código de seguridad**

*** Nombre impreso en la tarjeta**

*** Cédula de identidad**

Los campos marcados con asteriscos (*) son obligatorios.

ACEPTAR

COMERCIO

 Comercio de Prueba 1

ORDEN

123456

MONTO DEL PAGO

5000.0 Bs

Figura 39. Aplicación C-Pago: Formulario de datos del tarjetahabiente. (Elaboración propia).

3.4.2. Pantalla de confirmación de la transacción



Figura 40. Aplicación C-Pago: Pantalla de confirmación de la transacción. (Elaboración propia).

3.4.3. Pantalla de error de la aplicación

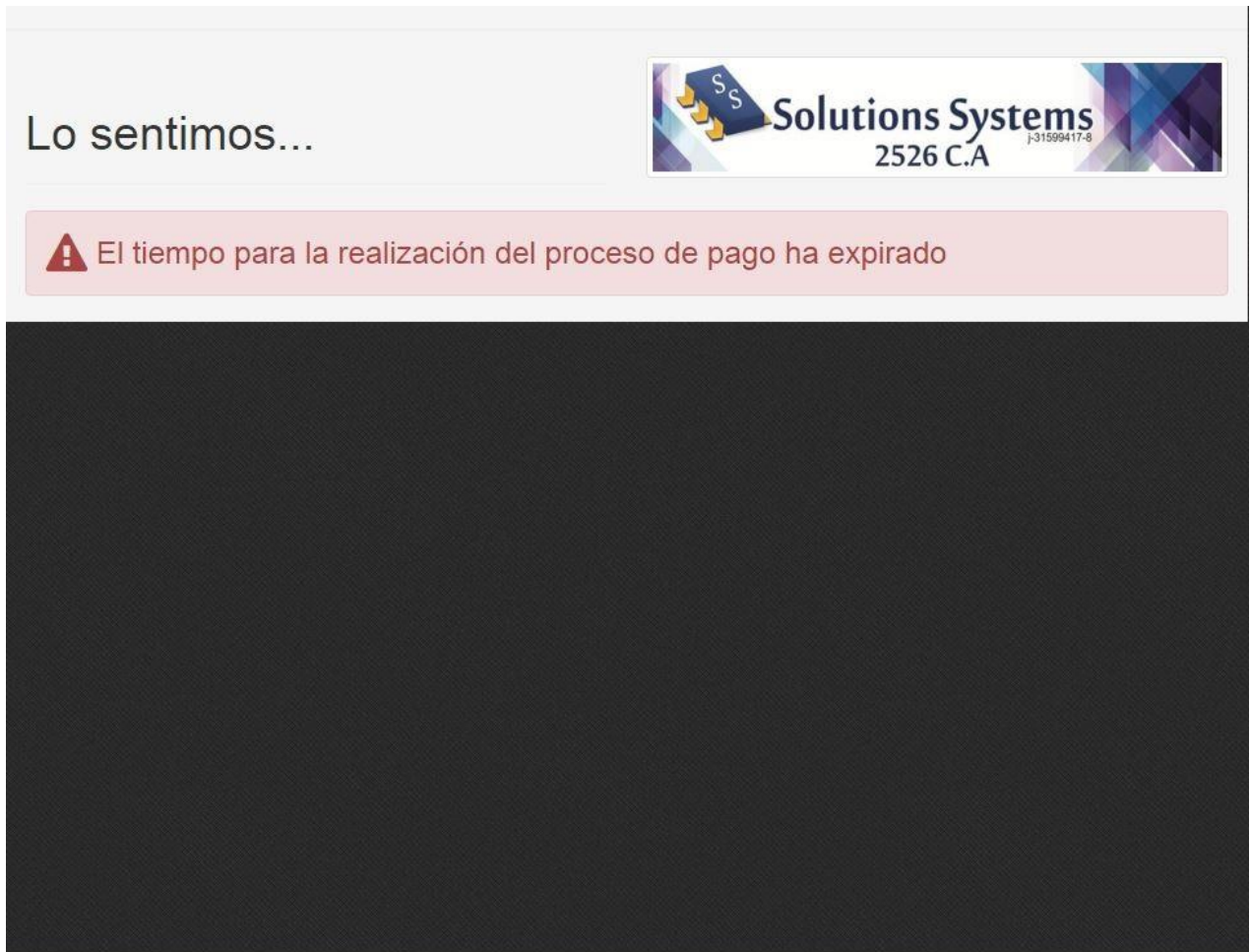



Figura 41. Aplicación C-Pago: Pantalla de error de la aplicación. (Elaboración propia).


3.4.4. Resultado con transacción aprobada

Resultado de la transacción

Compra



Solutions Systems
2526 C.A

 **APROBADA**

COMPROBANTE DE PAGO


Comercio de Prueba 1

J-12345467


Calle xhns, Edificio xyz piso 10

Caracas - Venezuela

0414335695 - es@hsy.com



Número de orden	1535475
Número de Tarjeta	123456****4567
Fecha y hora de la transacción	20/10/2015 03:40 pm.
Monto de la transacción	25639.19 Bs.




CONTINUAR


Figura 42. Aplicación C-Pago: Resultado con transacción aprobada. (Elaboración propia).

3.4.5. Resultado con transacción rechazada

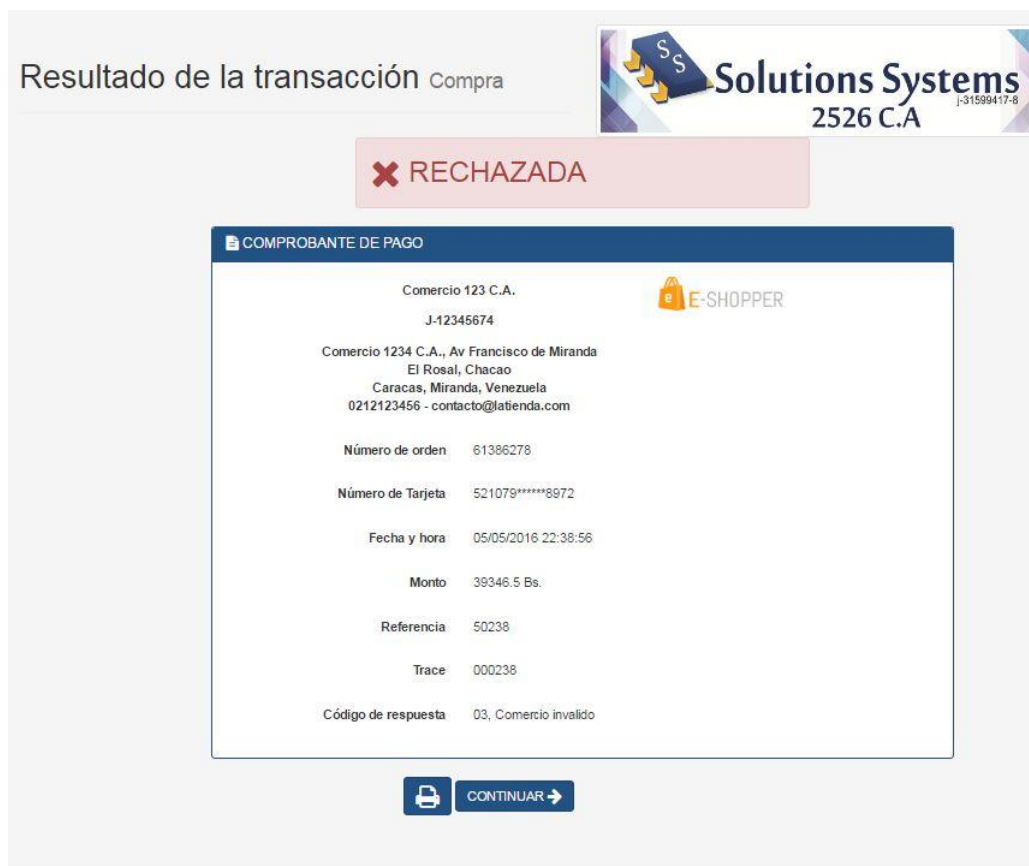


Figura 43. Aplicación C-Pago: Resultado con transacción rechazada. (Elaboración propia).

3.5. Módulos que comprenden el Motor Transaccional V-Financial

3.5.1. Núcleo Operacional

Agrupar las clases que realizan la construcción de las transacciones en base a la configuración y las reglas, además del análisis de la respuesta.

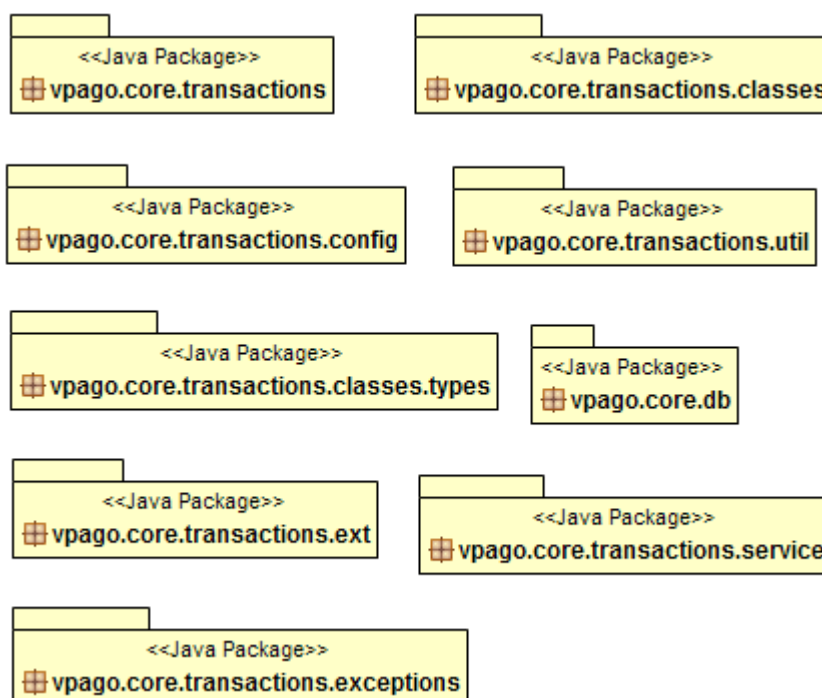


Figura 44. Paquetes que comprenden el módulo Núcleo Operacional (Elaboración propia).

Tabla 12. Descripción de los paquetes que comprenden el módulo Núcleo Operacional (Elaboración propia).

Paquete	Descripción
vpago.core.db	Clases para la configuración y gestión del framework Hibernate y la base de datos
vpago.core.transactions	Clases para la construcción y gestión de transacciones
vpago.core.transactions.classes	Clases para manejo de la lógica de las transacciones y la persistencia
vpago.core.transactions.classes.types	Tipos enumerados de uso común por las entidades y/o funciones del módulo transaccional
vpago.core.transactions.exceptions	Tipos de excepciones propias de la aplicación
vpago.core.transactions.ext	Contiene los paquetes con las funciones propias de cada Canal de Pago
vpago.core.transactions.service	Clases con funciones que proveen servicios a los métodos en el paquete de extensiones de los canales de pago
vpago.core.transactions.util	Contiene clases de uso común por todas las entidades y/o funciones del núcleo operacional

3.5.1.1. Paquete: vpago.core.db



Figura 45. Diagrama de clases del paquete vpago.core.db (Elaboración propia).

Tabla 13. Descripción de las clases del paquete `vpago.core.db` (Elaboración propia).

Clase	Descripción
DBInit	Métodos para inicializar tablas de la base de datos
DBTest	Métodos para pruebas con persistencia de objetos en la base de datos
HibernateUtil	Clase para el manejo de la configuración y funcionamiento del framework Hibernate

3.5.1.2. Paquete: `vpago.core.transactions`

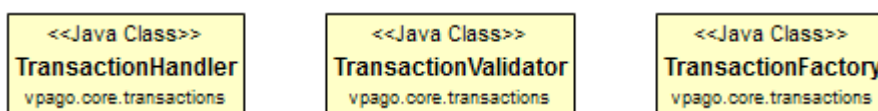


Figura 46. Diagrama de clases del paquete `vpago.core.transactions` (Elaboración propia).

Tabla 14. Descripción de las clases del paquete `vpago.core.transactions` (Elaboración propia).

Clase	Descripción
TransactionFactory	Clase que posee las funciones para construir y analizar tramas de transacciones
TransactionHandler	Bean de sesión para el manejo de las transacciones
TransactionValidator	Clase que posee las funciones para validar los elementos de una transacción

3.5.1.3. Paquete: `vpago.core.transactions.classes`

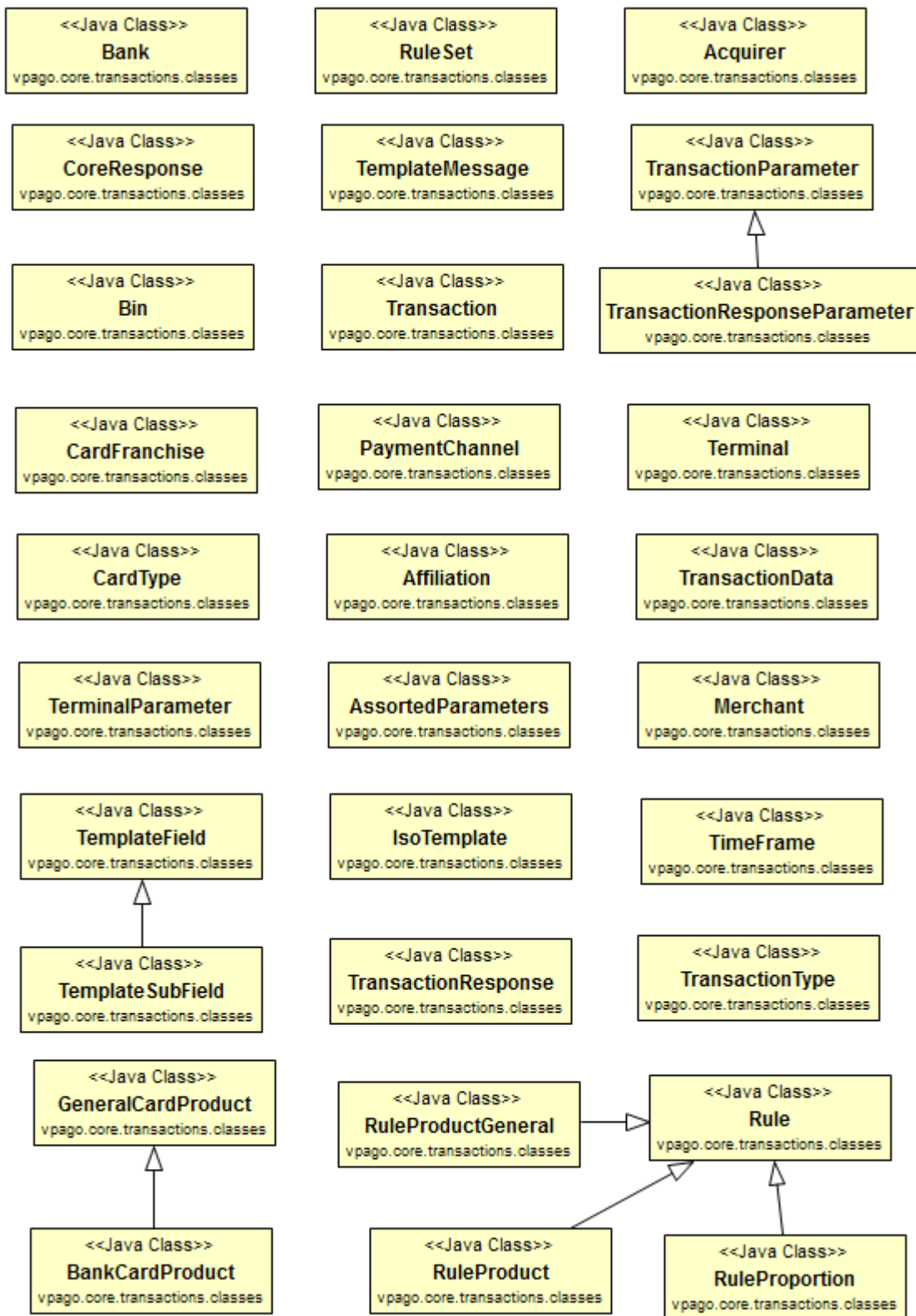


Figura 47. Diagrama de clases del paquete vpago.core.transactions.classes (Elaboración propia).

Tabla 15. Descripción de las clases del paquete `vpago.core.transaction.classes` (Elaboración propia).

Clase	Descripción
Acquirer	Clase persistente para representar un Adquiriente
Affiliation	Clase persistente para representar una Afiliación
AssortedParameters	Clase para almacenar de manera clasificada los parámetros de las transacciones
Bank	Clase persistente para representar un Banco
BankCardProduct	Clase persistente para representar un Producto de Tarjeta de Banco
Bin	Clase persistente para representar los rangos numéricos que identifican tipos de tarjetas
CardFranchise	Clase persistente para representar una franquicia procesadora de tarjetas
CardType	Clase persistente para representar Tipos de Tarjetas
CoreResponse	Clase persistente para representar la respuesta de una transacción dentro del sistema
GeneralCardProduct	Clase persistente para representar un Producto de Tarjeta General
IsoTemplate	Clase persistente para representar una Plantilla ISO que almacena los parámetros para mapear campos de una transacción a campos iso según el esquema elegido
Merchant	Clase persistente para representar un Comercio
PaymentChannel	Clase persistente para representar un Canal de Pago
Rule	Clase persistente para representar una Regla
RuleProduct	Clase persistente para representar una Regla de Producto
RuleProductGeneral	Clase persistente para representar una Regla General
RuleProportion	Clase persistente para representar una Regla por Proporción
RuleSet	Clase persistente para representar un Conjunto de Reglas
TemplateField	Clase persistente que representa una Plantilla de Campo con la descripción de un campo transaccional en la plantilla de un esquema almacenado en el motor
TemplateMessage	Clase persistente para representar una Plantilla de Mensajes
TemplateSubField	Clase persistente que representa una Plantilla de Subcampo utilizada para almacenar la configuración de los subcampos privados para los mensajes
Terminal	Clase persistente para representar un Terminal
TerminalParameter	Clase persistente para representar un Parámetro de Terminal
TimeFrame	Clase persistente para representar un Rango de Tiempo para la aplicación de una Regla

Transaction	Clase persistente para representar una Transacción
TransactionData	Clase persistente que representa la estructura de Datos de Transacción utilizada para proporcionar a TransactionHandler toda la información recibida en una invocación al servicio web que le es relevante
TransactionParameter	Clase persistente para almacenar parámetros asociados a las transacciones
TransactionResponse	Clase persistente para representar una Respuesta de Transacción
TransactionResponseParameter	Clase persistente para representar un Parámetro de Respuesta de la Transacción
TransactionType	Clase persistente para representar un Tipo de Transacción

3.5.1.4. Paquete: vpago.core.transactions.classes.types

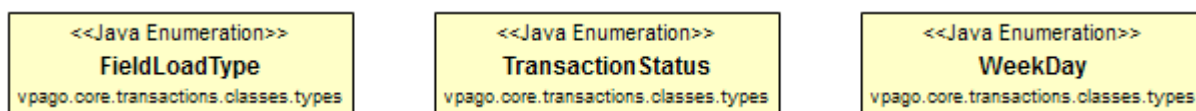


Figura 48. Diagrama de clases del paquete vpago.core.transactions.classes.types (Elaboración propia).

Tabla 16. Descripción de las clases del paquete vpago.core.transactions.classes.types (Elaboración propia).

Clase	Descripción
FieldLoadType	Tipo enumerado para representar los Tipos de Carga de Parámetros para una transacción
TransactionStatus	Tipo enumerado para representar un Tipo de Estado de Transacción
WeekDay	Tipo enumerado para representar los días de la semana

3.5.1.5. Paquete: vpago.core.transactions.exceptions

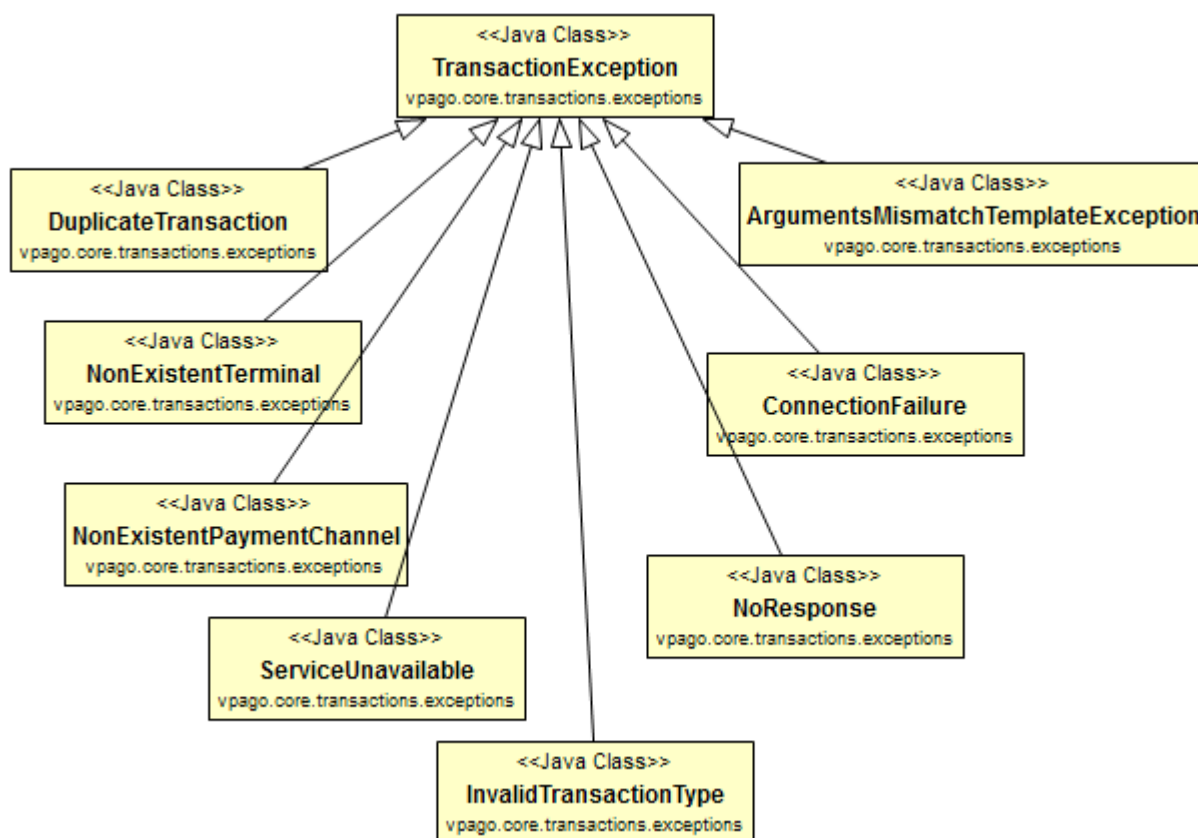


Figura 49. Diagrama de clases del paquete vpago.core.transactions.exceptions (Elaboración propia).

Tabla 17. Descripción de las clases del paquete vpago.core.transactions.exceptions (Elaboración propia).

Clase	Descripción
ArgumentsMismatchTemplateException	Excepción generada cuando no se encuentra una configuración de campo para un argumento
ConnectionFailure	Excepción generada cuando ocurre una falla en la conexión con el aprobador
DuplicateTransaction	Excepción generada al tratar de realizar una transacción duplicada
InvalidTransactionType	Excepción generada cuando el tipo de transacción solicitada es invalida
NonExistentPaymentChannel	Excepción generada cuando el canal de pago solicitado no existe
NonExistentTerminal	Excepción generada cuando se envía la transacción con un numero de terminal no existente
NoResponse	Excepción generada cuando se envía la transacción pero no se recibe respuesta
ServiceUnavailable	Excepción generada cuando no se puede acceder a algún

	componente de la plataforma
TransactionException	Excepción generada por el módulo de transacciones

3.5.1.6. Paquete: vpago.core.transactions.service

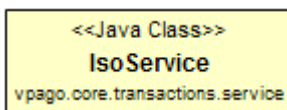


Figura 50. Diagrama de clases del paquete vpago.core.transactions.service (Elaboración propia).

Tabla 18. Descripción de las clases del paquete vpago.core.transactions.service (Elaboración propia).

Clase	Descripción
IsoService	Servicios para proveer parámetros ISO

3.5.1.7. Paquete: vpago.core.transactions.util

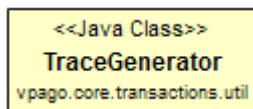


Figura 51. Diagrama de clases del paquete vpago.core.transaction.util (Elaboración propia).

Tabla 19. Descripción de las clases del paquete vpago.core.transaction.util (Elaboración propia).

Clase	Descripción
TraceGenerator	Generador de trace

3.5.2. Interfaz de Pago

Comprende las clases que se encargan del manejo del servicio web transaccional y los formatos relacionados al mismo.

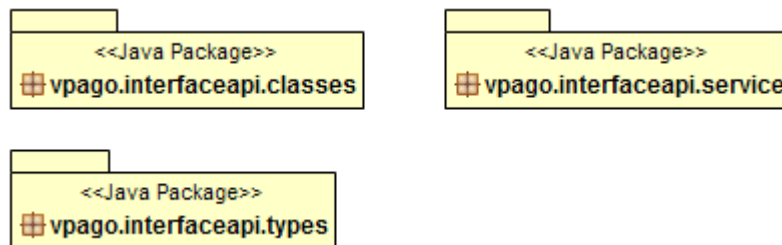


Figura 52. Paquetes que comprenden el módulo Interfaz de Pago (Elaboración propia).

Tabla 20. Descripción de los paquetes que comprenden el módulo Interfaz de Pago (Elaboración propia).

Paquete	Descripción
vpago.interfaceapi.classes	Clases para el manejo de las solicitudes, y sus respectivos parámetros, recibidas a través del servicio web
vpago.interfaceapi.service	Contiene todas las clases destinadas a implementar los métodos para los diferentes servicios que ofrece la Interfaz de Pago
vpago.interfaceapi.types	Tipos enumerados de uso común por las entidades y/o funciones del módulo de interfaz de pago

3.5.2.1. Paquete: vpago.interfaceapi.classes

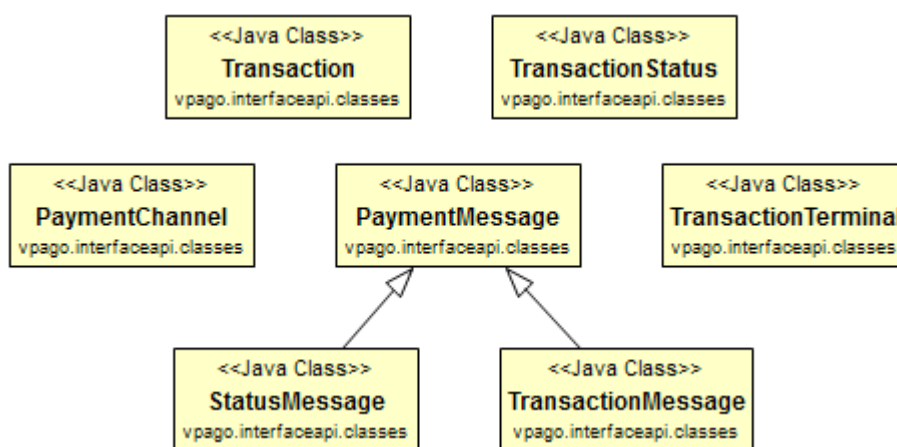


Figura 53. Diagrama de clases del paquete vpago.interfaceapi.classes (Elaboración propia).

Tabla 21. Descripción de las clases del paquete vpago.interfaceapi.classes (Elaboración propia).

Clase	Descripción
PaymentChannel	Clase para representar un Canal de Pago
PaymentMessage	Clase para aplicar formato a las entidades que serán devueltas por las funciones de la Interfaz de Pago
StatusMessage	Clase para aplicar formato a los mensajes no transaccionales.
Transaction	Clase para representar una Transacción
TransactionMessage	Representación de un mensaje transaccional
TransactionStatus	Estado de la transacción
TransactionTerminal	Punto terminal donde se origina la transacción

3.5.2.2. Paquete: vpago. interfaceapi.service

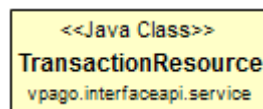


Figura 54. Diagrama de clases del paquete vpago.interfaceapi.service (Elaboración propia).

Tabla 22. Descripción de las clases del paquete vpago.interfaceapi.service (Elaboración propia).

Clase	Descripción
TransactionResource	Contiene todos los métodos para los servicios relacionados a la solicitud de transacciones

3.5.2.3. Paquete: vpago. interfaceapi.types

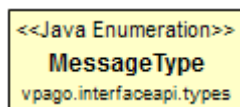


Figura 55. Diagrama de clases del paquete vpago.interfaceapi.types (Elaboración propia).

Tabla 23. Descripción de las clases del paquete vpago.interfaceapi.types (Elaboración propia).

Clase	Descripción
vpago.interfaceapi.types	Tipos enumerados de uso común por las entidades y/o funciones del módulo de interfaz de pago

3.5.3. API de Acceso a Datos

Contiene todas las clases que se encargan del manejo de todos los servicios web relacionados con la creación, modificación y consulta de los objetos y entidades de la plataforma, cuyo estado se encuentra almacenado en la base de datos. Comprende los paquetes:

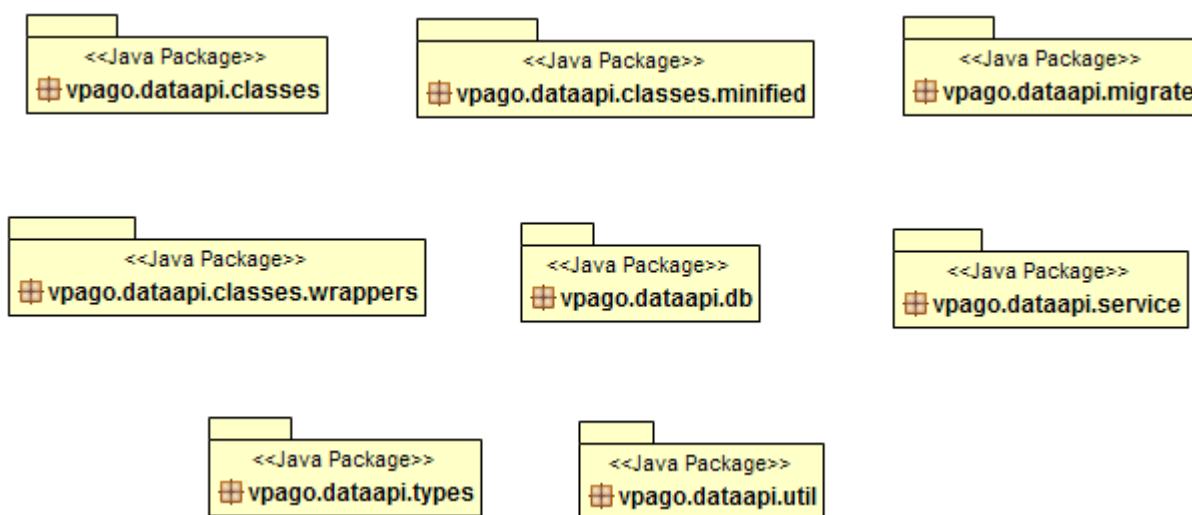


Figura 56. Paquetes que comprenden el módulo API de Acceso a Datos (Elaboración propia).

Tabla 24. Descripción de los paquetes que comprenden el módulo API de Acceso a Datos (Elaboración propia).

Paquete	Descripción
vpago.dataapi.classes	Clases para manejo de la lógica de las consultas, la persistencia y los datos enviados a través de los servicios web
vpago.dataapi.classes.minified	Clases utilitarias para mostrar subconjuntos de los datos de las entidades en los servicios de la API que retornen listas
vpago.dataapi.classes.wrappers	Clases utilitarias para construir listas de las entidades en los servicios de la API
vpago.dataapi.db	Clases para la configuración y gestión del framework Hibernate y la base de datos
vpago.dataapi.migrate	Clases para la gestión de las migraciones de la base de datos
vpago.dataapi.service	Contiene todas las clases destinadas a implementar los métodos para los diferentes servicios que ofrece la API
vpago.dataapi.types	Tipos enumerados de uso común por las entidades y/o servicios de la API
vpago.dataapi.util	Contiene clases de uso común por todas las entidades y/o servicios de la API

3.5.3.1. Paquete: vpago.dataapi.classes

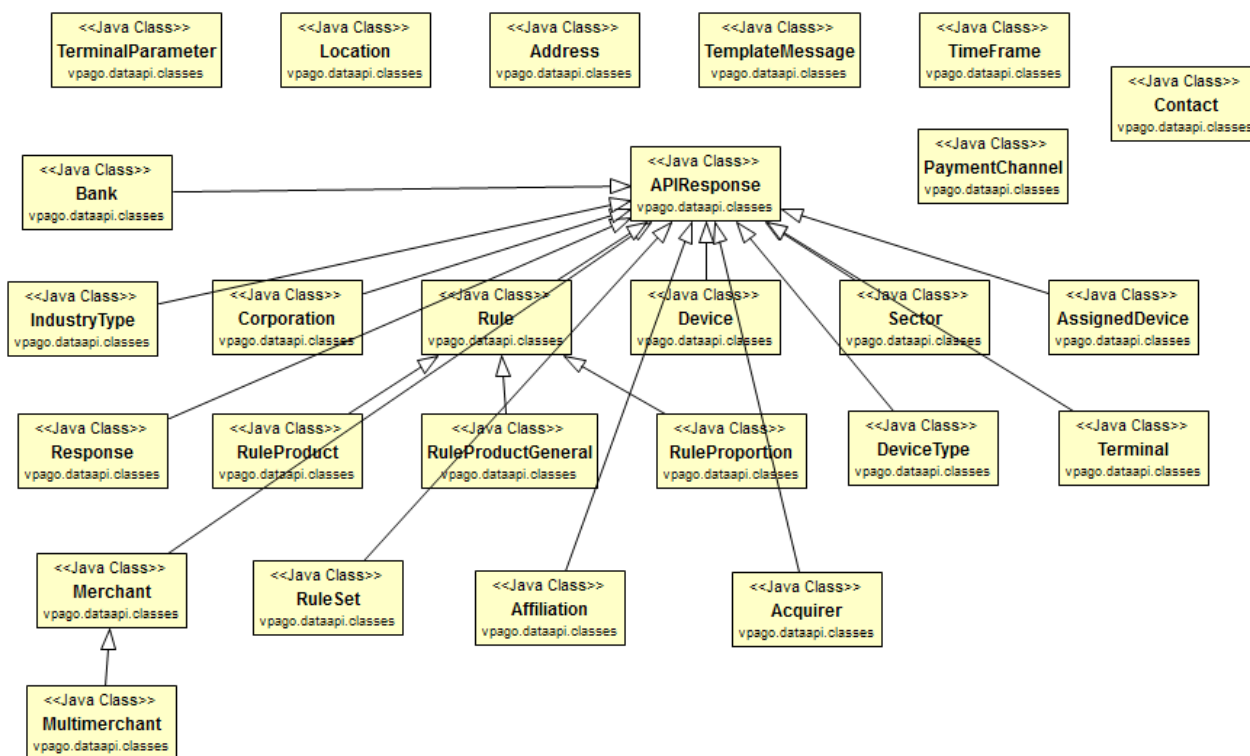


Figura 57. Diagrama de clases del paquete vpago.dataapi.classes (Elaboración propia).

Tabla 25. Descripción de las clases del paquete vpago.dataapi.classes (Elaboración propia).

Clase	Descripción
Acquirer	Clase persistente para representar un Adquiriente
Address	Clase persistente para representar una Dirección
Affiliation	Clase persistente para representar una Afiliación
APIResponse	Clase para aplicar formato a las entidades que serán devueltas por las funciones de servicios web
AssignedDevice	Clase persistente para representar un Dispositivo asignado a un Terminal
Bank	Clase persistente para representar un Banco
Contact	Clase persistente para representar un Contacto
Corporation	Clase persistente para representar una Corporación
Device	Clase persistente para representar un Dispositivo
DeviceType	Clase persistente para representar el Tipo de Dispositivo
IndustryType	Clase persistente para representar un Tipo de Industria segun el Banco Central de Venezuela
Location	Clase persistente para representar una Ubicación
Merchant	Clase persistente para representar un Comercio
Multimerchant	Clase persistente para representar un Multicomercio
PaymentChannel	Clase persistente para representar un Canal de Pago
Response	Clase para el envío de mensajes en la respuesta
Rule	Clase persistente para representar una Regla
RuleProduct	Clase persistente para representar una Regla de Producto

RuleProductGeneral	Clase persistente para representar una Regla General
RuleProportion	Clase persistente para representar una Regla por Proporción
RuleSet	Clase persistente para representar un Conjunto de Reglas
Sector	Clase persistente para representar un Sector económico que agrupa Tipos de Industrias según el Banco Central de Venezuela
TemplateMessage	Clase persistente para representar una Plantilla de Mensajes
Terminal	Clase persistente para representar un Terminal
TerminalParameter	Clase persistente para representar un Parámetro de Terminal
TimeFrame	Clase persistente para representar un Rango de Tiempo para la aplicación de una Regla

3.5.3.2. Paquete: vpago.dataapi.minified

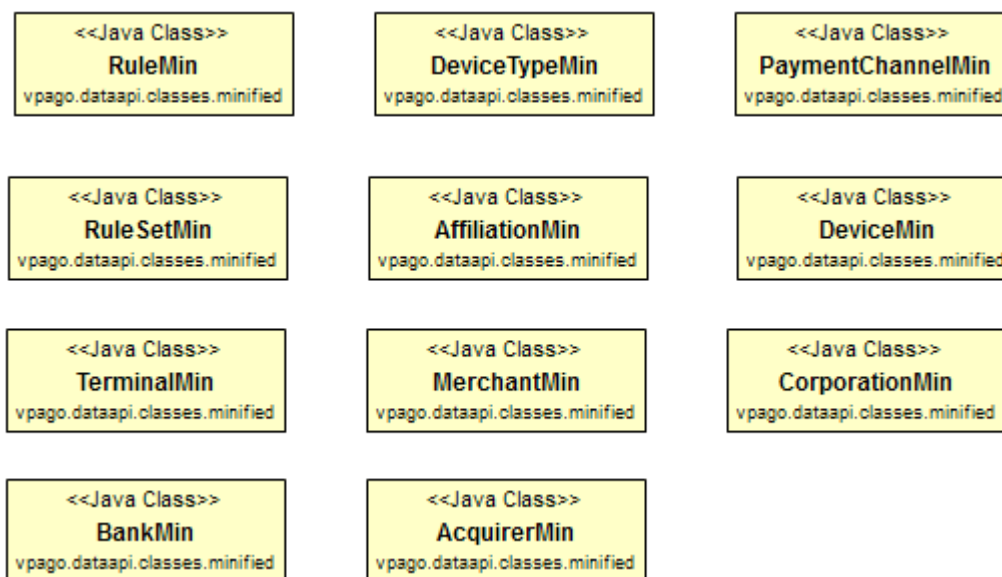


Figura 58. Diagrama de clases del paquete vpago.dataapi.classes.minified (Elaboración propia).

Tabla 26. Descripción de las clases del paquete vpago.dataapi.classes.minified (Elaboración propia).

Clase	Descripción
AcquirerMin	Clase utilitaria para mostrar un subconjunto de los datos de la entidad Adquiriente
AffiliationMin	Clase utilitaria para mostrar un subconjunto de los datos de la entidad Afiliación
BankMin	Clase utilitaria para mostrar un subconjunto de los datos de la entidad Banco
CorporationMin	Clase utilitaria para mostrar un subconjunto de los datos de la entidad Corporación
DeviceMin	Clase utilitaria para mostrar un subconjunto de los datos de la entidad Dispositivo
DeviceTypeMin	Clase utilitaria para mostrar un subconjunto de los datos de la entidad Tipo de Dispositivo
MerchantMin	Clase utilitaria para mostrar un subconjunto de los datos de la

	entidad Comercio
PaymentChannelMin	Clase utilitaria para mostrar un subconjunto de los datos de la entidad Canal de Pago
RuleMin	Clase utilitaria para mostrar un subconjunto de los datos de la entidad Regla
RuleSetMin	Clase utilitaria para mostrar un subconjunto de los datos de la entidad Conjunto de Reglas
TerminalMin	Clase utilitaria para mostrar un subconjunto de los datos de la entidad Terminal

3.5.3.3. Paquete: vpago.dataapi.wrappers

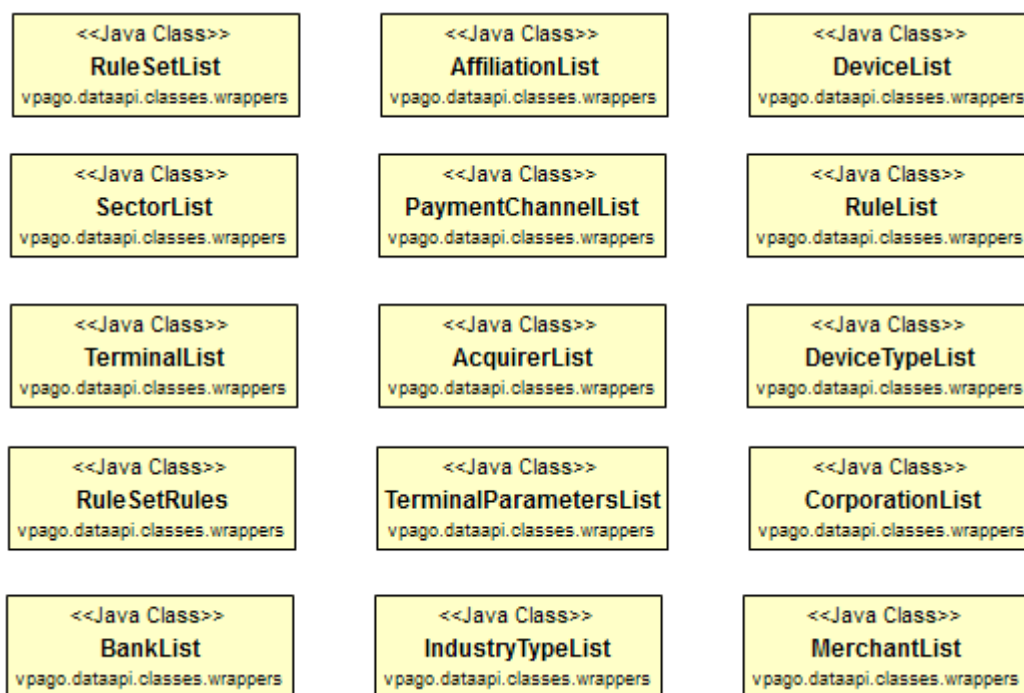


Figura 59. Diagrama de clases del paquete vpago.dataapi.classes.wrappers (Elaboración propia).

Tabla 27. Descripción de las clases del paquete vpago.dataapi.classes.wrappers (Elaboración propia).

Clase	Descripción
AcquirerList	Clase utilitaria para construir la lista de las entidades de la clase Adquiriente
AffiliationList	Clase utilitaria para construir la lista de las entidades de la clase Afiliación
BankList	Clase utilitaria para construir la lista de las entidades de la clase Banco
CorporationList	Clase utilitaria para construir la lista de las entidades de la clase Corporación
DeviceList	Clase utilitaria para construir la lista de las entidades de la clase Dispositivo
DeviceTypeList	Clase utilitaria para construir la lista de las entidades de la clase

	Tipo de Dispositivo
IndustryTypeList	Clase utilitaria para construir la lista de las entidades de la clase Tipo de Industria
MerchantList	Clase utilitaria para construir la lista de las entidades de la clase Comercio
PaymentChannelList	Clase utilitaria para construir la lista de las entidades de la clase Canal de Pago
RuleList	Clase utilitaria para construir la lista de las entidades de la clase Regla
RuleSetList	Clase utilitaria para construir la lista de las entidades de la clase Conjunto de Reglas
RuleSetRules	Clase utilitaria para construir la lista de las entidades de la clase Reglas asociadas a un Conjunto de Reglas
SectorList	Clase utilitaria para construir la lista de las entidades de la clase Sector
TerminalList	Clase utilitaria para construir la lista de las entidades de la clase Terminal
TerminalParametersList	Clase utilitaria para construir la lista de las entidades de la clase Parámetros de Terminal

3.5.3.4. Paquete: vpago.dataapi.db

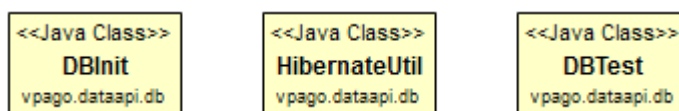


Figura 60. Diagrama de clases del paquete vpago.dataapi.db (Elaboración propia).

Tabla 28. Descripción de las clases del paquete vpago.dataapi.db (Elaboración propia).

Clase	Descripción
DBInit	Métodos para inicializar tablas de la base de datos
DBTest	Métodos para pruebas con persistencia de objetos en la base de datos
HibernateUtil	Clase para el manejo de la configuración y funcionamiento del framework Hibernate

3.5.3.5. Paquete: vpago.dataapi.migrate



Figura 61. Diagrama de clases del paquete vpago.dataapi.migrate (Elaboración propia).

Tabla 29. Descripción de las clases del paquete `vpago.dataapi.migrate` (Elaboración propia).

Clase	Descripción
DBControl	Métodos para la ejecución de las migraciones en la base de datos

3.5.3.6. Paquete: `vpago.dataapi.service`

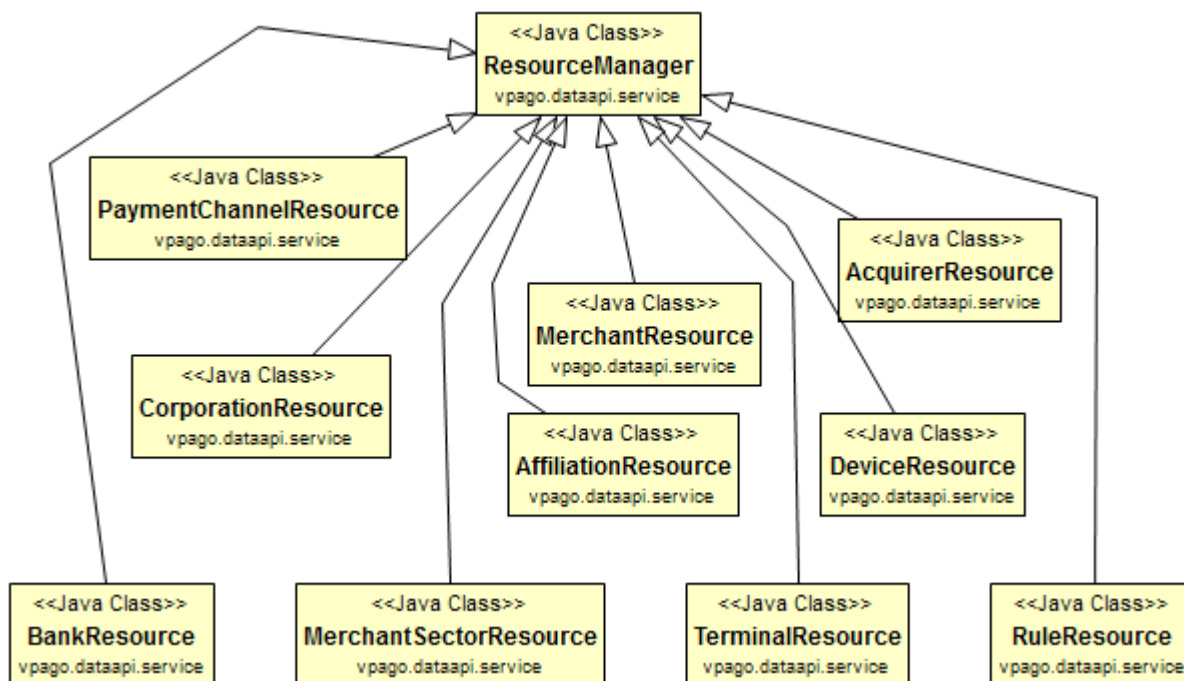


Figura 62. Diagrama de clases del paquete `vpago.dataapi.service` (Elaboración propia).

Tabla 30. Descripción de las clases del paquete `vpago.dataapi.service` (Elaboración propia).

Clase	Descripción
AcquirerResource	Contiene todos los métodos para los servicios relacionados al recurso Adquiriente (Acquirer)
AffiliationResource	Contiene todos los métodos para los servicios relacionados al recurso Afiliación (Affiliation)
BankResource	Contiene todos los métodos para los servicios relacionados al recurso Bank (Banco)
CorporationResource	Contiene todos los métodos para los servicios relacionados al recurso Corporación (Corporation)
DeviceResource	Contiene todos los métodos para los servicios relacionados a los recursos Dispositivo (Device), Tipo de Dispositivo (DeviceType), y AssignedDevice (Dispositivo Asignado)
MerchantResource	Contiene todos los métodos para los servicios relacionados al recurso Comercio (Merchant)
MerchantSectorResource	Contiene todos los métodos para los métodos para los servicios relacionados a los recursos Sector (Sector) y Tipo de Industria

	(IndustryType)
PaymentChannelResource	Contiene todos los métodos para los servicios relacionados al recurso Canal de Pago (PaymentChannel)
ResourceManager	Agrupar métodos y atributos comunes a las clases que manejan los recursos y servicios de la API
RuleResource	Contiene todos los métodos para los servicios relacionados a los recursos Regla (Rule) y Conjunto de Reglas (RuleSet)
TerminalResource	Contiene todos los métodos para los servicios relacionados al recurso Terminal (Terminal)

3.5.3.7. Paquete: vpago.dataapi.types



Figura 63. Diagrama de clases del paquete vpago.dataapi.types (Elaboración propia).

Tabla 31. Descripción de las clases del paquete vpago.dataapi.types (Elaboración propia).

Clase	Descripción
ResponseType	Tipo enumerado para representar los Tipos de Respuesta
WeekDay	Tipo enumerado para representar los días de la semana

3.5.3.8. Paquete: vpago.dataapi.util

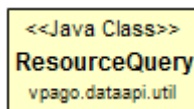


Figura 64. Diagrama de clases del paquete vpago.dataapi.util (Elaboración propia).

Tabla 32. Descripción de las clases del paquete vpago.dataapi.util (Elaboración propia).

Clase	Descripción
ResourceQuery	Contiene métodos para consultar recursos comunes en las funciones de la API

4. Despliegue y pruebas

4.1. Plataforma necesaria para utilizar la aplicación

4.1.1. Arquitectura básica de hardware propuesta

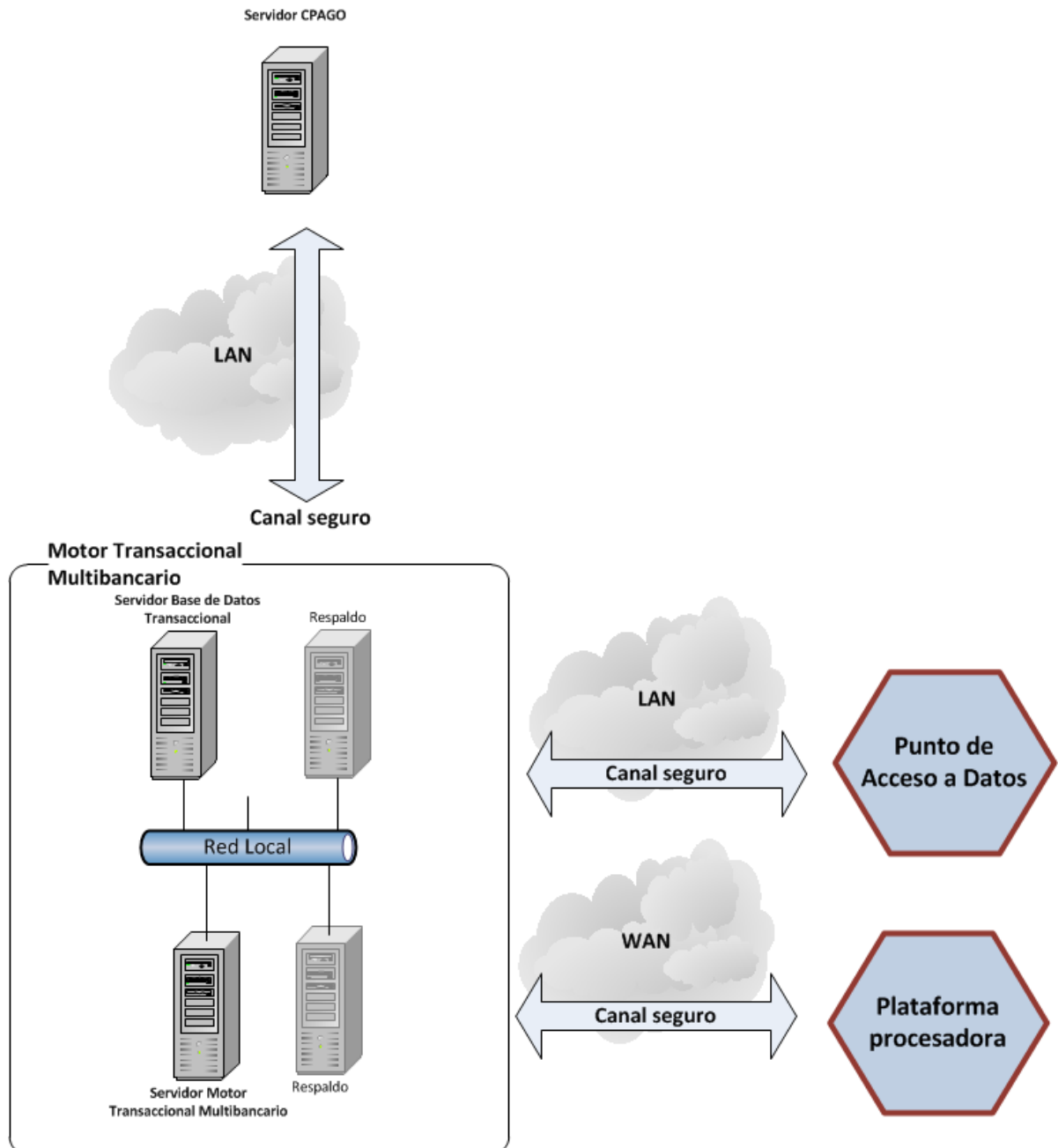


Figura 65. Arquitectura de hardware propuesta para el despliegue de la aplicación. (Elaboración propia).

4.1.2. Arquitectura de alta disponibilidad y seguridad propuesta

De conformidad con las normas de seguridad y requerimientos de la industria bancaria, los servicios relativos a manejo de transacciones deben estar cimentados sobre arquitecturas robustas que garanticen la continuidad del negocio. De acuerdo a lo anterior se propone un esquema de alta disponibilidad para un ambiente de producción, que, a su vez, permite cumplir con las regulaciones relativas a cifrado de los canales, monitoreo y protección de los sistemas.

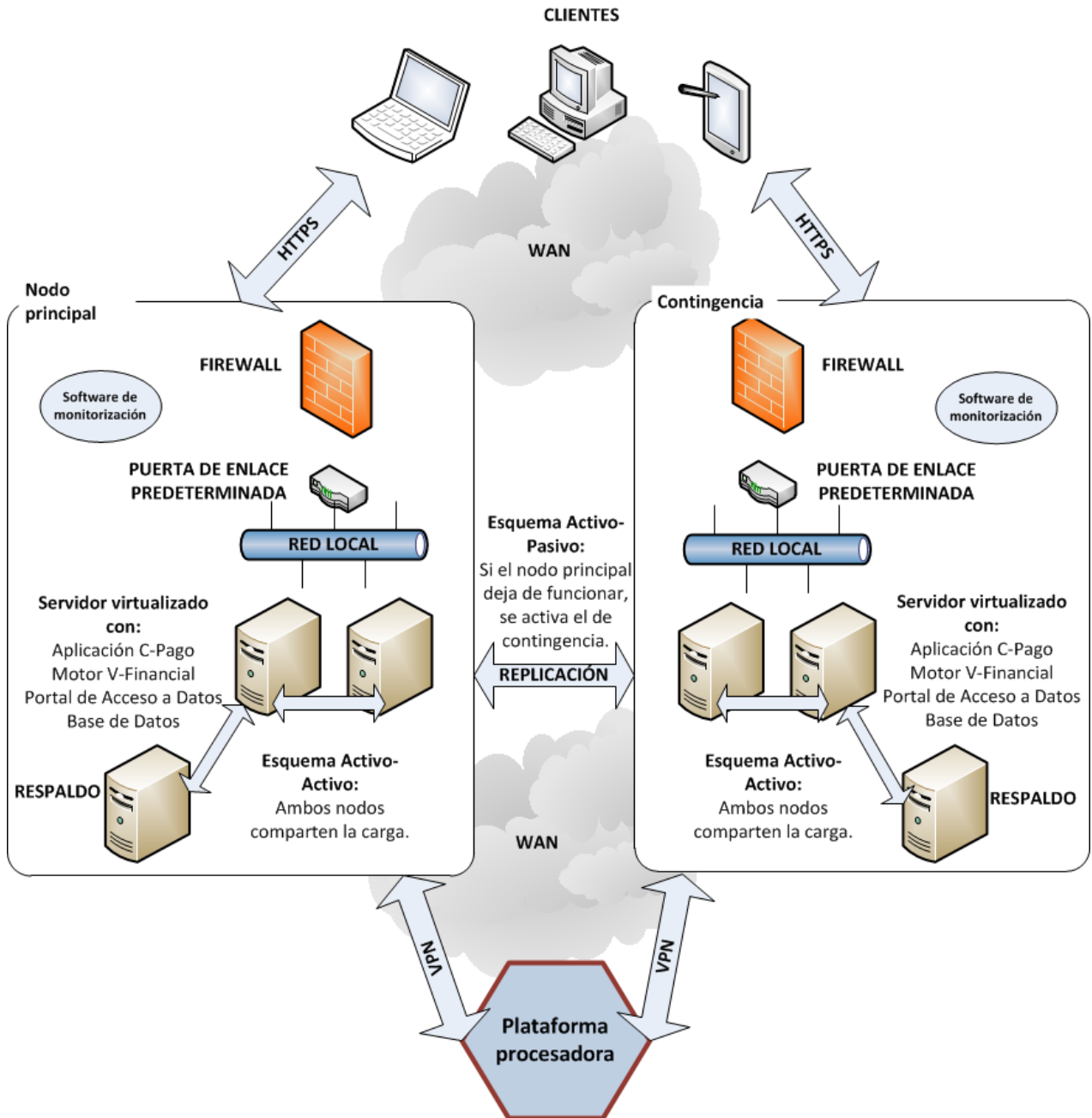


Figura 66. Arquitectura de alta disponibilidad propuesta. (Elaboración propia).

4.1.3. Software necesario

Tabla 33. Componentes de software necesarios para el despliegue de la aplicación.

Servidor de Motor Transaccional	Servidor de Base de Datos	Servidor CPAGO
<ul style="list-style-type: none"> • Sistema Operativo Centos 7 • Java JDK 8 • Glassfish Server 4.1 	<ul style="list-style-type: none"> • Sistema Operativo Centos 7 • PostgreSQL (u otro motor de base de datos compatible con JDBC) 	<ul style="list-style-type: none"> • Sistema Operativo Centos 7 • Java JDK 8 • Glassfish Server 4.1

4.2. Diagrama de despliegue

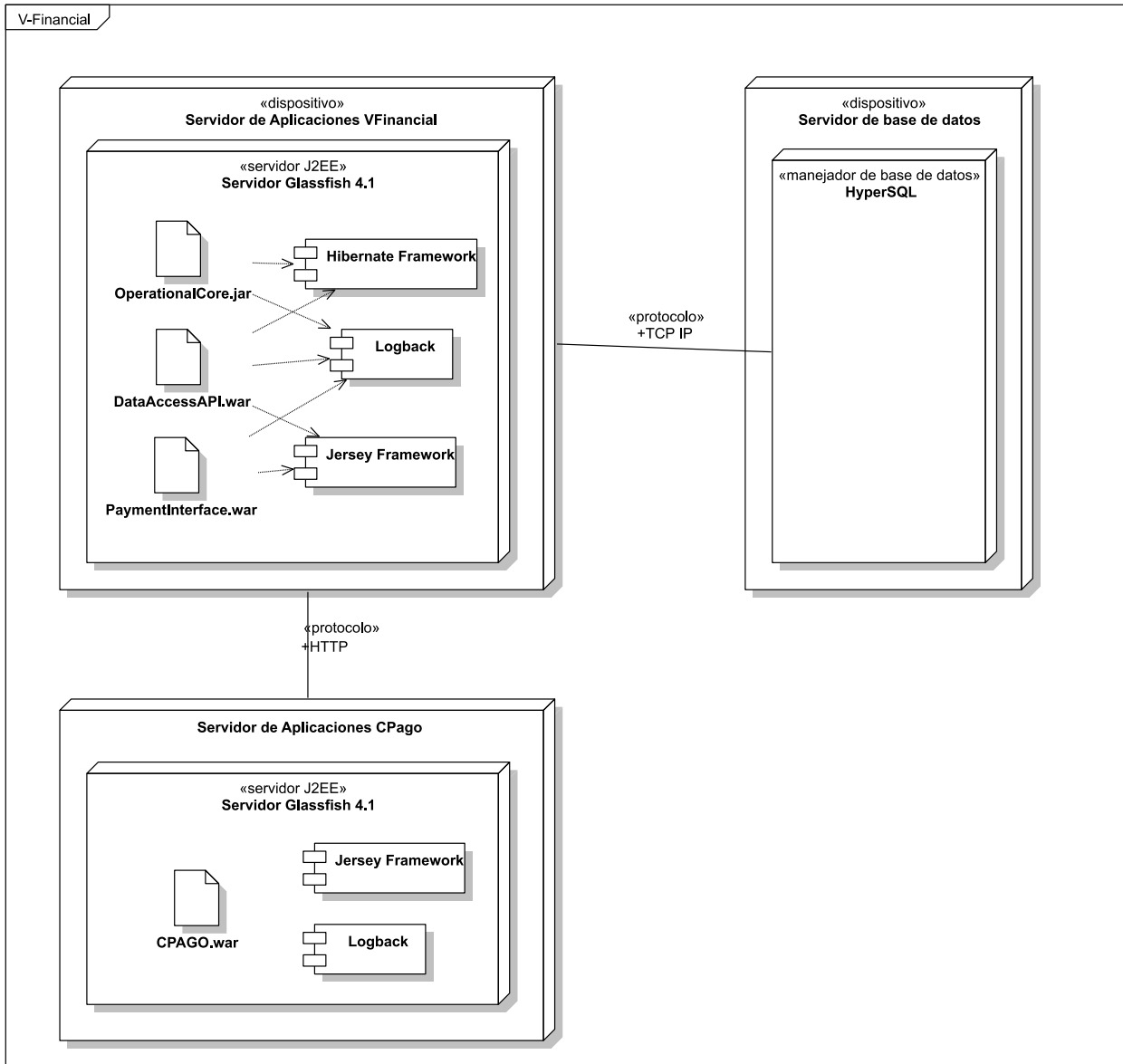


Figura 67. Diagrama de despliegue de los componentes de la Plataforma V-Financial (Elaboración propia).

4.3. Pruebas

Para la verificación del funcionamiento de los componentes de la plataforma se realizaron una serie de pruebas sobre cada uno de los elementos. Las pruebas ejecutadas las podemos dividir en tres conjuntos:

- Pruebas unitarias sobre la aplicación C-Pago: Contienen los casos de prueba que no involucran la comunicación con el Motor Transaccional
- Pruebas unitarias sobre el Motor Transaccional: Ejecutadas utilizando el software de pruebas SoapUI sobre cada uno de los servicios web proporcionados por el Motor Transaccional.
- Pruebas de integración: Casos de prueba que involucran la comunicación entre la aplicación C-Pago y el Motor Transaccional.

4.3.1. Conjunto de pruebas unitarias de aplicación C-Pago

Tabla 34. Tabla resumen de las instancias de casos de prueba de aplicación C-Pago. (Elaboración propia).

Caso de uso	Caso de prueba	Instancia de caso de prueba	Identificador de instancia de caso de prueba
C-Pago.1 - Realizar transacción de compra a través de interfaz gráfica	Invocación correcta de la interfaz	Invocación sin parámetro opcional merchantLogo	C-Pago.1.1.1
		Invocación con parámetro opcional merchantLogo	C-Pago.1.1.2
		Invocación con parámetro opcional merchantLogo con url no disponible	C-Pago.1.1.3
		Servidor V-Financial no disponible	C-Pago.1.1.4
	Solicitud incorrecta de envío del formulario de pago	Solicitud de envío del formulario de pago sin el campo "Número de la tarjeta"	C-Pago.1.2.1
		Solicitud de envío del formulario de pago sin el campo "Código de seguridad"	C-Pago.1.2.2
		Solicitud de envío del formulario de pago sin el campo "Nombre impreso en la tarjeta"	C-Pago.1.2.3
		Solicitud de envío del formulario de pago sin el campo "Cédula de identidad"	C-Pago.1.2.4
		Solicitud de envío del formulario de pago luego de 90 segundos	C-Pago.1.2.5
		Solicitud de envío del formulario de pago con menos de 12 dígitos en el campo "Número de la tarjeta"	C-Pago.1.2.6
		Solicitud de envío del formulario de pago con menos de 3 dígitos en el campo "Código de seguridad"	C-Pago.1.2.7
		Solicitud de envío del formulario de pago con	C-Pago.1.2.8

Caso de uso	Caso de prueba	Instancia de caso de prueba	Identificador de instancia de caso de prueba
		caracteres no dígitos en el campo "Cédula de identidad"	
Solicitud correcta de envío del formulario de pago		Solicitud de envío del formulario de pago luego de 90 segundos	C-Pago.1.3.1
		Solicitud de envío del formulario de pago antes de 90 segundos	C-Pago.1.3.2
Confirmación de datos		Confirmación de datos luego de 90 segundos	C-Pago.1.4.1
		Confirmación de datos antes de 90 segundos	C-Pago.1.4.2
Invocación de servicio web de venta		Servidor V-Financial no disponible	C-Pago.1.5.1
		Transacción aprobada por la entidad financiera	C-Pago.1.5.2
		Transacción rechazada por la entidad financiera	C-Pago.1.5.3
Invocación incorrecta de la interfaz		Invocación sin uno o más parámetros obligatorios	C-Pago.1.6.1
		Invocación con monto inválido	C-Pago.1.6.2
		Invocación con Identificador de terminal inexistente (terminalID)	C-Pago.1.6.3
		Invocación con Token de seguridad duplicado (secToken)	C-Pago.1.6.4
Notificación de resultado de transacción al comercio		Recepción de código HTTP 200	C-Pago.1.7.1
		Recepción de código HTTP diferente de HTTP 200	C-Pago.1.7.2
Servidor de C-Pago no disponible		Servidor de C-Pago no disponible	C-Pago.1.8.1
Cierre de interfaz durante el proceso		Cierre de formulario de datos de pago	C-Pago.1.9.1
		Cierre de pantalla de confirmación	C-Pago.1.9.2
		Cierre de pantalla de resultado	C-Pago.1.9.3
Redirección a url del comercio		Url de confirmación válida	C-Pago.1.10.1
		Url de confirmación inválida	C-Pago.1.10.2

4.3.2. Resumen de resultados de pruebas unitarias aplicación C-Pago

La descripción de los resultados de los casos de prueba más relevantes se muestra a continuación:

Caso de prueba	Invocación correcta de la interfaz: C-Pago.1.1.2 - Invocación con parámetro opcional merchantLogo
Resultado	En la sección del comercio debe mostrarse el logo y el nombre del mismo.

esperado	
Resultado	

Caso de prueba	Solicitud incorrecta de envío del formulario de pago: C-Pago.1.2.5 - Solicitud de envío del formulario de pago luego de 90 segundos
-----------------------	---

Resultado esperado	Debe mostrarse una pantalla de error donde se informe que ha expirado el tiempo límite para realizar la operación
---------------------------	---



Resultado	
------------------	--

Caso de prueba	Solicitud correcta de envío del formulario de pago: C-Pago.1.3.2 - Solicitud de envío del formulario de pago antes de 90 segundos
-----------------------	---

Resultado esperado	Debe mostrarse la pantalla de confirmación
---------------------------	--

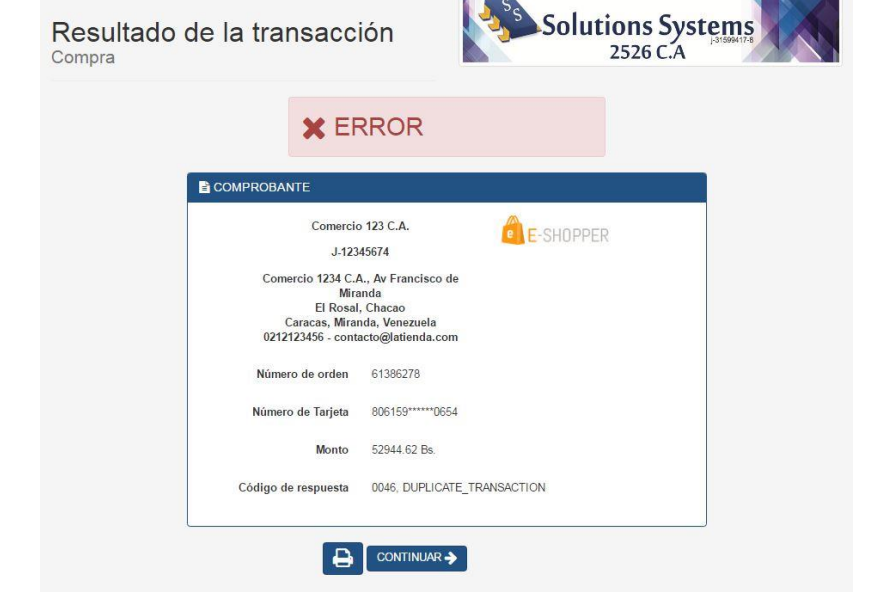
Resultado	
------------------	--

	<div style="text-align: right;">  </div> <p>Confirmación de la transacción Compra</p> <div style="border: 1px solid #ccc; padding: 10px; margin: 10px auto; width: 80%;"> <p style="text-align: center; background-color: #0056b3; color: white; padding: 5px;">DATOS DE LA TRANSACCIÓN</p> <div style="display: flex; justify-content: space-between; align-items: center;"> <div style="text-align: center;"> <p>Comercio 123 C.A. J-12345674</p> <p>Comercio 1234 C.A., Av Francisco de Miranda El Rosal, Chacao Caracas, Miranda, Venezuela 0212123456 - contacto@latienda.com</p> </div> <div style="text-align: center;">  </div> </div> <div style="margin-top: 10px;"> <p>Número de Tarjeta: 425317****2174</p> <p>Fecha de vencimiento: 01/16</p> <p>Nombre del tarjetahabiente: Emily Corro</p> <p>Cédula de identidad: 20191602</p> <p>Monto de la transacción: 37398.67 Bs.</p> </div> <div style="text-align: center; margin-top: 10px;"> ✕ CANCELAR ✓ CONTINUAR </div> </div>
--	--

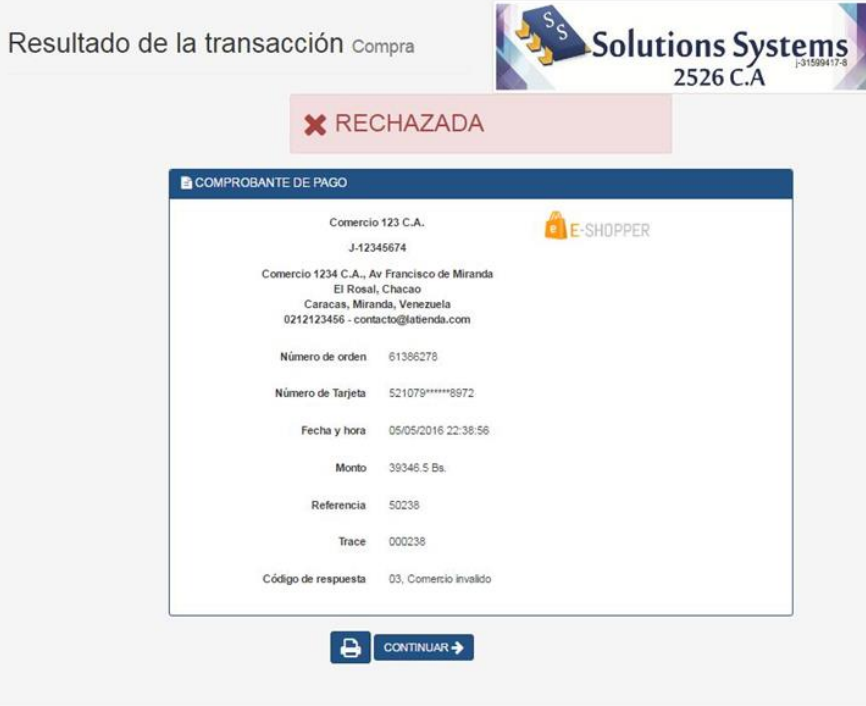
Caso de prueba	Confirmación de datos: C-Pago.1.4.2 - Confirmación de datos antes de 90 segundos
Resultado esperado	Debe mostrarse la pantalla con el resultado de la transacción
Resultado	<div style="text-align: right;">  </div> <p>Resultado de la transacción Compra</p> <div style="text-align: center; margin: 10px auto; width: 60%;"> <div style="background-color: #f08080; padding: 10px; display: inline-block; border: 1px solid #ccc;"> ✕ RECHAZADA </div> </div> <div style="border: 1px solid #ccc; padding: 10px; margin: 10px auto; width: 80%;"> <p style="text-align: center; background-color: #0056b3; color: white; padding: 5px;">COMPROBANTE DE PAGO</p> <div style="display: flex; justify-content: space-between; align-items: center;"> <div style="text-align: center;"> <p>Comercio 123 C.A. J-12345674</p> <p>Comercio 1234 C.A., Av Francisco de Miranda El Rosal, Chacao Caracas, Miranda, Venezuela 0212123456 - contacto@latienda.com</p> </div> <div style="text-align: center;">  </div> </div> <div style="margin-top: 10px;"> <p>Número de orden: 61386278</p> <p>Número de Tarjeta: 521079****8672</p> <p>Fecha y hora: 05/05/2016 22:38:56</p> <p>Monto: 39346.5 Bs.</p> <p>Referencia: 50238</p> <p>Trace: 000238</p> <p>Código de respuesta: 03, Comercio invalido</p> </div> <div style="text-align: center; margin-top: 10px;"> 🖨 CONTINUAR → </div> </div>


Caso de prueba	Invocación de servicio web de venta: C-Pago.1.5.3 - Transacción rechazada por la entidad financiera
Resultado esperado	Debe mostrarse la pantalla con el resultado de la transacción y el mensaje de "Rechazada"

Resultado	 <p>Resultado de la transacción Compra</p> <p>RECHAZADA</p> <p>COMPROBANTE DE PAGO</p> <p>Comercio 123 C.A. J-12345674</p> <p>Comercio 1234 C.A., Av Francisco de Miranda El Rosal, Chacao Caracas, Miranda, Venezuela 0212123456 - contacto@latienda.com</p> <p>Número de orden 61386278</p> <p>Número de Tarjeta 521079****8972</p> <p>Fecha y hora 05/05/2016 22:38:56</p> <p>Monto 39346.5 Bs.</p> <p>Referencia 50238</p> <p>Trace 000238</p> <p>Código de respuesta 03, Comercio invalido</p> <p>CONTINUAR →</p>
------------------	--

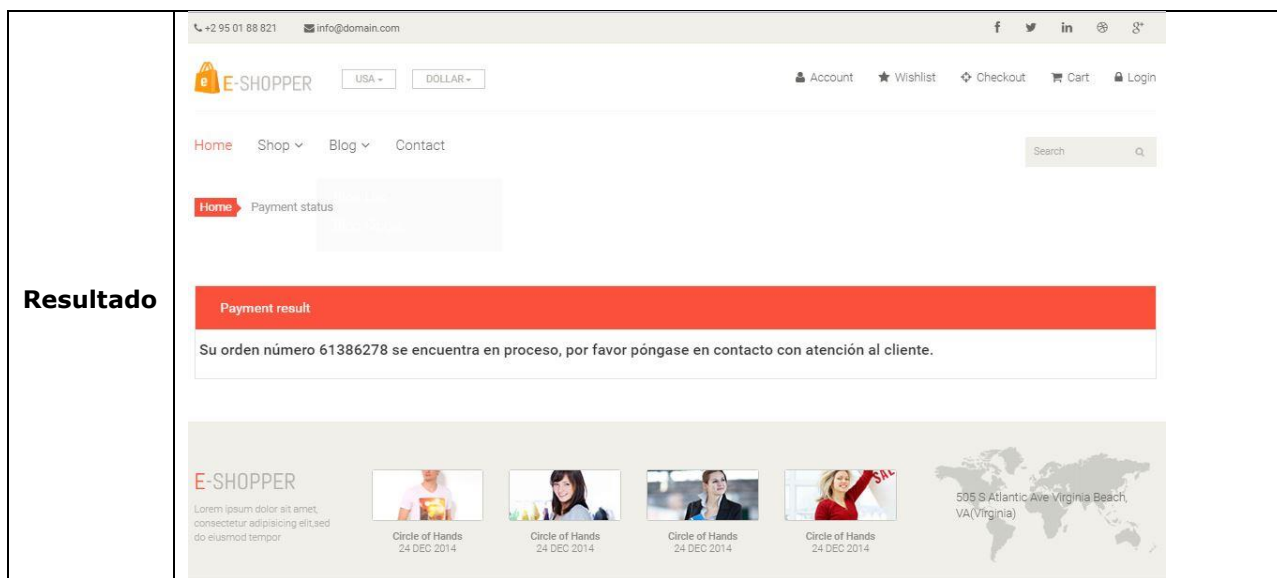
Caso de prueba	Invocación incorrecta de la interfaz: C-Pago.1.6.4 - Invocación con token de seguridad duplicado (secToken)
Resultado esperado	Debe mostrarse la pantalla de error indicando que la solicitud es incorrecta debido a que la transacción es duplicada
Resultado	 <p>Resultado de la transacción Compra</p> <p>ERROR</p> <p>COMPROBANTE</p> <p>Comercio 123 C.A. J-12345674</p> <p>Comercio 1234 C.A., Av Francisco de Miranda El Rosal, Chacao Caracas, Miranda, Venezuela 0212123456 - contacto@latienda.com</p> <p>Número de orden 61386278</p> <p>Número de Tarjeta 806159****0654</p> <p>Monto 52944.62 Bs.</p> <p>Código de respuesta 0046, DUPLICATE_TRANSACTION</p> <p>CONTINUAR →</p>

Caso de	Notificación de resultado de transacción al comercio: C-Pago.1.7.1 - Recepción de código
----------------	--

prueba	HTTP 200
Resultado esperado	La aplicación C-Pago recibe una respuesta HTTP 200 por parte del site del comercio y muestra la pantalla con el resultado de la transacción
Resultado	

Caso de prueba	Servidor de C-Pago no disponible: C-Pago.1.8.1 - Servidor de C-Pago no disponible
Resultado esperado	Ya que la aplicación no se encuentra disponible para recibir solicitudes debe mostrarse un mensaje de error HTTP 404 genérico
Resultado	

Caso de prueba	Redirección a url del comercio: C-Pago.1.10.1 - Url de confirmación válida
Resultado esperado	Debe mostrarse la interfaz de la url a donde fue redirigido el usuario



Resultado

4.3.3. Conjunto de pruebas unitarias Motor V-Financial

A continuación se listan los escenarios de pruebas para cada servicio web proporcionado por el motor transaccional.

Tabla 35. Tabla resumen de las instancias de casos de prueba del Motor V-Financial. (Elaboración propia).

Caso de uso	Caso de prueba	Instancia de caso de prueba	Identificador de instancia de caso de prueba
V-Financial.1.1.1 Listar todos los Comercios	Solicitud correcta de Listar todos los Comercios	Existe al menos un comercio registrado en el sistema	V-Financial.1.1.1.1.1
		No existe ningún comercio registrado en el sistema	V-Financial.1.1.1.1.2
V-Financial.1.1.2 Listar Comercios de Multicomercio	Solicitud correcta de Listar Comercios de Multicomercio	Multicomercio existe pero no tiene ningún comercio asociado	V-Financial.1.1.2.1.1
		Multicomercio existe y tiene al menos un comercio asociado	V-Financial.1.1.2.1.2
	Solicitud incorrecta de Listar Comercios de Multicomercio	El multicomercio indicado no existe	V-Financial.1.1.2.2.1
V-Financial.1.1.3 Consultar Comercio	Solicitud correcta de Consultar Comercio	El comercio solicitado existe	V-Financial.1.1.3.1.1
	Solicitud incorrecta de Consultar Comercio	El comercio solicitado no existe	V-Financial.1.1.3.2.1
V-Financial.1.1.4 Afiliar Comercio	Solicitud correcta de Afiliar Comercio	Solicitud con todos los parámetros requeridos	V-Financial.1.1.4.1.1
	Solicitud incorrecta de Afiliar Comercio	Solicitud sin uno o más parámetro(s) requerido(s)	V-Financial.1.1.4.2.1
V-Financial.1.1.4.1	Solicitud correcta	Solicitud con todos los	V-

Caso de uso	Caso de prueba	Instancia de caso de prueba	Identificador de instancia de caso de prueba
Afiliar Multicomercio	de Afiliar Multicomercio	parámetros requeridos	Financial.1.1.4.1.1.1
	Solicitud incorrecta de Afiliar Multicomercio	Solicitud sin uno o más parámetro(s) requerido(s)	V- Financial.1.1.4.1.2.1
V-Financial.1.1.5 Modificar datos de Comercio	Solicitud correcta de Modificar datos de Comercio	El comercio indicado existe	V-Financial.1.1.5.1.1
	Solicitud incorrecta de Modificar datos de Comercio	El comercio indicado no existe	V-Financial.1.1.5.2.1
V-Financial.1.1.5.1 Modificar datos de Multicomercio	Solicitud correcta de Modificar datos de Multicomercio	El multicomercio indicado existe	V- Financial.1.1.5.1.1.1
	Solicitud incorrecta de Modificar datos de Multicomercio	El multicomercio indicado no existe	V- Financial.1.1.5.1.2.1
V-Financial.1.1.6 Consultar Comercio por Terminal	Solicitud correcta de Consultar Comercio por Terminal	El comercio indicado existe	V-Financial.1.1.6.1.1
	Solicitud incorrecta de Consultar Comercio por Terminal	El comercio indicado no existe	V-Financial.1.1.6.2.1
V-Financial.1.1.7 Habilitar Comercio	Solicitud correcta de Habilitar Comercio	El comercio indicado existe	V-Financial.1.1.7.1.1
	Solicitud incorrecta de Habilitar Comercio	El comercio indicado no existe	V-Financial.1.1.7.2.1
V-Financial.1.1.8 Deshabilitar Comercio	Solicitud correcta de Deshabilitar Comercio	El comercio indicado existe	V-Financial.1.1.8.1.1
	Solicitud incorrecta de Deshabilitar Comercio	El comercio indicado no existe	V-Financial.1.1.8.2.1
V-Financial.1.1.9 Asociar Comercio o Multicomercio a Corporación	Solicitud correcta de Asociar Comercio o Multicomercio a Corporación	El comercio/multicomercio a asociar existe	V-Financial.1.1.9.1.1
	Solicitud incorrecta de Asociar Comercio o Multicomercio a Corporación	El comercio/multicomercio a asociar no existe	V-Financial.1.1.9.2.1
	Solicitud incorrecta de Asociar Comercio o Multicomercio a Corporación	La corporación a asociar no existe	V-Financial.1.1.9.2.2
V-Financial.1.1.10 Asociar Comercio a Multicomercio	Solicitud correcta de Asociar Comercio a Multicomercio	Tanto el comercio como el multicomercio a asociar existen y se colocan los parámetros en la url de manera correcta	V-Financial.1.1.10.1.1
	Solicitud incorrecta de Asociar Comercio a Multicomercio	El comercio a asociar no existe	V-Financial.1.1.10.2.1

Caso de uso	Caso de prueba	Instancia de caso de prueba	Identificador de instancia de caso de prueba
	de Asociar Comercio a Multicomercio	El multicomercio a asociar no existe	V-Financial.1.1.10.2.2
		En el parámetro identificador del comercio de la url se coloca uno de multicomercio	V-Financial.1.1.10.2.3
		En el parámetro identificador del multicomercio de la url se coloca uno de comercio	V-Financial.1.1.10.2.4
V-Financial.1.1.11 Listar todos los Multicomercios	Solicitud correcta Listar todos los Multicomercios	Existe al menos un multicomercio registrado en el sistema	V-Financial.1.1.11.1.1
		No existe ningún multicomercio registrado en el sistema	V-Financial.1.1.11.1.2
V-Financial.1.2 Gestión de Corporaciones	Solicitud correcta de Gestión de Corporaciones	Formato JSON válido	V-Financial.1.2.0.1.1
		Formato XML válido	V-Financial.1.2.0.1.2
	Solicitud incorrecta de Gestión de Corporaciones	Formato JSON inválido	V-Financial.1.2.0.2.1
		Formato XML inválido	V-Financial.1.2.0.2.2
V-Financial.1.2.1 Crear Corporación	Solicitud correcta de Crear Corporación	Solicitud con todos los parámetros requeridos	V-Financial.1.2.1.1.1
	Solicitud incorrecta de Crear Corporación	Solicitud sin uno o más parámetro(s) requerido(s)	V-Financial.1.2.1.2.1
V-Financial.1.2.2 Consultar Corporación	Solicitud correcta de Consultar Corporación	La corporación solicitada existe	V-Financial.1.2.2.1.1
	Solicitud incorrecta de Consultar Corporación	La corporación solicitada no existe	V-Financial.1.2.2.2.1
V-Financial.1.2.3 Listar Corporaciones	Solicitud correcta de Listar Corporaciones	Existe al menos una corporación registrada	V-Financial.1.2.3.1.1
	Solicitud incorrecta de Listar Corporaciones	No existe ninguna corporación registrada	V-Financial.1.2.3.2.1
V-Financial.1.2.4 Modificar datos de Corporación	Solicitud correcta de Modificar datos de Corporación	La corporación solicitada existe	V-Financial.1.2.4.1.1
	Solicitud incorrecta de Modificar datos de Corporación	La corporación solicitada no existe	V-Financial.1.2.4.2.1
V-Financial.1.2.5 Listar Comercios asociados a Corporación	Solicitud correcta de Listar Comercios asociados a Corporación	La corporación existe pero no tiene ningún comercio asociado	V-Financial.1.2.5.1.1
		La corporación existe y tiene al menos un comercio asociado	V-Financial.1.2.5.1.2
	Solicitud incorrecta de Listar Comercios asociados a Corporación	La corporación indicada no existe	V-Financial.1.2.5.2.1

Caso de uso	Caso de prueba	Instancia de caso de prueba	Identificador de instancia de caso de prueba
V-Financial.1.3 Gestión de Terminales	Solicitud correcta de Gestión de Terminales	Formato JSON válido	V-Financial.1.3.0.1.1
		Formato XML válido	V-Financial.1.3.0.1.2
	Solicitud incorrecta de Gestión de Terminales	Formato JSON inválido	V-Financial.1.3.0.2.1
		Formato XML inválido	V-Financial.1.3.0.2.2
V-Financial.1.3.1 Consultar Terminales de Comercio	Solicitud correcta de Consultar Terminales de Comercio	El comercio existe pero no tiene ningún terminal asociado	V-Financial.1.3.1.1.1
		El comercio existe y tiene al menos un terminal asociado	V-Financial.1.3.1.1.2
	Solicitud incorrecta de Consultar Terminales de Comercio	El comercio indicado no existe	V-Financial.1.3.1.2.1
V-Financial.1.3.2 Consultar Terminal	Solicitud correcta de Consultar Terminal	El terminal solicitado existe	V-Financial.1.3.2.1.1
	Solicitud incorrecta de Consultar Terminales de Comercio	El terminal solicitado no existe	V-Financial.1.3.2.2.1
V-Financial.1.3.3 Crear Terminal	Solicitud correcta de Crear Terminal	Solicitud con todos los parámetros requeridos	V-Financial.1.3.3.1.1
	Solicitud incorrecta de Crear Terminal	Solicitud sin uno o más parámetro(s) requerido(s)	V-Financial.1.3.3.2.1
V-Financial.1.3.4 Habilitar Terminal	Solicitud correcta de Habilitar Terminal	El terminal indicado existe	V-Financial.1.3.4.1.1
	Solicitud incorrecta de Habilitar Terminal	El terminal indicado no existe	V-Financial.1.3.4.2.1
V-Financial.1.3.5 Deshabilitar Terminal	Solicitud correcta de Deshabilitar Terminal	El terminal indicado existe	V-Financial.1.3.5.1.1
	Solicitud incorrecta de Deshabilitar Terminal	El terminal indicado no existe	V-Financial.1.3.5.2.1
V-Financial.1.3.6 Listar parámetros de terminal bancario de Terminal	Solicitud correcta de Listar parámetros de terminal bancario de Terminal	El terminal indicado existe pero no posee ningún parámetro de terminal bancario asociado	V-Financial.1.3.6.1.1
		El terminal indicado existe y posee al menos un parámetro de terminal bancario asociado	V-Financial.1.3.6.1.2
	Solicitud incorrecta de Listar parámetros de terminal bancario de Terminal	El terminal indicado no existe	V-Financial.1.3.6.2.1
V-Financial.1.3.7 Añadir parámetro de terminal bancario a Terminal	Solicitud correcta de Añadir parámetro de terminal bancario a Terminal	Solicitud con todos los parámetros requeridos	V-Financial.1.3.7.1.1

Caso de uso	Caso de prueba	Instancia de caso de prueba	Identificador de instancia de caso de prueba
	Solicitud incorrecta de Añadir parámetro de terminal bancario a Terminal	Solicitud sin uno o más parámetro(s) requerido(s)	V-Financial.1.3.7.2.1
V-Financial.1.4 Gestión de Bancos	Solicitud correcta de Gestión de Bancos	Formato JSON válido	V-Financial.1.4.0.1.1
		Formato XML válido	V-Financial.1.4.0.1.2
	Solicitud incorrecta de Gestión de Bancos	Formato JSON inválido	V-Financial.1.4.0.2.1
		Formato XML inválido	V-Financial.1.4.0.2.2
V-Financial.1.4.1 Listar Bancos	Solicitud correcta de Listar Bancos	Existe al menos un banco registrado en el sistema	V-Financial.1.4.1.1.1
		No existe ningún banco registrado en el sistema	V-Financial.1.4.1.1.2
V-Financial.1.4.2 Consultar Banco	Solicitud correcta de Consultar Banco	El banco solicitado existe	V-Financial.1.4.2.1.1
	Solicitud incorrecta de Consultar Banco	El banco solicitado no existe	V-Financial.1.4.2.2.1
V-Financial.1.5 Gestión de Afiliaciones	Solicitud correcta de Gestión de Afiliaciones	Formato JSON válido	V-Financial.1.5.0.1.1
		Formato XML válido	V-Financial.1.5.0.1.2
	Solicitud incorrecta de Gestión de Afiliaciones	Formato JSON inválido	V-Financial.1.5.0.2.1
		Formato XML inválido	V-Financial.1.5.0.2.2
V-Financial.1.5.1 Crear Afiliación	Solicitud correcta de Crear Afiliación	Solicitud con todos los parámetros requeridos	V-Financial.1.5.1.1.1
	Solicitud incorrecta de Crear Afiliación	Solicitud sin uno o más parámetro(s) requerido(s)	V-Financial.1.5.1.2.1
V-Financial.1.5.2 Listar Afiliaciones de Comercio	Solicitud correcta de Listar Afiliaciones de Comercio	El comercio existe pero no tiene ninguna afiliación asociada	V-Financial.1.5.2.1.1
		El comercio existe y tiene al menos una afiliación asociada	V-Financial.1.5.2.1.2
	Solicitud incorrecta de Listar Afiliaciones de Comercio	El comercio indicado no existe	V-Financial.1.5.2.2.1
V-Financial.1.6 Gestión de Adquirencia	Solicitud correcta de Gestión de Adquirencia	Formato JSON válido	V-Financial.1.6.0.1.1
		Formato XML válido	V-Financial.1.6.0.1.2
	Solicitud incorrecta de Gestión de Adquirencia	Formato JSON inválido	V-Financial.1.6.0.2.1
		Formato XML inválido	V-Financial.1.6.0.2.2
V-Financial.1.6.1 Listar Adquirientes de Comercio	Solicitud correcta de Listar Adquirientes de Comercio	El comercio existe pero no tiene ningún adquirente asociado	V-Financial.1.6.1.1.1
		El comercio existe y tiene al menos un adquirente asociado	V-Financial.1.6.1.1.2
	Solicitud incorrecta de Listar Adquirientes de Comercio	El comercio indicado no existe	V-Financial.1.6.1.2.1

Caso de uso	Caso de prueba	Instancia de caso de prueba	Identificador de instancia de caso de prueba
V-Financial.1.6.2 Listar Adquirientes de Multicomercio	Solicitud correcta de Listar Adquirientes de Multicomercio	El multicomercio existe pero no tiene ningún adquiriente asociado	V-Financial.1.6.2.1.1
	Solicitud incorrecta de Listar Adquirientes de Multicomercio	El multicomercio existe y tiene al menos un adquiriente asociado	V-Financial.1.6.2.1.2
	Solicitud incorrecta de Listar Adquirientes de Multicomercio	El multicomercio indicado no existe	V-Financial.1.6.2.2.1
V-Financial.1.6.3 Consultar Adquiriente	Solicitud correcta de Consultar Adquiriente	El adquiriente solicitado existe	V-Financial.1.6.3.1.1
	Solicitud incorrecta de Consultar Adquiriente	El adquiriente solicitado no existe	V-Financial.1.6.3.2.1
V-Financial.1.6.4 Crear Adquiriente	Solicitud correcta de Crear Adquiriente	Solicitud con todos los parámetros requeridos	V-Financial.1.6.4.1.1
	Solicitud incorrecta de Crear Adquiriente	Solicitud sin uno o más parámetro(s) requerido(s)	V-Financial.1.6.4.2.1
V-Financial.1.7 Gestión de Reglas	Solicitud correcta de Gestión de Reglas	Formato JSON válido	V-Financial.1.7.0.1.1
		Formato XML válido	V-Financial.1.7.0.1.2
	Solicitud incorrecta de Gestión de Reglas	Formato JSON inválido	V-Financial.1.7.0.2.1
		Formato XML inválido	V-Financial.1.7.0.2.2
V-Financial.1.7.1 Listar Conjuntos de Reglas de Comercio	Solicitud correcta de Listar Conjuntos de Reglas de Comercio	El comercio existe pero no tiene ningún conjunto de reglas asociado	V-Financial.1.7.1.1.1
		El comercio existe y tiene al menos un conjunto de reglas asociado	V-Financial.1.7.1.1.1
	Solicitud incorrecta de Listar Conjuntos de Reglas de Comercio	El comercio indicado no existe	V-Financial.1.7.1.2.1
V-Financial.1.7.2 Consultar Conjunto de Reglas	Solicitud correcta de Consultar Conjunto de Reglas	El conjunto de reglas solicitado existe	V-Financial.1.7.2.1.1
	Solicitud incorrecta de Consultar Conjunto de Reglas	El conjunto de reglas solicitado no existe	V-Financial.1.7.2.2.1
V-Financial.1.7.3 Crear Conjunto de Reglas	Solicitud correcta de Crear Conjunto de Reglas	Solicitud con todos los parámetros requeridos	V-Financial.1.7.3.1.1
	Solicitud incorrecta de Crear Conjunto de Reglas	Solicitud sin uno o más parámetro(s) requerido(s)	V-Financial.1.7.3.2.1

Caso de uso	Caso de prueba	Instancia de caso de prueba	Identificador de instancia de caso de prueba
V-Financial.1.7.4 Modificar Conjunto de Reglas	Solicitud correcta de Modificar Conjunto de Reglas	El conjunto de reglas solicitado existe	V-Financial.1.7.4.1.1
	Solicitud incorrecta de Modificar Conjunto de Reglas	El conjunto de reglas solicitado no existe	V-Financial.1.7.4.2.1
V-Financial.1.7.5 Asignar Conjunto de Reglas a Afiliación de Comercio	Solicitud correcta de Asignar Conjunto de Reglas a Afiliación de Comercio	El comercio, el conjunto de reglas y la afiliación existen. Tanto el conjunto de reglas como la afiliación pertenecen al mismo comercio	V-Financial.1.7.5.1.1
		El comercio a asociar no existe	V-Financial.1.7.5.2.1
		El conjunto de reglas a asociar no existe	V-Financial.1.7.5.2.2
	Solicitud incorrecta de Asignar Conjunto de Reglas a Afiliación de Comercio	La afiliación a asociar no existe	V-Financial.1.7.5.2.3
		El conjunto de reglas a asociar existe pero no pertenece al comercio indicado	V-Financial.1.7.5.2.4
	La afiliación a asociar existe pero no pertenece al comercio indicado	V-Financial.1.7.5.2.5	
V-Financial.1.7.6 Eliminar Conjunto de Reglas	Solicitud correcta de Eliminar Conjunto de Reglas	El conjunto de reglas existe y no se encuentra activo para una afiliación	V-Financial.1.7.6.1.1
	Solicitud incorrecta de Eliminar Conjunto de Reglas	El conjunto de reglas no existe	V-Financial.1.7.6.2.1
		El conjunto de reglas existe pero se encuentra activo para una afiliación	V-Financial.1.7.6.2.2
V-Financial.1.7.7 Listar Reglas de Conjunto de Reglas	Solicitud correcta de Listar Reglas de Conjunto de Reglas	El conjunto de reglas existe pero no tiene ninguna regla asociada	V-Financial.1.7.7.1.1
		El conjunto de reglas existe y tiene al menos una regla asociada	V-Financial.1.7.7.1.2
	Solicitud incorrecta de Listar Reglas de Conjunto de Reglas	El conjunto de reglas indicado no existe	V-Financial.1.7.7.2.1
V-Financial.1.7.8 Listar Reglas por Afiliación	Solicitud correcta de Listar Reglas por Afiliación	La afiliación existe pero no tiene ninguna regla asociada	V-Financial.1.7.8.1.1
		La afiliación existe y tiene al menos una regla asociada	V-Financial.1.7.8.1.2
	Solicitud incorrecta de Listar Reglas por Afiliación	La afiliación de reglas indicada no existe	V-Financial.1.7.8.2.1
V-Financial.1.7.9 Crear Regla	Solicitud correcta de Crear Regla	Solicitud con todos los parámetros requeridos	V-Financial.1.7.8.1.1
	Solicitud incorrecta	Solicitud sin uno o más	V-Financial.1.7.8.2.1

Caso de uso	Caso de prueba	Instancia de caso de prueba	Identificador de instancia de caso de prueba
	de Crear Regla	parámetro(s) requerido(s)	
V-Financial.1.7.10 Consultar Regla	Solicitud correcta de Consultar Regla	La regla indicada existe	V-Financial.1.7.10.1.1
	Solicitud incorrecta de Consultar Regla	La regla indicada no existe	V-Financial.1.7.10.2.1
V-Financial.1.7.11 Modificar Regla	Solicitud correcta de Modificar Regla	La regla indicada existe	V-Financial.1.7.11.1.1
	Solicitud incorrecta de Modificar Regla	La regla indicada no existe	V-Financial.1.7.11.2.1
V-Financial.1.7.12 Eliminar Regla	Solicitud correcta de Eliminar Regla	La regla indicada existe	V-Financial.1.7.12.1.1
	Solicitud incorrecta de Eliminar Regla	La regla indicada no existe	V-Financial.1.7.12.2.1
V-Financial.1.7.13 Agregar Regla a Conjunto de Reglas	Solicitud correcta de Agregar Regla a Conjunto de Reglas	La regla y el conjunto de reglas existen	V-Financial.1.7.13.1.1
	Solicitud incorrecta de Agregar Regla a Conjunto de Reglas	La regla a asociar no existe	V-Financial.1.7.13.2.1
		El conjunto de reglas a asociar no existe	V-Financial.1.7.13.2.2
V-Financial.1.7.14 Remover Regla de Conjunto de Reglas	Solicitud correcta de Remover Regla de Conjunto de Reglas	La regla y el conjunto de reglas existen y la regla se encuentra asociada a este último	V-Financial.1.7.14.1.1
		La regla a asociar no existe	V-Financial.1.7.14.2.1
	Solicitud incorrecta de Remover Regla de Conjunto de Reglas	El conjunto de reglas a asociar no existe	V-Financial.1.7.14.2.2
		La regla y el conjunto de reglas existen pero la regla no se encuentra asociada a este último	V-Financial.1.7.14.2.3
V-Financial.1.8 Gestión de Dispositivos	Solicitud correcta de Gestión de Dispositivos	Formato JSON válido	V-Financial.1.8.0.1.1
		Formato XML válido	V-Financial.1.8.0.1.2
	Solicitud incorrecta de Gestión de Dispositivos	Formato JSON inválido	V-Financial.1.8.0.2.1
		Formato XML inválido	V-Financial.1.8.0.2.2
V-Financial.1.8.1 Consultar Tipo de Dispositivo	Solicitud correcta de Consultar Tipo de Dispositivo	El tipo de dispositivo existe	V-Financial.1.8.1.1.1
	Solicitud incorrecta de Consultar Tipo de Dispositivo	El tipo de dispositivo no existe	V-Financial.1.8.1.2.1
V-Financial.1.8.2 Crear Tipo de Dispositivo	Solicitud correcta de Crear Tipo de Dispositivo	Solicitud con todos los parámetros requeridos	V-Financial.1.8.2.1.1
	Solicitud incorrecta de Crear Tipo de Dispositivo	Solicitud sin uno o más parámetro(s) requerido(s)	V-Financial.1.8.2.2.1
V-Financial.1.8.3	Solicitud correcta	El tipo de dispositivo existe	V-Financial.1.8.3.1.1

Caso de uso	Caso de prueba	Instancia de caso de prueba	Identificador de instancia de caso de prueba
Modificar Tipo de Dispositivo	de Modificar Tipo de Dispositivo		
	Solicitud incorrecta de Modificar Tipo de Dispositivo	El tipo de dispositivo no existe	V-Financial.1.8.3.2.1
V-Financial.1.8.4 Eliminar Tipo de Dispositivo	Solicitud correcta de Eliminar Tipo de Dispositivo	El tipo de dispositivo existe	V-Financial.1.8.4.1.1
	Solicitud incorrecta de Eliminar Tipo de Dispositivo	El tipo de dispositivo no existe	V-Financial.1.8.4.2.1
V-Financial.1.8.5 Listar Tipos de Dispositivos	Solicitud correcta de Listar Tipos de Dispositivos	Existe al menos un tipo de dispositivo registrado en el sistema	V-Financial.1.8.5.1.1
		No existe ningún tipo de dispositivo registrado en el sistema	V-Financial.1.8.5.2.1
V-Financial.1.8.6 Consultar Dispositivo	Solicitud correcta de Consultar Dispositivo	El dispositivo existe	V-Financial.1.8.6.1.1
	Solicitud incorrecta de Consultar Dispositivo	El dispositivo no existe	V-Financial.1.8.6.2.1
V-Financial.1.8.7 Crear Dispositivo	Solicitud correcta de Crear Dispositivo	Solicitud sin uno o más parámetro(s) requerido(s)	V-Financial.1.8.7.1.1
	Solicitud incorrecta de Crear Dispositivo	Solicitud con todos los parámetros requeridos	V-Financial.1.8.7.2.1
V-Financial.1.8.8 Modificar Dispositivo	Solicitud correcta de Modificar Dispositivo	El dispositivo existe	V-Financial.1.8.8.1.1
	Solicitud incorrecta de Modificar Dispositivo	El dispositivo no existe	V-Financial.1.8.8.2.1
V-Financial.1.8.9 Eliminar Dispositivo	Solicitud correcta de Eliminar Dispositivo	El dispositivo existe	V-Financial.1.8.9.1.1
	Solicitud incorrecta de Eliminar Dispositivo	El dispositivo no existe	V-Financial.1.8.9.2.1
V-Financial.1.8.10 Listar Dispositivos	Solicitud correcta de Listar Dispositivos	Existe al menos un dispositivo registrado en el sistema	V-Financial.1.8.10.1.1
		No existe ningún dispositivo registrado en el sistema	V-Financial.1.8.10.2.1
V-Financial.1.8.11 Asignar Dispositivo a Terminal	Solicitud correcta de Asignar Dispositivo a Terminal	El dispositivo y el terminal a asociar existen y el dispositivo no ha sido asignado a otro terminal	V-Financial.1.8.11.1.1
	Solicitud incorrecta de Asignar Dispositivo a	El dispositivo a asociar no existe	V-Financial.1.8.11.2.1
	Dispositivo a	El terminal a asociar no existe	V-Financial.1.8.11.2.2

Caso de uso	Caso de prueba	Instancia de caso de prueba	Identificador de instancia de caso de prueba
	Terminal	El dispositivo con el serial indicado ya se encuentra asociado a otro terminal	V-Financial.1.8.11.2.3
V-Financial.1.8.12 Consultar Asignacion de Dispositivo a Terminal	Solicitud correcta de Consultar Asignación de Dispositivo a Terminal	La asignación de dispositivo a terminal existe	V-Financial.1.8.12.1.1
	Solicitud incorrecta de Consultar Asignación de Dispositivo a Terminal	La asignación de dispositivo a terminal no existe	V-Financial.1.8.12.2.1
V-Financial.1.8.13 Revocar Dispositivo Asignado	Solicitud correcta de Revocar Dispositivo Asignado	La asignación de dispositivo existe	V-Financial.1.8.13.1.1
	Solicitud incorrecta de Revocar Dispositivo Asignado	La asignación de dispositivo no existe	V-Financial.1.8.13.2.1
V-Financial.1.9 Gestión de Clasificación de Comercios	Solicitud correcta de Gestión de Clasificación de Comercios	Formato JSON válido	V-Financial.1.9.0.1.1
		Formato XML válido	V-Financial.1.9.0.1.2
	Solicitud incorrecta de Gestión de Clasificación de Comercios	Formato JSON inválido	V-Financial.1.9.0.2.1
		Formato XML inválido	V-Financial.1.9.0.2.2
V-Financial.1.9.1 Obtener lista de sectores económicos	Solicitud correcta de Obtener lista de sectores económicos	Existe al menos un sector económico registrado en el sistema	V-Financial.1.9.1.1.1
		No existe ningún sector económico registrado en el sistema	V-Financial.1.9.1.1.2
V-Financial.1.9.2 Obtener tipos de industrias por sector económico	Solicitud correcta de Obtener tipos de industrias por sector económico	El sector económico existe y tiene al menos un tipo de industria asociada	V-Financial.1.9.2.1.1
		El sector económico existe pero no tiene ningún tipo de industria asociada	V-Financial.1.9.2.1.2
	Solicitud incorrecta de Obtener tipos de industrias por sector económico	El sector económico no existe	V-Financial.1.9.2.2.1
V-Financial.1.9.3 Consultar datos de sector	Solicitud correcta de Consultar datos de sector	El sector económico existe	V-Financial.1.9.3.1.1
	Solicitud incorrecta de Consultar datos de sector	El sector económico no existe	V-Financial.1.9.3.2.1
V-Financial.1.9.4 Consultar datos de	Solicitud correcta de Consultar datos	El tipo de industria existe	V-Financial.1.9.4.1.1

Caso de uso	Caso de prueba	Instancia de caso de prueba	Identificador de instancia de caso de prueba
tipo de industria	de tipo de industria		
	Solicitud incorrecta de Consultar datos de tipo de industria	El tipo de industria no existe	V-Financial.1.9.4.2.1
V-Financial.1.10 Gestión de Canales de Pago	Solicitud correcta de Gestión de Canales de Pago	Formato JSON válido	V-Financial.1.10.0.1.1
		Formato XML válido	V-Financial.1.10.0.1.2
	Solicitud incorrecta de Gestión de Canales de Pago	Formato JSON inválido	V-Financial.1.10.0.2.1
		Formato XML inválido	V-Financial.1.10.0.2.2
V-Financial.1.10.1 Listar Canales de Pago	Solicitud correcta de Listar Canales de Pago	Existe al menos un canal de pago registrado en el sistema	V-Financial.1.10.0.1.1
		No existe ningún canal de pago registrado en el sistema	V-Financial.1.10.0.1.2
V-Financial.2.1 Realizar transacción	Realizar transacción en formato correcto	Realizar transacción en formato JSON con el parámetro de tipo de transacción inválido (<i>type</i>)	V-Financial.2.1.1.1
		Realizar transacción en formato JSON con el parámetro de canal de pago inválido (<i>channelCode</i>)	V-Financial.2.1.1.2
		Realizar transacción en formato JSON con el parámetro de terminal inválido (<i>terminalId</i>)	V-Financial.2.1.1.3
		Realizar transacción en formato JSON con un parámetro de token de seguridad (<i>secToken</i>) que ya se encuentre en la base de datos	V-Financial.2.1.1.4
		Realizar transacción en formato JSON con un parámetro de token de seguridad (<i>secToken</i>) que no coincida con el calculado	V-Financial.2.1.1.5
		Realizar transacción en formato JSON desde el terminal de un comercio que no se encuentra habilitado	V-Financial.2.1.1.6
		Realizar transacción en formato JSON desde un terminal que no se encuentra habilitado	V-Financial.2.1.1.7
		Realizar transacción en formato JSON con todos los parámetros válidos	V-Financial.2.1.1.8
		Realizar transacción en formato XML con el parámetro de tipo de transacción inválido (<i>type</i>)	V-Financial.2.1.1.9
		Realizar transacción en formato XML con el parámetro de canal de pago inválido (<i>channelCode</i>)	V-Financial.2.1.1.10

Caso de uso	Caso de prueba	Instancia de caso de prueba	Identificador de instancia de caso de prueba
		Realizar transacción en formato XML con el parámetro identificador de terminal inválido (terminalId)	V-Financial.2.1.1.11
		Realizar transacción en formato XML con un parámetro de token de seguridad (secToken) que ya se encuentre en la base de datos	V-Financial.2.1.1.12
		Realizar transacción en formato XML con un parámetro de token de seguridad (secToken) que no coincida con el calculado	V-Financial.2.1.1.13
		Realizar transacción en formato XML desde el terminal de un comercio que no se encuentra habilitado	V-Financial.2.1.1.14
		Realizar transacción en formato XML desde un terminal que no se encuentra habilitado	V-Financial.2.1.1.15
		Realizar transacción en formato XML con todos los parámetros válidos	V-Financial.2.1.1.16
	Realizar transacción en formato incorrecto	Realizar transacción con formato JSON incorrecto	V-Financial.2.1.2.1
		Realizar transacción con formato XML incorrecto	V-Financial.2.1.2.2

4.3.3.1. Resultados de pruebas unitarias en módulo API de Acceso a Datos

A manera de resumen, se detallan los reportes de resultados para un subconjunto de las instancias de casos de pruebas definidas. De esta forma se comprueba el funcionamiento correcto de la serie de funciones provistas por el aplicativo. Todos los casos de prueba ejecutados correctamente sobre los módulos de la aplicación se marcan como "FINISHED"



TestSuite Results Report for Pruebas API de Acceso a Datos

Result Metrics

Result Metrics

Start Time	Thu May 05 12:44:07 GMT-04:00 2016
End Time	Thu May 05 12:45:28 GMT-04:00 2016
Time Taken	81194 ms
TestCase Count	64
Failed TestCase Count	0
TestStep Count	356
Failed TestStep Count	0
Assertion Count	562
Failed Assertion Count	0

TestCase Results

TestCase	Status	Start Time	Time Taken	Reason
V-Financial.1.1.1.1.1.1	FINISHED	12:44:07	119 ms	
V-Financial.1.1.2.1.2	FINISHED	12:44:07	46 ms	
V-Financial.1.1.3.1.1	FINISHED	12:44:07	45 ms	
V-Financial.1.1.4.1.1	FINISHED	12:44:07	439 ms	
V-Financial.1.1.4.1.1.1	FINISHED	12:44:08	85 ms	
V-Financial.1.1.5.1.1	FINISHED	12:44:08	967 ms	
V-Financial.1.1.5.1.1.1	FINISHED	12:44:09	187 ms	
V-Financial.1.1.6.1.1	FINISHED	12:44:09	1890 ms	
V-Financial.1.1.7.1.1	FINISHED	12:44:11	433 ms	
V-Financial.1.1.8.1.1	FINISHED	12:44:12	403 ms	
V-Financial.1.1.9.1.1	FINISHED	12:44:12	1186 ms	
V-Financial.1.1.10.1.1	FINISHED	12:44:13	282 ms	
V-Financial.1.1.11.1.1	FINISHED	12:44:14	31 ms	
V-Financial.1.2.1.1.1	FINISHED	12:44:14	84 ms	
V-Financial.1.2.2.1.1	FINISHED	12:44:14	51 ms	

created with Ready! API on Thu May 05 12:45:36 GMT-04:00 2016

Figura 68. Resultados de ejecución de casos de la API de Acceso a Datos en herramienta de pruebas SoapUI. (Elaboración propia).



TestSuite Results Report for Pruebas API de Acceso a Datos

TestCase Results

TestCase	Status	Start Time	Time Taken	Reason
V-Financial.1.2.3.1.1	FINISHED	12:44:14	45 ms	
V-Financial.1.2.4.1.1	FINISHED	12:44:14	48 ms	
V-Financial.1.2.5.1.2	FINISHED	12:44:14	897 ms	
V-Financial.1.3.1.1.2	FINISHED	12:44:15	1617 ms	
V-Financial.1.3.2.1.1	FINISHED	12:44:17	1759 ms	
V-Financial.1.3.3.1.1	FINISHED	12:44:19	1314 ms	
V-Financial.1.3.4.1.1	FINISHED	12:44:20	1906 ms	
V-Financial.1.3.5.1.1	FINISHED	12:44:22	2366 ms	
V-Financial.1.3.6.1.2	FINISHED	12:44:24	2793 ms	
V-Financial.1.3.7.1.1	FINISHED	12:44:27	2773 ms	
V-Financial.1.4.1.1.1	FINISHED	12:44:30	27 ms	
V-Financial.1.4.2.1.1	FINISHED	12:44:30	20 ms	
V-Financial.1.5.2.1.1	FINISHED	12:44:30	953 ms	
V-Financial.1.5.2.1.2	FINISHED	12:44:31	1441 ms	
V-Financial.1.6.1.1.2	FINISHED	12:44:33	2031 ms	
V-Financial.1.6.3.1.1	FINISHED	12:44:35	1699 ms	
V-Financial.1.6.4.1.1	FINISHED	12:44:37	1391 ms	
V-Financial.1.7.1.1.2	FINISHED	12:44:38	1825 ms	
V-Financial.1.7.2.1.1	FINISHED	12:44:40	1886 ms	
V-Financial.1.7.3.1.1	FINISHED	12:44:42	1470 ms	
V-Financial.1.7.4.1.1	FINISHED	12:44:44	2721 ms	
V-Financial.1.7.5.1.1	FINISHED	12:44:46	2733 ms	
V-Financial.1.8.11.1.1	FINISHED	12:44:49	2585 ms	
V-Financial.1.7.6.1.1	FINISHED	12:44:52	2034 ms	
V-Financial.1.7.7.1.2	FINISHED	12:44:54	4062 ms	
V-Financial.1.7.8.1.1	FINISHED	12:44:58	2386 ms	
V-Financial.1.7.9.1.1	FINISHED	12:45:01	2266 ms	
V-Financial.1.7.10.1.1	FINISHED	12:45:03	2580 ms	

created with Ready! API on Thu May 05 12:45:37 GMT-04:00 2016

Figura 69. Resultados de ejecución de casos de la API de Acceso a Datos en herramienta de pruebas SoapUI (cont.). (Elaboración propia).



TestSuite Results Report for Pruebas API de Acceso a Datos

TestCase Results

TestCase	Status	Start Time	Time Taken	Reason
V-Financial.1.7.11.1.1	FINISHED	12:45:06	3054 ms	
V-Financial.1.7.12.1.1	FINISHED	12:45:09	3081 ms	
V-Financial.1.7.13.1.1	FINISHED	12:45:12	3901 ms	
V-Financial.1.7.14.1.1	FINISHED	12:45:16	4092 ms	
V-Financial.1.8.1.1.1	FINISHED	12:45:20	68 ms	
V-Financial.1.8.2.1.1	FINISHED	12:45:20	74 ms	
V-Financial.1.8.3.1.1	FINISHED	12:45:21	240 ms	
V-Financial.1.8.4.1.1	FINISHED	12:45:21	63 ms	
V-Financial.1.8.5.1.1	FINISHED	12:45:21	22 ms	
V-Financial.1.8.6.1.1	FINISHED	12:45:21	63 ms	
V-Financial.1.8.7.1.1	FINISHED	12:45:21	83 ms	
V-Financial.1.8.8.1.1	FINISHED	12:45:21	96 ms	
V-Financial.1.8.9.1.1	FINISHED	12:45:21	118 ms	
V-Financial.1.8.10.1.1	FINISHED	12:45:21	29 ms	
V-Financial.1.8.12.1.1	FINISHED	12:45:22	2581 ms	
V-Financial.1.8.13.1.1	FINISHED	12:45:24	3120 ms	
V-Financial.1.9.1.1.1	FINISHED	12:45:27	16 ms	
V-Financial.1.9.2.1.1	FINISHED	12:45:27	59 ms	
V-Financial.1.9.3.1.1	FINISHED	12:45:28	39 ms	
V-Financial.1.9.4.1.1	FINISHED	12:45:28	58 ms	
V-Financial.1.10.0.1.1	FINISHED	12:45:28	12 ms	

created with Ready! API on Thu May 05 12:45:37 GMT-04:00 2016

Figura 70. Resultados de ejecución de casos de la API de Acceso a Datos en herramienta de pruebas SoapUI (cont.).
(Elaboración propia).

4.3.3.2. Resultados de pruebas unitarias en módulo Interfaz de Pago



TestSuite Results Report for Pruebas Interfaz de Pago

TestSuite Metrics

Overview

Project	V-Financial
Test suite name	Pruebas Interfaz de Pago
Description	Pruebas funcionales del módulo Interfaz de Pago del paquete de software V-Financial
Environment	Desarrollo remoto

Test Case Summary

✓ V-Financial.2.1.1.1	Realizar transacción en formato JSON con el parámetro de tipo de transacción inválido (type)
✓ V-Financial.2.1.1.2	Realizar transacción en formato JSON con el parámetro de canal de pago inválido (channelCode)
✓ V-Financial.2.1.1.3	Realizar transacción en formato JSON con el parámetro identificador de terminal inválido (terminalId)
✓ V-Financial.2.1.1.4	Realizar transacción en formato JSON con un parámetro de token de seguridad (secToken) que ya se encuentre en la base de datos
✓ V-Financial.2.1.1.5	Realizar transacción en formato JSON con un parámetro de token de seguridad (secToken) que no coincida con el calculado
✓ V-Financial.2.1.1.6	Realizar transacción en formato JSON desde el terminal de un comercio que no se encuentra habilitado
✓ V-Financial.2.1.1.7	Realizar transacción en formato JSON desde un terminal que no se encuentra habilitado
✓ V-Financial.2.1.1.8	Realizar transacción en formato JSON con todos los parámetros válidos
✓ V-Financial.2.1.1.9	Realizar transacción en formato XML con el parámetro de tipo de transacción inválido (type)
✓ V-Financial.2.1.1.10	Realizar transacción en formato XML con el parámetro de canal de pago inválido (channelCode)
✓ V-Financial.2.1.1.11	Realizar transacción en formato XML con el parámetro identificador de terminal inválido (terminalId)
✓ V-Financial.2.1.1.12	Realizar transacción en formato XML con un parámetro de token de seguridad (secToken) que ya se encuentre

created with Ready! API on Thu May 05 16:17:44 GMT-04:00 2016

Figura 71. Resumen de casos para la Interfaz de Pago en herramienta de pruebas SoapUI. (Elaboración propia).



TestSuite Results Report for Pruebas Interfaz de Pago

TestSuite Metrics

✓ V-Financial.2.1.1.13	Realizar transacción en formato XML con un parámetro de token de seguridad (secToken) que no coincida con el calculado
✓ V-Financial.2.1.1.14	Realizar transacción en formato XML desde el terminal de un comercio que no se encuentra habilitado
✓ V-Financial.2.1.1.15	Realizar transacción en formato XML desde un terminal que no se encuentra habilitado
✓ V-Financial.2.1.1.16	Realizar transacción en formato XML con todos los parámetros válidos
✓ V-Financial.2.1.2.1	Realizar transacción con formato JSON incorrecto
✓ V-Financial.2.1.2.2	Realizar transacción con formato XML incorrecto

Base Metrics

✓ Number of test cases	18
☰ Number of test steps	30
● Number of assertions	56
⊙ Number of load tests	0

TestSuite Properties

Name	Value
------	-------

created with Ready! API on Thu May 05 16:17:44 GMT-04:00 2016

Figura 72. Resumen de casos para la Interfaz de Pago en herramienta de pruebas SoapUI (cont.). (Elaboración propia).



TestSuite Results Report for Pruebas Interfaz de Pago

Result Metrics

Result Metrics

Start Time	Thu May 05 16:08:58 GMT-04:00 2016
End Time	Thu May 05 16:17:36 GMT-04:00 2016
Time Taken	518919 ms
TestCase Count	18
Failed TestCase Count	0
TestStep Count	30
Failed TestStep Count	0
Assertion Count	56
Failed Assertion Count	0

TestCase Results

TestCase	Status	Start Time	Time Taken	Reason
V-Financial.2.1.1.1	FINISHED	16:08:58	47 ms	
V-Financial.2.1.1.2	FINISHED	16:08:58	106 ms	
V-Financial.2.1.1.3	FINISHED	16:08:58	58 ms	
V-Financial.2.1.1.4	FINISHED	16:08:58	292 ms	
V-Financial.2.1.1.5	FINISHED	16:08:58	66 ms	
V-Financial.2.1.1.6	FINISHED	16:08:58	1671 ms	
V-Financial.2.1.1.7	FINISHED	16:09:00	1732 ms	
V-Financial.2.1.1.8	FINISHED	16:09:02	255225 ms	
V-Financial.2.1.1.9	FINISHED	16:13:17	98 ms	
V-Financial.2.1.1.10	FINISHED	16:13:17	19 ms	
V-Financial.2.1.1.11	FINISHED	16:13:17	52 ms	
V-Financial.2.1.1.12	FINISHED	16:13:17	182 ms	
V-Financial.2.1.1.13	FINISHED	16:13:18	74 ms	
V-Financial.2.1.1.14	FINISHED	16:13:18	1728 ms	
V-Financial.2.1.1.15	FINISHED	16:13:20	1609 ms	

created with Ready! API on Thu May 05 16:17:44 GMT-04:00 2016

Figura 73. Resultados de ejecución de casos de la Interfaz de Pago en herramienta de pruebas SoapUI. (Elaboración propia).



TestSuite Results Report for Pruebas Interfaz de Pago

TestCase Results

TestCase	Status	Start Time	Time Taken	Reason
V-Financial.2.1.1.16	FINISHED	16:13:21	255039 ms	
V-Financial.2.1.2.1	FINISHED	16:17:36	13 ms	
V-Financial.2.1.2.2	FINISHED	16:17:36	89 ms	

created with Ready! API on Thu May 05 16:17:45 GMT-04:00 2016

Figura 74. Resultados de ejecución de casos de la Interfaz de Pago en herramienta de pruebas SoapUI (cont.). (Elaboración propia).


4.3.4. Pruebas de integración

Realizadas mediante la aplicación de canal de pago C-Pago en conjunto con el Motor Transaccional para verificar la interacción entre ambos componentes.

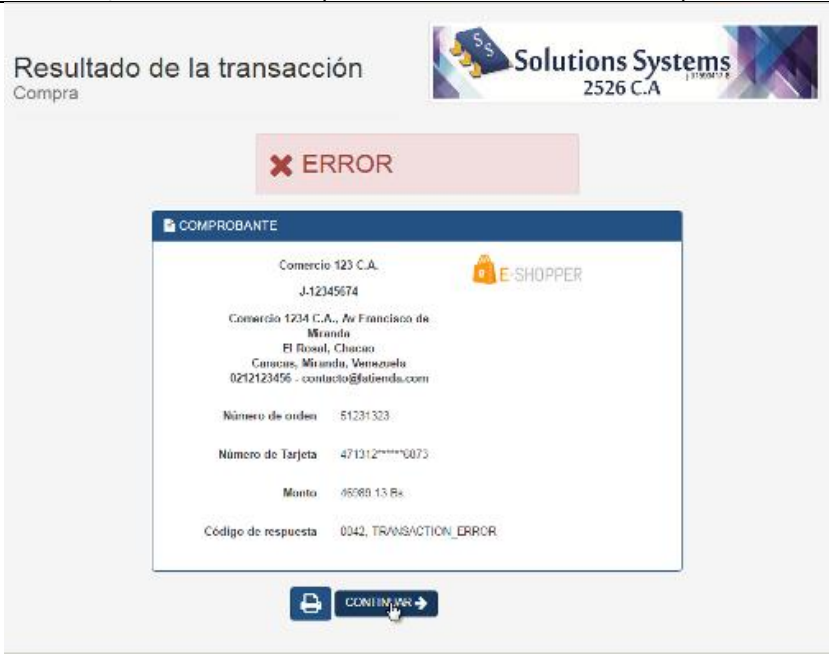
Tabla 36. Tabla resumen de las instancias de casos de prueba de aplicación C-Pago y Motor Transaccional. (Elaboración propia).

Caso de uso	Caso de prueba	Instancia de caso de prueba	Identificador de instancia de caso de prueba
C-Pago.1 - Realizar transacción de compra a través de interfaz gráfica	Invocación de servicio web de venta del Motor Transaccional	El Motor recibe respuesta de la entidad financiera	C-Pago.1.11.1
		El Motor no recibe respuesta de la entidad financiera	C-Pago.1.11.2

4.3.5. Resultados pruebas de integración

Caso de prueba	Invocación de servicio web de venta del Motor Transaccional: C-Pago.1.11.1 – El Motor recibe la respuesta de la entidad financiera		
Resultado esperado	El Motor Transaccional retorna el resultado de la transacción a la aplicación C-Pago y esta muestra el resultado.		
Resultado			

Caso de	Invocación de servicio web de venta del Motor Transaccional: C-Pago.1.11.2 – El Motor
----------------	---

prueba	recibe la respuesta de la entidad financiera
Resultado esperado	El Motor Transaccional reintenta la operación según la configuración y envía la transacción inversa, de no recibir respuesta notifica un error a la aplicación C-Pago.
Resultado	

RESULTADOS DEL TRABAJO ESPECIAL DE GRADO

- Las pruebas realizadas sobre el aplicativo de canal de pago C-Pago produjeron un resultado exitoso que permitió constatar la correctitud de la implementación de un botón de pago en línea.
- Las pruebas ejecutadas sobre el módulo de Interfaz de Pago del motor transaccional permitieron demostrar la funcionalidad de los servicios web configurables destinados a las solicitudes de transacciones de múltiples canales de pago.
- Se implementaron un conjunto de funciones para el manejo de los principales componentes de la plataforma: comercios, corporaciones, bancos, reglas de negocios, dispositivos, entre otros.
- El conjunto de pruebas realizadas sobre la API de Acceso a Datos proporcionada por el motor transaccional permitieron la validación de todas las funciones para gestionar los datos y configuraciones de los distintos actores que participan en el proceso de pago.
- La naturaleza personalizable de la mensajería implementada permitió la adición rápida de diversos tipos de transacciones. Esto se pudo constatar al realizar la integración de varios tipos de transacciones aceptados por una entidad procesadora de pagos.
- Las pruebas de integración de la aplicación de canal de pago con el motor transaccional permitieron constatar la adaptabilidad del esquema de configuración de transacciones sobre el cual se soporta la plataforma.
- La integración de un framework ORM permitió tener como resultado un motor transaccional que es capaz de adaptarse a múltiples sistemas manejadores de base de datos.

CONCLUSIONES

Se diseñó y desarrolló una plataforma transaccional que permite acoplar distintos canales y medios de pago asociados a diversas entidades bancarias. La solución implementada consiste en una plataforma configurable y adaptable al esquema transaccional de cada institución.

Mediante la implementación de servicios web con una estructura adaptable de acuerdo a la configuración de cada canal de pago, la plataforma es capaz de recibir solicitudes de transacciones con origen en diversos canales de pago y enviarlas a la entidad procesadora correspondiente, proporcionando de esta manera la capacidad de ser multibanco adquiriente.

Debido a las variaciones en el estándar ISO 8583 de cada entidad financiera, se desarrolló un módulo capaz de transformar las solicitudes de transacciones de servicios web en las tramas correspondientes según plantillas configurables para cada tipo de transacción. Así mismo es capaz de realizar el proceso inverso, recibir las tramas en los diversos formatos de cada procesador y transformarlos a una estructura de servicio web conocida por el canal de pago. De esto deriva la reducción en la complejidad del desarrollo de los canales de pago, ya que la responsabilidad de construir los mensajes en formato ISO 8583 esta delegada en el motor transaccional. De esta manera las aplicaciones que representan los canales de pago se abstraen conocer los formatos de las entidades procesadoras y, en su lugar, acceden a un formato y protocolo estándar. Esto trae como ventaja que nuevas implementaciones de canales de pago puedan acoplarse fácilmente, produciendo una plataforma escalable en el tiempo.

El desarrollo de una aplicación de botón de pago proporcionó la posibilidad de realizar las pruebas que permitieran verificar la integración del motor transaccional con un canal de pago, y de esta manera corroborar el funcionamiento y la adaptabilidad de los servicios web destinados a las transacciones.

Se implementaron una serie de servicios web que abarcan la gestión de diversos componentes de la plataforma. Esto da pie a la construcción de herramientas que otorgan a los comercios, bancos y procesadores capacidades de administración del comportamiento del sistema y reglas de negocio para las transacciones.

Se logró integrar un framework ORM al motor transaccional para conseguir el soporte a múltiples sistemas manejadores de base de datos. Esta capacidad está concebida para proporcionar una gran flexibilidad al momento del futuro despliegue de la plataforma.

Las características de escalabilidad y su naturaleza multiplataforma representan una ventaja competitiva en el mercado de medios y canales de pago actual, puesto que esta versatilidad permite que la plataforma sea adaptada a un abanico de situaciones sin la complejidad de realizar un desarrollo desde cero. Esto es especialmente favorable en el mercado venezolano, ya que existe una brecha en la demanda y la oferta de soluciones de pago.

TRABAJOS FUTUROS

Como desarrollo futuro se tiene previsto realizar un módulo de reportes y estadísticas que permita a los comercios, bancos y administradores de la plataforma consultar información relevante y resumida sobre los procesos y transacciones llevados a cabo por el motor transaccional a través de los distintos canales de pago conectados al mismo.

Es importante también el desarrollo de una aplicación que proporcione una interfaz gráfica para la configuración y personalización de la mensajería ISO 8583 de cada tipo de transacción, esto de acuerdo al formato proporcionado por la entidad procesadora correspondiente.

REFERENCIAS BIBLIOGRÁFICAS Y DIGITALES

- Authorize.Net. (s.f.). *Authorize.Net*. Obtenido de <http://www.authorize.net/>
- Capgemini's Strategic Analysis Group. (2012). *Global Trends in the Payment Card Industry 2012: Acquirers*. Capgemini.
- Capgemini's Strategic Analysis Group. (2012). *Global Trends in the Payment Card Industry 2012: Processors*. Capgemini.
- Chase Paymentech Solutions, LLC. (2012). *EMV: The New Way To Pay*. Chase Paymentech Solutions, LLC.
- Committee on Payment and Settlement Systems. (2003). *A glossary of terms used in payments and settlement systems*. Basel, Suiza: Bank for International Settlements.
- Dai, N., Mandel, L., & Ryman, A. (2007). *Eclipse Web Tools Platform: Developing Java Web Applications*. Addison-Wesley Professional.
- de Ruiter, J., & Poll, E. (2011). Formal Analysis of the EMV Protocol Suite. En T. conjunto, *Theory of Security and Applications* (pág. 225). Saarbrücken: Springer.
- EMVCo, L. (s.f.). *EMVCo*. Obtenido de <http://www.emvco.com/>
- European Central Bank. (2010). *The Payment System*. (T. Kokkola, Ed.) Frankfurt, Alemania.
- Gupta, S., & Roy, S. (2014). *Global Trends in the Payment Card Industry 2013: Acquirers*. Capgemini.
- Gupta, S., & Roy, S. (2014). *Global Trends in the Payment Card Industry 2013: Issuers*. Capgemini.
- Gupta, S., & Roy, S. (2014). *Global Trends in the Payment Card Industry 2013: Processors*. Capgemini.
- Karri, N. (2012). *Global Trends in the Payment Card Industry 2012: Issuers*. Capgemini.
- Microsoft Corporation. (2006). *La Arquitectura Orientada a Servicios (SOA) de Microsoft*.
- Nakajima, M. (2011). *Payment System Technologies and Functions: Innovations and Developments*. Estados Unidos: Business Science Reference.
- Navarro, A., Fernández, J., & Morales, J. (2013). *Revisión de metodologías ágiles para el desarrollo de software*.
- Optimal Payments Plc. (s.f.). *Navigating payment channels*.
- Paramá, J. (s.f.). *Modelo ER*.
- Payu Latam. (s.f.). *Payu Latam*. Obtenido de <http://www.payulatam.com/>
- PCI Security Standards Council, LLC. (2013). *Normas de Seguridad de Datos*.
- Schwaber, K., & Sutherland, J. (2013). *La Guía de Scrum*.
- Superintendencia de Bancos y Otras Instituciones Financieras. (23 de 12 de 2010). *Resolución 641-10: Normas que regulan el uso de los servicios de la banca electrónica*. Venezuela.
- Tutorials Point. (2014). *UML Tutorial*. Recuperado el Marzo de 2015, de <http://www.tutorialspoint.com/uml/>
- Wikipedia, c. d. (s.f.). *Wikipedia, La enciclopedia libre*. (L. e. Wikipedia, Editor) Recuperado el 4 de 8 de 2014, de ISO 8583: http://es.wikipedia.org/wiki/ISO_8583