# *Universidad de las Ciencias Informáticas*

# *CISED*

***Título:*** *Pasarela de pagos para la seguridad de transacciones bancarias en línea*

*Trabajo de diploma para optar por el título de Ingeniero en Ciencias Informáticas*

***Autores:*** *Damaris Solis Fonseca*

*Wilfredo Roque Pérez*

***Tutores:*** *Ing. María Lourdes Morilla Faurés*

*Lic. Miguel Ángel Hernández De la Rosa*

*La Habana, Junio 2012*

*"Año 54 de la Revolución"*

## DECLARACIÓN DE AUTORÍA

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmamos la presente a los ____ días del mes de _____ del año _____.

Damaris Solis Fonseca

_____

Firma del Autor

Wilfredo Roque Pérez

_____

Firma del Autor

Ing. María Lourdes Morilla Faurés

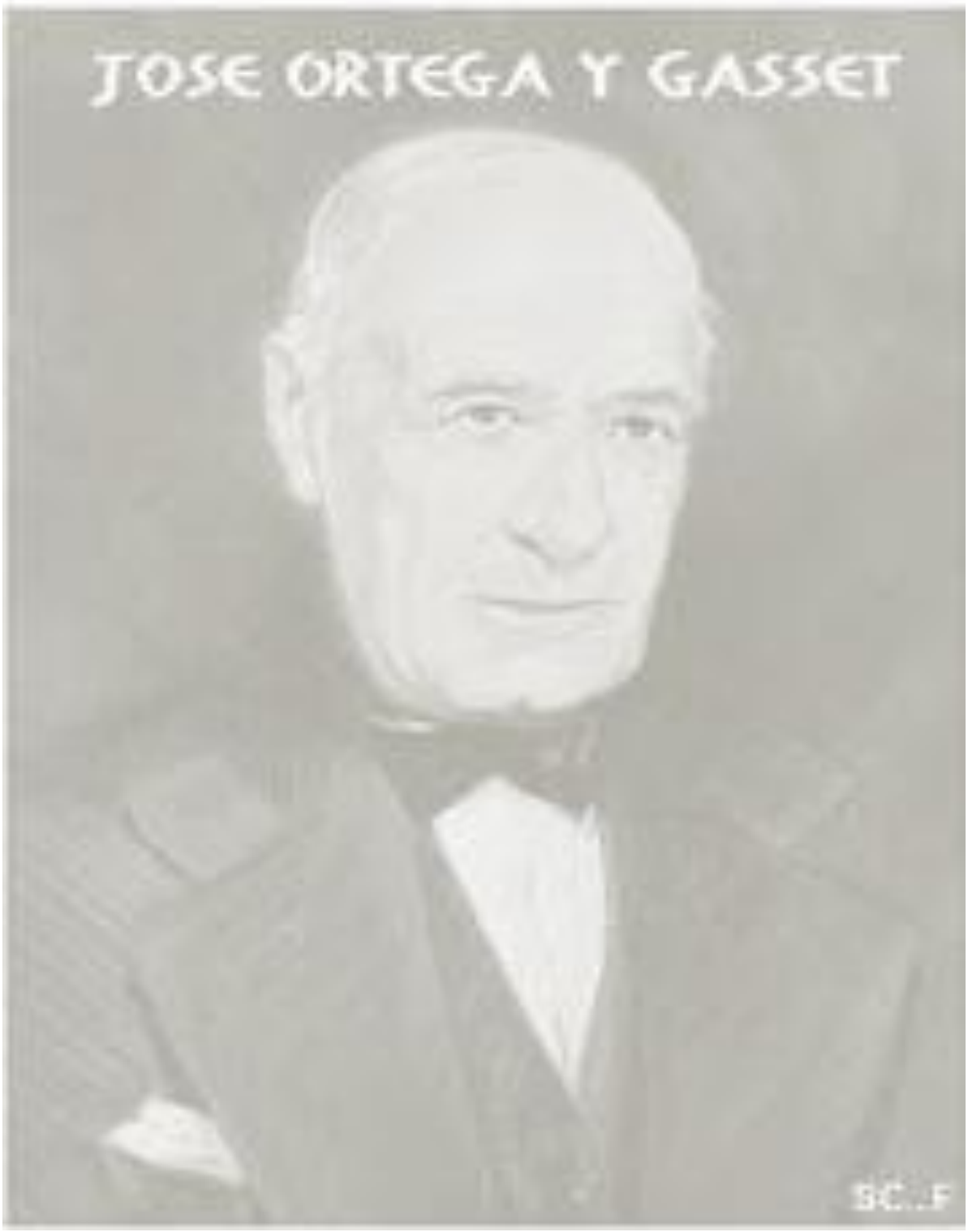
_____

Firma del Tutor

Lic. Miguel Ángel Hernández De la Rosa

_____

Firma del Tutor




*"Sólo es posible avanzar cuando se mira lejos. Sólo cabe progresar cuando se piensa en grande."*

**José Ortega y Gasset.**

**DATOS DE CONTACTO**

A continuación se detalla una breve descripción de los datos de los tutores:

La Ing. María Lourdes Morilla Faurés es graduada de Ingeniería en Ciencias Informáticas en la Universidad de las Ciencias Informáticas en el año 2008. Jefe de proyecto Pasarela de pagos. Se puede contactar por la dirección de correo electrónico: mlmorilla@uci.cu.

El Lic. Miguel Ángel Hernández de la Rosa se licenció en Educación Especialidad Informática, ISPETP 2006. Su categoría docente es de Profesor Asistente. Profesor del Departamento Ingeniería de Software y Práctica Profesional, Facultad 1, Universidad de las Ciencias Informáticas. Se puede contactar por medio del correo electrónico: miguelangel@uci.cu.

# *Dedicatoria*

*Dedicatoria de Damaris*

*Dedico mi trabajo de diploma a las personas más importantes de mi vida: mis padres Digna y Daniel. Yo sé que ellos están muy orgullosos de la persona que soy hoy. Representan el mejor ejemplo de amor, apoyo, respeto, consagración, voluntad y laboriosidad. A mis tías Inés, Doramis, Dignora y Lili que han sido para mí una segunda madre. A todos mis primos que los quiero como hermanos. A mis abuelitos que aunque dos de ellos no están conmigo físicamente yo sé que están presente en alma y pensamiento y sé que están orgullosos de mí. A mi querido Augusto que siempre confío en mí como una persona capaz de cruzar todas las barreras y me ha apoyado a lo largo de toda mi carrera. A mi hermano Reynier que no por mencionarlo de último es menos importante. Un abrazo y besos a todos los amo.*

*A mis padres que desde chiquito me educaron para llegar hasta aquí.*

*Dedicatoria de Wilfredo*

# *Agradecimientos*

*Agradecimientos de los autores*

*Es necesario agradecer a todas aquellas personas que nos han brindado su granito de arena en la realización de este trabajo. Les agradecemos a los tutores Lourdes y Miguel Ángel por saber orientarnos en todo el desarrollo de la investigación, por estar siempre dispuestos a ayudarnos en cualquier momento que los necesitábamos. A todos los profesionales del laboratorio 203 de seguridad digital por todos sus consejos y sugerencias.*

*Agradecimientos de Damaris*

*Agradecer a mis papas por apoyarme siempre y darme ánimo en todos los momentos de tensión que pasé. A mis familiares que siempre estuvieron al tanto de todo lo que yo estaba pasando. A mis amigas de la UCI Dary, Miladys, Adri y Yoci por apoyarme, confiar en mí y ayudarme en el desarrollo de la investigación. A todas las amistades de la UCI que me han ayudado. A mis queridas y eternas amigas de bayamo Yailín, Yisel, Dulcita, Elizabeth y la Gretel. A los profesores que me apoyaron y supieron darme buenos consejos cuando los necesite.*

*Quisiera agradecer en primer lugar a mi mamá, mi papá y mi hermana por siempre alentarme a seguir adelante, los quiero mucho. A mi novia Dianiseli, no creo poder encontrar palabras para agradecerle por su apoyo en todo momento. Quiero agradecer además a los berras, Onel, Dariel, Wilver, El mini y el Kinde por su ayuda durante el desarrollo de este trabajo. A los tutores por sus señalamientos, que perfeccionaron y nos condujeron hacia un trabajo de calidad.*

*Agradecimientos de Wilfredo*

**RESUMEN**

El trabajo de diploma desarrollado surge con motivo de apoyar el progreso del comercio electrónico en Cuba. Tal afirmación se refiere a la creación de un sistema de pago electrónico que permita la realización de pagos y transferencias entre tiendas electrónicas y bancos cubanos de manera segura. Estos sistemas son pasarelas de pagos que se encargan de cifrar la información confidencial que se requiere para ejecutar un pago o transferencia por las redes. En el país se utilizan pasarelas de pagos de otros países que comercializan con sus sitios de comercio electrónico pero cobran comisiones muy altas por sus servicios y no envían directamente efectivo a los bancos cubanos. Por lo tanto las pasarelas internacionales no manejan las monedas cubanas y sería provechoso contar con una propia que pueda ser utilizada por todas las tiendas electrónicas cubanas.

Con el uso de una pasarela de pagos se garantizan los pilares de la seguridad digital que son la autenticidad, la confidencialidad, el no repudio, la disponibilidad y la integridad de los datos que viajan por la red. La investigación consiste en una propuesta de pasarela de pagos que sea utilizada por sitios de comercio electrónico y bancos cubanos para realizar pagos y transferencias bancarias en tiempo real de forma segura. Para ello fue necesario trazar estrategias para el manejo correcto de los datos de los clientes de la pasarela utilizando estándares, mecanismos y protocolos de seguridad.

PALABRAS CLAVE: comercio electrónico, pago, pasarela de pagos, seguridad, transacciones bancarias *online*, transferencia.

# TABLA DE CONTENIDO

**INTRODUCCIÓN** ..... **1**
**CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.** ..... **7**
1.1 TRANSACCIONES BANCARIAS ONLINE ..... 7
*1.1.1 Seguridad en las transacciones bancarias online* ..... *8*
1.1.1.1 Seguridad de la información e intercambio electrónico de datos ..... 8
1.1.1.2 Estándares de seguridad ..... 9
1.1.1.3 Seguridad ante ataques ..... 10
1.1.1.4 Criptografía ..... 10
*1.1.2 Normas y procedimientos técnicos para garantizar la seguridad* ..... *12*
*1.1.3 Estándar de seguridad de datos en la industria de tarjetas de pago (PCI DSS)* ..... *14*
*1.1.4 Normas de seguridad de datos para las aplicaciones de pago (PA-DSS)* ..... *15*
*1.1.5 Requisitos de comunicación entre las entidades* ..... *17*
1.1.5.1 Norma ISO 8583 ..... 17
1.2 PASARELAS DE PAGOS ..... 18
*1.2.1 Pasarelas de pagos internacionales más utilizadas* ..... *18*
*1.2.2 Proyecto cubano de pasarela de pagos* ..... *21*
*1.2.3 Características de una pasarela de pagos segura* ..... *21*
1.3 TECNOLOGÍAS DE DESARROLLO DE SOFTWARE ..... 22
*1.3.1 Metodologías de desarrollo* ..... *22*
*1.3.2 Lenguajes de programación* ..... *23*
*1.3.3 Herramientas* ..... *26*
**CAPÍTULO 2: CARACTERÍSTICAS DE LA PASARELA DE PAGOS** ..... **29**
2.1 FASE VISIÓN Y ALCANCE ..... 29
*2.1.1 Modelo de dominio* ..... *29*
*2.1.2 Concepción de los módulos del sistema* ..... *30*
*2.1.3 Definición de personas* ..... *31*
2.2 FASE PLANEACIÓN ..... 32
*2.2.1 Especificación de escenarios de la pasarela de pagos* ..... *32*
*2.2.2 Requisitos de calidad del servicio* ..... *34*
2.3 PLAN DE ITERACIONES ..... 36
2.4 DESCRIPCIÓN DE ESCENARIOS ..... 36
*2.4.1 Especificación de tareas por escenarios.* ..... *40*
**CAPÍTULO 3: DESARROLLO Y ESTABILIZACIÓN.** ..... **43**
3.1 FASE DE DESARROLLO ..... 43
*3.1.1 Arquitectura del sistema* ..... *43*
*3.1.2 Patrones de diseño* ..... *45*
*3.1.3 Diagrama de aplicación* ..... *46*
*3.1.4 Diagrama lógico de centro de datos* ..... *46*
*3.1.5 Modelo de datos* ..... *47*
*3.1.6 Descripción de las clases controladoras* ..... *48*
*3.1.7 Estándares de codificación utilizados* ..... *51*
*3.1.8 Tratamiento de excepciones* ..... *51*
3.2 FASE DE ESTABILIZACIÓN ..... 52
*3.2.1 Pruebas del software* ..... *52*
**CONCLUSIONES** ..... **61**
**RECOMENDACIONES** ..... **62**
**REFERENCIAS BIBLIOGRÁFICAS** ..... **63**
**BIBLIOGRAFÍA** ..... **65**
**ANEXOS** ..... **68**
**GLOSARIO DE TÉRMINOS** ..... **115**

# Índice de figuras

Figura 1. Modelo de dominio. Fuente: elaboración propia. ..... 30
Figura 2. Módulos de la pasarela de pagos. Fuente: elaboración propia. ..... 31
Figura 3. Diagrama de la vista lógica de la arquitectura. Fuente: elaboración propia. ..... 44
Figura 4. Diagrama de aplicación. Fuente: elaboración propia. ..... 46
Figura 5. Diagrama lógico de centro de datos. Fuente: elaboración propia. ..... 47
Figura 6. Modelo de datos. Fuente: elaboración propia. ..... 47

# Índice de tablas

Tabla 1. Descripción de los módulos. Fuente: elaboración propia. ..... 31
Tabla 2. Descripción de personas. Fuente: elaboración propia. ..... 32
Tabla 3. Plan de iteraciones. Fuente: elaboración propia. ..... 36
Tabla 4. Descripción del escenario Registrar cliente. ..... 37
Tabla 5. Descripción del escenario Autenticar cliente. ..... 38
Tabla 6. Descripción del escenario Realizar pago. ..... 39
Tabla 7. Descripción del escenario Realizar transferencia. ..... 40
Tabla 8. Descripción de la tarea Listar cuentas bancarias. ..... 41
Tabla 9.Descripción de la tarea Mostrar reporte de historial de transacciones. ..... 41
Tabla 10. Clase persistente:"usuario". Fuente: elaboración propia. ..... 48
Tabla 11. Clase persistente: "transaccion". Fuente: elaboración propia. ..... 48
Tabla 12.Clase controladora: "TransaccionControl". Fuente: elaboración propia. ..... 49
Tabla 13.Clase controladora: "UsuarioControl". Fuente: elaboración propia. ..... 51
Tabla 14. Descripción de la prueba unitaria al método "TransferirDinero". ..... 54
Tabla 15. Resultado de las pruebas unitarias. Fuente: elaboración propia. ..... 55
Tabla 16. Caso de prueba Registrar cliente. Fuente: elaboración propia. ..... 56
Tabla 17. Descripción de variables del caso de prueba Registrar cliente. ..... 57
Tabla 18. Caso de prueba Configurar cuenta. Fuente: elaboración propia. ..... 58
Tabla 19. Descripción de variables del caso de prueba Configurar cuenta. ..... 58
Tabla 20.Caso de prueba Autenticar cliente. Fuente: elaboración propia. ..... 58
Tabla 21. Descripción de variables del caso de prueba Autenticar cliente. ..... 59
Tabla 22. Caso de prueba Realizar transferencia. Fuente: elaboración propia. ..... 59
Tabla 23. Descripción de variables del caso de prueba Realizar transferencia. ..... 59
Tabla 24. Resultados de las pruebas. Fuente: elaboración propia. ..... 60
Tabla 25. Descripción del escenario Configurar cuenta de administración. ..... 68
Tabla 26. Descripción de la tarea Mostrar usuario administración. ..... 69
Tabla 27. Descripción de la tarea Crear usuario de administración. ..... 70
Tabla 28. Descripción de la tarea Modificar usuario de administración. ..... 71
Tabla 29. Descripción de la tarea Eliminar usuario de administración. ..... 72
Tabla 30. Descripción de la tarea Cambiar contraseña de administrador. ..... 73
Tabla 31. Descripción del escenario Configurar cliente. ..... 74
Tabla 32. Descripción del escenario Mostrar cliente. ..... 75
Tabla 33. Descripción de la tarea Eliminar cliente. ..... 76
Tabla 34. Descripción de la tarea Buscar cliente. ..... 76
Tabla 35. Descripción del escenario Configurar pago. ..... 77
Tabla 36. Descripción de la tarea Mostrar operaciones de pago. ..... 78
Tabla 37. Descripción de la tarea Gestionar banco. ..... 79
Tabla 38. Descripción de la tarea Mostrar banco. ..... 79
Tabla 39. Descripción de la tarea Crear banco. ..... 80
Tabla 40. Descripción de la tarea Modificar banco. ..... 81
Tabla 41. Descripción de la tarea Eliminar banco. ..... 82
Tabla 42. Descripción del escenario Autenticar administrador. ..... 82
Tabla 43. Descripción del escenario Configurar cuenta. ..... 83
Tabla 44. Descripción de la tarea Modificar datos del cliente. ..... 84

Tabla 45. Descripción de la tarea Desactivar cliente. ..... 85
Tabla 46. Descripción del escenario Cambiar contraseña. ..... 86
Tabla 47. Descripción del escenario Configurar cuenta bancaria. ..... 86
Tabla 48. Descripción del escenario Notificar al cliente de la creación de la cuenta. ..... 87
Tabla 49. Descripción de la tarea Eliminar cuenta bancaria. ..... 88
Tabla 50. Descripción de la tarea Consultar saldo bancario. ..... 88
Tabla 51. Descripción de la clase persistente: “usuario”. ..... 89
Tabla 52. Descripción de la clase persistente: “transaccion”. ..... 89
Tabla 53. Descripción de la clase persistente: “transaccion_x_cuenta”. ..... 89
Tabla 54. Descripción de la clase persistente: “cuenta”. ..... 90
Tabla 55. Descripción de la clase persistente: “entidad_bancaria”. ..... 90
Tabla 56. Descripción de la clase persistente:”solicitudpago”. ..... 90
Tabla 57. Descripción de la clase controladora:”CuentaControl”. ..... 91
Tabla 58. Descripción de la clase persistente: “EntidadBancariaControl”. ..... 91
Tabla 59. Descripción de la clase persistente: “SolicitudPagoControl”. ..... 92
Tabla 60. Caso de prueba Configurar cuenta de administración. ..... 95
Tabla 61. Descripción de variables del caso de prueba Configurar cuenta de administración. . 95
Tabla 62. Caso de prueba Buscar cliente del módulo Administración. ..... 97
Tabla 63. Descripción de variables del caso de prueba Buscar cliente. ..... 97
Tabla 64. Caso de prueba Configurar cliente del módulo Administración. ..... 98
Tabla 65. Descripción de variables del caso de prueba Configurar cliente. ..... 99
Tabla 66. Caso de prueba Gestionar banco. Fuente: elaboración propia. ..... 100
Tabla 67. Descripción de variables del caso de prueba Gestionar banco. ..... 101
Tabla 68. Caso de prueba Autenticar administrador. ..... 101
Tabla 69. Descripción de variables del caso de prueba Autenticar administrador. ..... 101
Tabla 70. Caso de prueba Modificar cliente. Fuente: elaboración propia. ..... 103
Tabla 71. Descripción de variables del caso de prueba Modificar cliente. ..... 104
Tabla 72. Descripción de la prueba unitaria al método “RealizarPago”. ..... 105
Tabla 73. Descripción de la prueba unitaria al método “AutenticarCliente”. ..... 107
Tabla 74. Descripción de la prueba unitaria al método “AdicionarCliente”. ..... 108
Tabla 75. Descripción de la prueba unitaria al método “ConfigurarPago”. ..... 110
Tabla 76. Descripción de la prueba unitaria al método “ConsultarPago”. ..... 111
Tabla 77. Descripción de la prueba unitaria al método “ActualizarUsuario”. ..... 112
Tabla 78. Descripción de la prueba unitaria al método “ListarClientes”. ..... 113
Tabla 79. Descripción de la prueba unitaria al método “EliminarCliente”. ..... 114

# Introducción

El decursar de la humanidad ha traído consigo grandes avances tecnológicos. Los cuales han alcanzado cada una de las ramas de la sociedad. Producto a estos adelantos surgió una nueva ciencia, las ciencias informáticas. El desarrollo de la informática ha dado paso al surgimiento de las Tecnologías de la Información y las Comunicaciones (TICs), el constante uso de estas tecnologías han hecho imprescindible las prácticas diarias del Comercio Electrónico. El mismo consiste en la compra y venta de bienes y servicios utilizando medios electrónicos.

El comercio electrónico actualmente ha cobrado un impresionante auge en el mundo contemporáneo. Millones de personas a través de sus computadoras hacen uso de los sitios de comercio electrónico para realizar compras de bienes o productos, pagar servicios del hogar (por ejemplo: agua, corriente eléctrica, gas y renta), reservar viajes a lugares turísticos, venta o alquiler de casas (u otro inmueble), cobrar seguro de vida o de carro, entre otros servicios que les sean de interés.

Con el uso del comercio electrónico se pueden realizar transacciones bancarias *online*[1].Estas transacciones son operaciones bancarias que personas y empresas manejan desde sitios de comercio electrónico. Una vez que el cliente (persona que practica comercio electrónico) es usuario de un sitio web comercial (o tienda virtual como también se le llama), usando tarjetas de crédito o débito puede efectuar pagos de servicios solicitados o transferencias hacia otras cuentas bancarias. Estos procesos involucran a los bancos que gestionan las cuentas de los usuarios y de los vendedores (empresa que oferta los productos en las tiendas virtuales), y están asociadas a sus tarjetas de crédito o débito.

Los procesos de pago y transferencia bancaria *online* implican una alta seguridad en la transmisión de la información electrónica que se intercambia entre las entidades comerciales y los bancos.

A partir del desarrollo del comercio electrónico y de las transacciones bancarias *online,* se han evidenciado nuevas tendencias para mejorar la vida cotidiana y laboral. Igualmente se han identificado diferentes problemas de seguridad que han provocado la desconfianza en los usuarios al utilizar los sitios web comerciales como vía para gestionar sus cuentas bancarias.

Los autores Jordi Buch i Tarrats y Francisco Jordán (2000) refieren que internet tiene problemas de autenticidad, integridad, confidencialidad y repudio afectando a los requerimientos de las transacciones electrónicas u operaciones de banca virtual de la siguiente forma:

- *Robo de información*: los mal intencionados a través del correo electrónico o de la misma página comercial obtienen la información del usuario (números de tarjetas de crédito, información de cuenta bancaria);

[1] Significado en inglés de: *en línea.*

- *Suplantación de identidad*: los atacantes realizan acciones en nombre del verdadero dueño de la cuenta de usuario perjudicándolo, pues se apoderan de sus finanzas;
- Ataques de *"sniffers"* que adquieren la información de las operaciones y respaldan al robo de información y a la suplantación de identidad;
- *Modificación de la información*: va en contra de la seguridad de las transacciones ya que contribuyen a alterar la orden de compra, el pago y la cantidad de productos escogidos;
- Ataques por *Denegación de Servicio*: son frecuentes, muy fáciles de ejecutar y resulta difícil el reconocimiento del atacante, imposibilita la realización de operaciones ya que afecta al sistema bancario;
- El *repudio*: consiste en que una de las partes implicadas en el proceso, niegue haber realizado una transacción determinada y perjudique los sistemas de pago causándole costos adicionales de facturación.[1]

Para solucionar los problemas de seguridad existentes, distintas empresas se han dado la tarea de crear sistemas de pago electrónicos que garanticen la seguridad en transacciones bancarias. Estos sistemas son llamados pasarelas de pagos que constituyen mecanismos que impiden los fraudes y garantizan autenticidad, confidencialidad, integridad y el no repudio[2] en la red.

La pasarela de pagos es la herramienta que facilita el pago seguro entre los compradores y vendedores. Cuando se tiene una tienda *online*, la pasarela se utiliza para poder realizar los cobros. Por lo general se realiza a través de entidades bancarias, que facilitan el pago mediante tarjetas de crédito. Esta codifica la información sensible que viaja entre estas entidades y notifica todas las acciones efectuadas. Esta información pueden ser los datos personales del usuario como es su nombre, apellidos, dirección particular y el número de su tarjeta.[2]

Actualmente en Cuba existe muy poco desarrollo del comercio electrónico. Varias empresas trazan metas para crear una infraestructura tecnológica sólida y fomentar su desarrollo. Hace algunos años en el sector empresarial cubano se comenzaron a trazar estrategias con respecto a sus prácticas. A pesar de los problemas que han golpeado a la economía cubana, se facilitó la oferta de productos cubanos en sitios web comerciales propios y extranjeros.

En internet existen más de 350 sitios cubanos, con más de 16 000 páginas, en su gran mayoría en idioma español. De estos sitios, 175 poseen espejos fuera del país para asegurar una mayor velocidad de acceso a los visitantes desde el exterior. Internet como gran fuente de información y de posibilidades para el comercio, se considera un importante medio para que el mundo conozca a Cuba.[3]

Para lograr un correcto desarrollo de la actividad comercial en internet y específicamente sobre las pasarelas de pagos, en el territorio cubano se necesita además una adecuada formación y

[2] Evita que el usuario niegue que cierta transacción tuvo lugar.

capacitación acerca del tema, por lo que las universidades cubanas constituyen las instituciones vanguardias para una correcta formación en el panorama profesional.

La Universidad de las Ciencias Informáticas (UCI), constituye el más novedoso esfuerzo de Cuba en aras de llevar adelante la industria del software y potenciar la informatización de la sociedad. Para lograr esto, la UCI en su estructura, cuenta con centros de producción en los cuales se desarrollan proyectos que potencian tanto la vida social como laboral de la sociedad cubana.

En la Facultad 1 de la UCI se encuentra el Centro de Identificación y Seguridad Digital (CISED). En este se desarrollan diversos proyectos relacionados con la seguridad digital, especializados en el desarrollo de soluciones de seguridad para sistemas informáticos así como servicios de consultorías y certificaciones de seguridad digital. Uno de estos proyectos es: "Pasarela de pagos" el cual tiene como objetivo desarrollar una pasarela de pagos que garantice la seguridad en transacciones bancarias *online*.

A pesar de los intentos de expandir la actividad web comercial, en Cuba como país subdesarrollado, no existen los recursos financieros para explotar su desarrollo. A esto se le suman la brecha digital que existe a escala mundial, y la implantación del bloqueo económico[3]. Este constituye un obstáculo ya que no permiten las transacciones entre Cuba y otros bancos en el mundo.

Existen tres elementos básicos que garantizan el comercio electrónico: una plataforma competente para vender, las pasarelas de pagos y la distribución de mercancías. Lo primero se construyó por esfuerzo propio. Lo segundo es el cobro por internet a través de tarjetas de crédito de bancos internacionales, a los que Cuba no tiene acceso directo.

El país cuenta con varios sitios de comercio electrónico que utilizan pasarelas extranjeras, lo que implica:

- Que las tiendas virtuales operen hacia el exterior del país o para el sector turístico en la isla, ya que son los que tienen más acceso a internet.
- Por cada transacción realizada se cobra un porciento de lo pagado o una tasa fija que provoca al vendedor pérdidas monetarias por el uso de este servicio.
- Descontento por parte del cliente debido al pago por inscripción en la pasarela por el gasto de un servicio que en ocasiones no es seguro. Al ofrecerle un servicio a un cliente se debe asegurar su plena satisfacción para cerciorarse que el mismo vuelva a solicitarlo.

[3] Impuesto por parte del gobierno de EE.UU hacia Cuba.

- Se imposibilita realizar las operaciones con la moneda de intercambio cubana ya que estas pasarelas no se comunican con las instituciones bancarias cubanas debido al bloqueo económico por parte de los EE.UU hacia Cuba.
- En los bancos cubanos se dificulta el uso de sistemas de pago electrónico para ejecutar transacciones desde sitios comerciales. Esto implica que las personas realicen sus pagos y transferencias de la forma tradicional, asistiendo a las instituciones de forma personal, obviando las múltiples ventajas que trae consigo la banca y el comercio electrónico. Con esto se ven limitadas muchas operaciones de este tipo por afectar el tiempo e incluso la distancia a la que se encuentren los clientes del banco.

Por todo lo antes expuesto el **problema de la investigación** queda formulado de la siguiente manera: ¿Cómo facilitar los procesos de pagos y transferencias bancarias *online* entre bancos cubanos y aplicaciones de comercio electrónico, permitiendo el uso de la moneda de intercambio cubana de manera segura y sin costos?

Se enmarca como **objeto de estudio:** El proceso de las transacciones bancarias que se gestionan en pasarelas de pagos.

**El objetivo general** de la investigación consiste en: Desarrollar una pasarela de pagos que permita realizar los procesos de pagos y transferencias bancarias online entre bancos cubanos y aplicaciones de comercio electrónico, permitiendo el uso de la moneda de intercambio cubana de manera segura y sin implicar costos.

El **campo de acción** consiste en: Los procesos de pago y transferencia como transacciones bancarias gestionadas por pasarelas de pagos.

Las **tareas de la investigación** a realizar por la autora Damaris Solis Fonseca se relacionan a continuación:

- Análisis de los mecanismos de cifrado de datos.
- Determinación de las medidas de seguridad según las normas del pago con tarjeta en aplicaciones de pagos.
- Fundamentación de la metodología, herramientas y lenguajes a utilizar en el desarrollo de la pasarela de pagos.
- Elaboración de los artefactos necesarios que se corresponden con las fases: Visión, Planeación y Estabilización de la metodología *Microsoft Solutions Framework* (MSF) para el desarrollo de software ágil.

Las **tareas de la investigación** a realizar por el autor Wilfredo Roque Pérez son las siguientes:

- Análisis de los protocolos referentes a la seguridad de las transacciones bancarias *online*.
- Caracterización de pasarelas de pagos existentes en el mundo.

- Elaboración de los artefactos necesarios que se corresponden con la fase: Desarrollo de la metodología *Microsoft Solutions Framework* (MSF) para el desarrollo de software ágil.

Para realizar las tareas investigativas se emplearon métodos teóricos y empíricos de la investigación científica.

Los **métodos teóricos** utilizados para cumplir con las tareas a desarrollar son:

**Analítico-Sintético**: se consulta la bibliografía necesaria para dar cumplimiento a las tareas de la investigación y se realiza un resumen de los principales aspectos de cada una de ellas. Se realiza un estudio de los mecanismos y protocolos internacionales existentes referidos a la seguridad en las transacciones bancarias *online* así como la programación segura de aplicaciones web.

**Inductivo - Deductivo**: en el estudio del arte realizado este método posibilitó a través de la obtención de conocimientos generales, deducir conocimientos particulares para la solución de la investigación, aplicando conocimientos que estuvieran más acorde con el objetivo de la investigación.

**Modelación**: mediante la elaboración de diagramas permite hacer una reproducción simplificada del dominio del problema, de los datos que serán almacenados en el sistema y de las clases del sistema.

Los **métodos empíricos** utilizados son:

**Observación**: se utiliza para la detección de la situación problemática debido a los problemas que existen en el país al no contar con el funcionamiento de una pasarela de pagos propia capaz de efectuar transacciones bancarias *online* de forma eficiente y segura.

**Entrevista**: las mismas se realizan a especialistas que trabajan en los bancos cubanos con el fin de conocer cómo funciona la comunicación con las pasarelas de pagos y conciliar las formas de conexión y estándares de seguridad establecidos por dichas entidades.

Con el desarrollo de esta investigación se creó una pasarela de pagos que se podrá utilizar en el país para realizar pagos y transferencias bancarias a través de sitios web comerciales. De esta manera se brindará el servicio *online* de estos procesos en los bancos de Cuba al estar relacionados también con la pasarela de pagos. El proceso de manejo de la información de estas transacciones cuenta con requisitos de seguridad que impiden el acceso no deseado por atacantes al sistema. Con el uso de las tecnologías mencionadas se apoyará a los sistemas tradicionales de cobro, facilitándose a los usuarios de internet una mayor rapidez y accesibilidad a los mismos y de forma segura. Con la puesta en práctica de todo lo plateado anteriormente se apoyará a fomentar el comercio electrónico en el país.

El presente trabajo está estructurado en 4 capítulos:

El **Capítulo 1**, **Fundamentación teórica**, contiene un estudio del desarrollo alcanzado en el ámbito de las pasarelas de pagos a nivel nacional e internacional. Se presenta la

fundamentación teórica en la que se exponen los conceptos fundamentales que sustentan la investigación.

En el **Capítulo 2, Características de la pasarela de pagos**, se elabora una propuesta de la pasarela de pagos para la cual se estudiaron los procesos relacionados con el objeto de estudio y el campo de acción. Contiene además los requisitos funcionales y de calidad del servicio, estableciendo de esta forma las funcionalidades básicas del sistema y la vía para alcanzar una solución óptima para el problema planteado.

En el **Capítulo 3**, **Desarrollo y estabilización** se describe la arquitectura a utilizar, así como los patrones de diseño utilizados en el desarrollo de la pasarela. Cuenta con las clases persistentes que serán almacenadas en la base de datos. Además de relacionar los componentes técnicos que conforman la pasarela de pagos. Se realizan pruebas de caja blanca y caja negra que permiten verificar la calidad de la solución propuesta.

# Capítulo 1: Fundamentación teórica.

El presente capítulo aborda toda la información teórica de la investigación. Se abunda sobre las transacciones bancarias *online*, en qué consisten, definiciones de los términos, así como los aspectos que hay que tener en cuenta para lograr una mayor seguridad cuando se va a efectuar alguna transacción de este tipo. Recoge la información sobre las pasarelas de pagos como sistemas de pago electrónico seguro en transacciones bancarias *online*, se ejemplifican algunas de las más conocidas en el mundo y se presenta un proyecto de pasarela cubano. Por último se hace un análisis de las metodologías, lenguajes y herramientas a utilizar para la creación de la pasarela de pagos.

## 1.1 Transacciones Bancarias Online

El significado de transacción definido por el diccionario de la Real Academia Española (RAE) se refiere a la acción y efecto de transigir (acordar voluntariamente con otra parte algún punto litigioso para compartir la diferencia de la disputa, consentir a fin de terminar con una diferencia).

Para la economía, las finanzas o el comercio, una transacción es una operación de compra y venta. Es el traspaso de efectivo desde una cuenta bancaria hacia otra. Se da por la necesidad que tienen una o ambas partes y puede o no generar ganancias. Toma muchas formas, estilos y métodos, pero siempre implicará el intercambio de bienes o servicios a cambio del capital que le corresponda. Constituyen transacciones también las operaciones que realiza el cliente para invertir, reorganizar o conocer su capital.

Distintos autores de la materia se han referido a las transacciones bancarias en variados conceptos. Muratti (1972) afirma que la operación bancaria es una “interposición de crédito”. Staud (1976) establece que son “las que satisfacen necesidades de tráfico para la obtención y enajenación de dinero y de títulos valores”. Por su parte Joaquín Rodríguez (1976) comenta que “la operación bancaria es una operación de crédito realizada por una empresa bancaria en masa y con carácter profesional”. De todos los aspectos tratados anteriormente se llega a la conclusión de que las transacciones bancarias son las distintas operaciones que por medio del crédito se concretan en un banco.

El comercio electrónico siendo la compra por internet, es una de las grandes ventajas que ofrece la misma a los usuarios que la usan. Esta vía de comercio resulta decisiva a pesar de que algunas personas prefieran las compras tradicionales. Esto se incrementa todavía más en el caso de personas con dificultades para la movilidad y el desplazamiento, o simplemente para los muchos casos en los que los horarios de trabajo dificultan acceder a los establecimientos en sus horarios normales.[4]

Una transacción electrónica no es más que un contrato celebrado mediante medios electrónicos, a través de la red. La mayoría de estas son enajenaciones, definidas como

cualquier acto de disposición por el que se transmita la propiedad a título oneroso, entre las que se mencionan la compra-venta y el suministro.[5]

La Banca Electrónica (*E-Banking*) surge con el desarrollo del comercio electrónico, es el reflejo del banco tradicional pero desplegado a través de internet. Su uso permite un rápido y cómodo acceso a servicios bancarios como: revisar su saldo bancario, transferir dinero entre cuentas y pagar sus cuentas. Este intercambio de información financiera hacia los bancos electrónicos constituye lo que son las transacciones bancarias *online*. El banco virtual tiene ventajas sobre el tradicional pues permite: un amplio marco geográfico, rapidez y simplicidad en las transacciones, mayor control sobre las cuentas, mejor servicio al cliente, no requerimiento de presencia física y los servicios están disponibles todo el tiempo que se requiera sin importar hora o lugar donde se encuentre el cliente.[6]

### 1.1.1 Seguridad en las transacciones bancarias *online*

Cuando se realizan transacciones bancarias *online* se debe tener un estricto control de los mecanismos de seguridad que protegen el sistema de ataques a la autenticidad, confidencialidad, integridad, disponibilidad y el no repudio de la información (son los llamados pilares de la seguridad informática).[7]

La comunicación establecida entre las entidades y/o usuarios participantes en estas acciones debe realizarse sobre estrategias bien conocidas por ambas partes.

Utilizar conexiones a través de un servidor seguro haciendo uso de protocolos seguros permite que la información viaje cifrada entre el ordenador cliente y el servidor. Esta estrategia evita que los datos sean interceptados por terceras personas, comprometiendo datos sensibles. Al establecer una infraestructura de clave pública con sus procedimientos y mecanismos se fomenta la seguridad entre las entidades participantes cuando se realizan transacciones bancarias *online*. En el presente sub-epígrafe se caracterizan los aspectos importantes que se deben tener en cuenta cuando se efectúan transacciones de este tipo en la red.

#### 1.1.1.1 Seguridad de la información e intercambio electrónico de datos

La seguridad de la información se consigue implantando un conjunto adecuado de controles, que pueden ser políticas, prácticas, procedimientos, estructuras organizativas y funciones de software y hardware. Estos controles necesitan ser establecidos, implementados, monitoreados, revisados y mejorados donde sea necesario, para asegurar que se cumplan los objetivos específicos de seguridad y negocios de una organización.[8]

El intercambio electrónico de datos consiste en transmitir electrónicamente documentos comerciales y administrativo-contables entre aplicaciones informáticas en un formato normalizado, de forma que puedan procesarse dichos documentos entre las empresas sin intervención manual[8]. De esta forma se garantiza la automatización y estandarización de este importante proceso. El contar con un formato normalizado añade cierto nivel de seguridad, pues

un usuario común no podría generar un mensaje falsificado a menos que tenga conocimiento pleno de estas reglas.

#### 1.1.1.2 Estándares de seguridad

Los estándares de seguridad constituyen los protocolos que se utilizan en las transacciones del banco virtual.

El protocolo *Secure Socket Layer* (SSL) facilita la autenticación y privacidad de la información en internet mediante el uso de la criptografía. Sólo el servidor es autenticado (es decir, se garantiza su identidad) mientras que el cliente se mantiene sin autenticar; la autenticación mutua requiere un despliegue de infraestructura de claves públicas para los clientes. Los protocolos permiten a las aplicaciones cliente-servidor comunicarse de una forma diseñada para prevenir escuchas (*eavesdropping*), la falsificación de la identidad del remitente (*phishing*) y mantener la integridad del mensaje.

El SSL se ejecuta en una capa entre los protocolos de aplicación como el *Hypertext Transfer Protocol* (HTTP), el *Simple Mail Transfer Protocol* (SMTP), el *Network News Transfer Protocol* (NNTP) y sobre el protocolo de transporte *Transmission Control Protocol* (TCP), que forma parte de la familia de protocolos TCP/IP, este último es el protocolo de internet llamado *Internet Protocol*. Aunque pueda proporcionar seguridad a cualquier protocolo que use conexiones de confianza (tal como TCP), se usa en la mayoría de los casos junto a HTTP para formar el protocolo seguro de transferencia de hipertexto HTTPS, del significado en inglés *Hyper Text Transfer Protocol Secure*.[7]

Proporciona sus servicios de seguridad utilizando la criptografía de llave pública y privada. Para el intercambio de los datos entre el servidor y el cliente, utiliza algoritmos de cifrado simétrico. Para la autenticación, usa el algoritmo de cifrado de clave pública (RSA). La clave de sesión es la que se utiliza para cifrar los datos que vienen de él y van al servidor una vez establecido el canal seguro (usa criptografía simétrica).[7]

El protocolo HTTPS es la versión segura de HTTP. Fue desarrollado por *Enterprise Integration Technologies* (EIT). Permite el cifrado y autenticación digital igual que SSL. La diferencia está, en que HTTPS es un protocolo de nivel de aplicación, es decir, que extiende el protocolo HTTP por debajo. HTTPS es usado para asegurar páginas *World Wide Web* para aplicaciones de comercio electrónico, utilizando certificados de clave pública para verificar la identidad de los participantes.

Las características de HTTPS vienen dada por:

- Mejoras HTTPS (Firma, Encriptación y Autenticación).
- Algoritmos de Firma Digital (RSA, NIST-DSS).
- Algoritmos de *HASH* (MD2, MD5, SHS).
- Algoritmos de cifrado en bloque de clave simétrica.

- Algoritmos de cifrado de clave simétrica.[7]

#### 1.1.1.3 Seguridad ante ataques

El proyecto *Open Web Application Security Project* (OWASP) ofrece una guía para el desarrollo de aplicaciones web seguras, con el fin de lograr el progreso de la web libre de cibercriminales.

Para evitar los ataques al sistema es importante tener en cuenta las posibles brechas de seguridad que pudieran ser aprovechadas por estos para perpetrar sus ataques. Las herramientas existentes hoy en día permiten alcanzar un nivel de seguridad apropiado, aunque debe mantenerse una vigilancia estricta ante la posible ruptura de un mecanismo que derive en un ataque desastroso.

Las amenazas más comunes[9]:

1) Inyección de código SQL.

Se trata de una técnica con la cual un atacante puede ejecutar sentencias SQL en la base de datos mediante la manipulación de la entrada proporcionada a la aplicación[9]. Para mitigar esta vulnerabilidad es imprescindible comprender la importancia de la validación de todas las entradas tanto en el cliente como en el lado del servidor. El uso de parámetros en la construcción de las consultas también elimina en gran medida la posibilidad de recibir un ataque de este tipo.

2) Ejecución inter sitio (*Cross Site Scripting*).

La ejecución inter-sitio es un tipo de ataque que puede llevarse a cabo para robar información sensible perteneciente a los usuarios de un sitio web. Esta confía en que el servidor reflejará la entrada del usuario sin verificar si existe JavaScript incrustado[9]. Evitar este ataque es imprescindible pues la información que generalmente es robada a los usuarios es de importancia medular para estos (contraseñas, números de cuenta, PIN). Reemplazar antes de ser mostrados los caracteres especiales usados para construir sentencias script permite atenuar el peligro de este modo de ataque.

3) La manipulación de variables.

Hay ocasiones en las que la información es modificada antes de que llegue al servidor. Los atacantes que naveguen por la aplicación pueden modificar la información de una petición POST o GET[9]. Es posible además que puedan manipular los datos de formularios y variables URL. SSL brinda seguridad en este sentido cifrando el contenido de la página de modo que no viaje en texto claro por la red, aunque existen en la actualidad herramientas capaces de intervenir este tipo de conexiones.

#### 1.1.1.4 Criptografía

La criptografía es el arte o ciencia de cifrar y descifrar información mediante técnicas especiales y es empleada frecuentemente para permitir un intercambio de mensajes que sólo puedan ser leídos por personas a las que van dirigidos y que poseen los medios para descifrarlos. Las

aplicaciones fundamentales de la criptografía son el cifrado y la firma electrónica. Ambas aplicaciones son el núcleo del comercio electrónico y de cualquier transacción segura que se realice por Internet.

Dentro de la criptografía están el cifrado por bloques y el cifrado de flujos. El primero responde a los algoritmos simétricos de cifrado que opera en grupos de bits de longitud fija, llamados bloques, aplicándoles una transformación invariante. Cuando se realiza el cifrado toman un bloque de texto plano o claro como entrada y produce un bloque de igual tamaño de texto cifrado. Para descifrar estos mensajes, el receptor tiene que aplicar sobre el mensaje cifrado la misma clave que empleó el emisor para cifrar el mensaje original. Los algoritmos simétricos conocidos son: DES, Triple DES, IDEA, RC2, RC4 y RC5.

El cifrado de flujos corresponde a los algoritmos de cifrado asimétricos que pueden realizar el cifrado incrementalmente, convirtiendo el texto en claro en texto cifrado bit a bit. Los algoritmos asimétricos (o de clave pública como también se conocen) se basan en el uso de dos claves diferentes. Una clave puede descifrar lo que la otra ha cifrado. Las claves pública y privada tienen características matemáticas especiales, de tal forma que se generan siempre a la vez y por parejas. Ejemplos de algoritmos asimétricos: Diffie-Hellman, El Gamal, RSA.[7]

La diferencia entre el cifrado de flujo y el cifrado de bloque está en el método que usan, mientras los cifrados de bloque cifran bloques de varios bytes a la vez, los cifrados de flujo lo hacen byte a byte.

**Técnicas de cifrado**

Para mantener la información a salvo de todos, a excepción del emisor y el receptor legales de la misma y que permiten garantizar que el pago se ha realizado, se usan las técnicas de cifrado. No solamente van a ser utilizados para cifrar los datos, si no que van a permitir una explotación de sus posibilidades más amplia.

Las técnicas de cifrado tratan de asegurar que:

- Sólo el receptor debe ser capaz de acceder a los datos en claro (confidencialidad).
- Nadie ha podido añadir, quitar o cambiar los datos originales del mensaje, o los que puedan acompañarlo (integridad).
- El mensaje o los datos provienen de quien dice ser (autenticación).[10]

Para lograrlo se emplean los algoritmos de cifrado vistos anteriormente.

**Infraestructura de clave pública (PKI)**

El uso de PKI es un parámetro fundamental para lograr la seguridad en cualquier sistema. La PKI es una combinación de hardware y software, políticas y procedimientos de seguridad. Permite la garantía de la confidencialidad con el cifrado asimétrico de los datos, y mediante la firma digital protege la autenticidad, la disponibilidad, la integridad y el no repudio de la información[7]. Una correcta manipulación de las políticas de seguridad de esta tecnología es

muy importante, ya que si no ocurriese así ni los dispositivos más seguros ni los algoritmos de cifrado más fuertes sirviesen de nada.

**Firma Digital**

Aporta a la transmisión de mensajes telemáticos y en la gestión de documentos electrónicos, un método criptográfico que asocia la identidad de una persona o de un equipo al mensaje o documento, además de cumplir con la función de firmar y garantizar la identidad del firmante, puede asegurar la integridad del documento o mensaje. Esta se vincula a un documento para identificar al autor y garantizar que no se ha modificado su contenido tras ser firmado[7]. Se le aplica una función *hash* al contenido del mensaje, luego es aplicado un algoritmo de firma a dicho *hash* con la clave privada del firmante lo que da como resultado la firma digital. Esta aumenta la seguridad de las comunicaciones que se establecen en la red, además agilizan el proceso de comunicación al no requerir la presencia física de la persona.

**Certificado Digital**

Un certificado digital es un documento digital mediante el cual una autoridad de certificación (AC) o del inglés *Certificate Authority* (CA), garantiza la vinculación entre la identidad de un sujeto y su clave pública. El certificado contiene el nombre de la entidad certificada, el número de serie, la fecha de expiración, la copia de la clave pública del titular del certificado, y la firma digital de la autoridad emisora del certificado, de forma que el receptor pueda verificar que esta última ha establecido realmente la asociación. Cualquier individuo o institución puede generar un certificado digital, pero si este no es reconocido por quienes interactúen con el propietario del certificado, su valor es totalmente nulo. Los certificados digitales permiten establecer la identidad del portador, siendo emitido por una entidad reconocida con autoridad para emitirlos y revocarlos. Se suelen emitir siguiendo el estándar UIT-T X509. Existen varios tipos de certificados dependiendo siempre de quien sea el beneficiario del mismo.[7]

**Autoridad Certificadora**

La AC es la encargada de emitir certificados digitales para las partes que intervienen, da fe de quien presenta una clave pública es quien dice ser, es decir que da legitimidad a la relación de la clave pública con la identidad de un usuario o servicio. Establece también la revocación de los certificados, en caso de robo, pérdida o suspensión de claves privadas. La seguridad de la AC es crítica; un problema de seguridad que la afecte puede afectar a toda la infraestructura existente.[7]

### 1.1.2 Normas y procedimientos técnicos para garantizar la seguridad

Acciones de prevención sobre la gestión de un servidor web:

- La clausura de los servicios de comunicación del servidor, que no sean estrictamente necesarios, y los módulos de los aplicativos que soportan la identificación para acceso remoto (*login* remoto).
- La reducción del número de aplicaciones y archivos abiertos en los servidores.

- La implementación de políticas de control de acceso (deshabilitar las cuentas del sistema a usuarios que dejaron de laborar en la institución, personal con licencia, control de horario en cuentas y deshabilitar las cuentas de usuarios que no se conecten al sistema durante un período de tiempo determinado por el administrador).
- La instalación de un sistema de criptografía para claves de acceso o *password* que no pueden ser descifradas por personas ajenas a la institución.
- La implementación de un sistema de replicación de datos y servicios, para garantizar los servicios de 7 días por 24 horas.[11]

Se norma como acciones de detección:

- La revisión periódica de las últimas actividades realizadas en la base de datos en busca de acciones sospechosas, efectuadas por usuarios externos o internos.
- La configuración del sistema y guardar periódicamente sus resultados en un medio fiable, así como realizar comparaciones periódicas de la configuración operativa actual con la configuración inicial.
- La implementación de un sistema de monitoreo especializado para detectar cualquier evento fuera de lo normal en el sistema.
- La comprobación periódica de la integridad de los archivos importantes del sistema.
- La verificación periódica de los permisos de los archivos que se encuentren en los directorios de usuarios.[11]

Autentificación por contraseñas (identificación de una persona frente a un sistema informático)*:*
Consiste en la introducción mediante un teclado de un *password* que únicamente la persona que se autentifica conoce.

Teniendo en cuenta que un sistema informático moderno suele permitir el acceso de múltiples usuarios, con diversos niveles de privilegios sobre el mismo, almacenar las contraseñas en claro representa un riesgo demasiado elevado. El empleo de funciones resumen (funciones *hash*) permite validar las contraseñas sin necesidad de almacenarlas. Esta estrategia es más segura que el almacenamiento en claro de *password* pero también puede ser punto claro de un ataque sencillo como: el ataque de diccionario.[12]

Ataque de diccionario:

Un usuario malicioso puede construir una base de datos con millones de contraseñas. Una vez obtenido el archivo que contiene las signaturas de las contraseñas de cada usuario, bastaría con buscar en la base de datos el valor correcto. Para protegerse de los ataques de diccionario existen dos estrategias básicas:

- Concatenar a cada contraseña un trozo de información aleatorio (llamado sal) antes de calcular su signatura, y almacenar en la base de datos tanto la signatura como la sal. Esto

obligaría a un posible atacante a recalcular todo el diccionario cada vez que quisiera averiguar una contraseña, dificultando enormemente su tarea.

- Escoger contraseñas difíciles de adivinar es fundamental, estas deben ser lo suficientemente complejas como para no aparecer en un diccionario. Para ello es conveniente emplear programas específicos de generación de contraseñas.
- Incorporar rutinas de medición de la calidad de las contraseñas e impedir a los usuarios seleccionar contraseñas demasiado débiles.

Existe no obstante una serie de posibles problemas para un sistema basado en contraseñas, independientes de su implementación desde el punto de vista lógico, que conviene tener en cuenta: si el acceso se lleva a cabo desde un terminal remoto, la contraseña debe enviarse a través de un canal de comunicaciones, por lo que debería emplearse una conexión previamente cifrada. El empleo de un teclado para introducir la palabra secreta puede hacer el sistema susceptible de escuchas. Existen estudios que demuestran que a través de las radiaciones electromagnéticas de un teclado cualquiera, e incluso del simple sonido de cada tecla, es posible conocer lo que introduce el usuario en la consola. El modo más seguro de custodiar la contraseña es la propia memoria.

### 1.1.3 Estándar de seguridad de datos en la industria de tarjetas de pago (PCI DSS)

Es un conjunto de requisitos globales diseñados para asegurar que la información de la tarjeta de crédito o débito del titular permanece protegida independientemente de la forma o el lugar en que se recopile, procese, transmita y almacene. Desarrollado por los miembros fundadores del organismo PCI *Security Standards Council*. Pretende fomentar la adopción internacional de medidas sistemáticas de seguridad de datos.[13]

Las directivas y requisitos del Estándar PCI DSS incluyen:

- ➢ Creación y mantenimiento de una red de seguridad:
- ✓ Requisito 1: instalar y mantener una configuración de firewall para proteger los datos del titular de la tarjeta.
- ✓ Requisito 2: no usar los valores predeterminados facilitados por el proveedor para las contraseñas de sistema y otros parámetros de seguridad.
- ➢ Protección de los datos del titular de la tarjeta:
- ✓ Requisito 3: proteger los datos del titular de la tarjeta almacenados.
- ✓ Requisito 4: cifrar la transmisión de los datos del titular de la tarjeta a través de redes abiertas y públicas.
- ➢ Uso de un programa de administración de vulnerabilidades:
- ✓ Requisito 5: usar y actualizar regularmente software antivirus.
- ✓ Requisito 6: desarrollar y mantener sistemas y aplicaciones de seguridad.
- ➢ Implementación de medidas de control de acceso seguro:

- ✓ Requisito 7: limitar el acceso a los datos del titular de la tarjeta a las operaciones empresariales imprescindibles.
- ✓ Requisito 8: asignar un identificador único a cada persona que tenga acceso al equipo.
- ✓ Requisito 9: restringir el acceso físico a los datos del titular de la tarjeta.
- ➢ Control y prueba periódica de las redes:
- ✓ Requisito 10: seguir y controlar los accesos a los recursos de la red y a los datos del titular de la tarjeta.
- ✓ Requisito 11: probar regularmente los sistemas y procesos de seguridad.
- ➢ Mantenimiento de una directiva de seguridad de la información:
- ✓ Requisito 12: mantener una directiva que controle la seguridad de la información.

Los requisitos 9 y 12 no requieren la implementación de soluciones de tecnología. El requisito 9 exige controlar la seguridad física de las ubicaciones en las que se almacenan y procesan los datos del titular de la tarjeta. Esto puede incluir implementar medidas de seguridad de acceso a los edificios, instalar y mantener un equipo de vigilancia y exigir comprobaciones de identidad para las personas que trabajan en las instalaciones o las visitan. El requisito 12 demanda la creación de una directiva de seguridad de la información y la difusión entre los empleados, proveedores y otras personas de la organización que trabajen con los datos del titular de la tarjeta.[13]

### 1.1.4 Normas de seguridad de datos para las aplicaciones de pago (PA-DSS)

Las PA-DSS se derivan de los PCI-DSS. Esta guía recoge las responsabilidades y las características básicas para garantizar y mantener la seguridad en una aplicación de pago:

1. No retener la banda magnética, el código o valor de validación de la tarjeta ni los datos de bloqueo del PIN (*Personal Identification Number*).

- Para minimizar el riesgo, almacene solamente los elementos de datos que sean necesarios para el negocio.

2. Proteger los datos del titular de la tarjeta que fueron almacenados.

- Ocultar el PAN (*Personal Account Number*) cuando aparezca (los primeros cuatro y los últimos cuatro dígitos es la cantidad máxima de dígitos que aparecerá). Se utilizan los métodos:

  -Valores *hash* de una vía basados en criptografía sólida (el *hash* debe ser de todo el PAN).

  -Truncamiento (los valores *has*h no se pueden usar para reemplazar el segmento truncado del PAN)

  -*Tokens* y ensambladores de índices (los ensambladores se deben almacenar de manera segura).

  Criptografía sólida con procesos y procedimientos asociados para la gestión de claves.

-La aplicación de pago debe proteger las claves utilizadas para asegurar los datos de los titulares de tarjeta contra divulgación o uso indebido.

3. Proporcione funciones de autenticación segura.

- La aplicación de pago debe admitir y aplicar el uso de ID de usuario únicas y autenticación segura para todo el acceso administrativo y para todo acceso a los datos de titulares de tarjeta. La autenticación segura se debe aplicar para todas las cuentas, generadas o administrativas por la aplicación.
- La aplicación de pago emplea uno de los siguientes métodos para autenticar a todos los usuarios:

  -Algo que el usuario sepa, como una contraseña o frase de seguridad.

  -Algo que el usuario tenga, como un dispositivo *token* o una tarjeta inteligente.

  -Algo que el usuario sea, como un rasgo biométrico.
- La aplicación de pago requiere que se cambien las contraseñas de usuario por lo menos cada 90 días.
- La aplicación de pago requiere una longitud de contraseña mínima de siete caracteres.
- La aplicación de pago requiere que las contraseñas contengan caracteres numéricos y alfabéticos.
- La aplicación de pago limita los intentos de acceso repetidos bloqueando la cuenta del usuario después de más de seis intentos de inicio de sesión.
- Las contraseñas de la aplicación de pago deben ser ilegibles durante la transmisión y el almacenamiento, usando criptografía sólida.

4. Desarrollar aplicaciones de pago seguras.

- Desarrollar todas las aplicaciones de pago bajo las directrices de codificación segura. Considerar las vulnerabilidades de codificación comunes como inyección de código SQL, comunicaciones inseguras, manejo inadecuado de errores. La aplicación de pago debe utilizar servicios y protocolos seguros.

5. Probar a las aplicaciones de pago para tratar las vulnerabilidades.

- Los proveedores de software deben establecer un proceso para identificar y asignar una clasificación de riesgo a las vulnerabilidades de seguridad recientemente descubiertas y para probar sus aplicaciones a fin de determinar la presencia de vulnerabilidades. En este proceso, se debe incluir todo software o sistema subyacente que provea o requiera la aplicación de pago (por ejemplo, los servidores web, programas y bibliotecas de terceros).

6. Facilitar la implementación de una red segura.

- La aplicación de pago debe ser capaz de implementarse en un entorno de red seguro. La aplicación no debe interferir con el uso de dispositivos, aplicaciones ni configuraciones que se requieran para cumplir con las PCI-DSS.

7. Los datos de titulares de tarjetas nunca se deben almacenar en un servidor conectado a Internet.

- La aplicación de pago se debe desarrollar de manera que el servidor de base de datos y el servidor web no tengan que estar en el mismo servidor ni se requiera que el servidor de base de datos se encuentre en la DMZ (zona desmilitarizada).

8. Cifre el tráfico sensitivo de las redes públicas.

- Si la aplicación de pago envía o facilita el envío de datos de las tarjetas por redes públicas, la aplicación de pago debe respaldar el uso de cifrado sólido o de protocolos de seguridad (por ejemplo, SSL/TLS, seguridad del protocolo Internet) para proteger datos sensitivos del titular de la tarjeta durante la transmisión por redes públicas abiertas. [14]

### 1.1.5 Requisitos de comunicación entre las entidades

Para poder integrar una pasarela de pagos se debe poseer una tienda virtual que tenga como mínimo los siguientes requisitos: registro y autenticación de clientes, proceso de compra, determinación del costo, módulo integrador con la pasarela de pagos. Además la tienda virtual debe poseer políticas de venta, envíos, devoluciones y privacidad aprobadas por la pasarela de pago que se seleccione. Cada pasarela de pagos cumple con una serie de requisitos técnicos, que se verifican antes de auditar la tienda virtual. [15]

La comunicación entre una pasarela de pagos y el banco emisor de las tarjetas se puede realizar por medio de la norma ISO 8583 que proporciona un conjunto de reglas para la definición de protocolos en el intercambio de mensajes de transacciones financieras. Es un sistema de mensajes que adopta un formato uniforme para la integración, la interoperabilidad y el intercambio seguro de claves y otros propósitos administrativos.

#### 1.1.5.1 Norma ISO 8583

Una transacción con tarjeta por lo general tiene que viajar entre un número de sistemas. La operación lleva la información sobre el tipo de transacción, la tarjeta utilizada, el comerciante, el monto de la transacción, la información de seguridad, y así sucesivamente. La respuesta, de autorizar o rechazar la transacción, tiene que ser devuelto por la misma ruta a la terminal.

ISO 8583 define un formato de mensaje y un flujo de comunicación para que diferentes sistemas puedan intercambiar solicitudes de transacciones y de respuestas. Un mensaje de la norma ISO 8583 se compone de las siguientes partes: el MTI (Indicador de Tipo de mensaje), el mapa de bits y los datos reales del mensaje, agrupados en una serie de elementos.

El MTI es un código de cuatro dígitos numéricos que contiene información sobre el tipo de mensaje. Esto incluye información sobre la versión de la ISO, la función del mensaje y quién lo está enviando. El MTI es seguido por el mapa de bits que indica que los elementos están presentes en el cuerpo del mensaje. En implementaciones típicas, el mapa de bits es realmente un vector de bits individuales, 8 o 16 bytes de longitud y por lo tanto puede indicar la presencia o ausencia o 64 o 128 elementos de datos en el cuerpo del mensaje. El mapa de bits es

seguido por la serie de uno de los elementos de datos o campos, algunos de los cuales pueden subdividirse en subcampos. El modo específico de codificación de datos varía de un campo a otro.[16]

## 1.2 Pasarelas de pagos

Una pasarela de pagos es una página web que representa un servicio intermediario entre una página de comercio electrónico y un banco cuando se ejecutan transacciones bancarias *online*. Las pasarelas de pagos se integran a la tienda virtual y almacenan información del banco que maneja las cuentas de compradores y vendedores. Es una herramienta importante entre la compra y venta dentro del comercio electrónico. En el pago con tarjeta, la pasarela de pagos valida la veracidad de la tarjeta y organiza la transferencia del dinero de la cuenta del comprador a la cuenta del vendedor. Son comúnmente llamadas terminal de punto de venta (TPV) virtual, pero no son realmente terminales de punto de venta porque estas últimas sí pertenecen al banco y el vendedor debe tener una cuenta en el banco en donde esté implementado el TPV virtual. Para el proceso de pago o transferencias bancarias es necesaria la utilización de tarjetas de crédito o débito como medios electrónicos de pago.

El proceso de pago utilizando las pasarelas consta de varias fases. El cliente accede a un sitio de comercio electrónico y elige la lista de artículos a comprar. La aplicación calcula el importe a cobrar y cuando el cliente está listo para pagar, es redireccionado a la pasarela, la cual le muestra el monto a pagar y los datos a introducir como el número de tarjeta. La pasarela se encarga de codificar la información la cual viaja de forma segura hacia el banco. En este se comprueba rápidamente que la tarjeta sea válida (que no sea robada o que esté caducada) y que el cliente tenga los fondos suficientes para comprar los artículos. De estar todo en orden se ingresa el dinero en la cuenta del vendedor, la misma debe pertenecer al banco en cuestión o este debe tener relación con el banco que posee la cuenta del vendedor. La pasarela le comunica al comercio y al cliente el resultado de la transacción (si el pago se efectuó o no).

El proceso de transferencia sería similar: el cliente se encargará de informar a que cuenta desea ingresar la transferencia, la pasarela manipulará la información para que viaje confidencial hacia la cuenta destino, si no se presentan inconvenientes o si se presentaran, el cliente siempre será notificado. Finalmente todas las transacciones realizadas se actualizan en la base de datos de la pasarela.

### 1.2.1 Pasarelas de pagos internacionales más utilizadas

Actualmente existe gran diversidad en todo el planeta en cuanto a pasarelas de pagos se refiere. Cada una de ellas se diferencia entre sí por los requisitos que cumplen, el costo que cobran por transacciones, y los países con los cuales trabajan.

Paypal

*Paypal* es una pasarela de pagos de intermediación financiera que permite realizar compras y ventas *online* de forma segura. Es la más conocida a nivel mundial y pertenece a los Estados

Unidos (EE.UU). Paypal compone el procesador de pagos en línea ideal para hacer comercio electrónico y está extendida a más de 50 países, por lo que tiene un gran prestigio mundial. El servicio de Paypal permite la transferencia de dinero entre usuarios que tengan correo electrónico; es una alternativa a los tradicionales cheques o giros postales. Con el uso de esta pasarela se pueden realizar peticiones de pago en comercio electrónico de sitios web de terceros, lo cual implica un sistema de validación de pagos *online* portable y adaptable. Los métodos de transferencia y pago por medio del correo electrónico tienen asociado una tarjeta de crédito (tales como Visa, Mastercard, American Express y Diners de compradores de EE.UU y Europa). Estos procesos son rápidos y seguros, debido al protocolo de seguridad SSL. Aún no tiene información en español aunque se especula que en un futuro la tendrá (existen varias comisiones en otros idiomas).

Los costos por transacción no son altos y el registro es 100% confiable, no hay costos ocultos, ni mensualidades ni mucho menos costo de inscripción. No cobran comisión por enviar dinero, pero si por recibir. Contiene una cuenta *personal* que está pensada para compradores por la cual se podrá realizar de forma gratuita, los pagos de las compras realizadas. Sus principales funcionalidades son: pagar de forma segura en Internet y enviar y recibir dinero. Existe otra cuenta llamada *premier* que está pensada para vendedores. Se recibe, a tarifas reducidas, los pagos de las ventas en Internet. Esta cuenta permitirá: aceptar las principales formas de pago y recibir y gestionar pagos. Los costos por cada transacción son: 2.9% + US$ 0.25 por cada transacción menor a US$ 2900.00; 1.9% + US$ 0.25 por cada transacción mayor a US$ 2900.00.[17]

2Checkout (2CO)

*2CO* es la segunda pasarela a nivel mundial, por su uso, calidad, seguridad y prestigio. Esta pasarela trabaja con el 95% de los países del mundo. El pago que se emplea con su uso es de$ 49.00, para darse de alta. No hay pagos semanales, mensuales ni anuales. El pago se realiza al momento de registrarse como usuario. Por cada venta realizada, le cobra el 5.5% de la transacción más $ 0.45 por transacción.

Con esta pasarela se puede vender a todo el mundo y recibir dinero en cualquier banco. Le permite aceptar tarjetas de crédito como: Visa, Mastercard, American Express y Diners. El sistema realiza en tiempo real la verificación de la tarjeta de crédito. La información del comprador viaja encriptada con un nivel de seguridad de 128bits (protocolo SSL), y se utilizan certificados digitales que garantizan la autenticidad de las partes implicadas y con ello la seguridad de la transacción.

Esta pasarela es ideal para comerciantes que desean vender a EE.UU, Europa y Latinoamérica o que por razones comerciales no pueden trabajar con Paypal. Los países que no pueden usar 2CO son: Corea del Norte, Cuba, Irán, Sudán, Syria, Myanmar (Burma).[18]

E-Pagado

*E-Pagado* es una empresa que ha salido en español que gestiona los pagos con el correo electrónico y con el teléfono móvil del destinatario. Permite enviar y recibir dinero de forma segura inmediata y gratuita desde cualquier cuenta bancaria o tarjetas de crédito.

El servicio lo proporciona el banco Bankinter (banco Intercontinental Español). La pasarela trabaja con dos tipos de cuentas, una personal y otra comercial, está dirigida a pymes (pequeñas y medianas empresas) y empresas. El sistema de E-Pagado además de solicitar la cuenta de correo de los clientes, exige una contraseña de seguridad para acceder al servicio. Se pueden realizar compras desde los sitios web electrónicos asociados sin necesidad de enviar el número de cuenta o la tarjeta de crédito, esto supone un incremento en la seguridad y privacidad del sistema. La pasarela se puede descargar de forma instantánea desde la propia web.

Como forma de pago electrónico tiene múltiples ventajas frente a otros sistemas basados en el uso de tarjetas o cuentas corrientes. Para el comercio, las ventajas se basan en que no existe posibilidad de repudio, el cobro se produce al instante, no existe posibilidad de fraude, permite el cobro de cualquier importe por pequeño que sea y tiene unas comisiones inferiores a las del resto de sistemas de pago actuales. Sus comisiones son: completamente gratis para los particulares, enviar, pedir, comprar añadir o retirar fondos, si es un comercio y se da de alta cobran el 3% por recibir dinero, el 0,35 por retirar fondos, el 0,25% si la transferencia es periódica y el 35 de la venta por cobro-web, el resto es gratuito.[19]

AlertPay

Una de las pasarelas de pagos más conocida por la cantidad de países que pueden usarla es la pasarela canadiense *AlertPay*. Es un servicio que mediante el correo proporciona enviar y recibir dinero *online* desde y hacia cualquier parte del mundo, de manera segura, fácil y rápida, siendo esto su más alta prioridad. Esto viene siendo una gran ventaja sobre la pasarela Paypal pues esta última permite pagar pero no se pueden recibir pagos en varios países del mundo, en especial de Latinoamérica.

Las comisiones son 2.5% del importe de la venta + $0.25 fijo. El servicio se encuentra verificado por VISA, el servicio está certificado por BBBOnline, Verisign Secure, McAfee Secure. Su información confidencial es cifrada y usa el protocolo de cifrado SSL.[20]

SOLPAGOS

En Perú se desarrolló por ingenieros (peruanos) la pasarela *SOLPAGOS*. Es la primera pasarela de pagos integral con tarjetas de crédito y pago *online* exclusiva para micro y pequeñas empresas formales del país. Acepta pagos en línea y en tiempo real a través de Internet con tarjetas Visa y Mastercard, y le permite al usuario acceder y ver en su cuenta bancaria los montos de sus ventas cada semana. Se instala en minutos, es una solución de

pagos extremadamente intuitiva y no requiere de características especiales para su integración. Las comisiones por transacción son: por cada venta que se realice se abonan a SOLPAGOS el 5.90% del importe total de la misma. Este monto se descuenta automáticamente de sus ventas y es el monto sobre el que la pasarela emitirá la correspondiente factura. Presenta un sello de confianza que lo acredita como un comercio verificado y formal.

Para su completa seguridad, el sistema SOLPAGOS incluye, sin costo adicional, un exclusivo sistema anti-fraude que chequea y verifica parámetros tales como localización de la computadora desde la que se hizo el pago y distancia en kilómetros a la ciudad de envío de la mercadería, coincidencia del número telefónico, uso de *proxies* anónimos, correos gratuitos, país y código postal del cliente, ciudad donde se emitió la tarjeta de crédito, proporcionando una valoración de 0 a 10 del posible nivel de fraude del pedido.[21]

### 1.2.2 Proyecto cubano de pasarela de pagos

Desde el año 2007 se empezó a trabajar en Cuba en un proyecto para el desarrollo del Comercio Electrónico con tarjetas mayoristas llamado CE-Link. Fue introducido por el Banco Central de Cuba (BCC) para el pago en tiempo real de las transacciones de comercio electrónico entre los bancos comerciales, clientes y las tiendas virtuales que realizarán este tipo de comercio en el país. Posee un módulo de administración y otro de consultas con autentificación de usuario. En el módulo de consultas los bancos y tiendas virtuales pueden consultar las transacciones que han sido enviadas hacia cada una de estas entidades.

El CE-Link almacena toda la información relacionada con cada transacción de pago, y se conforman ficheros de compensación diariamente. La pasarela se probó y funciona correctamente pero aún no se ha hecho extensivo su uso a todo el país pues se necesitan los requisitos de seguridad desarrollados por el Ministerio del Interior (MININT) para garantizar autenticidad e integridad de la información entre Tienda – CE-Link y Cliente – CE-Link.[22]

### 1.2.3 Características de una pasarela de pagos segura

Después de realizado el estudio del estado del arte de las pasarelas extranjeras y del proyecto cubano para empresas mayoristas. Para el futuro desarrollo de la pasarela de pagos se pueden tener en cuenta los aspectos principales y más importantes definidos. Para lograr una correcta seguridad en la comunicación con las entidades que intercambian información con una pasarela de pagos, es recomendable el uso de los protocolos SSL y HTTPS que ya han sido probados y estudiados como protocolos seguros en pasarelas de pagos. Para vencer las desventajas que pudieran tener estos protocolos es necesaria la integración de los mismos con la PKI, proporcionando la codificación de la información sensible que viaja por la pasarela de pagos. Para garantizar la autenticidad de las partes implicadas en los procesos de la pasarela de pagos, se utilizan las firmas y certificados digitales. Como los medios de pagos más usados son las tarjetas electrónicas, las pasarelas trabajan con tarjetas de crédito y débito, estas podrían ser generadas por bancos cubanos y manejar las monedas cubanas. Además aspecto que no

podría faltar como servicio de una pasarela de pagos, es permitir generar una factura electrónica en formato pdf firmada electrónicamente por esta. La factura informaría sobre los reportes de todas las acciones efectuadas por los usuarios en la pasarela de pagos, evitando el no repudio.

## 1.3 Tecnologías de desarrollo de software

Se definen en los siguientes sub-epígrafes la metodología, herramientas de desarrollo de software y los lenguajes de programación y de modelado del mismo. Estos constituyen las tecnologías empleadas para la creación y funcionamiento de la pasarela de pagos.

### 1.3.1 Metodologías de desarrollo

Las metodologías de desarrollo son un conjunto de procedimientos, técnicas, herramientas y un soporte documental que ayuda a los desarrolladores a construir software. Actualmente han cobrado popularidad las metodologías ágiles por sus ventajas sobre las metodologías pesadas.[23]

Las metodologías ágiles de desarrollo están especialmente indicadas en proyectos con requisitos poco definidos o cambiantes. Estas metodologías se aplican bien en equipos pequeños que resuelven problemas concretos, lo que no está reñido con su aplicación en el desarrollo de grandes sistemas, ya que una correcta modularización de los mismos es fundamental para su exitosa implantación. Dividir el trabajo en módulos abordables minimiza los fallos y el coste. Las metodologías ágiles presentan diversas ventajas, entre las que se destacan son:

- Capacidad de respuesta a cambios de requisitos a lo largo del desarrollo.
- Entrega continua y en plazos breves de software funcional.
- Trabajo conjunto entre el cliente y el equipo de desarrollo.
- Importancia de la simplicidad, eliminado el trabajo innecesario.
- Atención constante a la excelencia técnica y al buen diseño.
- Mejora continua de los procesos y el equipo de desarrollo.

Con estas metodologías los pequeños grupos de desarrollo se centran en la tarea de construir software fomentando prácticas de fácil adopción y en un entorno ordenado que permiten que los proyectos finalicen exitosamente.

Sobre Microsoft Solutions Framework (MSF)

La metodología MSF es una metodología ágil que se compone de principios, modelos y disciplinas. Se conoce como la metodología de soluciones de Microsoft. MSF sirve como guía para administrar el equipo y los procesos en el desarrollo de software y contempla:

• El modelo de equipo enfocado a la administración de recursos.

Los equipos son pequeños y multidisciplinarios. Los miembros comparten responsabilidades y complementan sus habilidades para enfocarse al proyecto. Cada uno tiene un rol definido que adquiere relevancia en las distintas etapas del proceso de desarrollo.

• El modelo de la aplicación enfocado a la funcionalidad del desarrollo.

Este modelo contempla un diseño lógico en tres capas para el diseño de aplicaciones distribuidas multicapas. Define una aplicación como una red lógica de servicios distribuibles y reutilizables que cooperan en tareas comunes.

• El modelo de proceso enfocado a la programación del desarrollo.

Provee una estructura para el desarrollo de aplicaciones que consiste en 6 etapas distintas principales, también se pueden acortar las etapas en 4, la decisión de extender o no las etapas, depende del equipo de desarrollo y del cliente general. Cada una de las etapas culmina con una meta definida. Una primera etapa es la Visión en la que todo el equipo de desarrollo se reúne y definen la visión y el ámbito que el sistema va a tener. La Planeación es la segunda etapa en la que se especifican y definen las funcionalidades que compondrán al sistema. En la tercera etapa que es el Desarrollo es donde se crea y prueba la solución. En la Estabilización que es la siguiente etapa es donde se crea una solución piloto para el lanzamiento del producto. Finalmente se realizan la Instalación y el Soporte del producto en el lugar donde se vaya a usar.

MSF para Metodologías de Desarrollo Ágil (MSF for ASD) es un proceso de desarrollo formado por escenarios. Incorpora las prácticas probadas por Microsoft con respecto a los requerimientos, diseño, seguridad, rendimiento y pruebas. Presenta una guía recomendable para gestores y desarrolladores de proyecto software adaptable a la metodología de cada empresa. En la guía se incluyen: documentos de ejemplos, plantillas, archivos en blanco de Project, Excel y Word para la administración. El estudio realizado sobre metodologías ágiles permitió escoger a MSF for ASD como metodología de desarrollo para la creación del software. Sus modelos, principios y actividades son grandes ventajas de esta sobre las demás que se podían utilizar.[24]

### 1.3.2 Lenguajes de programación

Los lenguajes caracterizados a continuación son los lenguajes que serán usados por el equipo de desarrollo para la creación del software.

C Sharp (C#)

C# es un lenguaje de programación moderno, potente, flexible y orientado a objetos creado por Microsoft para la plataforma .NET (plataforma que permite desarrollar componentes software integrando varios lenguajes de programación). Es una combinación de los mejores elementos de lenguajes de amplia difusión como C++, Java, Visual Basic o Delphi.

Es el lenguaje que ha sido diseñado específicamente para ser utilizado en la plataforma .NET, por lo que programarla usando C# es mucho más sencillo e intuitivo que hacerlo con cualquiera de los otros lenguajes ya que carece de elementos heredados innecesarios en .NET. Por esta razón, se suele decir que es el lenguaje nativo de .NET. Se pueden crear sistemas: simples o complejos, aplicaciones de consola, de escritorio o para la Web, programas para computadoras personales o para dispositivos móviles.[25]

ASP.Net

El lenguaje ASP.Net es un lenguaje de programación de Microsoft del lado servidor utilizado para la generación de páginas web dinámicas. Surge del lenguaje *Active Server Pages* (ASP) que es menos complejo que el ASP.Net. Es un lenguaje adecuado para acceso a bases de datos y lectura de ficheros. Forma parte de .NET Framework y al codificar las aplicaciones ASP.NET tiene acceso a las clases en .NET Framework. El código de las aplicaciones puede escribirse en cualquier lenguaje compatible con el *Common Language Runtime* (CLR, entorno de ejecución en el que se cargan las aplicaciones desarrolladas en los distintos lenguajes), entre ellos Microsoft Visual Basic (VB.NET), C#, JScript .NET C++y J#. Estos lenguajes permiten desarrollar aplicaciones ASP.NET que se benefician del CLR, seguridad de tipos y herencia.

Las páginas creadas con la tecnología ASP.NET funcionan en tipos de navegadores como: Netscape, Opera, AOL, Internet Explorer. Incluye: marco de trabajo de página y controles, compilador de ASP.NET, infraestructura de seguridad, funciones de administración de estado, configuración de la aplicación, supervisión de estado y características de rendimiento, capacidad de depuración, marco de trabajo de servicios Web XML , entorno de diseñador extensible y entorno de host extensible y administración del ciclo de vida de las aplicaciones.

*Framework* ASP.Net MVC 2.0

El marco de ASP.NET MVC es un marco de presentación de poca complejidad y fácil de comprobar que se integra con las características de ASP.NET existentes.

ASP.NET MVC se concibió como alternativa a *Web Forms* y proporciona un modelo de programación basado en el popular patrón de arquitectura Modelo-Vista-Controlador (MVC).

El MVC es un estilo de arquitectura de software de llamada y retorno. Tiene como objetivo separar la interfaz gráfica del código que hace que funcione cualquier aplicación.[42]

Entre sus principales características destacan su completa integración con pruebas unitarias y su separación más clara entre la lógica de presentación (la interfaz de usuario), la lógica de negocio (la lógica de control) y la lógica de acceso a datos (los datos de una aplicación).[26]

Las ventajas de utilizar MVC se destacan a continuación:

- Se pueden agregar nuevas vistas si fuese necesario, al ser estas susceptibles de modificación con el MVC no crearía problemas en el sistema pues seguiría funcionando sin necesidad de paralizarse.
- Se podrían modificar los objetos de negocios para cambiar de tecnología.
- Se pueden aplicar opciones como el multilenguaje y distintos diseños de presentación sin alterar la lógica de negocio.

CSS

Las hojas en estilo de cascada es el nombre que se le atribuye al lenguaje CSS (*Cascading Style Sheets*). Es un lenguaje que proporciona estilo a documentos HTML y XML, separando el contenido de la presentación. Los estilos definen la forma de mostrar los elementos HTML y XML. Los estilos pueden ser en cuanto a fuentes, colores, márgenes, líneas, altura, anchura, imágenes de fondo, posicionamiento avanzado y muchos otros temas. Es posible usar HTML para añadir formato a los sitios web, sin embargo, CSS ofrece a los desarrolladores el control total sobre el estilo y formato de múltiples páginas web al mismo tiempo. Además es más preciso, sofisticado y está soportado por todos los navegadores. [27]

CSS fue toda una revolución en el mundo del diseño web. Entre los beneficios concretos de CSS se encuentran:

- Control de la presentación de muchos documentos desde una única hoja de estilo.
- Control más preciso de la presentación.
- Aplicación de diferentes presentaciones a diferentes tipos de medios (como pantalla, e impresión).
- Numerosas técnicas avanzadas y sofisticadas.[28]

JavaScript (JS)

JS es un lenguaje de programación que permite a los desarrolladores crear acciones en sus páginas web para interactuar con los usuarios. No requiere de compilación ya que el lenguaje funciona del lado del cliente, los navegadores son los encargados de interpretar estos códigos. Es soportado por la mayoría de estos como Internet Explorer, Netscape, Opera y Mozilla Firefox, entre otros.

JS posee varias características, entre ellas se pueden mencionar que es un lenguaje basado en acciones que posee menos restricciones. Además, es un lenguaje que utiliza Windows y gran parte de la programación en este lenguaje está centrada en describir objetos, escribir funciones que respondan a movimientos del mouse, aperturas, utilización de teclas, cargas de páginas entre otros.[29]

PL/pgSQL

PL/pgSQL es el lenguaje procedural nativo e interno del sistema gestor de bases de datos PostgreSQL. Este lenguaje es comparable al lenguaje procedural de Oracle, PL/SQL. Se usa para crear funciones y procedimientos disparados por eventos, añade estructuras de control al lenguaje SQL, puede realizar cálculos complejos, hereda todos los tipos definidos por el usuario, las funciones y los operadores, puede ser definido para ser fiable para el servidor y es fácil de usar.[30]

Excepto en el caso de funciones de conversión de entrada/salida y de cálculo para tipos definidos, cualquier cosa que pueda definirse en funciones de lenguaje C puede ser hecho con

PL/pgSQL. Es posible crear funciones complejas de cálculo y después usarlas para definir operadores o usarlas en índices funcionales.[31]

### 1.3.3 Herramientas

Las herramientas descritas a continuación son las escogidas entre el estudio realizado de varias existentes. Se determinaron las expuestas por su correcta adaptación, sus facilidades y ventajas para la implementación de la pasarela de pagos.

Entorno de desarrollo integrado (IDE)

Un IDE o del inglés *Integrated development environment*, es un programa compuesto por un conjunto de herramientas para un programador. Ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica. Pueden ser aplicaciones por si solas o pueden ser parte de aplicaciones existentes. Se dedican a uno o varios lenguajes de programación.

Los IDEs proveen un marco de trabajo amigable para la mayoría de los lenguajes de programación tales como C++, Java, C#, Basic y Object Pascal. Algunos ejemplos de IDE son: Eclipse, JBuilder de Borland, JDeveloper de Oracle, Delphi de Borland y Microsoft Visual Studio .NET de Microsoft. Este último fue escogido como el IDE para la programación de la pasarela por sus altas ventajas.

Microsoft Visual Studio Team System 2010 (VSTS) es un entorno de desarrollo integrado para sistemas Windows. Ofrece una interfaz común para trabajar de manera cómoda y visual con los lenguajes de la plataforma .NET (por defecto, C++, C#, Visual Basic.NET y JScript.NET, aunque pueden añadirse nuevos lenguajes mediante los *plugins* que proporcionen sus fabricantes). Permite a los desarrolladores crear aplicaciones de escritorio, sitios y aplicaciones web, así como servicios web en cualquier entorno que soporte la plataforma .NET. Su última versión, la 2010 usa *Windows Presentation Foundation* (WPF), incluye mejoras en la interfaz y el diseño de bases de datos, y proporciona una mejor configuración y adaptación para cualquier proceso de desarrollo ágil.

*Framework* 4.0

Un *framework* es una estructura de software compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación. Los objetivos principales que persigue son: acelerar el proceso de desarrollo, reutilizar código ya existente y promover buenas prácticas de desarrollo como el uso de patrones.[32]

El *framework* de .Net ofrece un entorno de ejecución altamente distribuido, que permite crear aplicaciones robustas y escalables. Los principales componentes de este entorno son: lenguajes de compilación, biblioteca de clases de .Net y CLR. Soporta más de 30 lenguajes de programación y aunque cada lenguaje tiene sus características propias, es posible desarrollar cualquier tipo de aplicación con cualquiera de estos lenguajes.[33]

El *framework* usado por el equipo de desarrollo es la versión 4.0 de *.NET Framework* pues es compatible con las aplicaciones que se han compilado con versiones anteriores. Admite más plataformas que en versiones anteriores y proporciona una implementación rápida de las aplicaciones. Puede identificar sistemas operativos y procesos de 64 bits con las propiedades. Presenta el MVC como nuevo modelo de programación para escribir código multiproceso y asincrónico que simplifica considerablemente el trabajo de los desarrolladores de aplicaciones y de bibliotecas.[34]

Sistema Gestor de Bases de Datos (SGBD)

Un SGBD o DBMA (*Data Base Management System*) es una colección de programas cuyo objetivo es servir de interfaz entre la base de datos, el usuario y las aplicaciones. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta. Un SGBD permite definir los datos a distintos niveles de abstracción y manipular dichos datos, garantizando la seguridad e integridad de los mismos. Algunos ejemplos de SGBD son DB2, Oracle, MySQL, MS SQL Server y PostgreSQL [35], este último fue seleccionado por el equipo de desarrollo como el gestor de base de datos a utilizar en el software.

PostgreSQL es un sistema gestor de base de datos relacional, robusto, con escalabilidad, su código fuente está disponible libremente e implementa los estándares SQL. Está considerado como la base de datos de código abierto más avanzada del mundo. Tiene versiones en distintos sistemas operativos tales como: Linux, Windows, Mac OS X, Solaris, BSD, Tru64 y otros más. Soporta la realización de transacciones seguras, vistas, uniones, claves extranjeras, procedimientos almacenados, *triggers*, tipos de datos definidos por el usuario y otras operaciones que se realizan en estos sistemas. El tamaño máximo de la base de datos es ilimitado; el de una tabla asciende a 32 TB, el de una fila a 1.6 TB y el de un campo de datos a 1 GB; el número de filas en una tabla es ilimitado, pero no el de columnas, que oscila entre 250 y 1600 columnas por tabla. El número de índices por tabla es también ilimitado.[36]

PostgreSQL 9.1 es la versión que incluye una cantidad de nuevas y mejoradas características, que les proporciona más comodidad a los diseñadores de aplicaciones, administradores de bases de datos y usuarios. Ofrece muchas características que los usuarios han estado solicitando por años, retirando obstáculos para el despliegue de aplicaciones nuevas o migradas en PostgreSQL. Estas incluyen:

- Replicación Sincrónica: permitiendo alta disponibilidad con consistencia sobre múltiples servidores.
- Regionalización por columna: soportando correctamente el ordenamiento por lenguaje en las bases de datos, tablas o columnas.
- Tablas *unlogged*: importante incremento del rendimiento para datos efímeros.[37]

Herramienta de modelado de procesos

Modelar es desarrollar una descripción lo más exacta posible de un sistema y de las actividades llevadas a cabo en él. Generalmente estos sistemas están compuestos por procesos y conjuntos de procesos que pueden resultar difíciles de comprender. Con las herramientas de modelado de procesos puede lograrse un mejor entendimiento del dominio del problema e incluso mejorarlo. Es por eso que una vez definida la metodología a utilizar el equipo de trabajo comienza el modelado de los procesos a través de la herramienta Altova UModel.

Altova Umodel 2010 es una herramienta de modelado que se basa en UML (lenguaje de modelado para el análisis y diseño de software, del inglés *Unified Modeling Language*). Puede generar modelos UML e ingeniería inversa. UModel tiene una interfaz visual, rápida y eficaz que proporciona un práctico diseño del software para programadores, maximizando los resultados. UModel elimina el misterio de UML con ayuda contextual a la introducción de datos, sintaxis coloreada, estilos en cascada, elementos de diseño personalizables, múltiples vistas de composición, función deshacer/rehacer ilimitada, y muchas otras funciones de usabilidad.

Soporta todos los diagramas UML como: diagrama de casos de uso, diagrama de actividad, diagrama de clases, diagrama de comunicación, diagrama de componentes. Es compatible con algunos entornos de desarrollo como : Eclipse, Borland, Jbuilder y Microsoft Visual Studio Team System.[38]

Herramienta de modelado de base de datos

ER/Studio

Embarcadero ER/Studio es una herramienta líder para el modelado de datos. Ayuda a las empresas a descubrir, documentar, y reutilizar los activos de datos. Brinda soporte completo a las bases de datos, proporcionando a los arquitectos de las mismas realizar fácilmente ingeniaría inversa. Analiza y optimiza bases de datos existentes. Con las grandes capacidades de colaboración la productividad del ER/Studio gana y refuerza los estándares organizacionales. Documenta y mejora las bases de datos existentes. Mejora la consistencia de los datos. Comunica eficientemente los modelos en una empresa. Traza los orígenes de los datos y mejora la integración y exactitud y modela más que sólo los datos.[39]

La fundamentación de los conceptos abordados en este capítulo arrojaron los siguientes resultados: el análisis sobre las operaciones bancarias permitió definir el pago y transferencia como transacciones bancarias en el comercio electrónico. El análisis de los protocolos seguros, mecanismos de seguridad y mecanismos de cifrado de datos permitió definir la seguridad de los pagos y transferencias bancarias *online*. El análisis de las tecnologías que se utilizan en la creación de un software propició la fundamentación de las herramientas, lenguajes y metodología de la pasarela de pagos.El estudio realizado sobre las pasarelas de pagos proporcionó una mayor comprensión de los requisitos que componen a la pasarela de pagos creada.

# Capítulo 2: Características de la pasarela de pagos.

En el presente capítulo se hace una propuesta de la pasarela de pagos para lo cual se estudiaron los procesos relacionados con el objeto de estudio y el campo de acción. Para una mayor compresión de cómo estará estructurado el sistema se construye el modelo de dominio. Además se especifican y describen los escenarios que contienen cada uno de los módulos. Seguidamente se identifican los requisitos de calidad del servicio.

## 2.1 Fase Visión y Alcance

En la fase visión y alcance se establece el dominio del problema, proporcionando al equipo de trabajo, una visión clara de los objetivos que persigue el desarrollo de la pasarela de pagos, una idea de la lógica del dominio y de las alternativas propuestas para solucionar de manera óptima el problema planteado. Seguidamente se define el modelo de dominio, se analizó este modelo pues la investigación consiste en crear una propuesta de un producto en este caso una pasarela de pagos y no existe un negocio en sí con un cliente en específico. Este modelo de dominio permitirá al analista un mejor entendimiento del entorno al cual la pasarela de pagos va a servir. Es tomado como el punto de partida para el posterior diseño del sistema.

### 2.1.1 Modelo de dominio

El modelo de dominio es una visualización de elementos de un dominio de interés en el mundo real. Los modelos del dominio favorecen la comprensión de los conceptos de un negocio o un dominio de problema. Disminuyen la brecha de representación entre cómo ven los clientes el problema y la representación en software de la solución, usando modelado Orientado por Objetos.[40]

Con la elaboración del modelo de dominio se intenta comprender los conceptos que intervienen en la pasarela de pagos. Este modelo se presenta en un diagrama UML que muestra las clases del dominio, con algunos atributos y como se relacionan unas con otras mediantes asociaciones.

Los conceptos que se muestran a continuación son las clases que intervienen en el modelo de dominio de la pasarela de pagos:

*Usuario*: es la persona que por medio de un sitio web de comercio accede a la pasarela de pagos para realizar un pago o transferencia bancaria.

*Transacción*: es toda acción que realiza el usuario en la pasarela de pagos. Se efectúan por medio de la cuenta del usuario. Pueden ser pagos o transferencias, si estos no se realizan correctamente por errores que se presentan en el transcurso de los procesos, también se consideran transacciones.

*Cuenta*: representa los datos de la tarjeta del usuario. Por medio de esta, la pasarela realiza la comunicación con el banco.

*Banco*: es el banco que maneja las cuentas del usuario y las tarjetas de crédito o débito.

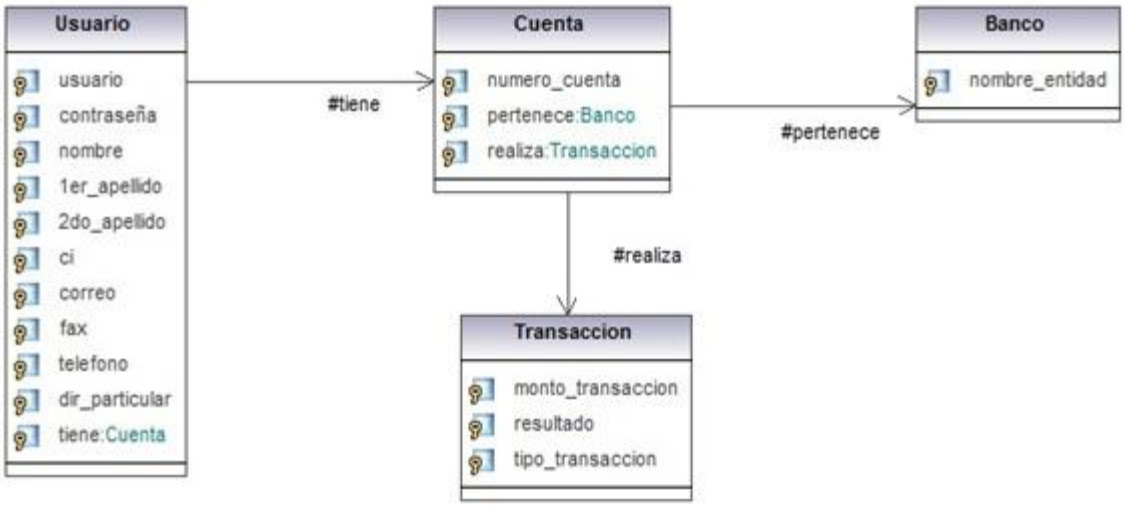



Figura 1. Modelo de dominio. Fuente: elaboración propia.

Descripción del modelo de dominio:

El modelo que se mostró anteriormente representa las entidades que interactúan en la pasarela de pagos para realizar transacciones bancarias en línea. Se muestra un usuario que realiza pagos o transferencias bancarias por medio de una cuenta que está asociada a la tarjeta de crédito o débito. Esta última es emitida por un banco que puede manejar las cuentas de los usuarios o se comunica con otros bancos que en ese caso controlaría estas cuentas. El usuario se registra insertando sus datos personales, creándose automáticamente una cuenta de este en la pasarela. La pasarela almacena además los datos de los bancos que emiten las tarjetas asociadas a las cuentas. Una vez que la pasarela contiene todos estos datos puede notificar a los usuarios y a los bancos de todas las acciones que se realizan por medio de esta.

### 2.1.2 Concepción de los módulos del sistema

Un módulo de un sistema es un conjunto de actividades que se realizan con el fin de darle seguimiento a un proceso y al mismo tiempo consistencia al sistema, desencadenando acciones a realizar por parte de los funcionarios. El sistema tiene como objetivo garantizar que los pagos y trasferencias que se realizan desde una tienda virtual hacia un banco sean de forma segura. Consta de 2 módulos que engloban las funcionalidades requeridas para la obtención exitosa de un sistema seguro y con calidad. La administración de la pasarela se estableció en una aplicación web separada de la que brinda los servicios de la pasarela de pagos, para lograr una mejor separación y organización de las funcionalidades. Estas aplicaciones contienen cada una un módulo con los escenarios necesarios para su correcto funcionamiento. Uno es el módulo “Administración” y el otro “Administración de clientes”.

A continuación se elabora el diagrama con los módulos anteriormente mencionados, describiéndose la relación que existe entre ellos.


Figura 2. Módulos de la pasarela de pagos. Fuente: elaboración propia.

El módulo “Administración” lo manejan los usuarios que se crean como administradores. Desde este los mencionados anteriormente, ejecutan algunas acciones sobre los clientes que tienen cuentas en la pasarela. En el módulo “Administración de clientes”, estos últimos administran sus cuentas personales y bancarias, una vez que el cliente inserta esta información, la misma persiste en la pasarela de forma adecuada para garantizar su seguridad. Con estos datos los clientes realizan los pagos y transferencias. En cada módulo se ejecutan un conjunto de reportes y notificaciones de cada acción realizada sobre la pasarela de pagos.

| Módulo | Descripción |
|---|---|
| **Administración** | El módulo contiene todos los escenarios de configurar cuentas de administración y de clientes, permite además buscar, mostrar y desactivar clientes y gestionar bancos. Especifica cómo la pasarela de pagos notifica al banco sobre las transacciones realizadas, cómo almacena internamente esos datos y cómo genera una factura electrónica con los resultados de cada transacción existente. |
| **Administración de clientes** | El módulo contiene los escenarios de registrar, modificar, desactivar y autenticar clientes. Describe además cómo los clientes una vez que registren los datos de su cuenta pueden ejecutar pagos y transferencias bancarias. |

Tabla 1. Descripción de los módulos. Fuente: elaboración propia.

### 2.1.3 Definición de personas

La metodología MSF for ASD establece el concepto “persona” que se refiere a los usuarios que interactúan con los sistemas que se desarrollan siguiendo sus pautas. La siguiente tabla muestra a las personas que interactúan con la pasarela de pagos.

| Personas | Descripción |
|---|---|

| | |
|---|---|
| Administrador | Es el responsable de buscar, listar, y eliminar cuentas de usuarios y bancos. Tiene el control total de las funcionalidades del sistema. |
| Cliente | Usa las funcionalidades que brinda el sistema para realizar transacciones bancarias. Tiene acceso a la creación, modificación y desactivación de su cuenta personal. |

Tabla 2. Descripción de personas. Fuente: elaboración propia.

## 2.2 Fase Planeación

En esta fase se determina la planeación del proyecto, el equipo de desarrollo prepara las especificaciones funcionales. Los principales artefactos de esta fase son los escenarios y los requerimientos de calidad del servicio que sirven de guía para todo el proceso de desarrollo.

### 2.2.1 Especificación de escenarios de la pasarela de pagos

Para la correcta elaboración de la pasarela de pagos y obtener un mejor razonamiento en el momento de la implementación se especificarán y describirán los escenarios necesarios de la propuesta de solución. Cuando los escenarios se componen de varias actividades se dividen en tareas para hacer menos complejo el trabajo.

Un escenario es un medio por el cual se define una interacción entre una persona y el sistema para lograr cumplir un objetivo específico. En la medida que la persona trata de alcanzar un objetivo, el escenario registra los pasos precisos que se toman en el intento de alcanzar ese objetivo.

A continuación se muestra el listado de los escenarios y sus respectivas tareas, divididos por módulos.

**Módulo Administración:**

- Configurar cuenta de administración.
  - ✓ Mostrar usuario de administración.
  - ✓ Crear usuario de administración:
    -Notificar al administrador de la creación de la cuenta.
  - ✓ Modificar usuario de administración.
  - ✓ Eliminar usuario de administración.
  - ✓ Cambiar contraseña de administrador.
- Configurar cliente.
  - ✓ Mostrar cliente.
  - ✓ Eliminar cliente (desactivarlo).

- ✓ Buscar cliente.
- ✓ Configurar pago.
- ✓ Mostrar operaciones de pago:
  -filtrar por fecha.

- Gestionar banco.
  - ✓ Mostrar banco.
  - ✓ Crear banco.
  - ✓ Modificar banco.
  - ✓ Eliminar banco (desactivar).
- Autenticar administrador.

**Módulo Administración de clientes:**

- Registrar cliente.
  -Notificar al cliente de la creación de la cuenta.
- Configurar cuenta.
  - ✓ Modificar datos del cliente.
  - ✓ Eliminar cliente (desactivar).
  - ✓ Cambiar contraseña.
- Autenticar cliente.
- Configurar cuenta bancaria.
  - ✓ Listar cuentas bancarias.
  - ✓ Registrar cuenta bancaria.
  - ✓ Eliminar cuenta bancaria.
  - ✓ Consultar saldo bancario.
  - ✓ Mostrar reporte del historial de transacciones:
    -filtrar por fecha.
- Realizar pago.
  -Enviar datos del pago.
  -Notificar a las entidades:
    -Al vendedor.
    -Al cliente.
  -Registrar resultado del pago.
- Realizar transferencia.
  -Enviar datos de la transferencia.
  -Notificar a las entidades:
    -Al vendedor.
    -Al cliente.

-Registrar resultado de la transferencia.

### 2.2.2 Requisitos de calidad del servicio

Los requisitos de calidad del servicio son las cualidades que el producto debe tener. Adoptan la forma de restricciones sobre cómo debería funcionar la pasarela de pagos. Estos requisitos son los que definen características del sistema como: rendimiento, ejecución, disponibilidad, confiabilidad, accesibilidad, utilidad y facilidad de mantenimiento entre otros.[24]

A continuación se definen los requisitos de calidad del servicio de la pasarela de pagos:

1. Usabilidad.

1.1. Sobre el uso del sistema.

El sistema podrá ser utilizado por cualquier usuario con conocimientos básicos sobre el uso de una computadora. Se les impartirá un entrenamiento básico en el uso de la aplicación a los administradores de la misma.

1.2. Sobre el diseño de las interfaces.

Las imágenes iconos o botones utilizados deben estar relacionadas con la función que realizan.

2. Fiabilidad.

Este requisito define las características de disponibilidad, confidencialidad, integridad (estas tres cualidades pertenecen a la seguridad del sistema), confiabilidad y capacidad de mantenimiento.

2.1Disponibilidad.

El sistema debe estar disponible las 24 horas los 7 días de la semana. El sistema no funcionará en caso de existir fallas o inestabilidad en las comunicaciones.

2.2 Confiabilidad.

La información manejada por el sistema está protegida de acceso no autorizado. Además se debe prevenir los posibles fallos y/o errores que pudieran presentarse así como posibilitar una rápida recuperación en dichos casos. El sistema registrará todas las acciones que se realicen.

2.3 Integridad.

La información manejada por el sistema permanece inalterada a menos que sea modificada por el personal autorizado, esta modificación es registrada, asegurando su precisión y confiabilidad. Se debe chequear la integridad de los datos.

2.4 Confidencialidad.

Sólo se accederá a la base de datos (BD) desde la aplicación, nunca directamente desde el gestor de BD. A las funcionalidades de la pasarela acceden sólo los usuarios que posean los permisos suficientes.

2.5 Mantenimiento.

Las reparaciones que se realicen en el sistema se deben ejecutar en el tiempo mínimo posible. Deberán realizarse validaciones y comprobaciones automáticas en todos los casos posibles para garantizar la consistencia de los datos.

3. Soporte.

Los desarrolladores de la pasarela de pagos una vez que se haya desplegado la misma en un banco ofrecerán servicios de mantenimiento y actualización.

4. Restricciones de diseño.

El sistema cuenta con un conjunto de patrones de asignación de responsabilidades que permiten la reutilización del código así como una mayor facilidad en la actualización del mismo. El código cuenta con unos estándares de codificación que fueron establecidos por el equipo de desarrollo.

4.1. Referentes a lenguajes de programación.

El sistema debe implementarse utilizando los lenguajes C#, ASP.Net, sobre la plataforma .Net y JS, CSS para propiciarle al sitio estilo y dinamismo.

4.2. Referentes a herramientas de desarrollo.

El sistema debe implementarse usando como IDE Microsoft Visual Studio Team System 2010. Como gestor de base de datos PostgreSQL 9.1.

5. Requisitos para la documentación y la ayuda del sistema.

En el procesos de desarrollo de la pasarela de pagos se generaron los artefactos necesarios que establece la metodología MSF for ASD para comprender como funciona la misma.

6. Interfaz

6.1. Interfaces de usuario.

Interfaz accesible, intuitiva y discreta. El manejo de las funcionalidades debe ser lo más intuitivo posible, de manera que sean muy claras las posibles acciones a llevar a cabo y la manera de hacerlas. Consistencia de la aplicación entre los distintos navegadores como Mozilla Firefox e Internet Explorer.

6.2. Interfaces de *hardware.*

En el cliente se requiere una máquina con 256Mb de RAM y un microprocesador de 1GHz como mínimo, puerto de conexión de red o en su defecto modem u otro dispositivo de interconexión inalámbrico. El servidor de base de datos debe contar con los siguientes requisitos mínimos: 2Gb de RAM, microprocesador 3.06 GHz en adelante y 40gb de espacio de almacenamiento en disco duro.

6.3. Interfaces de *software.*

- Sistema operativo Windows XP/Vista/7, Linux, MAC.
- Framework .NET 4.0 o superior.
- Gestor de base de datos PostgreSQL 9.1.
- En el cliente se requiere tener instalado Navegador web Internet Explorer 7 ó superior, Mozilla Firefox 6.02 o superior (es de vital importancia el cumplimiento de este aspecto en particular pues las características de seguridad incorporadas al producto pueden no funcionar correctamente en versiones inferiores a las especificadas).

6.4. Interfaces de comunicación.

Los servicios web poseerán una interfaz que permita manejar un alto nivel de seguridad haciendo uso del protocolo del nivel de aplicación HTTPS.

7. Requisitos de Licencia.

Para el desarrollo del sistema es necesario el uso de un conjunto de aplicaciones, que son propietarios, siendo indispensables para el buen desempeño de las mismas el tener las respectivas licencias:

- Windows 7 Ultimate edition.
- Internet Information Services 7.5
- Microsoft Visual Studio Team System 2010.

9. Requisitos Legales, de Derecho de Autor y otros.

El software puede ser utilizado por la UCI y en otro caso el CISED es el encargado de decidir cómo será el ambiente de despliegue del mismo.

10. Estándares Aplicables.

La Norma ISO-8583 se emplea como parte de la implementación del sistema. Es un estándar que se utiliza como formato de los mensajes que se intercambian entre la pasarela de pagos y los bancos cuando se generan las solicitudes y las respuestas de las transacciones.

## 2.3 Plan de iteraciones

La estimación del tiempo para codificar los escenarios es muy importante pues le permite al programador una mejor planificación del tiempo. La implementación de la pasarela de pagos se dividirá en dos iteraciones:

**Iteración#1:** se desarrollarán los escenarios pertenecientes al módulo Administración.

**Iteración #2:** se implementará el módulo Administración de clientes.

| **No** | **Módulo** | **Prioridad** | **Riesgo** | **E** | **I** |
|---|---|---|---|---|---|
| 1 | Administración | A | B | 40 | 1 |
| 2 | Administración de clientes | A | M | 60 | 1 |

Tabla 3. Plan de iteraciones. Fuente: elaboración propia.

**Leyenda:**

Esfuerzo (días): E; Iteración: I; Alto: A; Medio: M; Bajo: B.

## 2.4 Descripción de escenarios

Se especifican los escenarios “Registrar cliente”, “Autenticar cliente”, “Adicionar cuenta”, “Realizar transferencia” y “Realizar pago” del módulo “Administración de clientes”.

| | |
|---|---|
| ***Nombre del escenario:*** Registrar cliente. | ***Identificador:*** 5 |
| ***Objetivo del escenario:*** Registrar los datos personales y de su cuenta bancaria. | |
| ***Persona:*** cliente | |

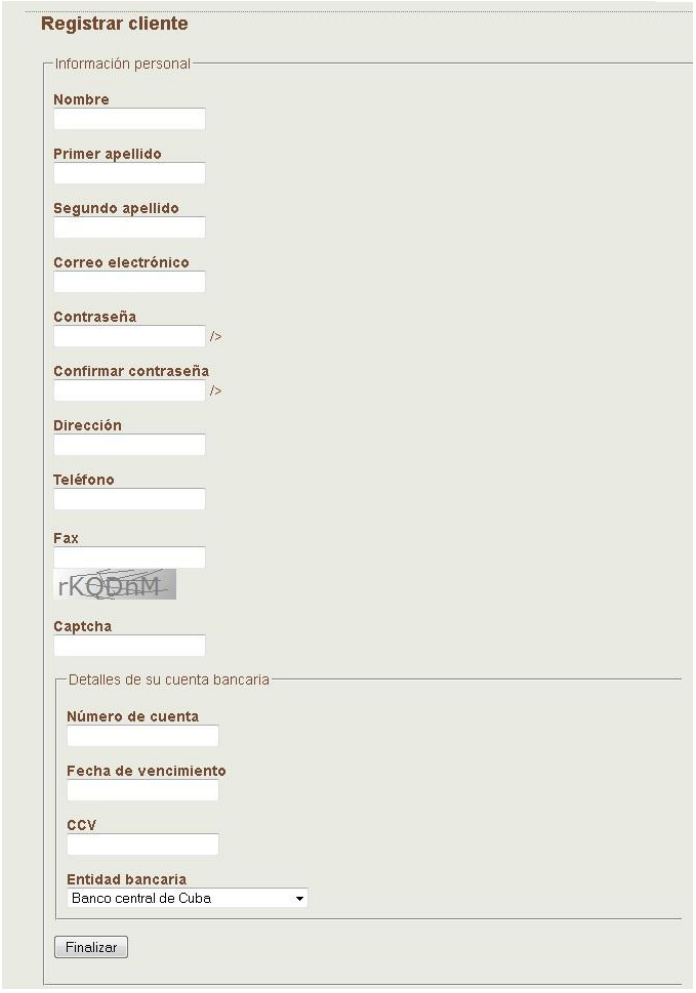
| ***Iteración:*** 2 | ***Prioridad:*** Crítico | ***Complejidad:*** 1 |
|---|---|---|
| ***Precondiciones:***- | | |
| **Descripción*:*** El cliente accede al sistema con el objetivo de registrar sus datos personales y de su cuenta bancaria. Los datos personales a registrar son: nombre, primer apellido, segundo apellido, correo electrónico, contraseña, confirmar contraseña, fax, teléfono (campo opcional), dirección (campo opcional), y catcha. Luego se requiere insertar los datos de la cuenta bancaria que son: número de cuenta, CCV, fecha de vencimiento y entidad bancaria. Se registran todos los datos insertados y se oprime en el botón “Finalizar”. Se notifica al cliente de la creación de su cuenta mediante el correo electrónico. Finaliza el escenario. | | |
| ***Validaciones*:**<br>• Que en cada campo se introduzcan los datos con el formato establecido.<br>• Los campos obligatorios no pueden estar vacíos. | | |
| ***Prototipo de interfaz de usuario:***  | | |



Tabla 4. Descripción del escenario Registrar cliente.
Fuente: elaboración propia.

| | | |
|---|---|---|
| ***Nombre del escenario:*** Autenticar cliente | | ***Identificador:*** 7 |
| ***Objetivo del escenario:*** El cliente se autentica en el sistema. | | |
| ***Persona:*** cliente | | |
| ***Iteración:*** 2 | ***Prioridad:*** Crítico | ***Complejidad:*** 2 |
| ***Precondiciones:*** El cliente debe estar registrado en el sistema. | | |
| **Descripción*:*** El cliente accede al sistema insertando los datos para la autenticación: correo electrónico y contraseña. El usuario se autentica en el sistema oprimiendo el botón “Iniciar sesión”. Finaliza el escenario. | | |
| ***Validaciones*:**<br>• Los datos tienen que introducirse con el formato establecido.<br>• Los campos no pueden estar vacíos.<br>• Validar existencia de la cuenta del cliente. | | |
| ***Prototipo de interfaz de usuario:***<br>  | | |

Tabla 5. Descripción del escenario Autenticar cliente.
Fuente: elaboración propia.

.

| | | |
|---|---|---|
| ***Nombre del escenario:*** Realizar pago. | | ***Identificador:*** 9 |
| ***Objetivo del escenario:*** Realizar un pago desde una tienda virtual. | | |
| ***Persona:*** cliente | | |
| ***Iteración:*** 2 | ***Prioridad:*** Crítico | ***Complejidad:*** 2 |
| ***Precondiciones:*** El cliente debe estar registrado en el sistema. | | |

| |
|---|
| **Descripción:** El cliente desde una tienda virtual accede al sistema, anteriormente introduce los datos de su tarjeta y el monto del pago. Estos son enviados de manera segura al sistema. En el sistema se muestra una interfaz con los datos anteriores. Además el cliente introduce los demás datos de la tarjeta: la fecha de vencimiento, CCV y PIN y oprime el botón “Aceptar”. El sistema envía los datos del pago de forma cifrada hacia la entidad bancaria que emite los datos de la tarjeta. El sistema recibe el resultado del pago desde el banco emisor. El sistema notifica al vendedor y al cliente del resultado del pago. El sistema registra los resultados. Finaliza el escenario. |
| ***Validaciones*:**<br>• Que los datos se introduzcan con el formato establecido.<br>• Que no existan campos vacíos. |
| ***Prototipo de interfaz de usuario:***<br>  |

Tabla 6. Descripción del escenario Realizar pago.
Fuente: elaboración propia.

| | | |
|---|---|---|
| ***Nombre del escenario:*** Realizar transferencia. | | ***Identificador:*** 10 |
| ***Objetivo del escenario:*** Realizar una transferencia de efectivo hacia otra cuenta. | | |
| ***Persona:*** cliente | | |
| ***Iteración:*** 2 | ***Prioridad:*** Crítico | ***Complejidad:*** 2 |
| ***Precondiciones:*** El cliente debe estar autenticado como usuario del sistema. | | |
| **Descripción:** El cliente accede al sistema para realizar una transferencia hacia otra cuenta. El sistema muestra la interfaz de realizar transferencia. Se muestran los campos a introducir: número de la tarjeta, fecha de vencimiento de la tarjeta, CCV, PIN, número de la cuenta destino y monto. El cliente oprime el botón “Aceptar”. El sistema envía los datos cifrados hacia el banco emisor de las tarjetas. Este envía la respuesta de la validación de los datos. El sistema notifica al cliente del resultado de la transferencia. El sistema almacena los datos. Finaliza el escenario. | | |

| |
|---|
| ***Validaciones*:-** |
| ***Prototipo de interfaz de usuario:***   |

Tabla 7. Descripción del escenario Realizar transferencia.
Fuente: elaboración propia.

### 2.4.1 Especificación de tareas por escenarios.

Se especifican las tareas “Listar cuentas bancarias” y “Mostrar reporte de historial de transacciones” del escenario “Configurar cuenta bancaria” del módulo “Administración de clientes”.

| | | |
|---|---|---|
| ***Nombre de la tarea:*** Listar cuentas bancarias. | | ***Identificador:*** 8.1 |
| ***Objetivo de la tarea:*** El cliente lista sus cuentas bancarias. | | |
| ***Persona:*** cliente | | |
| ***Iteración:*** 2 | ***Prioridad:*** Crítico | ***Complejidad:*** 1 |
| ***Precondiciones:*** El cliente debe haber registrado los datos de sus cuentas bancarias. | | |
| **Descripción*:*** El cliente selecciona la opción “Mostrar cuentas bancarias”. El sistema muestra el listado de las cuentas bancarias registradas con los datos: número de cuenta (se muestran sólo los 4 primeros y últimos dígitos), fecha de vencimiento y CCV. Finaliza la tarea. | | |
| ***Validaciones*:-** | | |
| ***Prototipo de interfaz de usuario:*** | | |


Tabla 8. Descripción de la tarea Listar cuentas bancarias.
Fuente: elaboración propia.

| | | |
|---|---|---|
| ***Nombre del a tarea:*** Mostrar reporte del historial de transacciones. | | ***Identificador:*** 8.5 |
| ***Objetivo de la tarea:*** Mostrar el historial de transacciones realizadas por el cliente en el sistema. | | |
| ***Persona:*** cliente | | |
| ***Iteración:***2 | ***Prioridad:*** Crítico | ***Complejidad:*** 1 |
| ***Precondiciones:*** El cliente debe de haber realizado algún pago o transferencia en el sistema. | | |
| **Descripción*:*** El cliente selecciona la opción "Operaciones". El sistema muestra el listado de las transacciones realizadas por ese cliente. Se muestran los pagos y transferencias con los datos: monto, resultado, tipo de transacción y fecha. El sistema además brinda la opción de filtrar por fecha. Finaliza la tarea. | | |
| ***Validaciones*:** - | | |
| ***Prototipo de interfaz de usuario:*** | | |

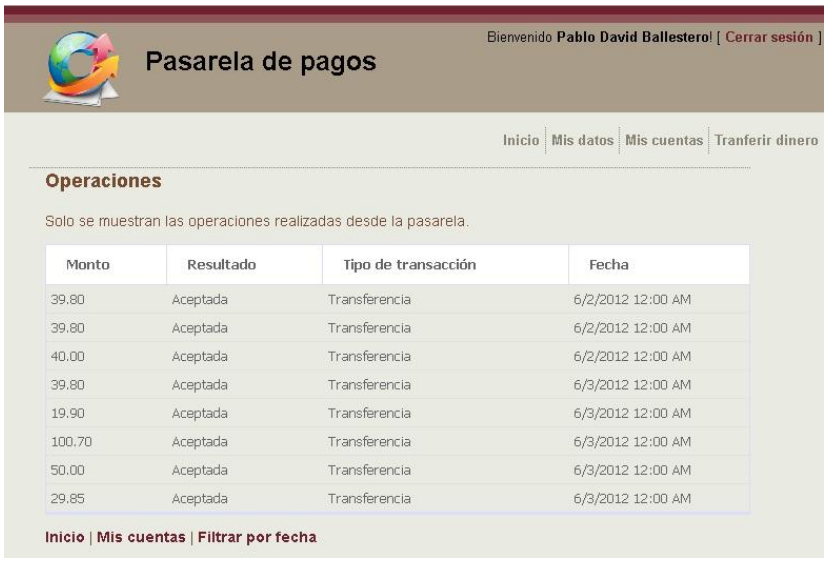



Tabla 9.Descripción de la tarea Mostrar reporte de historial de transacciones.
Fuente: elaboración propia.

Este capítulo propició la elaboración del modelo de dominio permitiendo mantener una visión clara de los conceptos que interactúan en la pasarela de pagos. La especificación de los escenarios y sus tareas divididos por módulos permitió una mejor descripción de las funcionalidades que brinda la pasarela de pagos. En el módulo “Administración” se identificaron un total de 4 escenarios incluyendo en los más extensos un conjunto de tareas, dando un total de 14 de ellas. En el módulo “Administración de clientes” se identificaron 6 escenarios y 7 tareas específicamente. La planificación por iteraciones de los días estimados para realizar cada módulo facilitó una mejor estimación de la prioridad y del tiempo de los mismos. La descripción de los requisitos de calidad del servicio brindó un esclarecimiento del ambiente de desarrollo y buen funcionamiento de la pasarela de pagos.

# Capítulo 3: Desarrollo y estabilización.

El presente capítulo representa el más importante y extenso de la investigación. Se desarrollan las fases de desarrollo y de estabilización del producto. La primera fase recoge los diagramas necesarios para la correcta comprensión del lector sobre el funcionamiento de la pasarela de pagos. En la fase de estabilización se realizan las pruebas de calidad del software para demostrar su correcto funcionamiento.

## 3.1 Fase de Desarrollo

En esta fase se describe la arquitectura y patrones empleados para la ejecución de la pasarela de pagos definiéndose la vista lógica de la arquitectura. Se modelan los diagramas de aplicación y lógico de centro de datos exigidos por la metodología establecida. Se construye el modelo de datos con las clases persistentes que componen la base de datos. También se elabora el diagrama de clases de la pasarela de pagos y se establecen los estándares de codificación utilizados.

### 3.1.1 Arquitectura del sistema

Sobre la arquitectura de software se puede decir que el surgimiento está dado por el artículo de Dewayne Perry y Alexander Wolf en 1992 (*Foundations for the study of software architecture*). Es previa al diseño y a la implantación del código.

Existe una definición oficial de arquitectura  a partir del año 2000 de la IEEE 1471-2000 en la que cita:

La arquitectura de software es la organización fundamental de un sistema encarnada en sus componentes, las relaciones entre ellos y el ambiente y los principios que orientan su diseño y evolución.[41]

**Estilo arquitectónico utilizado**

La arquitectura en tres capas es uno de los estilos arquitectónicos tradicionales más usados a la hora de establecer cómo estará estructurado el software. Se representa el sistema en tres capas: presentación, lógica de negocio o negocio y acceso a datos. La primera capa permite que el usuario pueda observar en su ordenador los datos que se muestran. En la capa de negocios es donde está la lógica, se reciben las peticiones del usuario, y tras ejecutar una acción se le envía las respuestas del proceso. Desde la última capa es de donde se accede a los datos, haciendo referencia a un gestor de BD que realiza el almacenamiento, modificación y consulta de los datos.

El uso de la arquitectura en tres capas permite que el sistema se pueda descomponer en varios niveles de abstracción. Además facilita la evolución del sistema, ya que los cambios sólo deben de afectar a la capa donde se encuentre la modificación. [41]

**Vista lógica de la arquitectura**

La arquitectura de la pasarela de pagos está representada por 3 capas lógicas, lo que permite disminuir al máximo el acoplamiento, aumentar la reutilización entre las mismas, facilitar la modularidad, reusabilidad, el cambio y la portabilidad. Esta distribución de capas permite que se realicen grandes cambios sin tener que realizar alteraciones en las demás capas. Al tener las capas una correcta definición, la comunicación entre ellas se realizará sólo a nivel de interfaces, permitiendo trabajar de manera transparente a las instancias reales.

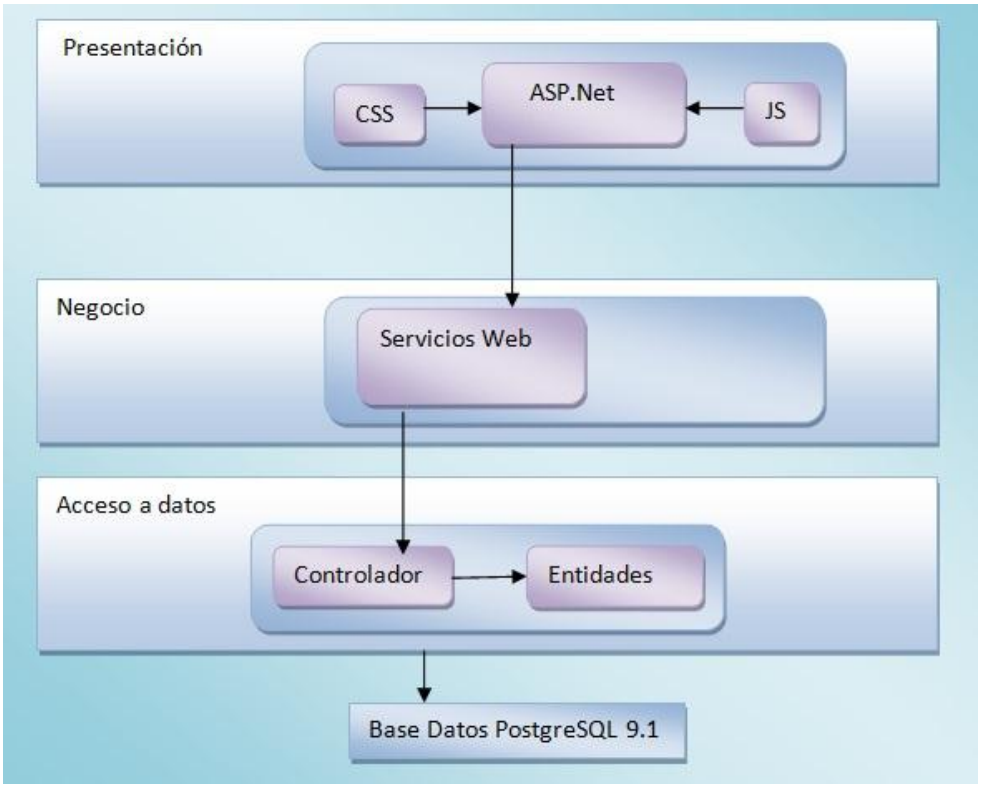



Figura 3. Diagrama de la vista lógica de la arquitectura. Fuente: elaboración propia.

En la figura 3 se muestra la vista lógica de la arquitectura de la pasarela de pagos. La capa de presentación contiene las interfaces de usuario y los componentes para su correcto funcionamiento. Estos elementos son ficheros *JavaScript* y CSS. Además de contar con un proyecto web desarrollado con el *framework* ASP.NET MVC en su versión 2.0, teniendo comunicación directa con la capa de negocio. Con su utilización se aprovechan en esta capa las ventajas del patrón MVC.

En la capa de negocio se recogen las funcionalidades necesarias para darle solución a los requerimientos definidos estableciendo los servicios web que brinda la pasarela de pagos. Las funcionalidades se encuentran definidas según el contexto en el que se desenvuelven. Contiene las clases controladoras, que son las que manejan todas las operaciones sobre las entidades del dominio definidas. Además se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de acceso a datos, para solicitar al gestor de base de datos almacenar o recuperar datos de él.

En la capa de acceso a datos se encuentran un conjunto de librerías que permiten la directa relación con las funcionalidades definidas en el dominio. Para hacer posible esta relación se utilizaron las clases interfaces y controladoras que definen a la capa de negocio, posibilitando de esta forma que se puedan hacer cambios en esta capa sin afectar a las demás capas. Su función principal es realizar la implementación de las interfaces y trabajar al mismo tiempo directamente con la fuente de datos establecida.

Por último la capa de acceso a datos se comunica con la base de datos de la pasarela de pagos la cual está constituida por todo el conjunto de tablas y procedimientos que permiten el almacenamiento de la información recogida y procesada.

### 3.1.2 Patrones de diseño

Un patrón de diseño es una descripción de clases cuyas instancias colaboran entre sí. Cada patrón es adecuado para ser adaptado a un cierto tipo de problema y permite que algunos aspectos de la estructura del sistema puedan cambiar independientemente de otros aspectos. Con el uso de los patrones se facilitan la reusabilidad, extensibilidad y mantenimiento de los sistemas. En la creación y manipulación de las clases de implementación de la pasarela de pagos se utilizaron algunos patrones de asignación de responsabilidades.

Los patrones GRASP (*General Responsibilit y Assignment Software Patterns*) son principios, técnicas o buenas prácticas para mejorar los diseños orientados a objetos. [40]

El patrón creador se utilizó cuando se crearon las instancias de las clases. Este patrón contribuye a identificar quién debe ser el responsable de la creación (o instanciación) de nuevos objetos o clases.

El patrón experto se usó en las clases encargadas de realizar operaciones sobre otras; son los objetos que tienen el acceso a la información necesaria para la ejecución de las tareas.

El patrón bajo acoplamiento está presente también en el desarrollo del proyecto, pues al contar con una arquitectura en tres capas se aprovechan sus ventajas. Las clases interfaz “IUsuarioControl”, “ITransaccionControl”, ”ICuentaControl”, “ISolicitudPagoControl” e “IEntidadBancariaControl” delegan sus responsabilidades en las clases controladoras “UsuarioControl”, “TransaccionControl”, ”CuentaControl”, “SolicitudPagoControl” y “EntidadBancariaControl” respectivamente.

El patrón alta cohesión se emplea en las clases “Encriptador”, “MailSender”, “ISOMessage” y “PuntoSalida”, estas clases realizan pocas funcionalidades y no siempre tienen que ser ejecutadas.

El patrón controlador se representa en todas las clases controladoras implementadas: “UsuarioControl”, “TransaccionControl”, “CuentaControl” y “EntidadBancariaControl”. Estas clases se encargan de realizar las tareas necesarias para el correcto funcionamiento de la pasarela de pagos.

### 3.1.3 Diagrama de aplicación

A continuación se muestra el diagrama de aplicación, es uno de los diagramas indicados por la metodología utilizada. El mismo define los componentes que se relacionan con la pasarela de pagos y sus conexiones. El diagrama muestra un navegador por medio del cual los usuarios interactúan con la pasarela. Además existe otra aplicación que es la tienda virtual donde los usuarios ejecutan sus transacciones y a la vez esta tienda mantiene conexión con la pasarela a través de las API (*Application Programming Interface*). Son métodos que el desarrollador de cualquier aplicación ofrece a otros desarrolladores para que distintas aplicaciones puedan interactuar con su aplicación.[43] Es decir que mediante las API de la tienda virtual el cliente accede mediante un servicio web de manera segura a la pasarela de pagos. Esta a su vez se conecta a la base de datos que almacena la información manejada en ella. Por otra parte se encuentra la aplicación de Administración de la pasarela de pagos que accede a la base de datos para consultar los mismos dependiendo de las acciones a ejecutar.

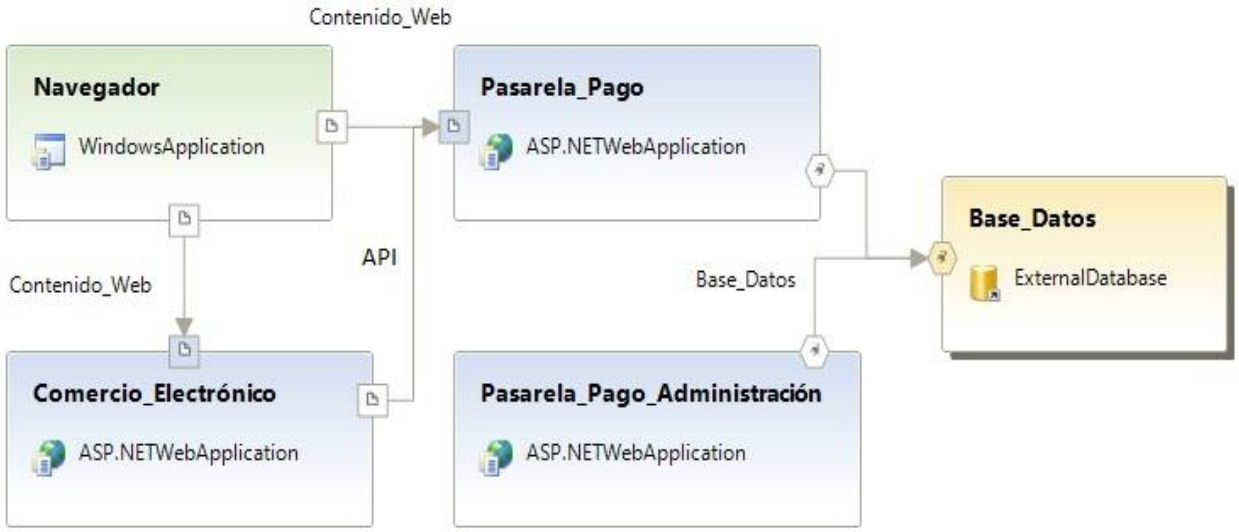



Figura 4. Diagrama de aplicación. Fuente: elaboración propia.

### 3.1.4 Diagrama lógico de centro de datos

El diagrama lógico de centro de datos es otro de los diagramas establecidos a modelar por la metodología MSF for ASD. Conforma las configuraciones de los servidores involucrados en la conexión con la pasarela de pagos. La figura muestra a un cliente (página web desde donde el usuario practica comercio electrónico) que se conecta por un modo seguro al servidor de aplicación de la pasarela de pagos. Al mismo tiempo el servidor de aplicación de la pasarela de pagos tiene conexión directa con el servidor de base de datos de la misma. Este último es consultado por el servidor de Administración de la pasarela de pagos. Para las conexiones entre el servidor de aplicación, el servidor de base de datos y el servidor de Administración se utilizó el protocolo seguro de transporte TCP/IP. Además el servidor de aplicaciones de la pasarela de pagos se comunica a través de un canal seguro con el banco, utilizando SSL de 128 bits, el cual garantiza la integridad de los datos que viajan. Este banco es el que maneja las cuentas de los

usuarios. El protocolo de transporte utilizado es TCP/IP y los mensajes que se intercambian entre estos es por medio de la norma de intercambio de datos electrónicos ISO 8583.

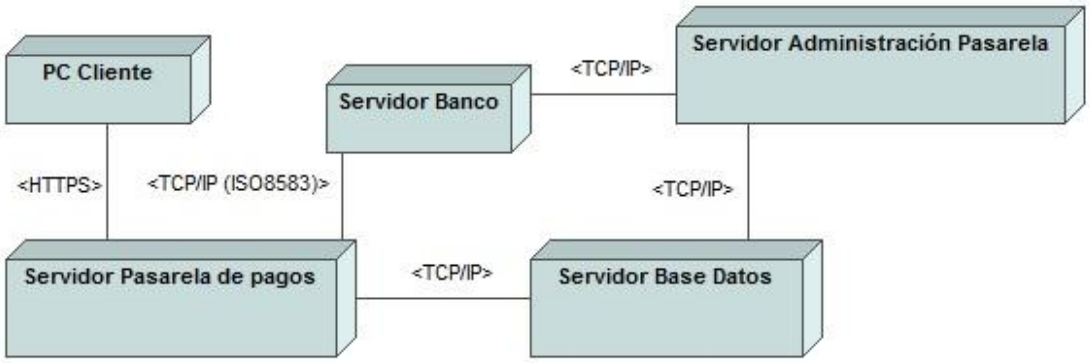



Figura 5. Diagrama lógico de centro de datos. Fuente: elaboración propia.

### 3.1.5 Modelo de datos

Es el conjunto de conceptos que permiten describir los datos, las relaciones que existen entre ellos, la semántica, las restricciones de integridad o condiciones que los datos cumplen para reflejar la correcta realidad; y las operaciones de manipulación de los datos (insertar, borrar, modificar y recuperar los datos). Se utiliza para describir la representación lógica y física de la información persistente manejada por la pasarela de pagos.[44]

El siguiente diagrama representa el modelo de datos de la pasarela de pagos que como antes se ha mencionado se han aprovechado todas las potencialidades del SGBD PostgreSQL.

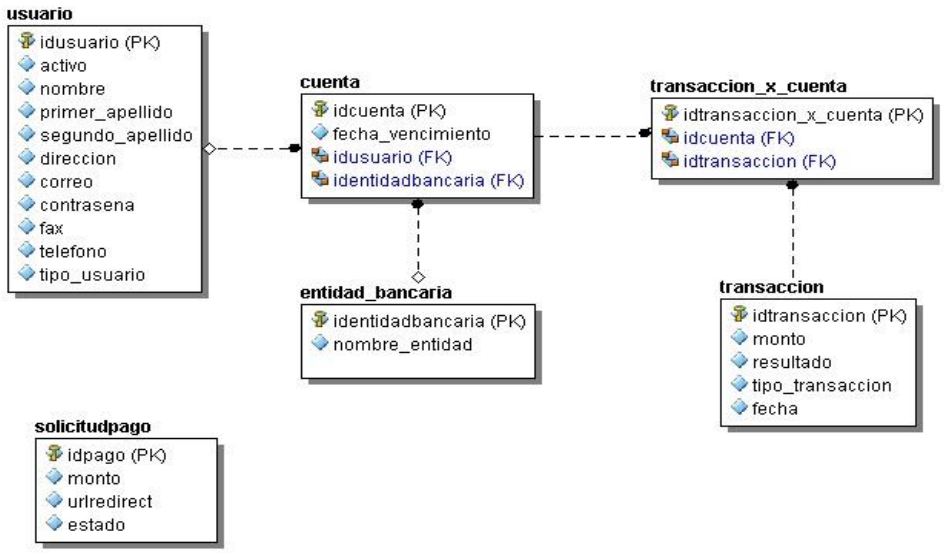



Figura 6. Modelo de datos. Fuente: elaboración propia**.**

**Descripción de las clases persistentes**

En esta sección se describirán las clases persistentes que presenta el modelo de datos de la pasarela de pagos mostrado anteriormente. Las clases persistentes se definen para conocer la información real representada en las tablas de la base de datos. Las siguientes tablas describen

las clases del modelo de datos "*usuario"* y "*transaccion"*. Las demás se encuentran en el Anexo 2.

| **Nombre** | | usuario |
|---|---|---|
| **Descripción** | | Contiene todos los datos de los usuarios. |
| **Atributo** | **Tipo** | **Descripción** |
| id_ usuario | string | Identificación del usuario. |
| contrasena | string | Contraseña del usuario. |
| fax | string | Dato del fax (opcional). |
| telefono | string | Número de teléfono. |
| correo | string | Dirección de correo electrónico. |
| direccion | string | Dirección particular. |
| nombre | string | Nombre propio. |
| primer_apellido | string | Primer apellido. |
| segundo_apellido | string | Segundo apellido. |
| activo | boolean | Indica si el usuario existe en la pasarela. |
| tipo_usuario | string | Define los permisos que tiene los usuarios para accionar en la pasarela. |

Tabla 10. Clase persistente:"usuario". Fuente: elaboración propia.

| **Nombre** | | transaccion |
|---|---|---|
| **Descripción** | | Contiene todos los datos de las transacciones. |
| **Atributo** | **Tipo** | **Descripción** |
| idtransaccion | string | Identificación de la transacción. |
| monto | double | Cantidad de efectivo a pagar o transferir. |
| fecha | DateTime | Fecha en la que se realizó la transacción. |
| resultado | bool | Resultado de la transacción (si se efectúo o no) . |
| tipo_transaccion | string | Tipo de transacción (pago o transferencia). |

Tabla 11. Clase persistente: "transaccion". Fuente: elaboración propia.

### 3.1.6 Descripción de las clases controladoras

Se describen las clases controladoras con las funcionalidades establecidas para el funcionamiento de la pasarela de pagos. Las restantes descripciones se especifican en el Anexo 3.

| **Nombre:**TransaccionControl | | |
|---|---|---|
| **Tipo de clase:** Controladora | | |
| **Descripción:** Controla las funcionalidades para realizar transacciones. | | |
| **Atributo:-** | **Tipo:-** | **Descripción:-** |
| **Métodos:** | **Descripción:** | |
| TransaccionDadoId(string id) | El método dado un identificador devuelve una transacción. | |

| | |
|---|---|
| RealizarPago(string cuentaOrigen, string idtienda, double monto, string pin, string ccv, string fechavenc) | El método dado los parámetros de entrada realiza un pago desde una cuenta hacia otra. |
| TransferirDinero(string cuentaOrigen, string cuentaDestino, string pin, string ccv, double monto, string fechVenc) | El método dado los parámetros de la cuenta destino y los datos necesarios de la cuenta origen, realiza una transferencia de efectivo. |
| Adicionartransaccion(string cuenta, double monto, bool resultado, string tipoTransaccion, DateTime fecha) | El método adiciona una transacción. |
| InterpretarAuthResponseNumero(string[] message) | El método devuelve la respuesta dada por el banco del mensaje que le fue enviado sobre la validación de la cuenta. |
| InterpretarFinancialResponse(string[] message) | El método devuelve la respuesta dada por el banco del mensaje que le fue enviado sobre el resultado de la transacción. |
| VerificarCCV(string cuenta, string ccv) | El método verifica si es correcto el CCV dado este dato y el número de cuenta. |
| InterpretarAuthorizationResponseSaldo(string[] message) | El método devuelve el resultado del saldo correspondiente a la cuenta del cliente. |
| InterpretarISOMessage(string message) | El método devuelve la respuesta de los datos que se envían y reciben del banco. |

Tabla 12.Clase controladora: “TransaccionControl”. Fuente: elaboración propia.

| **Nombre:**UsuarioControl | | |
|---|---|---|
| **Tipo de clase:** Controladora | | |
| **Descripción:** Controla las funcionalidades para gestionar usuarios. | | |
| **Atributo:-** | **Tipo:-** | **Descripción:-** |
| **Métodos:** | **Descripción:** | |
| TestConnection() | El método establece la conexión con la base de datos. | |
| ExisteUsuario(string email) | El método dado el correo devuelve si existe o no el usuario que se corresponde con ese dato. | |
| AdicionarUsuario (string nombre, string primer_apellido, string segundo_apellido, string direccion, string correo, string fax, string telefono, string tipo_usuario, bool activo, string contrasena) | El método inserta los datos establecidos para adicionar un usuario. | |
| ListarUsuariosAdmin() | El método muestra un listado de usuarios que juegan el rol de administrador. | |
| ActualizarUsuario(string idusuario, string nombre, string primer_apellido, string segundo_apellido, string direccion, string correo, string fax, string telefono, bool activo) | El método actualiza los datos de un usuario. | |

| | |
|---|---|
| EliminarUsuarioAdmin(string id) | El método dado un identificador elimina un usuario de administración. |
| UsuarioDadoId(string id) | El método dado un identificador devuelve los datos del usuario (administrador) que se corresponde con ese dato. |
| UsuarioDadoCorreo(string correo) | El método dado el correo electrónico devuelve los datos del usuario (administrador) que se corresponde con ese dato. |
| AutenticarUsuario(string correo, string password) | El método permite autenticar a un usuario como administrador en el sistema. |
| CambiarContrasena(string nombre, string oldpassword, string newpassword) | El método dado el dato correo, la contraseña vieja y la contraseña nueva permite actualizar la misma. |
| UsuarioDadoNombreApellidos(string correo) | El método dado el correo devuelve el nombre y los apellidos del usuario que se corresponde con ese dato. |
| FechasTransacciones(List<transaccion> lista, DateTime desde, DateTime hasta) | El método muestra un listado con las transacciones que se efectuaron entre un rango de fechas insertados. |
| ListarClientesActivos() | El método devuelve un listado de usuarios que son clientes y están activos en el sistema. |
| ListarClientes() | El método devuelve un listado de los usuarios que son clientes. |
| ClienteDadoId(string id) | El método dado un identificador devuelve el cliente activo que se corresponde con ese dato. |
| ClienteDadoCorreo(string correo) | El método dado el correo devuelve el cliente que se corresponde con ese dato. |
| EliminarCliente(string id) | El método dado un identificador desactiva la cuenta del cliente con todos sus datos. |
| CuentasCliente(string id) | El método dado un identificador muestra un listado con todas las cuentas del cliente que se corresponde con ese dato. |
| CuentasCliente(string idtienda) | El método devuelve la cuenta del vendedor. |
| TransaccionDadoId(string id) | El método devuelve la transacción que se corresponde con el identificador recibido. |
| TransaccionesXCliente(usuario u) | El método dado un cliente devuelve un listado de todas las transacciones realizadas por este. |
| BuscarCliente(string criterio, string datos) | El método devuelve un listado de clientes que se corresponden con el criterio de búsqueda. |
| AutenticarCliente(string correo, string pass) | El método permite autenticar a un cliente. |
| AdicionarCliente(string nombre, string primer_apellido, string segundo_apellido, string direccion, string correo, string fax, string telefono, string tipo_usuario, bool activo, string contrasena) | El método inserta los datos de un cliente, añadiéndolo al listado de clientes del sistema. |
| CheckPaswwordHistory(string pass, usuario u) | El método chequea el historial de contraseñas del cliente. |
| GetPasswordHistory(string passwordHistory) | El método devuelve el historial de contraseñas del cliente. |
| ConfigurarPago(string correo, string idcuenta) | El método crea un identificador de la tienda virtual para consumir el servicio de pago. |
| ExisteIdTienda(string idtienda) | El método devuelve si existe o no la tienda con el identificador entrado por parámetro. |

Tabla 13.Clase controladora: “UsuarioControl”. Fuente: elaboración propia.

### 3.1.7 Estándares de codificación utilizados

Las convenciones o estándares de codificación son un conjunto de directrices que especifican cómo debe escribirse el código fuente. Algunos ejemplos de directrices que se encuentran son aquellas que pautan el nombrado de variables, clases y estructuras de control o incluso qué información incluir en los comentarios.[45]

La estrategia de codificación utilizada son las guías de diseño de la plataforma .Net. Estas son convenciones de nomenclatura plasmadas en la plataforma. Para la declaración de las clases entidades se comienza con minúscula, las demás clases siguen el estándar de comenzar con mayúscula y si el nombre es compuesto se mantendría la mayúscula. Para la declaración de atributos se comienza con guión bajo seguido del nombre con letra minúscula. Las propiedades se escriben con mayúsculas y si son compuestas las demás palabras van con mayúsculas también. Los métodos igualmente, comienzan con mayúscula y las variables auxiliares con minúsculas.

El uso de estrategias de codificación garantiza un mejor entendimiento a la hora de leer el código. Asegura la reutilización del código y la eficacia en el trabajo, pues de ser necesaria la incorporación de nuevos integrantes del equipo de desarrollo, no le costaría entender el código existente.

### 3.1.8 Tratamiento de excepciones

El tratamiento de errores o excepciones, como también se le conoce, es uno de los elementos más importantes que se deben tener en cuenta en la implementación de un sistema. Una excepción es un evento que ocurre durante la ejecución de un programa y detiene el flujo normal de la secuencia de instrucciones de ese programa; es una condición anormal que surge en una secuencia de código durante su ejecución. Las excepciones interrumpen el procesamiento normal porque no cuentan con la información necesaria para resolver el problema, en el contexto en que sucedió.[46]

Para evitar que se produzcan excepciones es necesario realizar validaciones en todos los módulos que se estén implementando. Es muy importante tener en cuenta que se debe validar la captura de información tanto del lado del cliente como en el servidor. Con la utilización de ASP.Net se le puede dar una mejor visión al usuario para capturar información, además de ahorrar tiempo, también hay que tener en cuenta que el servidor es el responsable de validar todo lo que recibe, por lo que se deben hacer validaciones en el mismo y mostrar los errores ocasionados.

En la implementación de los módulos Administración, Administración de usuario y Administración de transacciones se hacen validaciones que permiten controlar la entrada de datos del usuario, además de controlar la realización de operaciones válidas. Estas validaciones

fueron hechas tanto del lado del cliente como del lado del servidor. A nivel de programación cuando el usuario realiza sobre la aplicación una acción no válida, automáticamente se genera un mensaje de error, mensaje que es mostrado al usuario indicando cuál fue el error cometido y cómo puede solucionarlo. También a nivel de interfaz gráfica se ha validado que los datos entrados sean del tipo requerido, además de validar que no queden campos de texto vacíos y que no se envíe al servidor información innecesaria. Si se violan alguna de estas restricciones el usuario de alguna manera es advertido, ya sea a través de mensajes o utilizando las bondades que proporciona ASP.Net en este sentido.

## 3.2 Fase de Estabilización

En esta fase se definen las pruebas de calidad del software. Al software se le realizaron distintas pruebas empleando los métodos de caja blanca y caja negra.

### 3.2.1 Pruebas del software

Una prueba es un proceso de ejecución de un programa con la intención de descubrir errores. Un buen caso de prueba es aquel que tiene una alta probabilidad de mostrar un error no descubierto hasta entonces. Una prueba tiene éxito si se descubre un error no detectado hasta entonces.La prueba no puede asegurar la ausencia de defectos; sólo pueden demostrar que hay defectos en cualquier software.[47]

Dentro de cada una de las etapas de desarrollo de un software las pruebas son fundamentales ya que a partir de ellas es posible controlar que los productos cumplan requisitos mínimos de operatividad además de garantizar la calidad de estos productos.

**Métodos de prueba**

- ✓ Pruebas de caja negra: pruebas que se llevan a cabo sobre la interfaz del software. El objetivo es demostrar que las funciones del software son operativas, que las entradas se aceptan de forma adecuada y se produce un resultado correcto, y que la integridad de la información externa se mantiene (no se ve el código).
- ✓ Pruebas de caja blanca: se comprueban los caminos lógicos del software. Se puede examinar el estado del programa en varios puntos para determinar si el estado real coincide con el esperado (sobre el código).

Estos métodos de pruebas se llevaron a la práctica para probar el funcionamiento de la pasarela de pagos. Se usaron dos estrategias de pruebas, las pruebas unitarias hacia las funcionalidades de mayor complejidad, y la estrategia de prueba del sistema, usando la técnica partición de equivalencia donde se validaron las entradas válidas e inválidas de datos utilizando los casos de pruebas.

**Pruebas unitarias.**

La prueba de unidad es la primera fase de las pruebas dinámicas y se realizan sobre cada módulo del software de manera independiente. El objetivo es comprobar que el módulo,

entendido como una unidad funcional de un programa independiente, está correctamente codificado. Las pruebas unitarias se le aplicaron a las funcionalidades de más peso, o sea a los métodos más complejos de la pasarela de pagos. Para ello se utilizó la herramienta VSTS 2010. Se muestra el resultado del método “TransferirDinero”, las demás pruebas unitarias se encuentran en el Anexo 5.

```
public string[] TransferirDinero(string cuentaOrigen, string cuentaDestino, string pin, string ccv, double monto, string fechVenc)
{
    var e = new Encriptador();
    bool aceptada = false;
    bool otra = false;
    string[] retornar = new string[2];
    string[] result = new string[3];
    var isoMessage = new ISOMessage();
    try
    {
        var mensaje = isoMessage.BuildAuthRequest(e.GetPasswordHash(cuentaDestino));
        var aux = PuntoSalida.EnviarISO(mensaje);
        result[2] = PuntoSalida.EnviarISO("HandShake");
        if (result[2] == "No se pudo conectar a {0}:9898 localhost")
        {
            retornar[0] = "Conexión con el banco interrumpida";
            return retornar;
        }
        if (InterpretarAuthResponseNumero(isoMessage.ParseMessage(result[2])) == "OK")
        {
            mensaje = isoMessage.BuildMessageTrans("CuentaOrigen", cuentaOrigen, monto, pin, fechVenc);
            aux = PuntoSalida.EnviarISO(mensaje);
            result[0] = PuntoSalida.EnviarISO("HandShake");
            if (InterpretarFinancialResponse(isoMessage.ParseMessage(result[0])) == "OK")
            {
                retornar[0] = InterpretarFinancialResponse(isoMessage.ParseMessage(result[0]));
                aceptada = true;
                var mensaje2 = isoMessage.BuildMessageTrans("CuentaDestino", e.GetPasswordHash(cuentaDestino), monto, "", "");
                aux = PuntoSalida.EnviarISO(mensaje2);
                result[1] = PuntoSalida.EnviarISO("HandShake");
                retornar[1] = InterpretarFinancialResponse(isoMessage.ParseMessage(result[1]));

                if (InterpretarFinancialResponse(isoMessage.ParseMessage(result[1])) == "OK")
                    otra = true;
                AdicionarTransaccion(cuentaOrigen, monto, aceptada, "Transferencia", DateTime.Now);
                AdicionarTransaccion(e.GetPasswordHash(cuentaDestino), monto, otra, "Transferencia", DateTime.Now);
            }
            else
            {
                retornar[0] = InterpretarFinancialResponse(isoMessage.ParseMessage(result[0]));
            }
        }
        else
        {
            var det = isoMessage.ParseMessage(result[2])[38];
            switch (det)
            {
                case "PIN   ":
                    retornar[0] = "PIN incorrecto";
                    break;
                case "DATE  ":
                    retornar[0] = "Fecha de vencimiento incorrecta";
                    break;
                case "WRONG#":
                    retornar[0] = "Numero incorrecto";
                    break;
            }
        }
    }
    catch (Exception ex)
    {
        retornar[0] = ex.Message;
    }
    return retornar;
}
```

| **Prueba Unitaria** | |
|---|---|
| **Nombre de la Prueba** | *TransferirDinero* |

| Estado | Tipo | Ultima Ejecución |
|---|---|---|
| Satisfactoria | Caja Blanca | 10/05/2012 |
| **Ejecutado por** | **Verificado por** | |
| Damaris Solis Fonseca | Wilfredo Roque Pérez | |
| **Descripción** | Para poder ejecutar la prueba se deben pasar varios tipos de datos string para los datos: número de cuenta origen, número cuenta destino, PIN, CCV y fecha de vencimiento y un double para el monto. Con esos datos se realiza la transferencia del monto establecido desde la cuenta origen hacia la destino si todos los datos son válidos. Una vez realizada la transferencia se devuelve el resultado de la misma. En caso de no realizarse muestra un mensaje indicando el error cometido en la operación. | |
| **Entrada** | número de cuenta origen, número cuenta destino, PIN, CCV, monto, fecha vencimiento. | |
| **Criterio de aceptación** | Retorna un arreglo con dos posiciones en cual se encuentra el resultado de la transacción para la cuenta destino y la cuenta origen. | |

**Resultado:**

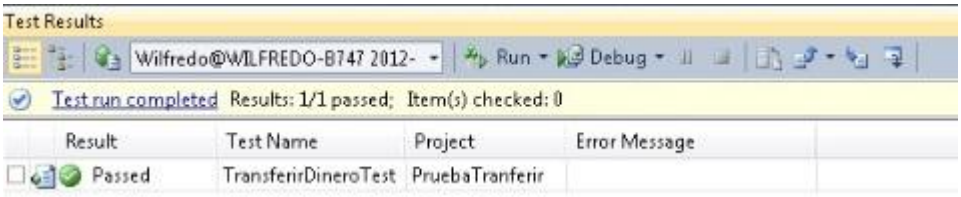



Tabla 14. Descripción de la prueba unitaria al método “TransferirDinero”.

**Resultados de las pruebas unitarias**

Se efectuaron dos iteraciones de pruebas unitarias a los métodos de mayor complejidad de la pasarela de pago, estos pertenecen al módulo Administración de clientes. En una primera iteración se realizaron 5 pruebas de las cuales todas fueron aceptadas. En la segunda iteración se le aplicó la prueba de unidad a 4 funcionalidades de las cuelas ninguna falló. Los resultados se muestran en la tabla 15.

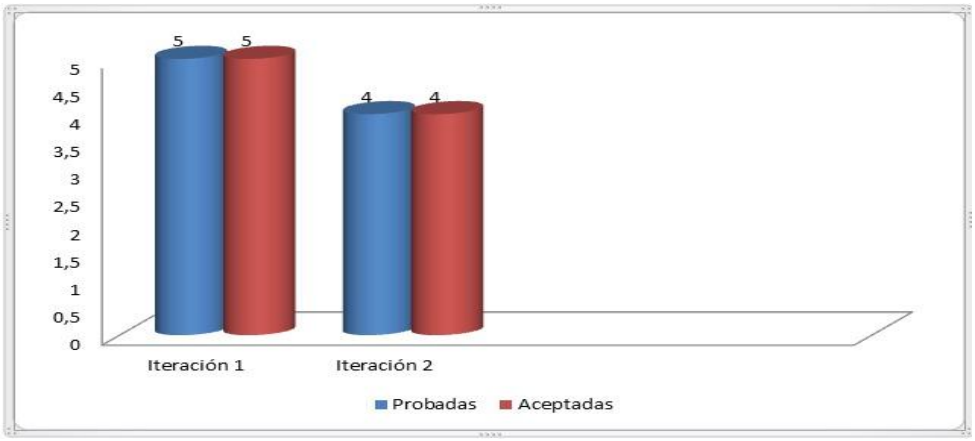



Tabla 15. Resultado de las pruebas unitarias. Fuente: elaboración propia.

**Pruebas de caja negra**

Las pruebas de caja negra a la pasarela de pagos se efectuaron en dos iteraciones mediante la realización de casos de prueba a cada uno de los escenarios y sus tareas correspondientes. Para ello se detallaron las clases válidas e inválidas teniendo en cuenta las entradas de datos para cada una de las interfaces que se muestran.

Las siguientes tablas muestran los casos de pruebas realizados a los escenarios “Registrar cliente”, “Configurar cuenta” y “Autenticar cliente” del módulo Administración de clientes. Además se describe el caso de prueba “Realizar transacción” de este módulo. Las restantes tablas se encuentran en el Anexo 4.

| **Escenario** | **Descripción** | **nombre** | **1er_apellido** | **2do_apellido** | **correo** | **contraseña** | **dirección** | **teléfono** | **fax** | **Respuesta del sistema** | **Flujo central** |
|---|---|---|---|---|---|---|---|---|---|---|---|
| EC 1. Registrar cliente. | Se crea la cuenta del cliente insertando los datos correspondientes. | V wilfredo | V roque | V perez | V wroque | V Pe*p@it7o | V Pinar calle3 %7 y 9 | V 3456 | V 67895 | Se almacenan todos los campos insertados. | 1. Dar clic en la opción “Registrar”.<br>2. Se muestra la interfaz con los campos a llenar.<br>3. Se llenan los campos obligatorios.<br>4. Se pulsa en el botón “Aceptar”.<br>5. Se muestra un mensaje |
| | | V wilfredo | V roque | V perez | V wroque | V Pe*p@it7o | V Pinar calle3 %7 y 9 | V | V | | |
| | | V wilfredo | V roque | V perez | V wroque | V Pe*p@it7o | V Pinar calle3 %7 | V 3456 | V | | |
| | | I | I | I | I | I | I | I | I | Se muestra un mensaje indicando que los | |

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | campos obligatorios no pueden estar vacíos. | notificando que la cuenta se creó correctamente. |
| | | V Wilfredo | V Roque | V Perez | V wroque | I Pe*p@ | V Pinar calle3 %7 y 9 | V 3456 | V 67895 | Se muestra un mensaje indicando que algún campo no cumple con el formato establecido. | |

Tabla 16. Caso de prueba Registrar cliente. Fuente: elaboración propia.

| No | Nombre de campo | Clasificación | Valor Nulo | Descripción |
|---|---|---|---|---|
| 1 | nombre | campo de texto | No | Se aceptan letras minúsculas y mayúsculas. |
| 2 | 1er_apellido | campo de texto | No | Se aceptan letras minúsculas y mayúsculas. |
| 3 | 2do_apellido | campo de texto | No | Se aceptan letras minúsculas y mayúsculas. |
| 4 | correo | campo de texto | No | Se aceptan letras minúsculas y mayúsculas, el carácter @ es obligatorio luego el dominio al que pertenece el nombre del usuario del correo electrónico. |
| 5 | contraseña | campo de texto | No | Se aceptan letras minúsculas, mayúsculas, números, signos y debe exceder de los 6 caracteres. |
| 6 | dirección | campo de texto | No | Letras y números. |
| 7 | teléfono | campo de texto | Sí | Sólo números. Sólo números. |

| | | | | |
|---|---|---|---|---|
| 8 | fax | campo de texto | Sí | |

Tabla 17. Descripción de variables del caso de prueba Registrar cliente.
Fuente: elaboración propia.

| **Escenario** | **Descripción** | **número de tarjeta** | **CCV** | **Fecha validación** | **Respuesta del sistema** | **Flujo central** |
|---|---|---|---|---|---|---|
| Tarea 2. Adicionar cuenta. | Adicionar una cuenta con los datos correspondientes. | V<br>456987125123 | V<br>1254 | V<br>12/14 | Se adiciona una cuenta bancaria del cliente. | 1. Dar clic en la opción “Registrar cuenta”.<br>2. Se muestra la interfaz con todos los campos.<br>3. Se insertan los datos.<br>4. 5. Se pulsa en el botón “Finalizar”.<br>5. Se muestra un mensaje indicando que se creó la cuenta correctamente. |
| | | V<br>4yu987125123 | V<br>1254 | V<br>12/15 | Se muestra un mensaje de error indicando que algún campo no cumple con el formato establecido. | |
| | | I | I | I | Se muestra un mensaje indicando que debe llenar los campos obligatorios. | |
| Tarea 3. Eliminar cuenta bancaria | Eliminar una cuenta bancaria de un cliente. | N/A | N/A | N/A | Se muestra un mensaje informando que se eliminó la cuenta seleccionada. | 1. Dar clic en la opción “Eliminar”.<br>2. Se muestra la interfaz para eliminar la cuenta.<br>3. Se pulsa en |

| | | | | | | |
|---|---|---|---|---|---|---|
| | | | | | | el botón “Eliminar”.<br>4. Se muestra una alerta de verificación de la eliminación.<br>5. Se pulsa en “Aceptar” y se elimina la cuenta. |

Tabla 18. Caso de prueba Configurar cuenta. Fuente: elaboración propia.

| No | Nombre de campo | Clasificación | Valor Nulo | Descripción |
|---|---|---|---|---|
| 1<br>2<br>3 | Número de la tarjeta<br>fecha validación<br>CCV | campo de texto<br>campo de texto<br>campo de texto | No<br>No<br>No | Sólo números.<br>Números con el formato mm/aa<br>Sólo números. |

Tabla 19. Descripción de variables del caso de prueba Configurar cuenta.
Fuente: elaboración propia.

| Escenario | Descripción | correo | contraseña | Respuesta del sistema | Flujo central |
|---|---|---|---|---|---|
| EC 4. Autenticarcliente. | Autenticar cliente insertando los datos establecidos. | V<br>dsfonseca@uci.cu | V<br>46dam*erl | Se autentica el usuario como cliente. | 1. Se muestra una interfaz de autenticación.<br>2. Se insertan el correo y la contraseña.<br>2. Clic en el botón “Iniciar sesión”.<br>3. Se autentica el cliente. |
| | | I<br>Dsfonsec.uci.cu | V<br>DA*34hyl | Se muestra un mensaje informando que el correo o la contraseña están incorrectos. | |
| | | V<br>damaris | I<br>P*A34 | | |

Tabla 20.Caso de prueba Autenticar cliente. Fuente: elaboración propia.

| No | Nombre de campo | Clasificación | Valor Nulo | Descripción |
|---|---|---|---|---|

| | | | | |
|---|---|---|---|---|
| 1 | correo | campo de texto | No | Se aceptan letras minúsculas y mayúsculas, el carácter @ es obligatorio luego el dominio al que pertenece el nombre del usuario del correo electrónico. |
| 2 | contraseña | campo de texto | No | Se aceptan letras minúsculas, mayúsculas, números, signos y debe exceder de los 6 caracteres. |

Tabla 21. Descripción de variables del caso de prueba Autenticar cliente.
Fuente: elaboración propia.

| **Escenario** | **Descripción** | **Cuenta destino** | **PIN** | **Monto** | **Respuesta del sistema** | **Flujo central** |
|---|---|---|---|---|---|---|
| EC 5. Realizar transferencia | Se realiza envío de efectivo desde una cuenta origen hacia una cuenta destino. | V 459871236547 | V 4561 | V 4527 | Se realiza la transferencia del monto a la cuenta destino. | 1. Clic en realizar transferencia.<br>2. Se muestra la interfaz para ejecutar la acción.<br>3. Se introducen los datos establecidos.<br>4. Se codifican los datos.<br>5. Se envían dichos datos, oprimiendo el botón “Enviar”. |
| | | V459871236547 | I 34ddfff | V 4527 | Se muestra un mensaje de error notificando que algún campo está escrito incorrectamente. | |
| | | V459871236547 | V 4561 | I 456ddd | | |
| | | I Ffhfjfkfkf789 | V 4561 | V 4527 | | |
| | | I | I | I | Se muestra un mensaje informando que debe llenar los campos obligatorios. | |

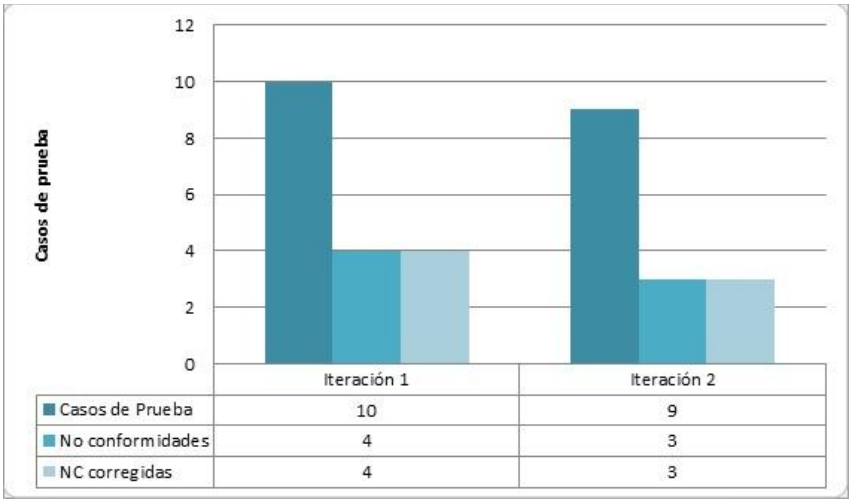
Tabla 22. Caso de prueba Realizar transferencia. Fuente: elaboración propia.

| **No** | **Nombre de campo** | **Clasificación** | **Valor Nulo** | **Descripción** |
|---|---|---|---|---|
| 1 | cuenta destino | Campo de texto. | No | Sólo números. |
| 2 | PIN | Campo de texto. | No | Sólo números. |
| 3 | monto | Campo de texto. | No | Sólo números. |

Tabla 23. Descripción de variables del caso de prueba Realizar transferencia.
Fuente: elaboración propia.

### Resultados de las pruebas de caja negra

En las pruebas de caja negras realizadas, de los 10 escenarios y las 22 tareas existentes, se identificaron 19 casos de pruebas, constituyendo estos las interfaces que requieren entrada de datos. En la primera iteración se efectuaron 10 casos de pruebas detectándose 4 no conformidades a las cuales se les dio solución. En la segunda y última iteración se detectaron 3 no conformidades de los 9 casos de prueba restantes, las mismas fueron resueltas en su totalidad. En las dos iteraciones efectuadas se detectaron un total de 13 no conformidades, las cuales en su mayoría respondían a errores de bajo impacto en el correcto funcionamiento de la aplicación y todas tuvieron solución un tiempo máximo de 2 días, lo que indica que la pasarela de pagos desarrollada presenta buena calidad.




| | Iteración 1 | Iteración 2 |
|---|---|---|
| Casos de Prueba | 10 | 9 |
| No conformidades | 4 | 3 |
| NC corregidas | 4 | 3 |

Tabla 24. Resultados de las pruebas. Fuente: elaboración propia.

En este capítulo la definición de las vistas de la arquitectura de la pasarela de pagos en el desarrollo de la misma propició la comprensión de las funcionalidades, servicios y conexiones para su correcto desarrollo. El uso de los patrones de diseño, estándares de código, manejo de excepciones y la descripción de las clases de programación facilitaron en gran medida la implementación de la pasarela de pagos. El modelado de la base de datos permitió la representación lógica y física de la información que se almacena en la pasarela. Mediante los casos de prueba y las pruebas unitarias se probaron las funcionalidades, identificándose un conjunto de no conformidades que fueron resueltas en las iteraciones realizadas. Este resultado demostró que la solución presenta un buen funcionamiento.

# Conclusiones

- La aplicación de los métodos teóricos y el análisis bibliográfico permitieron concretar el marco teórico en correspondencia con el proceso de transacciones bancarias que se gestionan mediante pasarelas de pagos.
- El análisis de las pasarelas de pagos estudiadas permitió definir parte de las funcionalidades en el desarrollo de la pasarela de pagos que ejecuta transferencias y pagos bancarios *online* de manera segura.
- La pasarela de pagos favorece la ejecución de transferencias y pagos bancarios *online* garantizando los niveles de seguridad de la información, reduciendo a un mínimo el tiempo de respuesta.
- El análisis de los mecanismos de seguridad en transacciones de comercio electrónico permitió crear una aplicación de pago para garantizar la confianza en el uso de su solución.
- El estudio de las tecnologías propuestas permitió la utilización de lenguajes, herramientas y metodología de avanzada para el desarrollo de la pasarela de pagos.
- La solución se desarrolló en el período de tiempo establecido y cumple con todas las funcionalidades que se describieron en la Fase de Planeación.
- Se realizó el diseño de la pasarela de pagos logrando un mayor entendimiento de la misma para su posterior implementación.
- Se validó la solución empleando los métodos de pruebas de caja negra y caja blanca arrojándose resultados satisfactorios como base para reconocer que la solución está en condiciones para ser liberada.

  Por todo lo anterior expuesto se concluye que las tareas investigativas facilitaron una adecuada ejecución de la investigación.

# Recomendaciones

Una vez concluida la investigación se recomienda:

- Crear nuevos escenarios en el módulo Administración de transacciones con otras operaciones que además del pago y transferencia se pueden efectuar desde una pasarela de pagos hacia los bancos.
- Dar continuidad a este trabajo en función de mejorar y ampliar los servicios brindados por la pasarela de pagos. Se sugiere además: probar la pasarela de pagos en un entorno real, realizar la configuración de los protocolos de comunicación de los bancos con los cuales intercambiará información la pasarela y generar una factura electrónica en formato pdf de todas las transacciones ejecutadas sobre la misma, para garantizar el no repudio de los datos.

# Referencias Bibliográficas


1. i Tarrats, J.B. and F. Jordán, 2000, LA SEGURIDAD DE LAS TRANSACCIONES BANCARIAS EN INTERNET. Informes SEIS, p. 133.
2. www.tecnologiapyme.com, [citado 7-11-2011].
3. Anónimo, (2001), SITUACIÓN ACTUAL Y PERSPECTIVAS DEL COMERCIO ELECTRÓNICO EN LA REGIÓN.
4. Eduardo Berrocal, R.D., 2009, Juan Giménez, Manuel Sala, Nacho Somalo, Libro Blanco del Comercio Electrónico.
5. Texto Unico Ordenado de la Ley del Impuesto a la Renta - Decreto Legislativo N° 774, artículo 5°.
6. www.alfa-redi.org, Revista de Derecho Informático [citado 10-11-2011].
7. Tellez, 2009, La Seguridad del entorno tecnológico y en las Transacciones para el Comercio Electrónico.
8. Anónimo, 2007, EDI. Tecnología de la información. Código de buenas prácticas para la gestión de la seguridad de la información.
9. Pakala., S., 2005, The Open Web Application Security Project (OWASP). The Open Web Application Security Project (OWASP).
10. Bonastre, J.A.L., Seguridad en el comercio electrónico.
11. Loarte, A.S., 2005, Normas y procedimientos técnicos para garantizar la seguridad. (de la información publicada por las Entidades de la Administracion Pública peruana a través de Internet).
12. López, M.J.L., Criptografía y Seguridad en Computadores.
13. technet.microsoft.com, 2011, Guía de planeación para el cumplimiento del Estándar de seguridad de datos en la industria de tarjetas de pago. [citado 30-4-2012].
14. Anónimo, 2010, Requisitos y procedimientos de evaluación de seguridad.
15. Anónimo, 2011, APOYO DIGITAL S.A.C.
16. Kumar, S., 2010, Code project.
17. www.paypal.com, [citado 10-12-2011].
18. www.apoyodigital.com.pe, ebusiness/2checkout.html. [citado 10-12-2012].
19. epagado.com, [citado 15-12-2012].
20. www.alertpay.com, [citado 12-12-2012].
21. solpagos.com, [citado 14-12-2012].
22. Anónimo, 2007, PROYECTO PARA EL PAGO DE OPERACIONES DE COMERCIO ELECTRONICO CON TARJETAS MAYORISTAS CE-LINK.
23. Isaías Carrillo Pérez, 2008, METODOLOGIA DE DESARROLLO DEL SOFTWARE.
24. Anónimo, Guía de MSF for Agile Software Development.

25. www.hhsistemas.com.ar, C#, [citado 17-12-2012].
26. http://msdn.microsoft.com, [citado 16-12-2011].
27. www.w3c.es, [citado 28-12-2012].
28. es.html.net, [citado 8-12-2012].
29. www.maestrosdelweb.com, [citado 5-1-2012].
30. msdn.microsoft.com, [citado 9-1-2012].
31. www.ibiblio.org, [citado 7-1-2012].
32. Gutiérrez., J.J. ¿Qué es un framework web?
33. www.desarrolloweb.com, [citado 15-1-2012].
34. msdn.microsoft.com, [citado 12-12-2012].
35. Gastelú, P.D.C.A.T., 2011, SMDB:(Sistemas manejadores de base de datos).
36. microbuffer.wordpress.com, que-es-postgresql (2011), [citado 19-12-2011].
37. www.postgresql.org, [citado 19-12-2011].
38. www.altova.com, [citado 5-1-2012].
39. Anónimo. 2009, ER/Studio Modelado de datos empresarial. [citado 12-12-2011].
40. Craig Larman, 2003, UML y Patrones.
41. Pressman, R., 1997, Ingeniería de software. Un enfoque práctico.
42. blogdeaitor.wordpress.com, 2008, model-view-controller. [citado 15-2-2012].
43. google.dirson.com, [citado 7-5-2012].
44. Rivera, J.F., [citado 13-3-2012].
45. www.utopicainformatica.com, convenciones estándares de codificación. 2010 [citado 20-3-2012].
46. Martínez, R.S.Q., 2007, Manejo de Excepciones.
47. Pressman, R., Un enfoque práctico. Capítulo 17 (Técnicas de Software) de la página 281-304 y Capítulo 18 (Estrategias de Pruebas de Software) de la página 305-322.

# Bibliografía


1. Texto Único Ordenado de la Ley del Impuesto a la Renta - Decreto Legislativo N° 774, artículo 5.
2. Lucena López Manuel J., 2001, Criptografía y Seguridad en Computadores Tercera Edición.
3. Ahmed M, Garrett C, Faircloth J., 2002, Web Developer 's Guide.
4. Ahon, El Comercio Electrónico en América Latina. Realidades y Perspectivas.
5. Alfaro FM., ¿Qué es el Comercio Electrónico?
6. Amaro Calderón SDVRJC, 2007, Metodologías Ágiles.
7. Anónimo, Guía de MSF for Agile Software Development.
8. Anónimo, 2001, SITUACIÓN ACTUAL Y PERSPECTIVAS DEL COMERCIO ELECTRÓNICO EN LA REGIÓN.
9. Anónimo, 2005, NTP-ISO/IEC 17799. Comisión de Reglamentos Técnicos y Comerciales peruano -INDECOPI.
10. Anónimo, 2007, PROYECTO PARA EL PAGO DE OPERACIONES DE COMERCIO ELECTRONICO CON TARJETAS MAYORISTAS CE-LINK.
11. Anónimo, 2007, Electronic Banking. Microsoft Dynamics GP.
12. Anónimo, 2009, ER/Studio Modelado de datos empresarial.
13. Anónimo, 2011, APOYO DIGITAL S.A.C.
14. Anónimo, 2007, EDI. Tecnología de la información. Código de buenas prácticas para la gestión de la seguridad de la información.
15. Anónimo, 2010, Requisitos y procedimientos de evaluación de seguridad.
16. Bipin Joshi PD, Kevin Hoffman, Professional ADO.NET Programming.
17. blogdeaitor.wordpress.com, 2008, model-view-controller, [citado 15-2-2012].
18. Bonastre JAL, Seguridad en el comercio electrónico.
19. Craig Larman PH., 2003, UML y Patrones.
20. Eduardo Berrocal RD, Juan Giménez,Manuel Sala,Nacho Somalo, 2009, Libro Blanco del Comercio Electrónico.
21. emadrid2011.es, [citado 26-12-2011].
22. epagado.com, [citado 15-1-2012].
23. es.html.net, [citado 8-1-2012].
24. Fiedler MS.,2003, El E-Banking: Una Realidad Financiera.
25. G. Schneider., 2006, Commerce electronic.
26. García EB., 2007, BIM-ISO8583.NET.dll, www.codeproject.com.
27. Gastelú PDCAT., 2011, SMDB: (Sistemas manejadores de base de datos).

28. GreyWyvern.com, 2012, Brian Huisman AkA GreyWyvern, [citado 15-3-2012].
29. Gutiérrez. JJ., ¿Qué es un framework web?
30. google.dirson.com, [citado 7-5-2012].
31. microbuffer.wordpress.com, 2011, que-es-postgresql, [citado19-12-2011].
32. msdn.microsoft.com, [citado 16-12-2011].
33. www.postgresql.org, [citado 19-12-2011].
34. i Tarrats JB, Jordán F., 2003, LA SEGURIDAD DE LAS TRANSACCIONES BANCARIAS EN INTERNET.
35. Carrillo Pérez Isaías, Martín ADR, 2008, METODOLOGIA DE DESARROLLO DEL SOFTWARE.
36. J. Wang., 2006, Computer Network Security: Theory and Practice.
37. K. Laudon CT., 2006, E-Commerce: Business, Technology, Society.
38. Kumar S., 2010, Code project.
39. Leininger MG. ,1976, Operaciones Bancarias.
40. Martínez Gil Lisbet, 2009, EXPERIENCIAS EN EL USO DE UN SOFTWARE LIBRE COMO MEDIO DE ENSEÑANZA EN LA ASIGNATURA COMERCIO ELECTRÓNICO EN LA UNIVERSIDAD DE PINAR DEL RÍO.
41. Loarte AS., 2005, Normas y procedimientos técnicos para garantizar la seguridad. (De la información publicada por las Entidades de la Administración Pública peruana a través de Internet).
42. López MJL., Criptografía y Seguridad en Computadores.
43. Manfred., 2010, Simple CAPTCHA Mechanism for ASP.NET MVC 2.
44. Martínez RSQ., 2007, Manejo de Excepciones.
45. Mestras JP., 2008, Estructura de las Aplicaciones Orientadas a Objetos.
    El patrón Modelo-Vista-Controlador (MVC).
46. Moreno ED, Pereira FD, Chiaramonte RB., 2005, Criptografía en software y hardware.
47. msdn.microsoft.com, [citado 12-1-2012].
48. Obregón LGADS. “Sistema de Personalización de Documentos de Identificación de la República de Cuba”. Universidad de las Ciencias Informáticas, Facultad 1, 2010.
49. Pakala. S., 2005, The Open Web Application Security Project (OWASP). The Open Web Application Security Project (OWASP).
50. Pressman R., Un enfoque práctico, Capítulo 17 (Técnicas de Software) de la página 281-304 y Capítulo 18 (Estrategias de Pruebas de Software) de la página 305-322.
51. Pressman R., 1997, Ingeniería de software. Un enfoque práctico.
52. Rivera JF., www.aurea.es, [citado 13-3-2012].
53. Rodríguez JR.,1976, Derecho bancario. Porrúa M.

54. sites.google.com, [citado 20-3-2012].
55. Slideshare, Manejo de excepciones.
56. solpagos.com, [citado 14-12-2012].
57. technet.microsoft.com, 2011, Guía de planeación para el cumplimiento del Estándar de seguridad de datos en la industria de tarjetas de pago, [citado 30-4-2012].
58. Tellez IRSR., 2009, La Seguridad del entorno tecnológico y en las Transacciones para el Comercio Electrónico.
59. Veiga AT., Módulo de Reportes del Sistema Único de Identificación Nacional, Universidad de las Ciencias Informáticas, 2010.
60. www.alertpay.com, [citado 12-12-2012].
61. www.alfa-redi.org, Revista de Derecho Informático, [citado 10-11-2011].
62. www.altova.com, [citado 5-11-2011].
63. www.apoyodigital.com.pe, [citado 10-12-2011].
64. www.desarrolloweb.com, [citado 15-1-2012].
65. www.hhsistemas.com, C#, [citado 17-12-2012].
66. www.ibiblio.org, [citado 7-1-2012].
67. www.maestrosdelweb.com, [citado 5-1-2012].
68. www.paypal.com, [citado 10-12-2011].
69. www.tecnologiapyme.com, [citado 7-11-2011].
70. www.utopicainformatica.com, 2010, convenciones estándares de codificación, [citado 20-3-2012].
71. www.w3c.es, [citado 28-12-2012].

# Anexos

Anexo1 Descripción de los escenarios y sus tareas.

**Módulo Administración.**

| | | |
|---|---|---|
| ***Nombre del escenario:*** Configurar cuenta de administración. | | ***Identificador:***1 |
| ***Objetivo del escenario:*** El administrador realiza la configuración de su cuenta en el sistema. | | |
| ***Persona:*** administrador | | |
| ***Iteración:*** 1 | ***Prioridad:*** Crítico | ***Complejidad:*** 3 |
| ***Precondiciones:*** El administrador debe estar autenticado en el sistema. | | |
| **Descripción*:*** El sistema puede contar con varios administradores, pero existe uno que es el principal, este puede ver los datos de los demás y los puede eliminar. Cada uno controla y administra su cuenta. El administrador una vez autenticado en el sistema puede realizar varias operaciones en el sistema. Para mostrar usuario de administración ver tarea 1.1, para crear usuario de administración ver tarea 1.2, para modificar usuario de administración ver tarea 1.3, para eliminar usuario de administración ver tarea 1.4 y para cambiar contraseña de administrador ver tarea 1.5. Finaliza el escenario. | | |
| ***Validaciones*:**- | | |
| ***Prototipo de interfaz de usuario:***   | | |

Tabla 25. Descripción del escenario Configurar cuenta de administración.
Fuente: elaboración propia

El escenario Configurar cuenta de administración fue dividido en 5 tareas: “Mostrar usuario de administración”, “Crear usuario de administración”, “Modificar usuario de administración”, “Eliminar usuario de administración” y “Cambiar contraseña”.

| | |
|---|---|
| ***Nombre de la tarea:*** Mostrar usuario de administración. | ***Identificador:***1.2 |
| ***Objetivo de la tarea:*** Mostrar detalles de un usuario de administración. | |
| ***Persona:*** administrador | |

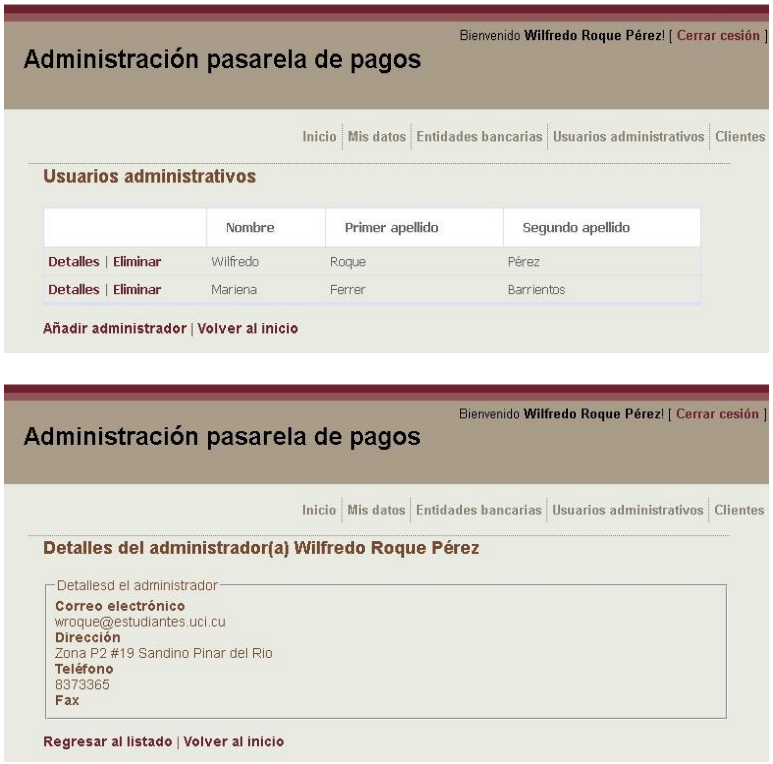
| ***Iteración:*** 1 | ***Prioridad:*** Medio | ***Complejidad:*** 1 |
|---|---|---|
| ***Precondiciones:*** El administrador debe estar autenticado en el sistema. | | |
| **Descripción*:*** El sistema brinda la opción de mostrar listado de usuarios de administración. El administrador elije esta alternativa. El sistema muestra el listado de los administradores existentes que se muestran con los datos: nombre, primer apellido y segundo apellido. El administrador selecciona en “Detalles” para ver los datos del administrador seleccionado, se muestran los datos: correo electrónico, fax , teléfono y dirección. Finaliza la tarea. | | |
| ***Validaciones*:**- | | |
| ***Prototipo de interfaz de usuario:***     | | |

Tabla 26. Descripción de la tarea Mostrar usuario administración.
Fuente: elaboración propia.

| ***Nombre de la tarea:*** Crear usuario de administración. | | ***Identificador:*** 1.2 |
|---|---|---|
| ***Objetivo de la tarea:*** El usuario se registra como administrador. | | |
| ***Persona:*** administrador | | |
| ***Iteración:*** 1 | ***Prioridad:*** Crítico | ***Complejidad:*** 1 |
| ***Precondiciones:***- | | |
| **Descripción*:*** El usuario accede al sistema con el objetivo de registrar sus datos personales. Se registra como administrador. Los datos a insertar son: nombre, primer apellido, segundo apellido, | | |

| |
|---|
| correo electrónico, contraseña, confirmar contraseña, teléfono (campo opcional), fax, dirección (campo opcional). El administrador oprime el botón “Aceptar”. El sistema envía una notificación al correo electrónico del administrador de la creación de su cuenta. Finaliza la tarea. |
| ***Validaciones*:**<br>• Que en cada campo se introduzcan los datos con el formato establecido.<br>• Los campos obligatorios no pueden estar vacíos. |
| ***Prototipo de interfaz de usuario:***<br>  |

Tabla 27. Descripción de la tarea Crear usuario de administración.

Fuente: elaboración propia.

| | | |
|---|---|---|
| ***Nombre de la tarea:*** Modificar usuario de administración. | | ***Identificador:*** 1.3 |
| ***Objetivo de la tarea:*** El administrador se registra para modificar sus datos. | | |
| ***Persona:*** administrador | | |
| ***Iteración:*** 1 | ***Prioridad:*** Crítico | ***Complejidad:*** 1 |
| ***Precondiciones:*** El usuario debe estar registrado y autenticado como administrador en el sistema. | | |
| **Descripción*:*** El administrador accede al sistema y elige la opción modificar datos. El sistema muestra los datos. El administrador puede modificar cualquier dato que le sea de su interés. Oprime el botón “Guardar”. Finaliza la tarea. | | |
| ***Validaciones*:**<br>• Que en cada campo se introduzcan los datos con el formato establecido.<br>• Los campos no pueden estar vacíos. | | |

***Prototipo de interfaz de usuario:***

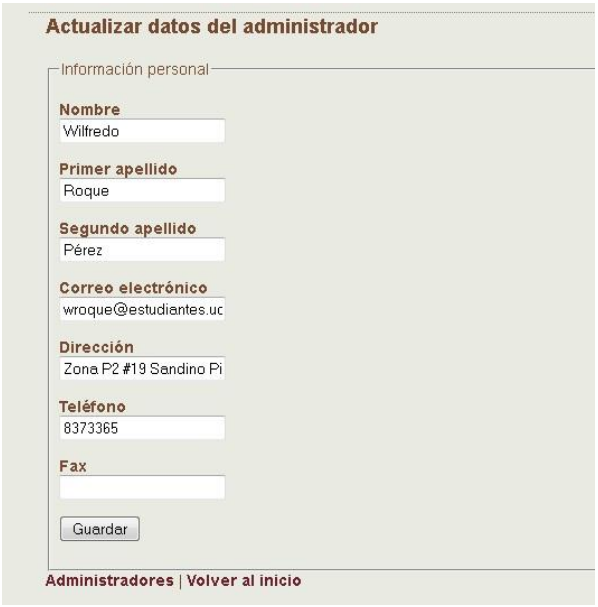



Tabla 28. Descripción de la tarea Modificar usuario de administración.
Fuente: elaboración propia.

| | | |
|---|---|---|
| ***Nombre de la tarea:*** Eliminar usuario de administración. | | ***Identificador:*** 1.4 |
| ***Objetivo de la tarea:*** El administrador elimina su cuenta. | | |
| ***Persona:*** administrador | | |
| ***Iteración:*** 1 | ***Prioridad:*** Crítico | ***Complejidad:*** 1 |
| ***Precondiciones:*** El usuario debe estar registrado y autenticado como administrador en el sistema. | | |
| **Descripción*:*** El administrador accede al sistema y elige la opción eliminar. El sistema pide verificación de la eliminación en un cuadro de texto. El administrador elimina su cuenta de las cuentas de administración. Finaliza la tarea. | | |
| ***Validaciones*:**- | | |

*Prototipo de interfaz de usuario:*

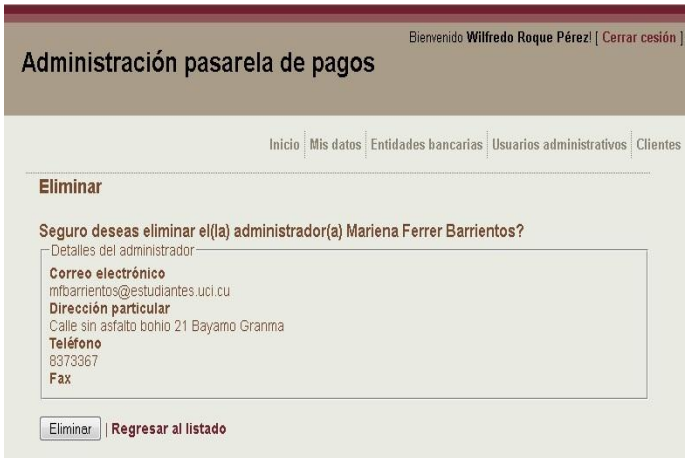




Tabla 29. Descripción de la tarea Eliminar usuario de administración.
Fuente: elaboración propia.

| | | |
|---|---|---|
| ***Nombre de la tarea:*** Cambiar contraseña de administrador. | | ***Identificador:*** 1.5 |
| ***Objetivo de la tarea:*** El administrador cambia su contraseña. | | |
| ***Persona:*** administrador | | |
| ***Iteración:*** 1 | ***Prioridad:*** Crítico | ***Complejidad:*** 1 |
| ***Precondiciones:*** El administrador debe estar autenticado en el sistema. | | |
| **Descripción*:*** El administrador accede al sistema y elige la opción cambiar contraseña. El sistema muestra los datos: contraseña actual, nueva contraseña y confirmar contraseña. El administrador oprime el botón “Aceptar”. Finaliza la tarea. | | |
| ***Validaciones*:**- | | |

*Prototipo de interfaz de usuario:*

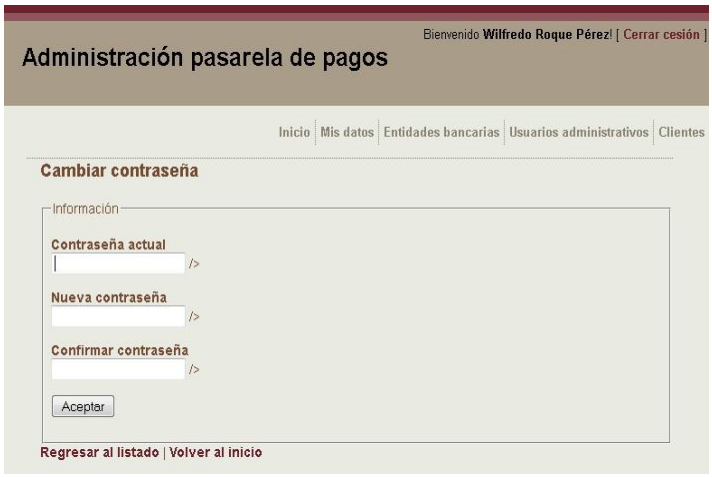



Tabla 30. Descripción de la tarea Cambiar contraseña de administrador.
Fuente: elaboración propia.

| | | |
|---|---|---|
| ***Nombre del escenario:*** Configurar cliente. | | ***Identificador:*** 2 |
| ***Objetivo del escenario:*** Configura la cuenta del cliente. | | |
| ***Persona:*** administrador | | |
| ***Iteración:*** 1 | ***Prioridad:*** Crítico | ***Complejidad:*** 3 |
| ***Precondiciones:*** El administrador debe estar autenticado en el sistema. | | |
| **Descripción*:*** El administrador selecciona la opción “Clientes”. El sistema muestra las distintas opciones que puede realizar el administrador sobre el cliente. Para Mostrar cliente ver tarea 2.1, para Eliminar cliente (desactivarlo) ver tarea 2.2, para ver tarea Buscar cliente ver tarea 2.3, para configurar pago ver tarea 2.4 y para Mostrar operaciones de pago ver tarea 2.5. Finaliza el escenario. | | |
| ***Validaciones*:**- | | |
| ***Prototipo de interfaz de usuario:*** | | |


Tabla 31. Descripción del escenario Configurar cliente.
Fuente: elaboración propia.

| | | |
|---|---|---|
| ***Nombre de la tarea:*** Mostrar cliente. | | ***Identificador:*** 2.1 |
| ***Objetivo de la tarea:*** El administrador muestra los datos de un cliente en el sistema. | | |
| ***Persona:*** administrador | | |
| ***Iteración:*** 1 | ***Prioridad:*** Crítico | ***Complejidad:*** 1 |
| ***Precondiciones:*** El administrador debe estar autenticado en el sistema. | | |
| **Descripción*:*** El administrador selecciona la opción “Clientes”. El sistema muestra el listado de los clientes. El administrador elije un cliente. El sistema muestra los detalles del cliente seleccionado: correo electrónico, teléfono, fax y dirección. Finaliza la tarea. | | |
| ***Validaciones*:**- | | |
| ***Prototipo de interfaz de usuario:*** | | |



Tabla 32. Descripción del escenario Mostrar cliente.
Fuente: elaboración propia.

| | | |
|---|---|---|
| ***Nombre de la tarea:*** Eliminar cliente. | | ***Identificador:*** 2.2 |
| ***Objetivo de la tarea:*** Desactivar un cliente. | | |
| ***Persona:*** administrador | | |
| ***Iteración:*** 1 | ***Prioridad:*** Medio | ***Complejidad:*** 1 |
| ***Precondiciones:*** El administrador debe estar autenticado en el sistema. | | |
| **Descripción*:*** El sistema brinda la opción de eliminar cliente. El administrador elije esta alternativa. El sistema muestra el listado de los clientes. El administrador selecciona el cliente a desactivar. El sistema muestra una alarma advirtiendo si desea eliminar al cliente seleccionado. El administrador oprime el botón “Eliminar”. Se desactiva al cliente. Finaliza la tarea. | | |
| ***Validaciones*:**- | | |
| ***Prototipo de interfaz de usuario:*** | | |


Tabla 33. Descripción de la tarea Eliminar cliente.
Fuente: elaboración propia.

| | | |
|---|---|---|
| ***Nombre de la tarea:*** Buscar cliente. | | ***Identificador:*** 2.3 |
| ***Objetivo de la tarea:*** El administrador busca un cliente en el sistema. | | |
| ***Persona:*** administrador | | |
| ***Iteración:*** 1 | ***Prioridad:*** Crítico | ***Complejidad:*** 1 |
| ***Precondiciones:*** El administrador debe estar autenticado en el sistema. | | |
| **Descripción*:*** El administrador selecciona la opción "Buscar cliente". El sistema muestra las opciones de búsqueda: nombre, nombre y apellidos y correo electrónico. El administrador selecciona la opción de búsqueda. Finaliza la tarea. | | |
| ***Validaciones*:** - | | |
| ***Prototipo de interfaz de usuario:***  | | |



Tabla 34. Descripción de la tarea Buscar cliente.
Fuente: elaboración propia.

| | |
|---|---|
| ***Nombre de la tarea:*** Configurar pago. | ***Identificador:*** 2.4 |

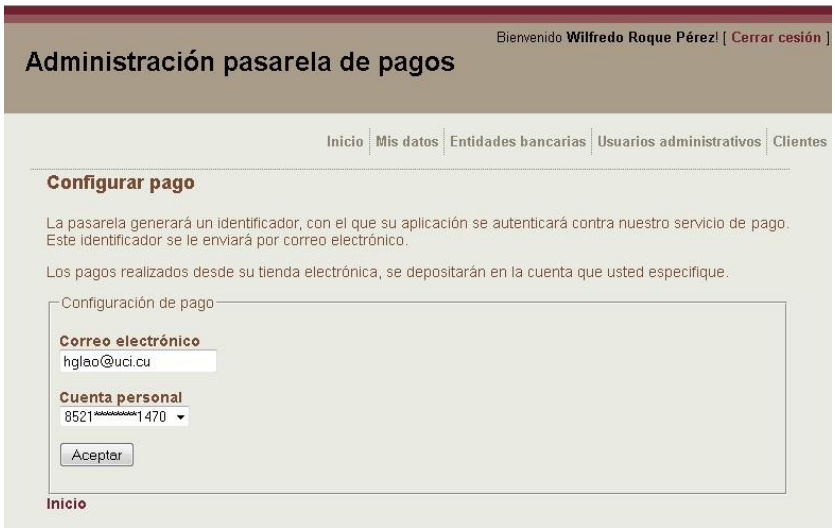
| | | |
|---|---|---|
| ***Objetivo de la tarea:*** El administrador gestiona los datos para configurar pago. | | |
| ***Persona:*** administrador | | |
| ***Iteración:*** 1 | ***Prioridad:*** Crítico | ***Complejidad:*** 1 |
| ***Precondiciones:*** El administrador debe estar autenticado en el sistema. | | |
| **Descripción*:*** El administrado recibe de la tienda electrónica el número de la tarjeta de la cuenta del vendedor. El sistema genera un identificador que representa el número de la tienda en el sistema. Esta cuenta será la cuenta donde se depositarán todos los pagos efectuados por los clientes. Además se requiere insertar el campo correo electrónico para enviar las notificaciones de los pagos efectuados hacia esa cuenta. Se oprime el botón “Aceptar”. Finaliza la tarea. | | |
| ***Validaciones*:**- | | |
| ***Prototipo de interfaz de usuario:***   | | |

Tabla 35. Descripción del escenario Configurar pago.
Fuente: elaboración propia.

| | | |
|---|---|---|
| ***Nombre de la tarea:*** Mostrar operaciones de pago. | | ***Identificador:*** *2.*5 |
| ***Objetivo de la tarea:*** Mostrar las transacciones realizadas por un usuario. | | |
| ***Persona:*** administrador | | |
| ***Iteración:*** 1 | ***Prioridad:*** Crítico | ***Complejidad:*** 1 |
| ***Precondiciones:*** El administrador debe estar autenticado en el sistema. | | |
| **Descripción*:*** El administrador accede al sistema para consultar las transacciones de un cliente. El sistema muestra el listado de clientes. El administrador selecciona la opción “Mostrar operaciones de pago”. El administrador selecciona el cliente al cual requiere consultar sus transacciones. Se muestra un listado de las transacciones realizadas. Se muestra los datos: tipo de transacción, | | |

| |
|---|
| monto, resultado y fecha. El sistema brinda además la opción de filtrar por fechas las transacciones. Finaliza la tarea. |
| ***Validaciones*:**- |
| ***Prototipo de interfaz de usuario:***   |

Tabla 36. Descripción de la tarea Mostrar operaciones de pago.
Fuente: elaboración propia.

El escenario Configurar banco fue dividido en varias tareas: “Mostrar banco”, “Crear banco”, “Modificar banco” y “Eliminar banco (desactivarlo)”.

| | | |
|---|---|---|
| ***Nombre del escenario:*** Gestionar banco. | | ***Identificador:*** 3 |
| ***Objetivo del escenario:*** El administrador gestiona los bancos emisores que se comunican con la pasarela de pago. | | |
| ***Persona:*** administrador | | |
| ***Iteración:*** 1 | ***Prioridad:*** Crítico | ***Complejidad:*** 3 |
| ***Precondiciones:*** El administrador debe estar autenticado en el sistema. | | |
| **Descripción*:*** El administrador selecciona la opción “Entidades bancarias”. Para mostrar banco ver tarea 3.1, para crear banco ver tarea 3.2, para editar banco ver tarea 3.3 y para eliminar banco ver tarea 3.4. Finaliza el escenario. | | |
| ***Validaciones*:**- | | |

***Prototipo de interfaz de usuario:***


Tabla 37. Descripción de la tarea Gestionar banco.
Fuente: elaboración propia.

| | | |
|---|---|---|
| ***Nombre de la tarea:*** Mostrar banco. | | ***Identificador:*** 3.1 |
| ***Objetivo de la tarea:*** El administrador muestra los detalles de un banco. | | |
| ***Persona:*** administrador | | |
| ***Iteración:*** 1 | ***Prioridad:*** Crítico | ***Complejidad:*** 1 |
| ***Precondiciones:*** El administrador debe estar autenticado en el sistema. | | |
| **Descripción*:*** El administrador del sistema selecciona la opción “Entidades bancarias”. Se muestra un listado de los bancos emisores de las tarjetas que existen en el sistema. Se muestran con los datos: nombre y código. Finaliza la tarea. | | |
| ***Validaciones*:** - | | |
| ***Prototipo de interfaz de usuario:***  | | |



Tabla 38. Descripción de la tarea Mostrar banco.
Fuente: elaboración propia.

| | |
|---|---|
| ***Nombre de la tarea:*** Crear banco. | ***Identificador:*** 3.2 |

| | | |
|---|---|---|
| ***Objetivo de la tarea:*** El administrador inserta los datos del banco. | | |
| ***Persona:*** administrador | | |
| ***Iteración:*** 1 | ***Prioridad:*** Crítico | ***Complejidad:*** 1 |
| ***Precondiciones:*** El administrador debe estar autenticado en el sistema. | | |
| **Descripción*:*** El administrador del sistema selecciona la opción “Añadir banco”. El sistema muestra los campos a insertar: nombre y código. El administrador oprime el botón “Aceptar. Finaliza la tarea. | | |
| ***Validaciones*:**<br>• Los datos tienen que introducirse con el formato establecido.<br>• Todos los campos deben estar llenos. | | |
| ***Prototipo de interfaz de usuario:***<br>  | | |

Tabla 39. Descripción de la tarea Crear banco.
Fuente: elaboración propia.

| | | |
|---|---|---|
| ***Nombre de la tarea:*** Modificar banco. | | ***Identificador:*** 3.3 |
| ***Objetivo de la tarea:*** El administrador modifica los datos del banco. | | |
| ***Persona:*** administrador | | |
| ***Iteración:*** 1 | ***Prioridad:*** Crítico | ***Complejidad:*** 1 |
| ***Precondiciones:*** El administrador debe estar autenticado en el sistema. | | |
| **Descripción*:*** El administrador del sistema selecciona la opción “Modificar banco”. El sistema muestra el listado de los bancos ya creados. El administrador selecciona el banco a modificar. El sistema muestra los datos que se pueden modificar. El administrador actualiza los datos deseados y oprime el botón “Guardar”. Se actualiza la información, emitiéndose un mensaje notificando que la modificación se realizó satisfactoriamente. Finaliza la tarea. | | |

| |
|---|
| ***Validaciones*:**<br>• Los datos tienen que introducirse con el formato establecido.<br>• Todos los campos a actualizar deben estar llenos. |
| ***Prototipo de interfaz de usuario:***<br>  |

Tabla 40. Descripción de la tarea Modificar banco.
Fuente: elaboración propia.

| | | |
|---|---|---|
| ***Nombre de la tarea:*** Eliminar banco. | | ***Identificador:*** 3.4 |
| ***Objetivo de la tarea:*** El administrador desactiva un banco. | | |
| ***Persona:*** administrador | | |
| ***Iteración:*** 1 | ***Prioridad:*** Crítico | ***Complejidad:*** 1 |
| ***Precondiciones:*** El administrador debe estar autenticado en el sistema. | | |
| **Descripción*:*** El administrador selecciona la opción “Eliminar banco”. El sistema muestra el listado de los bancos. El administrador selecciona el banco a eliminar. El sistema muestra en un cuadro de texto la verificación de la eliminación. El sistema elimina el banco seleccionado, emitiéndose un mensaje informando que la eliminación se realizó correctamente. La eliminación es una desactivación del banco, pues los datos de las transacciones realizadas hacia el permanecen en el sistema. Mientras esté desactivado no se pueden realizar transacciones hacia esta entidad. Finaliza la tarea. | | |
| ***Validaciones*:** - | | |
| ***Prototipo de interfaz de usuario:*** | | |


Tabla 41. Descripción de la tarea Eliminar banco.
Fuente: elaboración propia.

| | | |
|---|---|---|
| ***Nombre del escenario:*** Autenticar administrador. | | ***Identificador:*** 4 |
| ***Objetivo del escenario:*** El usuario se autentica como administrador del sistema. | | |
| ***Persona:*** administrador | | |
| ***Iteración:*** 1 | ***Prioridad:*** Crítico | ***Complejidad:*** 1 |
| ***Precondiciones:*** El administrador debe estar registrado en el sistema. | | |
| **Descripción*:*** El administrador accede al sistema. Introduce los datos de autenticación: correo electrónico y contraseña. Finaliza el escenario. | | |
| ***Validaciones*:**<br>• Los datos tienen que introducirse con el formato establecido.<br>• Los campos no pueden estar vacíos.<br>• Validar existencia de la cuenta del usuario. | | |
| ***Prototipo de interfaz de usuario:***<br>  | | |



Tabla 42. Descripción del escenario Autenticar administrador.
Fuente: elaboración propia.

**Módulo Administración de clientes.**

| | | |
|---|---|---|
| ***Nombre del escenario:*** Configurar cuenta. | | ***Identificador:*** 6 |
| ***Objetivo del escenario:*** Configurar la cuenta del cliente en el sistema. | | |
| ***Persona:*** cliente | | |
| ***Iteración:*** 2 | ***Prioridad:*** Crítico | ***Complejidad:*** 3 |
| ***Precondiciones:*** | | |
| **Descripción*:*** El cliente accede al sistema con el objetivo de configurar su cuenta. El sistema muestra las opciones de configuración: para mostrar cliente ver tarea modificar datos del cliente ver tarea 6.1, para desactivar cliente ver tarea 6.2 y para cambiar contraseña ver tarea 6.3. Finaliza el escenario. | | |
| ***Validaciones*:**- | | |
| ***Prototipo de interfaz de usuario:***   | | |

Tabla 43. Descripción del escenario Configurar cuenta.
Fuente: elaboración propia.

El escenario Configurar cuenta fue dividido en varias tareas: “Modificar datos”, “Desactivar cliente” y “Cambiar contraseña”.

| | | |
|---|---|---|
| ***Nombre de la tarea:*** Modificar datos. | | ***Identificador:*** 6.1 |
| ***Objetivo de la tarea:*** El cliente modifica los datos de su cuenta personal. | | |
| ***Persona:*** cliente | | |
| ***Iteración:*** 2 | ***Prioridad:*** Crítico | ***Complejidad:*** 1 |
| ***Precondiciones:*** El cliente debe estar registrado y autenticado en el sistema. | | |
| **Descripción*:*** El cliente accede al sistema y elige la opción “Actualizar”. El sistema muestra una interfaz | | |

| |
|---|
| con los campos a actualizar: nombre, primer apellido, segundo apellido, fax, teléfono, dirección y correo electrónico. El cliente actualiza los datos de interés y oprime el botón “Guardar”. Finaliza el escenario. |
| ***Validaciones*:**<br>• Que los campos se introduzcan con el formato establecido.<br>• Que los campos obligatorios no este vacíos. |
| ***Prototipo de interfaz de usuario:***<br>  |

Tabla 44. Descripción de la tarea Modificar datos del cliente.
Fuente: elaboración propia.

| | | |
|---|---|---|
| ***Nombre de la tarea:*** Desactivar cliente. | | ***Identificador:*** 6.2 |
| ***Objetivo del escenario:*** El cliente deshace su cuenta en el sistema. | | |
| ***Persona:*** cliente | | |
| ***Iteración:*** 2 | ***Prioridad:*** Crítico | ***Complejidad:*** 1 |
| ***Precondiciones:*** El cliente debe estar registrado y autenticado en el sistema. | | |
| **Descripción*:*** El cliente accede al sistema y elige la opción desactivar. El sistema muestra una verificación de la desactivación. El cliente acepta y ya no puede efectuar transacciones en el sistema. La cuenta no se elimina completamente, esta información puede hacer falta a los administradores para realizar reportes. Finaliza el escenario. | | |
| ***Validaciones*:**- | | |
| ***Prototipo de interfaz de usuario:*** | | |


Tabla 45. Descripción de la tarea Desactivar cliente.
Fuente: elaboración propia.

| | | |
|---|---|---|
| ***Nombre de la tarea:*** Cambiar contraseña. | | ***Identificador:*** 6.3 |
| ***Objetivo de la tarea:*** El cliente cambia su contraseña de autenticación. | | |
| ***Persona:*** cliente | | |
| ***Iteración:*** 2 | ***Prioridad:*** Crítico | ***Complejidad:*** 1 |
| ***Precondiciones:*** El cliente debe estar registrado y autenticado en el sistema. | | |
| **Descripción*:*** El cliente accede al sistema y elige la opción cambiar contraseña. El sistema muestra los datos: contraseña actual, contraseña nueva y confirmar contraseña. El cliente llena los campos y elije aceptar. Se cambia la contraseña. Finaliza el escenario. | | |
| ***Validaciones*:**- | | |
| ***Prototipo de interfaz de usuario:*** | | |


Tabla 46. Descripción del escenario Cambiar contraseña.
Fuente: elaboración propia.

| | | |
|---|---|---|
| ***Nombre del escenario:*** Configurar cuenta bancaria. | | ***Identificador:*** 8 |
| ***Objetivo del escenario:*** El cliente gestiona los datos de su cuenta bancaria. | | |
| ***Persona:*** cliente | | |
| ***Iteración:*** 2 | ***Prioridad:*** Crítico | ***Complejidad:*** 3 |
| ***Precondiciones:*** El cliente debe estar autenticado en el sistema. | | |
| **Descripción*:*** El cliente una vez autenticado en el sistema desea realizar varias operaciones sobre su cuenta bancaria. Para mostrar listado de cuentas bancarias ver tarea 8.1. Para registrar cuenta bancaria ver tarea 8.2. Para eliminar cuenta bancaria ver tarea 8.3. Para consultar saldo ver tarea 8.4 y para mostrar reporte del historial transacciones ver tarea 8.5. Finaliza el escenario. | | |
| ***Validaciones*:**- | | |
| ***Prototipo de interfaz de usuario:*** | | |


Tabla 47. Descripción del escenario Configurar cuenta bancaria.
Fuente: elaboración propia.

El escenario Configurar cuenta bancaria fue dividido en 4 tareas: “Mostrar listado de cuentas bancarias”, “Registrar cuenta bancaria”, “Eliminar cuenta bancaria”, “Consultar saldo bancario”, “Mostrar reporte de historial de transacciones”.

| | | |
|---|---|---|
| ***Nombre de la tarea:*** Registrar cuenta bancaria. | | ***Identificador:*** 8.2 |
| ***Objetivo del escenario:*** se insertan los datos de la tarjeta del cliente. | | |
| ***Persona:*** cliente | | |
| ***Iteración:*** 2 | ***Prioridad:*** Crítico | ***Complejidad:*** 1 |
| ***Precondiciones:*** El cliente debe estar autenticado en el sistema. | | |
| **Descripción*:*** El cliente selecciona la opción “Mis cuentas”. El sistema muestra una interfaz con el listado de las cuentas bancarias que tiene el cliente. El mismo selecciona “Adicionar cuenta bancaria”. Se muestra los campos a llenar: número de la tarjeta, CCV y fecha de validación. El cliente oprime el botón “Aceptar”. Finaliza la tarea. | | |
| ***Validaciones*:**-<br>• Que el número de la tarjeta no se repita.<br>• Que los campos se introduzcan con el formato establecido.<br>• Que los campos no estén vacíos. | | |
| ***Prototipo de interfaz de usuario:***<br>  | | |

Tabla 48. Descripción del escenario Notificar al cliente de la creación de la cuenta.
Fuente: elaboración propia.

| | | |
|---|---|---|
| ***Nombre de la tarea:*** Eliminar cuenta bancaria. | | ***Identificador:*** 8.3 |
| ***Objetivo de la tarea:*** Eliminar una cuenta bancaria. | | |
| ***Persona:*** cliente | | |
| ***Iteración:*** 2 | ***Prioridad:*** Medio | ***Complejidad:*** 1 |
| ***Precondiciones:*** El cliente debe estar autenticado en el sistema. | | |
| **Descripción*:*** El cliente lista las cuentas bancarias que están a su nombre. El sistema muestra el listado. El cliente selecciona una cuenta a eliminar y oprime el botón “Eliminar”. El sistema muestra una alarma informando que se eliminara la cuenta. El cliente acepta y se elimina la cuenta. Finaliza la tarea. | | |
| ***Validaciones*:** - | | |

***Prototipo de interfaz de usuario:***

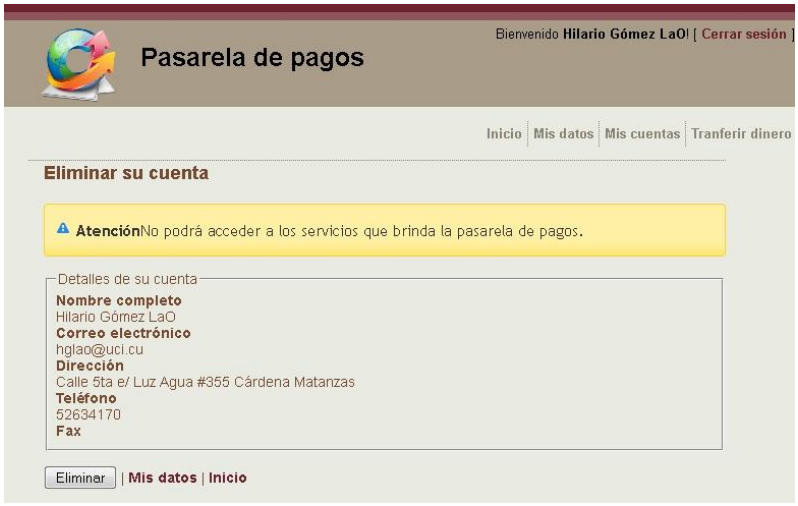



Tabla 49. Descripción de la tarea Eliminar cuenta bancaria.
Fuente: elaboración propia.

| | | |
|---|---|---|
| ***Nombre de la tarea:*** Consultar saldo bancario. | | ***Identificador:*** 8.4 |
| ***Objetivo de la tarea:*** El cliente consulta su saldo bancario. | | |
| ***Persona:*** cliente | | |
| ***Iteración:*** 2 | ***Prioridad:*** Medio | ***Complejidad:*** 1 |
| ***Precondiciones:*** El cliente debe estar autenticado en el sistema. | | |
| **Descripción*:*** El cliente selecciona la opción mostrar saldo bancario. El sistema muestra las cuentas registradas por ese cliente. Este selecciona la cuenta de la que quiere conocer el saldo. El sistema muestra los campos a llenar: PIN, fecha de vencimiento y CCV. El cliente inserta los datos y oprime el botón “Aceptar”. El sistema muestra el monto de esa cuenta. Finaliza la tarea. | | |
| ***Validaciones*:** <br>• Que en el sistema exista la cuenta a la que pertenece el saldo bancario a mostrar. | | |
| ***Prototipo de interfaz de usuario:***  | | |



Tabla 50. Descripción de la tarea Consultar saldo bancario.

Fuente: elaboración propia.

Anexo 2 Descripción de las clases persistentes.

| **Nombre** | usuario | |
|---|---|---|
| **Descripción** | Contiene todos los datos de los usuarios. | |
| **Atributo** | **Tipo** | **Descripción** |
| id_ usuario | string | Identificación del usuario. |
| contrasena | string | Contraseña del usuario. |
| fax | string | Dato del fax (opcional). |
| telefono | string | Número de teléfono. |
| correo | string | Dirección de correo electrónico. |
| direccion | string | Dirección particular. |
| nombre | string | Nombre propio. |
| primer_apellido | string | Primer apellido. |
| segundo_apellido | string | Segundo apellido. |
| activo | bool | Indica si el usuario existe en la pasarela. |
| tipo_usuario | string | Define los permisos que tiene los usuarios para accionar en la pasarela. |

Tabla 51. Descripción de la clase persistente: “usuario”.
Fuente: elaboración propia.

| **Nombre** | transaccion | |
|---|---|---|
| **Descripción** | Contiene todos los datos de las transacciones. | |
| **Atributo** | **Tipo** | **Descripción** |
| idtransaccion | string | Identificación de la transacción. |
| monto | double | Cantidad de efectivo a pagar o transferir. |
| fecha | DateTime | Fecha en la que se realizó la transacción. |
| resultado | bool | Resultado de la transacción (si se efectúo o no). |
| tipo_transaccion | string | Tipo de transacción (pago o transferencia). |

Tabla 52. Descripción de la clase persistente: “transaccion”.
Fuente: elaboración propia.

| **Nombre** | transaccion | |
|---|---|---|
| **Descripción** | Contiene todos los datos de las transacciones. | |
| **Atributo** | **Tipo** | **Descripción** |
| idtransaccion_x_cuenta | string | Identificación de la transacción. |

Tabla 53. Descripción de la clase persistente: “transaccion_x_cuenta”.
Fuente: elaboración propia.

| **Nombre** | cuenta |
|---|---|

| **Descripción** | Contiene todos los datos de las cuentas de los usuarios. | |
|---|---|---|
| **Atributo** | **Tipo** | **Descripción** |
| idcuenta | string | Identificación de la cuenta del banco. |
| fecha_vencimiento | Datetime | Indica si la tarjeta es válida. |

Tabla 54. Descripción de la clase persistente: “cuenta”.
Fuente: elaboración propia.

| **Nombre** | entidad_bancaria. | |
|---|---|---|
| **Descripción** | Contiene todos los datos del banco que maneja las cuentas de los usuarios. | |
| **Atributo** | **Tipo** | **Descripción** |
| identidadbancaria | string | Identificación del banco. |
| nombre_entidad | string | Nombre del banco. |

Tabla 55. Descripción de la clase persistente: “entidad_bancaria”.
Fuente: elaboración propia.

| **Nombre** | solicitudpago | |
|---|---|---|
| **Descripción** | Contiene todos los datos de los usuarios. | |
| **Atributo** | **Tipo** | **Descripción** |
| idpago | string | Identificación del pago. |
| monto | double | Monto del pago. |
| urlredirect | string | Dirección de la tienda virtual. |
| estado | string | Representa el estado en que está la transacción: iniciada, terminada o suspendida. |

Tabla 56. Descripción de la clase persistente:”solicitudpago”.
Fuente: elaboración propia.

Anexo 3 Descripción de las clases controladoras.

| **Nombre:**CuentaControl | | |
|---|---|---|
| **Tipo de clase:** Controladora | | |
| **Descripción:** Controla las funcionalidades de gestionar cuenta bancaria. | | |
| **Atributo:-** | **Tipo:-** | **Descripción:-** |
| **Métodos:** | | **Descripción:** |
| AdicionarCuenta(string idcuenta, string fecha_vencimiento, string idusuario, string identidadbancaria, string ccv) | | El método inserta una cuenta con los parámetros recibidos. |
| EliminarCuenta(string id) | | El método elimina una cuenta dado un identificador. |
| CuentaDadoId(string id) | | El método busca una cuenta dado un identificador. |
| NumeroCuenta(string id) | | El método devuelve un listado con las cuentas añadidas. |
| CuentaCliente(string id) | | El método devuelve un listado de las cuentas de un usuario dado su identificador. |
| NumeroCuenta(string id) | | El método muestra el PAN del cliente dado un |

| | |
|---|---|
| | identificador. Sólo muestra los primeros y 4 últimos dígitos. |
| TransaccionesXcuenta(string id) | El método muestra el listado de transacciones pertenecientes al identificador introducido. |
| ConsultarSaldo(string id, string ccv, string pin, string fechaVenc) | El método devuelve la respuesta dada por el banco sobre el saldo del usuario. |
| InterpretarAuthorizationResponseSaldo(string[] message) | El método devuelve la respuesta del mensaje enviado al banco sobre el saldo del cliente. |

Tabla 57. Descripción de la clase controladora:"CuentaControl".
Fuente: elaboración propia.

| **Nombre:**EntidadBancariaControl | | |
|---|---|---|
| **Tipo de clase:** Controladora | | |
| **Descripción:** Controla las funcionalidades para gesitonar los bancos. | | |
| **Atributo:-** | **Tipo:-** | **Descripción:-** |
| **Métodos:** | **Descripción:** | |
| ExisteEntidadBancaria(string nombre) | El método dado un nombre devuelve si existe o no el banco que se corresponde con ese nombre. | |
| AdicionarEntidadBancaria(string nombre_entidad, string codigo) | El método añade un banco insertando el nombre. | |
| ListarEntidades() | El método lista los bancos existentes. | |
| EditarEntidadBancaria(string id, string nombre_entidad, string codigo) | El método muestra una entidad bancaria que se corresponde con el nombre y el identificador recibidos. | |
| EntidadBancariaDadoId(string id) | El método devuelve el banco que se corresponde con el identificador recibido. | |
| EliminarEntidadBancaria(string id) | El método elimina un banco según un identificador dado. | |

Tabla 58. Descripción de la clase persistente: "EntidadBancariaControl".
Fuente: elaboración propia.

| **Nombre:**SolicitudPagoControl | | |
|---|---|---|
| **Tipo de clase:** Controladora | | |
| **Descripción:** Controla las funcionalidades para configurar la solicitud de los pagos. | | |
| **Atributo:-** | **Tipo:-** | **Descripción:-** |
| **Métodos:** | **Descripción:** | |
| ExisteSolicitud(string idpago) | El método devuelve si existe o no una solicitud de pago. | |
| AdicionarSolicitud(string idpago, double monto, string urlredirect) | El método adiciona en una lista una solicitud de pago siempre y cuando esta exista. | |
| EliminarSolicitud(string idpago) | El método elimina la solicitud de pago que coincide con el identificador introducido. | |
| ProcesarSolicitud(string idpago) | El método devuelve si la solicitud fue procesada o no. | |
| ListarSolicitudes() | El método devuelve un listado de todas las solicitudes de pago existentes. | |

| | |
|---|---|
| ObtenerSolicitud(string idpago) | El método devuelve la solicitud de pago que coincide con el identificador pasado por parámetro. |

Tabla 59. Descripción de la clase persistente: “SolicitudPagoControl”.
Fuente: elaboración propia.

Anexo 4 Casos de prueba.

| **Escenario** | **Descripción** | **nombre** | **1er_apellido** | **2do_apellido** | **correo** | **contraseña** | **dirección** | **teléfono** | **fax** | **Respuesta del sistema** | **Flujo central** |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Tarea 6. Mostrar listado de usuarios de administración. | Mostrar cuenta de usuario de administración con los datos establecidos. | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | Del listado de usuarios de administración que existen se selecciona el usuario deseado que se muestra con todos sus datos. | 1. Dar clic en la opción Mostrar listado de usuarios de administración.<br>2. Se muestra la interfaz con el listado de usuarios de administración.<br>3. Se selecciona un usuario.<br>4. Se pulsa en el botón “Detalles”.<br>5. Se muestra toda la información del usuario. |
| Tarea 7. Crear usuario de administración. | Crear usuario de administración insertando los | V<br>wilfredo | V<br>roque | V<br>perez | V<br>wroque | V<br>Pe*p@it7o | V<br>Pinar calle3 %7 y 9 | V<br>3456 | V<br>67895 | Se crea la cuenta del usuario con los datos insertados. | 1. Dar clic en la opción Añadir administrador. |
| | | V | V | V | V | V | V | V | V | | |

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | datos establecidos. | wilfredo | roque | perez | wroque | Pe*p@it7o | Pinar calle3 %7 y 9 | | | | 2. Se muestra la interfaz con los campos a llenar.<br>3. Se pulsa en el botón Aceptar.<br>4. Se informa que se insertaron los datos correctamente |
| | | V wilfredo | V roque | V perez | V wroque | V Pe*p@it7o | V Pinar calle3 %7 y 9 | V 3456 | V | | |
| | | V wilfredo | V roque | V perez | V wroque | V Pe*p@it7o | V Pinar calle3 %7 y 9 | V | V 67895 | | . |
| | | I D34mer5 | V solis | I fons3ca | I dsfon | I 34rts | I 34re | V | V | Se muestra un mensaje de error informando que algún campo está escrito incorrectamente. Se informa que la contraseña debe exceder los 6 caracteres. | |
| | | I | I | I | I | I | V | V | V | Se muestra un mensaje de error informando que debe llenar los campos obligatorios. | |
| Tarea 8. Modificar usuario de administración. | Modificar datos de un usuario de administ | V damaris | V solis | V perez | V dsfonseca@uci.cu | V D2m345r | V | V | V | Se modifican los datos establecidos. | 1. Dar clic en la opción “Actualizar datos”. |
| | | I | I | I | I | I | V | V | V | Se muestra | |

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | ración insertando los datos requeridos. | | | | | | | | | un mensaje de erro r informando que llene los campos obligatorios. | 2. Se muestra la interfaz con los datos del administrador (sólo puede modificar sus datos ese administrador). 3. Se introducen los nuevos datos. 4. Se pulsa en el botón "Guardar". 5. Se muestra un mensaje informando que los datos fueron modificados correctamente. |
| Tarea 9. Eliminar usuario de administración. | El administrador elimina su cuenta. (pueden existir varios administradores) | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | Se muestra un mensaje informando que fue eliminada su cuenta de administrador. | 1. Dar clic en la opción eliminar. 2. Se muestra la interfaz para realizar la acción. 3. Clic en el botón eliminar. 4. Se muestra un mensaje |

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | de verificación de la eliminación.<br><br>5. Se pulsa en "Aceptar".<br><br>6. Se elimina el usuario. |

Tabla 60. Caso de prueba Configurar cuenta de administración.
Fuente: elaboración propia.

| No | Nombre de campo | Clasificación | Valor Nulo | Descripción |
|---|---|---|---|---|
| 1 | nombre | campo de texto | No | Se aceptan letras minúsculas y mayúsculas. |
| 2 | 1er_apellido | campo de texto | No | Se aceptan letras minúsculas y mayúsculas. |
| 3 | 2do_apellido | campo de texto | No | Se aceptan letras minúsculas y mayúsculas. |
| 4 | correo | campo de texto | No | Se aceptan letras minúsculas y mayúsculas, el carácter @ es obligatorio luego el dominio al que pertenece el nombre del usuario del correo electrónico. |
| 5 | contraseña | campo de texto | No | Se aceptan letras minúsculas, mayúsculas, números, signos y debe exceder de los 6 caracteres. |
| 6 | dirección | campo de texto | No | Letras y números. |
| 7 | teléfono | campo de texto | Si | Sólo números. |
| 8 | fax | campo de texto | Si | Sólo números. |

Tabla 61. Descripción de variables del caso de prueba Configurar cuenta de administración.
Fuente: elaboración propia.

| Escenario | Descripción | nombre | 1er_apellido | 2do_apellido | correo | Respuesta del sistema | Flujo central |
|---|---|---|---|---|---|---|---|

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| EC 10. Buscar cliente. | Buscar un cliente siguiendo un criterio de búsqueda. | V miladys | V vazquez | V Fernandez | V mvfdez@estudiantes.uci.cu | Se muestra el o los clientes que se corresponden con las entradas realizadas. | 1. Dar clic en la opción Buscar clientes.<br>2.Se muestra la interfaz para buscar al cliente<br>3. Se introducen los datos solicitados.<br>4. Se muestra el listado con el o los clientes correspondientes a los datos insertados. |
| | | V miladys | V vazquez | V Fernandez | V | | |
| | | V | V | V | V dsfonseca@estudiantes.uci.cu | | |
| | | V wilfredo | V roque | V perez | V dsFonseca32@estudiantes.uci.cu | | |
| | | V wilfredo | V | V | V | | |
| | | V dary | V arcia | V medina | V damedina@uci.cu | Se muestra un mensaje informando que el cliente no existe. | |
| | | I Dam3ris | I Solis | I Fons2c6 | I Dfonseca.uci.cu | Se muestra un mensaje de error informando que los campos nombre, apellidos o correo electrónico están escritos incorrectamente. | |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | I | I | I | I | Se muestra un mensaje de error informando que los campos están vacíos. | |

Tabla 62. Caso de prueba Buscar cliente del módulo Administración.
Fuente: elaboración propia.

| No | Nombre de campo | Clasificación | Valor Nulo | Descripción |
|---|---|---|---|---|
| 1 | Nombre | campo de texto | No | Se aceptan letras minúsculas y mayúsculas. |
| 2 | 1er_apellido | campo de texto | No | Se aceptan letras minúsculas y mayúsculas. |
| 3 | 2do_apellido | campo de texto | No | Se aceptan letras minúsculas y mayúsculas. |
| 4 | correo | campo de texto | No | Se aceptan letras minúsculas y mayúsculas, el carácter @ es obligatorio luego el dominio al que pertenece el nombre del usuario del correo electrónico. |

Tabla 63. Descripción de variables del caso de prueba Buscar cliente.
Fuente: elaboración propia.

| **Escenario** | **Descripción** | **nombre** | **1er_apellido** | **2do_apellido** | **correo** | **contraseña** | **dirección** | **teléfono** | **fax** | **Respuesta del sistema** | **Flujo central** |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Tarea 11. Mostrarcliente. | Mostrar una cuenta de un cliente con los datos establecidos. | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | Del listado de clientes que existen se selecciona el cliente deseado mostrándose con todos sus datos. | 1. Dar clic en la opción “Clientes”.<br>2. Se muestra la interfaz con el listado de |

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | clientes.<br>3. Se seleccion a un cliente 4. Se pulsa en el botón “Detalles” .<br>5. Se muestra toda la informaci ón del cliente. |
| Tarea 12. Elimi narcli ente | Desactiv ar una cuenta de un cliente. | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | Se muestra un mensaje informando que se desactivó la cuenta del cliente. | 1. Dar clic en la opción “Eliminar” del listado de clientes.<br>2. Se muestra la interfaz para desactiva r al cliente.<br>3. Se pulsa en el botón “Eliminar”<br>4. Se muestra una alerta de verificaci ón de la desactiva ción.<br>5. Se pulsa en “Aceptar” y se desactiva el cliente. |

Tabla 64. Caso de prueba Configurar cliente del módulo Administración.
Fuente: elaboración propia.

| No | Nombre de campo | Clasificación | Valor Nulo | Descripción |
|---|---|---|---|---|

| | | | | |
|---|---|---|---|---|
| 1 | nombre | campo de texto | No | Se aceptan letras minúsculas y mayúsculas. |
| 2 | 1er_apellido | campo de texto | No | Se aceptan letras minúsculas y mayúsculas. |
| 3 | 2do_apellido | campo de texto | No | Se aceptan letras minúsculas y mayúsculas. |
| 4 | correo | campo de texto | No | Se aceptan letras minúsculas y mayúsculas, el carácter @ es obligatorio luego el dominio al que pertenece el nombre del usuario del correo electrónico. |
| 5 | contraseña | campo de texto | No | Se aceptan letras minúsculas, mayúsculas, números, signos y debe exceder de los 6 caracteres. |
| 6 | dirección | campo de texto | No | Letras y números. |
| 7 | teléfono | campo de texto | Si | Sólo números. |
| 8 | fax | campo de texto | Si | Sólo números. |

Tabla 65. Descripción de variables del caso de prueba Configurar cliente.
Fuente: elaboración propia.

| **Escenario** | **Descripción** | **nombre** | **Respuesta del sistema** | **Flujo central** |
|---|---|---|---|---|
| Tarea 13. Mostrar banco | Mostrar los datos de un banco seleccionado del listado de bancos. | N/A | Del listado de bancos existentes se selecciona el banco deseado que se muestra con sus datos. | 1. Dar clic en la opción Entidades bancarias.<br>2. Se muestra la interfaz con el listado de los bancos.<br>3. Se selecciona el banco de interés.<br>4. Se pulsa en el botón “Detalles”.<br>5. Se muestra el nombre del banco seleccionado. |

| | | | | |
|---|---|---|---|---|
| Tarea 14. Crear banco | Crear un banco insertando los datos establecidos. | V<br>Metropolitano | Se crea el banco con el dato nombre. Se muestra un mensaje informando que se creó el banco correctamente. | 1. Dar clic en la opción “Registrar entidad bancaria”.<br>2. Se muestra la interfaz con el campo nombre que se debe llenar.<br>3. Se pulsa en el botón “Aceptar”.<br>4. Se informa que se insertó el banco correctamente. |
| | | I<br>B32 | Se muestra un mensaje de error informando que el campo está escrito incorrectamente. | |
| | | I | Se muestra un mensaje de error alertando que debe llenar el campo nombre. | |
| Tarea 15. Modificar banco. | Modificar el nombre de un banco seleccionado. | V<br>BPA | Se muestra un mensaje informando que se modificó el nombre del banco antiguo por el actual. | 1. Dar clic en la opción “Actualizar entidad bancaria”.<br>2. Se muestra la interfaz con el campo a modificar.<br>3. Se pulsa en el botón “Aceptar”.<br>4. Se informa que se modificó correctamente el banco. |
| | | I<br>B23et5 | Se muestra un mensaje de error informando que el campo no cumple con el formato establecido. | |
| | | I | Se muestra un mensaje de error informando que debe llenar el campo nombre. | |
| Tarea 16. Eliminar banco. | Eliminar el banco seleccionado. | N/A | Se elimina el banco seleccionado. | 1. Dar clic en la opción “Eliminar”.<br>2. Se muestra la interfaz con la ventana de eliminación.<br>3. Clic en el botón “Eliminar”.<br>4. Se muestra una ventana de verificación de la eliminación.<br>5. Se oprime en “Aceptar”.<br>6. Se muestra un mensaje informando que se eliminó el banco. |

Tabla 66. Caso de prueba Gestionar banco. Fuente: elaboración propia.

| No | Nombre de campo | Clasificación | Valor Nulo | Descripción |
|---|---|---|---|---|
| 1 | nombre | campo de texto | No | |

Tabla 67. Descripción de variables del caso de prueba Gestionar banco.
Fuente: elaboración propia.

| Escenario | Descripción | correo | contraseña | Respuesta del sistema | Flujo central |
|---|---|---|---|---|---|
| EC 17. Autenticar administrador | Autenticar administrador insertando los datos establecidos. | V<br>dsfonseca@uci.cu | V<br>46dam*erl | Se autentica el usuario como administrador.<br>Se muestra un mensaje informando que el correo o la contraseña están incorrectos. | 1. Se muestra una interfaz de autenticación.<br>2. Se insertan el correo y la contraseña.<br>2. Clic en el botón “Iniciar sesión”.<br>3. Se autentica el administrador. |
| | | I<br>Dsfonsec.uci.cu | V<br>DA*34hyl | Se muestra un mensaje informando que el correo o la contraseña están incorrectos. | |
| | | V<br>damaris | I<br>P*A34 | | |

Tabla 68. Caso de prueba Autenticar administrador.
Fuente: elaboración propia.

| No | Nombre de campo | Clasificación | Valor Nulo | Descripción |
|---|---|---|---|---|
| 1 | correo | campo de texto | No | Se aceptan letras minúsculas y mayúsculas, el carácter @ es obligatorio luego el dominio al que pertenece el nombre del usuario del correo electrónico. |
| 2 | contraseña | campo de texto | No | Se aceptan letras minúsculas, mayúsculas, números, signos y debe exceder de los 6 caracteres. |

Tabla 69. Descripción de variables del caso de prueba Autenticar administrador.
Fuente: elaboración propia.

| Escenario | Descripción | nombre | 1er_apellido | 2do_apellido | correo | contraseña | dirección | teléfono | fax | Respuesta del sistema | Flujo central |
|---|---|---|---|---|---|---|---|---|---|---|---|
| EC 18. Modificarcliente. | Modifica los datos del cliente. | V damaris | V solis | V fonseca | V dsfonseca@uci.cu | V Pap1*987 | V Edificio 9 apto33 bayamo | V 481478 | V 457892 | Se modifican los datos de los campos seleccionados a modificar. | 1. Dar clic en la opción “Actualizar datos”.<br>2. Se muestran todos los campos que se pueden modificar(todos)<br>3. Se seleccionan los campos a modificar.<br>4. Se introducen los nuevos datos.<br>5. Se pulsa en el botón “Guardar”.<br>6. Se muestra un mensaje indicando que se modificaron los datos correctamente. |
| | | V damaris | V solis | V fonseca | V dsfonsecauci.cu | V Pap1*987 | V Edificio 9 apto33 bayamo | V 481478 | V 457892 | Se muestra un mensaje de error indicando que algún campo no cumple con el formato establecido. | |
| EC 19 Desactivar | Desactivar una cuenta de un | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | Se muestra un mensaje informando | 1. Dar clic en la opción |

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| cuenta de un cliente. | cliente. | | | | | | | | | que se desactivó la cuenta del cliente. | "Eliminar".<br>2. Se muestra la interfaz para desactivar al cliente.<br>3. Se pulsa en el botón "Eliminar"<br>4. Se muestra una alerta de verificación de la desactivación.<br>5. Se pulsa en "Aceptar" y se desactiva el cliente. |

Tabla 70. Caso de prueba Modificar cliente. Fuente: elaboración propia.

| No | Nombre de campo | Clasificación | Valor Nulo | Descripción |
|---|---|---|---|---|

| | | | | |
|---|---|---|---|---|
| 1 | nombre | campo de texto | No | Se aceptan letras minúsculas y mayúsculas. |
| 2 | 1er_apellido | campo de texto | No | Se aceptan letras minúsculas y mayúsculas. |
| 3 | 2do_apellido | campo de texto | No | Se aceptan letras minúsculas y mayúsculas. |
| 4 | correo | campo de texto | No | Se aceptan letras minúsculas y mayúsculas, el carácter @ es obligatorio luego el dominio al que pertenece el nombre del usuario del correo electrónico. |
| 5 | contraseña | campo de texto | No | Se aceptan letras minúsculas, mayúsculas, números, signos y debe exceder de los 6 caracteres. |
| 6 | dirección | campo de texto | No | Letras y números. |
| 7 | teléfono | campo de texto | Si | Sólo números. |
| 8 | fax | campo de texto | Si | Sólo números. |

Tabla 71. Descripción de variables del caso de prueba Modificar cliente.
Fuente: elaboración propia.

Anexo 5 Pruebas unitarias.

```
public string RealizarPago(string cuentaOrigen, string idtienda, double monto, string pin, string ccv, string fechavenc)
{
    var e = new Encriptador();
    var result = "";
    var usurioControl = new UsuarioControl();
    var cuentadestino = usurioControl.CuentaCliente(idtienda).idcuenta;
    var aux = TransferirDinero(cuentaOrigen, e.GetPasswordClearly(cuentadestino), pin, ccv, monto, fechavenc, "Pago");

    if (aux[0] == "Monto insuficiente" || aux[0] == "PIN incorrecto" || aux[0] == "Fecha de vencimiento incorrecta"
        || aux[0] == "Numero incorrecto" || aux[0] == "Conexión con el banco interrumpida" || aux[0] == "Codigo CCV incorrecto")
    {
        result = aux[0];
    }
    if (aux[0] == "OK" && aux[1] == "OK")
    {
        result = "OK";
    }
   EnviarCorreo(cuentaOrigen, "Pago");
    return result;
}
```

| **Prueba Unitaria** | |
|---|---|
| **Nombre de la Prueba** | *RealizarPago* |

| Estado | Tipo | Ultima Ejecución |
|---|---|---|
| Satisfactoria | Caja Blanca | 20/05/2012 |
| **Ejecutado por** | **Verificado por** | |
| Damaris Solis Fonseca | Wilfredo Roque Pérez | |
| **Descripción** | Para poder ejecutar la prueba se deben pasar varios tipos de datos string para los datos: número de cuenta origen, identificador de la tienda, PIN, CCV y fecha de vencimiento y un double para el monto. Con esos datos se realiza el pago. Una vez realizado el pago se devuelve el resultado del mismo. En caso de no realizarse muestra un mensaje indicando el error cometido en la operación. | |
| **Entrada** | número de cuenta, identificador de la tienda, monto, PIN, CCV y fecha vencimiento. | |
| **Criterio de aceptación** | Retorna una cadena de caracteres con el resultado de la transacción. | |

**Resultado:**

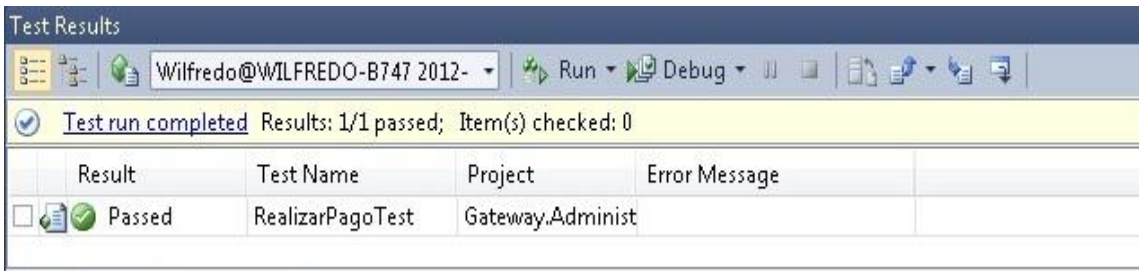



Tabla 72. Descripción de la prueba unitaria al método “RealizarPago”.
Fuente: elaboración propia.

```
public bool AutenticarCliente(string correo, string pass)
{

    var e = new Encriptador();
    var password = e.GetPasswordHash(pass);
    var lista = ListarClientesActivos();
    for (int i = 0; i < lista.Count; i++)
    {
        if (lista[i].correo == correo)
        {
            if (lista[i].contrasena == password )
            {
                if (lista[i].tipo_usuario == "Cliente")
                {
                      return true;
                }

            }

        }
    }
    return false;
}
```

| **Prueba Unitaria** | | |
|---|---|---|
| **Nombre de la Prueba** | *AutenticarCliente* | |
| **Estado** | **Tipo** | **Ultima Ejecución** |
| Satisfactoria | Caja Blanca | 10/04/2012 |
| **Ejecutado por** | **Verificado por** | |
| Damaris Solis Fonseca | Wilfredo Roque Pérez | |
| **Descripción** | Para poder ejecutar la prueba se deben pasar varios tipos de datos string para los datos: correo y contraseña. Con esos se autentica el cliente. En caso de introducirse mal esos datos se muestra un mensaje indicando el error cometido. | |
| **Entrada** | correo y contraseña. | |
| **Criterio de aceptación** | Retorna una variable booleana con el resultado de la autenticación. | |

**Resultado:**

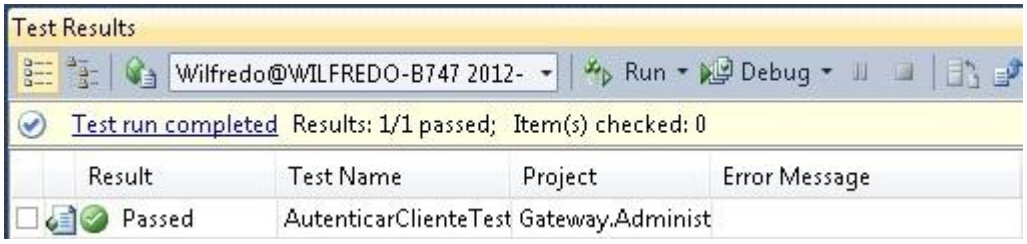



Tabla 73. Descripción de la prueba unitaria al método “AutenticarCliente”.
Fuente: elaboración propia.

```
public bool AdicionarCliente(string nombre, string primer_apellido, string segundo_apellido, string direccion,
    string correo, string fax, string telefono, string tipo_usuario, bool activo, string contrasena)
{
    var e = new Encriptador();
    var usuarios = new UsuarioDb();
    var passhistory = "1@" + e.GetPasswordHash(contrasena);
    var _mailsender = new MailSender("smtp.uci.cu", 25, "wroque@estudiantes.uci.cu",
        e.GetPasswordClearly("5bqhtL0KwXQ/AQh7sFHpRA=="), true);
    try
    {
        if (ExisteUsuario(correo))
        {
            var aux = ClienteDadoCorreo(correo);
            if (aux.activo)
            {
                return false;
            }
            else
            {
                var u = new usuario
                            {
                                idusuario = aux.idusuario,
                                activo = false,
                                contrasena = e.GetPasswordHash(contrasena),
                                correo = correo,
                                direccion = direccion,
                                fax = fax,
                                nombre = nombre,
                                primer_apellido = primer_apellido,
                                segundo_apellido = segundo_apellido,
                                telefono = telefono,
                                tipo_usuario = tipo_usuario,
                                password_history =  passhistory
                            };
                usuarios.UpdateCl(u);
                _mailsender.SendMail("wroque@estudiantes.uci.cu", correo, _mailsender.Asunto[1], _mailsender.Message[1]);
                return true;
            }
        }
        var user = new usuario
        {
            idusuario = Guid.NewGuid().ToString(),
            activo = false,
            contrasena = e.GetPasswordHash(contrasena),
            correo = correo,
            direccion = direccion,
            fax = fax,
            nombre = nombre,
            primer_apellido = primer_apellido,
            segundo_apellido = segundo_apellido,
            telefono = telefono,
            tipo_usuario =  tipo_usuario,
            password_history =  passhistory
        };
        usuarios.Insert(user);
        _mailsender.SendMail("wroque@estudiantes.uci.cu", correo, _mailsender.Asunto[1], _mailsender.Message[1]);
        return true;
    }
    catch (Exception)
    {
        return false;
    }
}
```

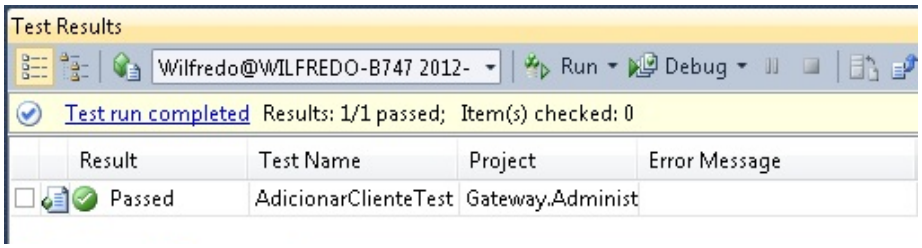
<table>
<tr><th colspan="3">Prueba Unitaria</th></tr>
<tr><td>Nombre de la Prueba</td><td colspan="2">AdicionarCliente</td></tr>
<tr><td>Estado</td><td>Tipo</td><td>Ultima Ejecución</td></tr>
<tr><td>Satisfactoria</td><td>Caja Blanca</td><td>12/04/2012</td></tr>
<tr><td>Ejecutado por</td><td colspan="2">Verificado por</td></tr>
<tr><td>Damaris Solis Fonseca</td><td colspan="2">Wilfredo Roque Pérez</td></tr>
<tr><td>Descripción</td><td colspan="2">Para poder ejecutar la prueba se deben pasar varios tipos de datos string para los datos: nombre, primer apellido, segundo apellido, fax, teléfono, dirección, correo, contraseña y tipo usuario. Se espera además un parámetro de tipo bool activo. Con esos datos se registran los usuarios. En caso de introducirse mal esos datos se muestra un mensaje indicando el error cometido.</td></tr>
<tr><td>Entrada</td><td colspan="2">nombre, primer apellido, segundo apellido, fax, teléfono, dirección, correo, contraseña, tipo usuario y activo.</td></tr>
<tr><td>Criterio de aceptación</td><td colspan="2">Retorna una cadena de caracteres con los datos del usuario.</td></tr>
<tr><td colspan="3">Resultado:<br> 
</td></tr>
</table>

Tabla 74. Descripción de la prueba unitaria al método “AdicionarCliente”.
Fuente: elaboración propia.

```
public bool ConfigurarPago(string correo, string idcuenta)
{
    var result = false;
    var e = new Encriptador();
    var usuarios = new UsuarioDb();
    var _mailsender = new MailSender("smtp.uci.cu", 25, "wroque@estudiantes.uci.cu", e.GetPasswordClearly("5bqhtL0KwXQ/AQh7sFHpRA=="),true);
    try
    {
        var u = ClienteDadoCorreo(correo);
        u.idtienda = e.GetPasswordHash(correo + " " + idcuenta);
        _mailsender.SendMail("wroque@estudiantes.uci.cu", correo, _mailsender.Asunto[2], _mailsender.Message[2] + u.idtienda);
        usuarios.ConfigPayment(u);
        result = true;
    }
    catch (Exception)
    {
        return false;
    }
    return result;
}
```

| **Prueba Unitaria** | | |
|---|---|---|
| **Nombre de la Prueba** | *ConfigurarPago* | |
| **Estado** | **Tipo** | **Ultima Ejecución** |
| Satisfactoria | Caja Blanca | 16/05/2012 |
| **Ejecutado por** | **Verificado por** | |
| Damaris Solis Fonseca | Wilfredo Roque Pérez | |
| **Descripción** | Para poder ejecutar la prueba se deben pasar varios tipos de datos string para los datos: correo e idcuenta. Con esos datos se configura el pago para poder realizar esta operación pago desde la tienda virtual. En caso de introducirse mal esos datos se muestra un mensaje indicando el error cometido. | |
| **Entrada** | Correo e idcuenta. | |
| **Criterio de aceptación** | Retorna en una variable booleana el resultado de la configuración del pago. | |

**Resultado:**

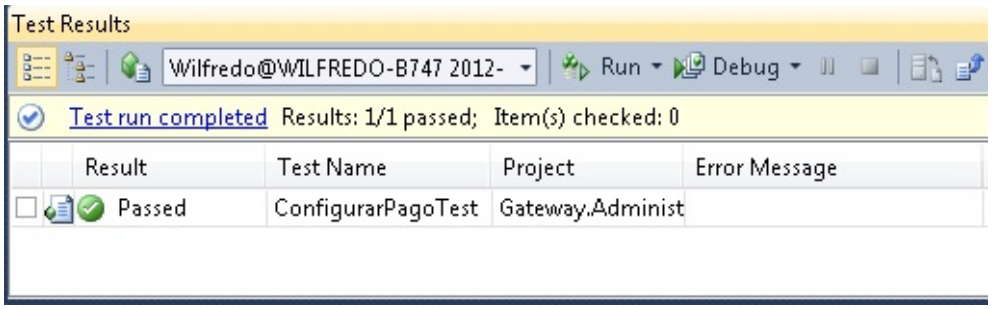



Tabla 75. Descripción de la prueba unitaria al método “ConfigurarPago”.
Fuente: elaboración propia.

```
public string ConsultarSaldo(string id, string ccv, string pin, string fechaVenc)
{
    string result = "";
    var p = new PuntoSalida();
    var isoMessage = new ISOMessage();
    var t = new TransaccionControl();
    var e = new Encriptador();
    try
    {
        var mensaje = isoMessage.BuildAuthRequest(id, "100010", pin, fechaVenc);
        var aux = p.EnviarISO(e.GetPasswordHash(mensaje));
        var response = p.EnviarISO(e.GetPasswordHash("HandShake"));
        var textoclaro = e.GetPasswordClearly(response);
        if (textoclaro == "No se pudo conectar a {0}:9898 localhost")
        {
            return "Conexión con el banco interrumpida";
        }
        result = t.InterpretarAuthResponseNumero(isoMessage.ParseMessage(textoclaro));
        if (result == "OK")
        {
            result = t.InterpretarAuthorizationResponseSaldo(isoMessage.ParseMessage(textoclaro));
        }
        else
        {
            result = t.InterpretarAuthResponseNumero(isoMessage.ParseMessage(textoclaro));
        }
    }
    catch
    {
        result = "No se ha podido contactar con el banco";
    }
    return result;
}
```

| **Prueba Unitaria** | | |
|---|---|---|
| **Nombre de la Prueba** | *ConsultarSaldo* | |
| **Estado** | **Tipo** | **Ultima Ejecución** |
| Satisfactoria | Caja Blanca | 15/05/2012 |
| **Ejecutado por** | **Verificado por** | |
| Damaris Solis Fonseca | Wilfredo Roque Pérez | |

| | |
|---|---|
| **Descripción** | Para poder ejecutar la prueba se deben pasar varios tipos de datos string para los datos: id, CCV, PIN y fecha de vencimiento. Con esos datos el cliente consulta el saldo. En caso de introducirse mal esos datos se muestra un mensaje indicando el error cometido. |
| **Entrada** | id, CCV, fecha vencimiento y PIN |
| **Criterio de aceptación** | Retorna en una variable booleana el resultado de la consulta del pago. |

**Resultado:**

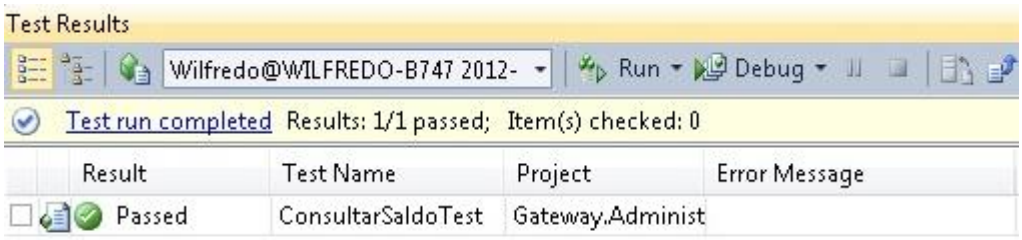



Tabla 76. Descripción de la prueba unitaria al método “ConsultarPago”.
Fuente: elaboración propia.

```
public bool ActualizarUsuario(string idusuario, string nombre, string primer_apellido, string segundo_apellido,
    string direccion, string correo, string fax, string telefono, bool activo)
{
    try
    {
        var u = new usuario();
        u.idusuario = idusuario;
        u.nombre = nombre;
        u.primer_apellido = primer_apellido;
        u.segundo_apellido = segundo_apellido;
        u.correo = correo;
        u.fax = fax;
        u.direccion = direccion;
        u.telefono = telefono;
        u.activo = activo;
        //u.tipo_usuario = tipo_usuario;

        var usuarios = new UsuarioDb();
        usuarios.Update(u);
        return true;

    }
    catch (Exception)
    {
        return false;
    }
}
```

| **Prueba Unitaria** | |
|---|---|
| **Nombre de la Prueba** | *ActualizarUsuario* |

| Estado | Tipo | Ultima Ejecución |
|---|---|---|
| Satisfactoria | Caja Blanca | 16/04/2012 |
| **Ejecutado por** | **Verificado por** | |
| Damaris Solis Fonseca | Wilfredo Roque Pérez | |
| **Descripción** | Para poder ejecutar la prueba se deben pasar varios tipos de datos string para los datos: idusuario, nombre, primer apellido, segundo apellido, dirección, fax, teléfono y correo. Se espera además el dato bool activo. Con esa información se actualizan los datos del cliente. En caso de introducirse mal esos datos se muestra un mensaje indicando el error cometido. | |
| **Entrada** | idusuario, nombre, primer apellido, segundo apellido, dirección, fax, teléfono, correo y activo. | |
| **Criterio de aceptación** | Retorna en una variable booleana el resultado de la actualización. | |

**Resultado:**

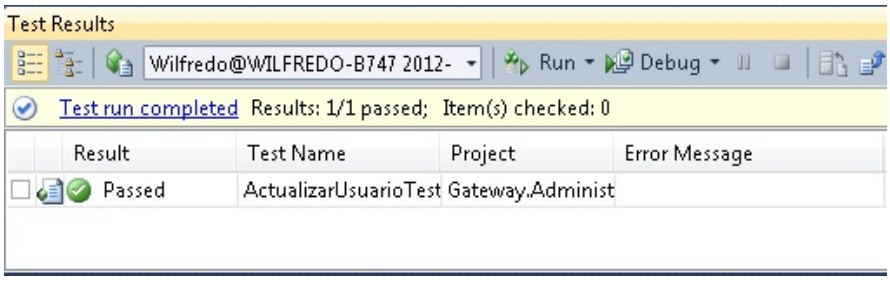



Tabla 77. Descripción de la prueba unitaria al método “ActualizarUsuario”.
Fuente: elaboración propia.

```
public List<usuario> ListarClientes()
{
    var usuarios = new UsuarioDb();
    var lista = usuarios.Load();
    var result = new List<usuario>();
    for (int i = 0; i < lista.Count; i++)
    {
        if (lista[i].tipo_usuario == "Cliente")
            result.Add(lista[i]);
    }
    return result;
}
```

| Prueba Unitaria | | |
|---|---|---|
| **Nombre de la Prueba** | *ListarClientes* | |
| **Estado** | **Tipo** | **Ultima Ejecución** |
| Satisfactoria | Caja Blanca | 16/04/2012 |
| **Ejecutado por** | **Verificado por** | |
| Damaris Solis Fonseca | Wilfredo Roque Pérez | |
| **Descripción** | Para poder ejecutar la prueba sólo se debe acceder a la lista de los clientes y mostrar esa lista. | |
| **Entrada** | - | |
| **Criterio de aceptación** | Retorna una lista de clientes con sus datos. | |

**Resultado:**

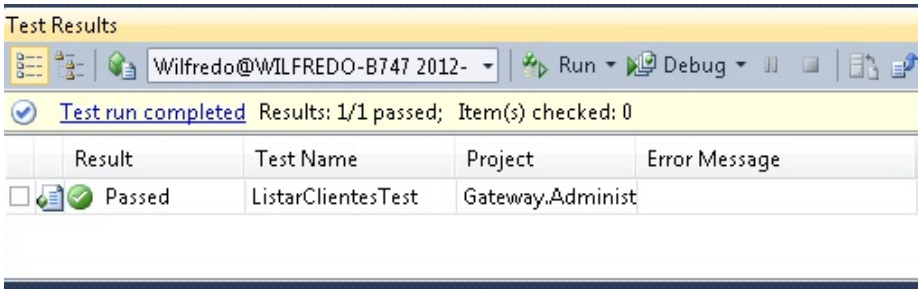



Tabla 78. Descripción de la prueba unitaria al método “ListarClientes”.
Fuente: elaboración propia.

```
public bool EliminarCliente(string id)
{
    try
    {
        var clientes = new UsuarioDb();
        var cuentaControl = new CuentaControl();
        var cuentas = cuentaControl.CuentaCliente(id);
        for (int i = 0; i < cuentas.Count; i++)
            cuentaControl.EliminarCuenta(cuentas[i].idcuenta);
        var c = ClienteDadoId(id);
        c.activo = false;
        clientes.Update(c);
        return true;
    }
    catch (Exception)
    {
        return false;
    }

}
```

| Prueba Unitaria | | |
|---|---|---|
| **Nombre de la Prueba** | *EliminarCliente* | |
| **Estado** | **Tipo** | **Ultima Ejecución** |
| Satisfactoria | Caja Blanca | 18/04/2012 |
| **Ejecutado por** | **Verificado por** | |
| Damaris Solis Fonseca | Wilfredo Roque Pérez | |
| **Descripción** | Para poder ejecutar la prueba se espera el identificador del cliente que se va a eliminar el dato es id de tipo string. Si existe ese id se elimina el cliente. | |
| **Entrada** | id | |
| **Criterio de aceptación** | Retorna en una variable booleana el resultado de la eliminación. | |

**Resultado:**

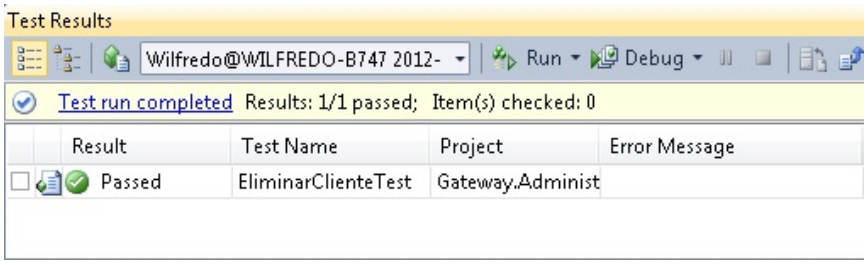



Tabla 79. Descripción de la prueba unitaria al método “EliminarCliente”.
Fuente: elaboración propia.

# Glosario de términos

Autenticidad: tener la certeza de que el mensaje, documento o transacción provienen exactamente de la persona que dice ser y es recibido por la persona a quien va destinado.

Catcha: en inglés *Completely Automated Public Turing test to tell Computers and Humans Apart*. Prueba desafío-respuesta utilizada para determinar cuándo el usuario es o no humano.

CCV: Código valor de validación o código valor de verificación en las tarjetas de débito o crédito.

Confidencialidad: garantía de que el contenido solo será conocido por el emisor y el destinatario.

Disponibilidad: consiste en asegurar que los usuarios autorizados tengan acceso continuo a información y recursos en el momento deseado.

DMZ: arquitectura *Screened Subnet*, conocida como red perimétrica o De-*Militarized Zone* (DMZ) es con diferencia la más utilizada e implantada hoy en día, ya que añade un nivel de seguridad en las arquitecturas de cortafuegos.

DSS: normas de seguridad de datos.

Factura electrónica: es un documento electrónico que tiene las mismas características que la factura tradicional y que garantiza la autenticidad de su origen y la integridad de su contenido.

Función hash: es un algoritmo matemático para identificar probabilísticamente un gran conjunto de información, dando como resultado un conjunto imagen finito.

Integridad: se refiere a las variaciones por supresión, adición o modificación del contenido enviado por el emisor y el destinatario

No repudio: asegurar técnicamente que el emisor y receptor no puedan negar el envío y la recepción de un documento, lo cual es conocido como no repudio en origen y destino para impedir perjuicios por la simple excusa de que nunca se recibió el mensaje o de que determinado texto nunca fue enviado.

PAN: *Personal Account Number*, es el número de la cuenta del usuario de la tarjeta. Es un campo numérico de 16 dígitos, que identifica al usuario como persona autorizada para acceder al servicio *online*.

PIN: en inglés *Personal Identification Number*. Número de identificación personal que es utilizado para obtener acceso a algo, o identificarse.

TICs: tecnologías de la información y las comunicaciones. Son aquellas herramientas computacionales e informáticas que procesan, almacenan, sintetizan, recuperan y presentan información representada de la más variada forma.