

**ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**



**Facultad de Ingeniería en Electricidad y Computación**

**Maestría En Sistemas De Información Gerencial**

**“IMPLEMENTACIÓN DE UNA APLICACIÓN TIPO BOTÓN DE PAGO DEL  
BANCO CENTRAL COMO UNA ALTERNATIVA DE PAGO A LAS  
TARJETAS DE CRÉDITO EN LOS COMERCIOS ELECTRÓNICOS”**

**TRABAJO DE TITULACIÓN**

**Previo la obtención del título de:**

**MAGÍSTER EN SISTEMAS DE INFORMACIÓN GERENCIAL**

**ANA KATIUSKA CRIOLLO CASSAGNE**

**GUAYAQUIL – ECUADOR**

**AÑO 2017**

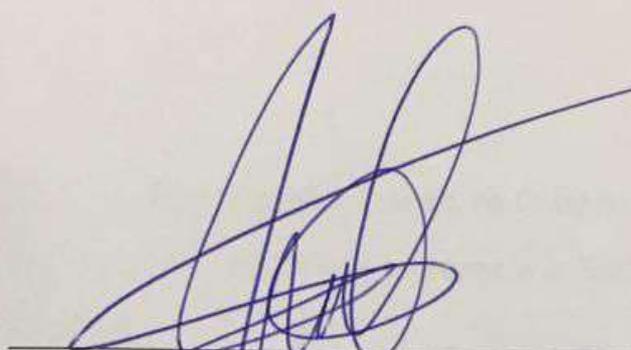
## **AGRADECIMIENTO**

Agradezco a Dios por guiar mis pasos y la oportunidad de estudiar en el MSIG. A mis padres Ana Cassagne y Oswaldo Criollo, por apoyarme en los momentos más difíciles. A mi esposo José López por darme el apoyo y motivación necesarios para culminar esta etapa. A mi amigo Daniel, mis profesores y compañeros de la XVII Promoción por compartir el conocimiento y experiencias muy enriquecedoras, que aportaron significativamente para alcanzar este objetivo.

## DEDICATORIA

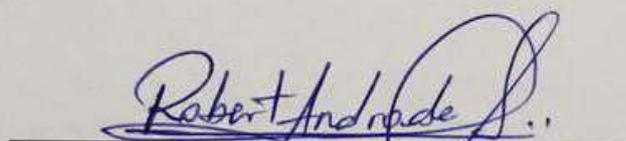
Dedico este proyecto a mi mamá y a mi hija Kaitlyn Elizabeth, que este título académico sea un estímulo en su vida. Las cosas **buenas** llegan a los que saben esperar. Las **mejores**, a los que no se rinden y luchan. Y las grandes **BENDICIONES**, a los que creen...

TRIBUNAL DE SUSTENTACIÓN



---

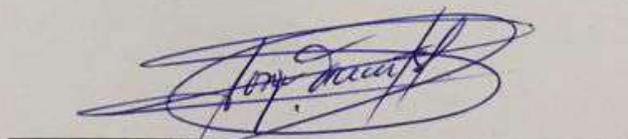
Mgs. Lenin Freire  
DIRECTOR MSIG



---

Mgs. Robert Andrade

DIRECTOR DEL PROYECTO DE  
GRADUACIÓN



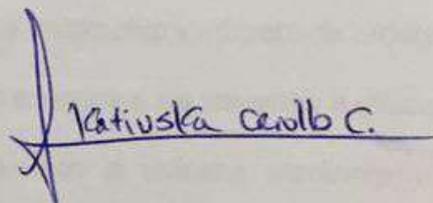
---

Mgs. Jorge Luis Fuentes  
MIEMBRO DEL TRIBUNAL

## DECLARACIÓN EXPRESA

"La responsabilidad del contenido de este Trabajo de titulación, nos corresponde exclusivamente; y el patrimonio intelectual de la misma a la ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL"

(Reglamento de Graduación de ESPOL)

  
Katiuska Cevallo C.

## RESUMEN

El objetivo de este proyecto es en esencia, proponer una solución que permita implementar una aplicación tipo Botón de pago del Banco Central como una alternativa de pago de producto o servicio con cargo a saldos en las tarjetas de crédito o débito en los comercios electrónicos.

Una de las opciones más utilizada por los comercios electrónicos es el cobro a través de tarjeta de crédito; actualmente se está incluyendo a la tarjeta de débito como forma de pago. La tarjeta de crédito es una identificación de material plástico, con banda magnética, microchip y número de tarjeta, emitida por un banco o entidad financiera que autoriza a los usuarios a utilizarla como medio de pago en los negocios adheridos al sistema, mediante su firma y exhibición. Esta modalidad de financiación obliga al usuario a devolver el importe dispuesto por las entidades autorizadas. Por otro lado las tarjetas de débito, identificación utilizada para efectuar operaciones financieras activas (incrementar el saldo), pasivas (disminuir el saldo). En este tipo de tarjeta el dinero se debita el saldo disponible de la cuenta del titular; en este caso no es el dinero que presta el banco como en las tarjetas de crédito.

El Dinero electrónico aun no muestra presencia significativa en el país; es una innovación en los servicios de pago del comercio electrónico, su potencial difusión y crecimiento es de alto impacto en la eficiencia de los procesos productivos de los hábitos de compra y consumo de la sociedad en general. Dinero electrónico es una forma de pago, implementado por el Banco Central del Ecuador, cuyo respaldo es el dinero físico de los usuarios que acceden al sistema abriendo una cuenta de Efectivo Desde Mi Celular (dinero electrónico), a través de su dispositivo móvil.

Sin duda alguna las formas de pago son un problema para los comercios electrónicos, muchos de los compradores se quejan de que no existe una forma de pago de fácil integración. Por tal motivo se realizará la implementación de este proyecto que fue dividido en dos etapas. La primera etapa es para desarrollar el aplicativo web botón de pago con dinero electrónico denominado “Botón de Pago”, un proceso de servicio de pago de una compra electrónica acoplada al sitio web del comercio electrónico. La segunda etapa es realizar las pruebas de CERTIFICACIÓN, ejecutando el PLAN DE PRUEBAS. En el cual se detallarán los tipos de pruebas a realizar, tales como las pruebas de SISTEMAS, CARGA, CONCURRENCIA Y RENDIMIENTO.

## ÍNDICE GENERAL

AGRADECIMIENTO.....	i
DEDICATORIA.....	ii
TRIBUNAL DE SUSTENTACIÓN.....	iii
DECLARACIÓN EXPRESA.....	iv
RESUMEN .....	v
ÍNDICE GENERAL.....	vii
ÍNDICE DE FIGURAS .....	xiii
ÍNDICE DE TABLAS .....	xv
INTRODUCCIÓN .....	xvi
CAPÍTULO 1.....	1
1.1. ANTECEDENTES .....	1
1.2. DESCRIPCIÓN DEL PROBLEMA .....	3
1.3. SOLUCIÓN PROPUESTA.....	5
1.4. OBJETIVO GENERAL.....	11
1.5. OBJETIVOS ESPECÍFICOS.....	11
1.6. METODOLOGÍA.....	12
CAPÍTULO 2.....	15
2.1. COMERCIO ELECTRÓNICO .....	15
2.2. MEDIOS DE PAGO .....	17

2.3.	CONCEPTOS DE DINERO ELECTRÓNICO.....	20
2.4.	CONCEPTOS Y BENEFICIOS DEL DINERO ELECTRÓNICO DEL BANCO CENTRAL.....	22
2.5.	HERRAMIENTAS OPEN SOURCE PARA LA IMPLEMENTACIÓN DE BOTÓN DE PAGO.....	23
2.6.	METODOLOGÍAS DE PRUEBAS DE SOFTWARE.....	32
2.7.	METODOLOGÍAS PARA DISEÑOS DE API.....	38
2.8.	METODOLOGÍAS PARA DISEÑOS DE WEB SERVICES.....	41
CAPÍTULO 3.....		46
3.1.	LEVANTAMIENTO DE REQUERIMIENTOS.....	46
3.2.	LEVANTAMIENTO DE INFORMACIÓN.....	49
3.3.	DEFINICIÓN DE LA SITUACIÓN ACTUAL.....	51
3.4.	ALCANCE DEL PROYECTO.....	51
CAPÍTULO 4.....		53
4.1.	FASE I: ANÁLISIS Y DISEÑO DEL BOTÓN DE PAGO CON DINERO ELECTRÓNICO.....	53
4.1.1.	ANÁLISIS DE REQUERIMIENTOS.....	53
4.1.2.	DISEÑO DE COMPONENTES.....	58
4.1.3.	DISEÑO DE INTERFACES DE USUARIOS.....	59
4.1.4.	ARQUITECTURA DE APLICACIÓN.....	61

4.1.5. DESARROLLO DE LA SOLUCIÓN PARA BOTÓN DE PAGO CON DINERO ELECTRÓNICO .....	61
4.2. FASE II: PLAN DE PRUEBAS .....	64
4.2.1. INTRODUCCIÓN .....	64
4.2.2. ARTEFACTOS DE PRUEBA.....	65
4.2.3. CARACTERÍSTICAS A SER PROBADAS .....	66
4.2.4. CARACTERÍSTICAS QUE NO SERÁN PROBADAS .....	67
4.2.5. TIPOS DE PRUEBAS .....	67
4.2.6. PROCESOS DE PRUEBA .....	68
CAPÍTULO 5.....	70
6.1. DESARROLLO E IMPLEMENTACIÓN DE CLIENTE DE BOTÓN DE PAGO .....	70
6.2. DESCRIPCIÓN DEL ENTORNO DE PRUEBAS .....	72
6.3. PRUEBAS INTERNAS DE RENDIMIENTO .....	74
6.4. PRUEBAS INTERNAS DE CONCURRENCIA.....	77
6.5. PRUEBAS INTERNAS DE CARGA .....	79
6.6. PRUEBAS DE ACEPTACIÓN CON LOS USUARIOS .....	82
CAPÍTULO 6.....	84
6.1. EVALUACIÓN Y ANÁLISIS DE RESULTADOS .....	84
6.2. PROPUESTAS INICIALES DE ALTERNATIVAS DE MEJORA .....	86
CONCLUSIONES Y RECOMENDACIONES .....	88

BIBLIOGRAFÍA.....	90
GLOSARIO.....	92
ANEXOS .....	94

## ABREVIATURAS Y SIMBOLOGÍA

**API** Application Programming Interface, Interfaz de programación de aplicaciones.

**CIL** Common Intermediate Language, Lenguaje intermedio común.

**CLR** Common Language Runtime, Entorno en tiempo de ejecución de lenguaje común.

**EDI** Electronic Data Interchange, Intercambio electrónico de datos.

**HTTP** Hypertext Transfer Protocol, Protocolo de transferencia de hipertexto.

**ISTQB** International Software Testing Qualifications Board, Organización Internacional de Certificación de Calidad del Software.

**JDBC** Java Database Connectivity. API para ejecución de operaciones sobre bases de datos.

**OCDE** Organización para la Cooperación y el Desarrollo Económicos.

**OLTP** OnLine Transaction Processing, Procesamiento de transacciones en línea.

**SMS** Short Message Service, Servicio de mensajes cortos.

**SMTP** Simple Mail Transfer Protocol, Protocolo para transferencia simple de correo.

**SOAP** Simple Object Access Protocol, Protocolo simple de acceso a objetos.

**SSL** . Secure Sockets Layer, Protocolo para transmitir información a las aplicaciones.

**TPV** Terminal Punto de Venta.

**UDDI** Universal Description, Discovery and Integration, Catálogo de negocios de Internet.

**URI** Uniform Resource Identifier, Identificador de recursos uniformes.

**URL** Uniform Resource Locator, Localizador Uniforme de Recursos.

**VPN** Virtual Private Network, Red privada virtual.

**WCF** Windows Communication Foundation, Marco de trabajo para aplicaciones orientadas a servicios.

**WSDL** Web Services Description Language, Lenguaje de descripción del servicio web.

**XML** eXtensible Markup Language, Lenguaje de marcas extensible.

## ÍNDICE DE FIGURAS

Figura 1.1 Usuario comprando en línea .....	2
Figura 1.2 Comercios electrónicos.....	2
Figura 1.3 Prácticas comerciales fraudulentas.....	4
Figura 1.4 Mecanismos de cobro del comercio.....	4
Figura 1.5 Clientes del comercio sin atender .....	5
Figura 1.6 Servicio de pago con Dinero electrónico .....	6
Figura 1.7 Forma de pago Dinero electrónico .....	6
Figura 1.8 Diseño de arquitectura del API Botón de pago.....	9
Figura 1.9 Modelo ciclo de vida de Desarrollo .....	10
Figura 1.10 Metodología ágil SCRUM.....	13
Figura 2.1 Modelo ciclo de vida de Desarrollo - V .....	34
Figura 4.1 Diagrama de objetos API Botón de Pagos .....	54
Figura 4.2 Diagrama UML Compra en línea y pago con dinero electrónico.....	55
Figura 4.3 Diagrama de Componente API Botón de pago .....	59
Figura 4.4 Pantalla de compras en línea diseño Botón de pago .....	59
Figura 4.5 Pantalla ingreso de datos para pago.....	60
Figura 4.6 Pantalla de consulta de transacciones con dinero electrónico .....	60
Figura 4.7 Diagrama de Arquitectura API Botón de pago dinero electrónico.....	61
Figura 4.8 Product Backlog y Sprint Backlog del proyecto .....	65
Figura 4.9 Taskboard del proyecto .....	66
Figura 4.10 Burndown Chart del proyecto.....	66
Figura 5.1 Servicio de cobro de dinero electrónico .....	74
Figura 5.2 Trama de ejemplo.....	75

Figura 5.3 Envío de transacciones en REST Console .....	76
Figura 5.4 Consulta de transacciones en Base de datos .....	77
Figura 5.5 Vista de resultados JMETER .....	78
Figura 5.6 Árbol de resultados JMETER.....	78
Figura 5.7 Reporte sumariada JMETER.....	79
Figura 5.8 Vista de servicio de Dinero electrónico en WCF Test Client.....	79
Figura 5.9 Trama de ejemplo 1 .....	80
Figura 5.10 Trama de ejemplo 2 .....	81
Figura 5.11 Trama de ejemplo 3.....	81
Figura 5.12 Pantalla de compra en línea y botón de pago .....	82
Figura 5.13 Pantalla de datos de pago de transacción .....	83
Figura 5.14 Pantalla de mensaje satisfactorio.....	83
Figura 6.1 Gráfico de transacciones realizadas con dinero electrónico.....	85
Figura 6.2 Beneficios de dinero electrónico .....	86
Figura 6.3 Seguridad de sistema .....	86
Figura 6.4 Inclusión financiera .....	87

## ÍNDICE DE TABLAS

Tabla 1 Actores, Roles y Caso de uso de la transacción .....	56
Tabla 2 Caso de uso 1 .....	56
Tabla 3 Caso de uso 2 .....	56
Tabla 4 Caso de uso 3 .....	57
Tabla 5 Caso de uso 4 .....	57
Tabla 6 Caso de uso 5 .....	57
Tabla 7 Caso de uso 6 .....	57
Tabla 8 Objetos de API Botón de Pagos dinero electrónico .....	63
Tabla 9 Métodos del servicio de pago de la solución.....	68
Tabla 10 Hardware del ambiente de pruebas .....	72
Tabla 11 Software del sistema.....	73
Tabla 12 Otras configuraciones del sistema.....	73
Tabla 13 Herramientas de pruebas.....	73
Tabla 14 Tramas de ejemplo REST Console .....	76
Tabla 15 Código de respuesta de transacciones.....	77

## INTRODUCCIÓN

Al término de la maestría, en noviembre del 2016, me reuní con el Magister Juan Carlos García a quien le presente la propuesta de realizar la implementación de una aplicación tipo Botón de Pagos con forma de pago Dinero electrónico para incluir a usuarios desatendidos por los comercios electrónicos de nuestro país.

Luego de presentar el tema aprobado me reuní con mi tutor de tesis asignado el Magister Robert Andrade quien mostró interés en la propuesta de la nueva forma de pago con dinero electrónico, para comercios electrónicos, aplicación tipo Botón de Pago del Banco Central. La compra en línea de artículos y servicios, resulta atractiva para los compradores, sin embargo, es importante tomen las debidas precauciones y así evitar ser víctimas de prácticas comerciales fraudulentas. Los compradores aún no encuentran una forma de pago totalmente segura y confiable, para poder realizar compras en línea. Las formas de pago existentes de los comercios electrónicos no logran brindar las garantías necesarias, referidas a las transferencias de dinero y manipulación de datos de sensibles, requeridos para el proceso de compra en línea.

Del análisis realizado a esta problemática se propone como solución la implementación de un Botón de Pago con dinero electrónico acoplado al proceso de compra en línea en un sitio web del comercio electrónico. Esta forma de pago con Dinero electrónico, permite al usuario del comercio pagar la compra realizada, mediante la transferencia de dinero desde las cuentas asociadas de origen hacia el destino, en Banco Central del Ecuador. Al dar clic en el botón de pago se solicitará ingresar número de identificación y número celular, esta información es enviada al servicio del Banco Central para realizar el cobro de la transacción. El comprador recibe la notificación de confirmación de pago vía SMS, quien deberá responder con su clave única, para la autorizar el cobro. Esto hace que el sistema sea seguro y confiable porque brinda al usuario del comercio el control total de la transacción desde su dispositivo móvil.

# **CAPÍTULO 1**

## **GENERALIDADES**

### **1.1. ANTECEDENTES**

El comercio electrónico (e-commerce) permite la compra y venta de productos a través de internet, utiliza como forma de pago los medios electrónicos, como las tarjetas de crédito o débito. Permite realizar transferencia de fondos electrónicos, procesamiento de transacciones en línea (OLTP), intercambio electrónico de datos (EDI), entre otros.

Los sitios web de los comercios electrónicos ofertan sus productos para que el usuario puede realizar las compras en línea desde la tranquilidad de su hogar, sin la asistencia o presión de un vendedor. Brinda al usuario facilidades para la entrega del producto, ahorrando tiempo y esfuerzo, el usuario prefiere este servicio por comodidad.



*Figura 1.1 Usuario comprando en línea*

El comercio electrónico ha crecido de manera extraordinaria debido al acceso al Internet desde los hogares, la tenencia de Smartphone o dispositivos móviles y la reducción del analfabetismo digital en los últimos años. América Latina es una de las regiones del mundo con mayor crecimiento en sus flujos comerciales a través de internet, representa el 7% del comercio electrónico global. En 2013 el e-commerce en Latam se duplicó, registrando ventas por US\$43 mil millones.



*Figura 1.2 Comercios electrónicos*

En el Ecuador el comercio electrónico toma fuerza, puesto que los servicios que ofrecen las aerolíneas, hoteles, entidades financieras y otras, se

multiplican en las páginas web. En el 2015 los dueños de los comercios electrónicos que ganaron el E-Commerce Awards Ecuador fueron LAN.COM, COMANDATO.COM, CINEMARK, MOVISTAR, PACIFICARD, JOLIE. GRUPO CENTRICO, CUPONASO, BANCO BOLIVARIANO; ellos obtuvieron la mayoría de usuario que realizan transacciones o compras a través de computador o teléfonos móviles, este tipo de transacciones crecen a una tasa promedio entre el 18% y 35% según el sector desde 2013.

## **1.2. DESCRIPCIÓN DEL PROBLEMA**

El comercio electrónico es definido por los estudios de la Organización para la Cooperación y el Desarrollo Económicos (OCDE) como el proceso de compra, venta o intercambio de bienes, servicios e información a través de las redes de comunicación. Representa la posibilidad de adquirir bienes o servicios ofrecidos por proveedores en diversas partes del mundo.

La compra en línea de artículos y servicios resulta atractiva para los usuarios del comercio, debido a las facilidades que ofrece, sin embargo, es importante tomar las debidas precauciones, para evitar ser víctimas de prácticas comerciales fraudulentas. El usuario del comercio electrónico no encuentra una forma de pago totalmente segura y confiable, los mecanismos de seguridad ofrecidos actualmente, exponen los datos personales en la red, motivo por el cual los usuarios deciden no realizar compras en línea. Las formas de pago existentes no brindan las garantías necesarias, referidas a las transferencias de dinero y manipulación de datos sensibles requeridos en el proceso de compra en línea.



*Figura 1.3 Prácticas comerciales fraudulentas*

Los proveedores ofrecen a los usuarios del comercio electrónico, mecanismos de cobros que integran componentes, entre estos están PayPal o Google Wallet. Estos mecanismos requieren de un esquema de autenticación, cuentas de correos, registro de tarjetas y números de control que forman parte del plástico que identifica al usuario. Obtener toda la información necesaria para realizar el pago de la transacción, en ocasiones tiende a confundir al usuario porque no logra completar los datos requeridos. Quedándose sin la posibilidad de realizar las compras en línea.



*Figura 1.4 Mecanismos de cobro del comercio*

En Ecuador aproximadamente el 40% de la población económicamente activa (PEA) no está incluida financieramente; es decir, no forma parte activa del

sistema financiero nacional de comercios electrónicos. Esto quiere decir que hay un nicho de mercado desatendido, el cual por obvias razones no realiza compras en línea y podría estar interesado en nuevas formas de pago que brinden seguridad y sean de fácil utilización, de esta manera este servicio será repotenciado.



*Figura 1.5 Clientes del comercio sin atender*

### **1.3. SOLUCIÓN PROPUESTA**

Dinero electrónico es una innovación en los servicios de pago del comercio electrónico, su potencial difusión y crecimiento es de alto impacto en la eficiencia de los procesos productivos de los hábitos de compra y consumo de la sociedad en general. La forma de pago dinero electrónico, aún no muestra una presencia significativa; sin embargo el uso de dispositivos móviles y el impulso de iniciativas internacionales como la inclusión financiera, es de apoyo relevante para la cobertura de los servicios financieros y pagos en la población.



*Figura 1.6 Servicio de pago con Dinero electrónico*

El usuario del comercio electrónico, requiere de su número de cédula, número celular, para realizar el pago de la compra con su dinero electrónico. El servicio del Banco Central del Ecuador recibe y envía información de la cuenta del usuario asociada. El usuario recibe un SMS para que realice confirmación del pago de la transacción, enviando como respuesta su clave única de usuario que autoriza el cobro de la transacción.



*Figura 1.7 Forma de pago Dinero electrónico*

En el informe de Estado de la Industria 2013, servicios financieros móviles para los no bancarizados de la GSM Association, el dinero electrónico podría

ser una plataforma de acceso a servicios financieros, que transforme por entero la economía de los países. Hasta la fecha, al menos 110 sistemas de pagos móviles se han desarrollado en todo el mundo con alrededor de 40 millones de usuarios. Según el Rastreador de Servicios de Dinero Móvil de GSMA, el dinero móvil continúa profundizando la inclusión financiera. En 2015 el número de servicios de dinero móvil se incrementó. De acuerdo con los datos del Banco Mundial sobre inclusión financiera global, los servicios de dinero móvil están disponibles en un 85% de los países donde el número de personas con una cuenta en una institución financiera es inferior al 20%.

Del análisis de la descripción del problema, se propone como solución la implementación de una aplicación tipo Botón de Pago del Banco Central (API de servicio de pago), que permitirá realizar el proceso de cobro de transacciones en línea desde un sitio web de comercio electrónico.

La implementación se realizará con tecnología Microsoft .NET Framework 4.5, con un Servidor de aplicaciones Windows Server 2012, un Motor de acceso a datos de SQL Server 2008 R2, y un Motor de servidor web se utilizará de forma nativa el servidor IIS versión 8.0 donde se habilitarán las comunicaciones HTTP y HTTPS para la ejecución de servicios Web y WCF.

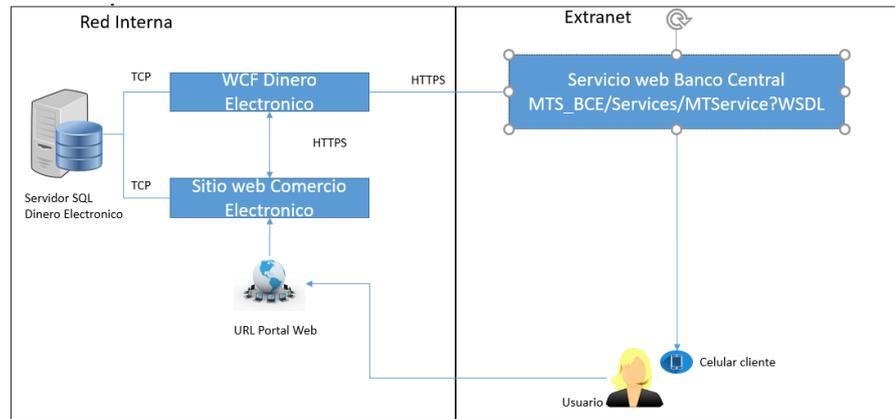
## **FASE I**

Se desarrollará un aplicativo denominado “Botón de Pago” acoplado al sitio web de comercio electrónico, el cual consistirá en la realización de un proceso de servicio de pago con Dinero electrónico de una compra en línea.

El proceso de servicio de pago con Dinero electrónico es el siguiente:

- El usuario del comercio accede al portal del comercio electrónico y selecciona la lista de productos a comprar, agregando cada ITEM al carrito de compra.
- Se realiza el proceso de confirmación del pedido y el CheckOut de la compra.
- Se selecciona la forma de pago con Dinero electrónico, se mostrara la Interface API para el ingreso de datos del usuario, estos son el número de identificación y el número de celular, información que está asociada a la cuenta del usuario en Banco Central.
- El API de servicio de pago se conecta con el servicio web del Banco Central, el cual realiza la petición al servicio de cobro con los datos ingresados anteriormente.
- El Banco Central solicita la confirmación al usuario del comercio mediante un mensaje de texto enviado al número de celular registrado en su cuenta asociada.
- El usuario del comercio confirma el pago de la compra en línea, con la respuesta del mensaje con su clave única (número de PIN) al Banco Central.
- El API de servicio de pago mediante un sensor está esperando respuesta del cobro de la transacción, la cual puede ser exitosa o fallida. Una vez que el Banco Central confirma el pago mediante el servicio web. El Portal mostrará al usuario un mensaje de proceso satisfactorio, en el caso de que la respuesta del Banco fue transacción

aprobada. Caso contrario se mostrará una respuesta de mensaje de error si la respuesta del Banco fue transacción rechazada.



*Figura 1.8 Diseño de arquitectura del API Botón de pago*

## FASE II

Las pruebas se realizarán en un ambiente de CERTIFICACIÓN, en el que se ejecutará el PLAN DE PRUEBAS REQUERIDO. En el cual se detallarán los tipos de pruebas a realizar, tales como las pruebas FUNCIONALES, DE CARGA, CONCURRENCIA Y RENDIMIENTO del APLICATIVO Botón de Pagos, además de la INTEGRACIÓN de cada uno de los COMPONENTES, evaluando que el SOFTWARE sea 100% de CALIDAD.

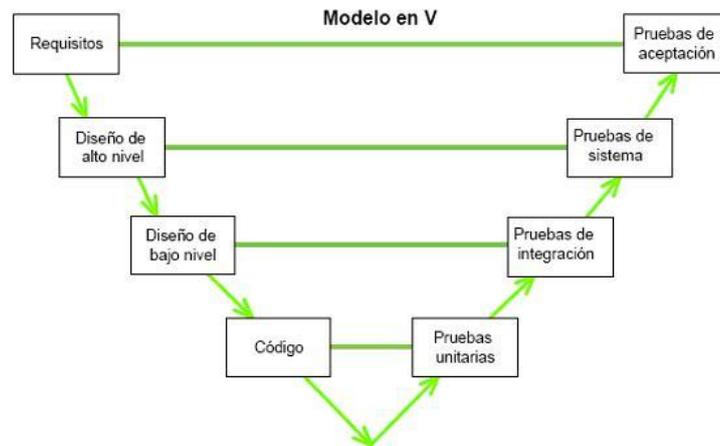


Figura 1.9 Modelo ciclo de vida de Desarrollo

Entre los **BENEFICIOS** de la solución Botón de pago con dinero electrónico tenemos:

- Permitirá realizar el **pago de las compras en línea** de una forma más **rápida, segura**, otorgando al **usuario del comercio** el control del proceso desde su inicio, **en el aplicativo** y posterior confirmación del pago desde su **dispositivo móvil**.
- El botón de pagos con **dinero electrónico es una forma de pago segura**, debido a la encriptación de los datos sensibles del usuario que viajarán por la red hasta obtener la respuesta del servicio del Banco Central del Ecuador.
- Este servicio que ofrece el Banco Central del Ecuador estará **disponible las 24 horas** del día, sin limitación de tiempo y podrá realizar el pago de las compras en línea desde la comodidad de su hogar a través de su dispositivo móvil celular.

- El gobierno dispuso que la **utilización del dinero electrónico como forma de pago abaratará costos, como la reducción de la tarifa del IVA**, pues se trata de un servicio público, el cual permitirá la inclusión financiera.
- El servicio de cobro al ser administrado por el Banco Central del Ecuador garantiza la interoperabilidad, es decir **puede acceder desde un número de teléfono de cualquier operadora**, sin consumir saldo del celular, ni mensajes.

#### 1.4. OBJETIVO GENERAL

Implementar una aplicación tipo Botón de Pago del Banco Central como una alternativa de pago a las tarjetas de crédito en los comercios electrónicos.

#### 1.5. OBJETIVOS ESPECÍFICOS

- Establecer las generalidades de la tesis.
- Definir conceptos y herramientas sobre la propuesta realizada.
- Definir la situación actual y Realizar el levantamiento de información.
- Implementar un aplicativo de Botón de Pago mediante Dinero Electrónico.
- Realizar las pruebas al Botón de Pago.
- Analizar los resultados de las pruebas realizadas.

## 1.6. METODOLOGÍA

La metodología ÁGIL seleccionada para el desarrollo de este proyecto es la más conocida, se denomina “SCRUM”; tiene la ventaja de ser flexible a los cambios dentro de cualquier etapa. Presentando al usuario un avance preliminar del sprint evaluado por el equipo de trabajo, de esta manera maximizamos los beneficios. SCRUM tiene las siguientes características:

1. Individuos e interacciones sobre procesos y herramientas.
2. Software funcionando sobre documentación extensiva.
3. Colaboración con el cliente sobre negociación contractual.
4. Respuesta ante el cambio sobre seguir un plan.

Adicionalmente SCRUM tiene los siguientes principios esenciales:

- a) Priorización: ordenamiento periódico de las necesidades y deseos contradictorios, es vital para el éxito.
- b) Responsabilidad: todos responden por sus decisiones o acciones realizadas.
- c) Inspeccionar y Adaptar: en intervalos regulares de tiempo el equipo revisará su progreso y realizará ajustes en sus avances.
- d) Ritmo: los miembros de los equipos se esforzarán en desarrollar una cadencia regular de sus acciones.
- e) Retroalimentación: los miembros del equipo aceptan y reciben nueva información sobre sus circunstancias en el entorno.

- f) Colaboración: La cooperación, el esfuerzo por construir se da utilizando los talentos y las ideas de cada uno de los miembros del equipo.
- g) Auto organización: Los miembros del equipo podrán movilizar sus esfuerzos con respecto a sus metas y mantenerse centrados en su objetivo.
- h) Foco: se debe dar tiempo y espacio a los miembros del equipo para que se concentren.



Figura 1.10 Metodología ágil SCRUM

SCRUM no proporciona una lista completa de actividades, artefactos y roles para entregar un producto terminado, se basa en el mínimo marco necesario para que el proyecto salga adelante con la eficacia del equipo, entregando iterativamente un producto exitoso. Mientras realiza las mejoras en el enfoque del desarrollo del producto en la organización.

SCRUM está asentado en: 1. Transparencia, 2. Inspección, 3. Adaptación; eliminar o disminuir, alguno de estos tres elementos del control del proceso empírico, debilita la eficacia de SCRUM.

## CAPÍTULO 2

### MARCO TEÓRICO

#### 2.1. COMERCIO ELECTRÓNICO

El comercio electrónico se refiere a las transacciones de compra y/o venta de un producto y/o servicio a través del Internet. También conocido como e-commerce, el cual aprovecha las herramientas tecnológicas para realizar las transacciones.

En el comercio electrónico intervienen cuatro agentes:

**El vendedor**, quien ofrece sus productos o servicios a través de internet.

**El comprador**, quien adquiere los productos o servicios ofertados por el proveedor a través de Internet.

**El gestor de medios de pago** permite establecer los mecanismos de pago para que el vendedor reciba el dinero que transfiere el cliente al vendedor a cambio de los productos o servicios.

**La entidad de certificación** garantiza mediante un certificado electrónico que los agentes que intervienen en el proceso de la transacción electrónica son quienes dicen ser. En el sistema de comercio electrónico, el comprador selecciona un producto en la página web que ofrece el servicio, luego de ingresa sus datos y elige el método de pago correspondiente.[1]

El e-commerce ha motivado a las empresas a este modelo de negocio, introduciendo nuevas clasificaciones a partir de los participantes y objetivos de cada compra o venta.

1. **B2C o Business to Client (Negocio a Cliente)**, Este modelo hace referencia a las transacciones realizadas entre una empresa y el cliente final de forma directa. Podemos observar ejemplos de comercio electrónico B2C cuando adquirimos productos o servicios a través de un catálogo en una tienda virtual.
2. **B2B o Business to Business (Negocio a Negocio)**, En este tipo de negocio, tanto los clientes como proveedores del producto o servicio son empresas. Los negocios suelen usar este tipo de e-commerce para reducir los costos, lograr mejoras en la comunicación y obtener un mayor control de las ventas.
3. **B2E o Business to Employee (Negocio a empleado)**, en cambio, se enfoca en la relación comercial entre la misma empresa y sus colaboradores a través de una plataforma interna. Los negocios aprovechan este tipo de e-commerce para motivar al trabajador y fidelizarlo con la empresa.

4. **C2C (Consumer to consumer)**, por otro lado, los consumidores también pueden participar de transacciones con otros consumidores a través del comercio. Este tipo de e-commerce se ha visto popularizado por páginas web que sirven como vitrina para que las personas expongan sus propios productos, por lo general, de segunda mano.
5. **G2C o Government to Consumer (Gobierno a Consumidor)**, el cual se refiere a los trámites en línea que realizan los consumidores con su gobierno municipal o estatal a través de portales online. [2]

## **2.2. MEDIOS DE PAGO**

El éxito del comercio electrónico se debe a la facilidad de realizar el pago para el cliente, cuanto más sencillo sea, más favorable será la compra. Existen los siguientes tipos de pago:

### **2.2.1. PAGOS ONLINE**

Estos medios de pago permiten realizar transferencias de dinero entre usuarios comprador/vendedor que deseen ofrecer/adquirir un producto/servicio en internet. Procesa pagos con tarjetas de crédito/debito, cuentas de ahorro/corriente y efectivo, a través de una plataforma que canaliza el pago, verificando y validando cada transacción.

#### **1. TRANSFERENCIA BANCARIA**

Es un método de pago que permite realizar movimientos de dinero de una cuenta a otra, traspasando fondos entre cuentas bancarias

sin sacar físicamente el dinero. Al realizar un pedido en una tienda online por transferencia bancaria se envía al cliente un código de pedido que deberá incluir en la orden de traspaso.

Este método es utilizado por usuarios que no se fían de introducir los datos de su tarjeta de crédito en la página web.

## **2. PASARELA DE PAGO**

Es un proveedor de servicios para comercios electrónicos, que autoriza pagos. Equivalente a un TPV (Terminal Punto de Venta) de una tienda física.

Las opciones más comunes son Redsys (anteriormente llamada Sermepa) para Servired y la pasarela Pasat 4B. Es el método de pago online más común.

## **3. PAGO POR MÓVIL**

Este método de pago se va instaurando en el mercado, existen modelos para la validación de las compras online desde el móvil. Mediante chip NFC (Near Field Communications) y técnicas de acercamiento entre dispositivos se puede proceder con mecanismos de pagos seguros.

Google Wallet permite transferir dinero desde la cuenta de usuario del sistema, cuando se realiza una operación, se hace separando las finanzas personales del cliente. Es muy útil, para pagos pequeños.

### **2.2.2. PAGOS OFFLINE**

Algunos clientes prefieren medios offline para pagar las compras que realizan por internet. Los medios más utilizados por los usuarios son: el contra reembolso un formato arriesgado para la empresa que aumenta las ventas por la confianza que genera en el cliente y algo menos usados son la domiciliación o giro bancario.

#### **1. CONTRA REEMBOLSO**

Este método de pago se efectúa una vez se ha recibido la mercancía, es necesario que se compruebe la mercancía antes de realizar el pago.

Habitualmente, enviar mercancía contra reembolso supone un coste adicional que factura la empresa de transporte. Es decisión del comercio electrónico asumir este coste o imputarlo al cliente final.

#### **2. GIRO POSTAL**

El tradicional servicio de giro postal que ofrece el correo. El comprador paga el servicio y el vendedor envía la mercancía contra recepción.

#### **3. DOMICILIACIÓN BANCARIA**

El vendedor cobra directamente del banco del comprador que ha dado autorización previa.

### **2.2.3. INTEGRADORES**

Herramientas online, estas soluciones para aplicaciones web, permiten que las transacciones a través de medios electrónicos sean seguras para los usuarios que registran datos como número de tarjetas de crédito/débito. Los integradores más conocidos son PayPal, SafetyPay.

#### **1. PAYPAL**

PayPal es una empresa estadounidense propiedad de eBay, se ha convertido en un medio de pago muy difundido.

En cada país PayPal matiza su comportamiento, pero en general ofrece estas modalidades. Trabajar con dinero depositado en la cuenta de PayPal previamente. Trabajar como una pasarela de pago bancaria, solicitando los datos de la tarjeta de crédito/débito.

#### **2. SAGEPAY**

Sagepay pasarela de pago con control antifraude propiedad del grupo Sage.

A diferencia de otras pasarelas de pago, Sagepay únicamente factura una cuota mensual fija de 30 dólares y permite hasta mil transacciones. Para un volumen de ventas superior la cuota se incrementaría. [3]

### **2.3. CONCEPTOS DE DINERO ELECTRÓNICO**

El dinero electrónico es un medio de pago electrónico, implementado por el Banco Central del Ecuador, cuyo respaldo es el dinero físico.

El ciudadano que desee acceder a este sistema debe abrir una cuenta de Efectivo desde mi celular (dinero electrónico), a través de su teléfono móvil.

### **2.3.1. ACTIVACIÓN DE CUENTA**

La cuenta de dinero electrónico se activa en Ecuador marcando al \*153# desde su teléfono celular. Responda las preguntas de validación para que el sistema registre sus datos y le envíe su clave de seguridad a través de un mensaje de texto.

### **2.3.2. COSTO DEL PAGO DE IMPUESTOS**

El costo del pago del servicio en el caso de personas naturales el costo por transacción es de 0,05 centavos y para personas jurídicas es de 0.20 centavos.

### **2.3.3. ACREDITAR EFECTIVO A LA CUENTA**

Para acreditar el efectivo a las cuentas de dinero electrónico el ciudadano acude a una de las instituciones financieras o establecimientos autorizados por el Banco Central, con su cédula y el dinero físico que desea acreditar a su cuenta de *Efectivo Desde Mi Celular*.

El sistema valida su identificación, procesa la acreditación y envía un mensaje a su celular, confirmando el abono realizado.

Los establecimientos para acreditación de dinero electrónico o cambio de Efectivo Desde Mi Celular (dinero electrónico) a físico, son:

Instituciones financieras:

Banco Central del Ecuador, Banco del Pacífico, Banco Amazonas, Cooperativa Jep, Cooperativa 29 de Octubre, Mutualista Imbabura, Servipagos entre otros.

Establecimientos autorizados:

Almacenes Tía, Farmacias Sana entre otros.

#### **2.3.4. RETIRAR EFECTIVO A LA CUENTA**

Retire el dinero físico de su cuenta de Efectivo Desde Mi Celular (dinero electrónico) en una de las instituciones financieras o establecimientos autorizados por el Banco Central presentando su cédula de identidad.

#### **2.3.5. VENTAJAS DE USO DE DINERO ELECTRÓNICO**

- Es un sistema rápido y seguro.
- Las transacciones están protegidas por claves personales.
- Se conoce con claridad el origen y el destino de los recursos.
- La transacción no consume el saldo de llamadas ni mensajes SMS de su celular.
- No es necesario tener un teléfono inteligente. [4]

### **2.4. CONCEPTOS Y BENEFICIOS DEL DINERO ELECTRÓNICO DEL BANCO CENTRAL**

Los beneficios del dinero electrónico de acuerdo con el Banco Central son los siguientes:

- No requiere de conexión a internet para realizar el pago.
- Bajas tarifas de costo por transacción.

- No depende de ningún modelo de celular en específico se puede realizar pagos a través de cualquier teléfono celular.
  - El servicio funciona 24 horas al día, durante los 7 días de la semana.
  - Evita realizar largas colas en ventanillas para realizar el pago.
  - No requiere cuenta bancaria para utilizar el servicio.
  - Podrá ser canjeado por billetes en todo momento.
  - Devolución de 2 puntos del IVA por compras en efectivo desde mi celular y 1 punto por compras hechas con tarjetas de crédito y débito.
  - Nos brinda seguridad debido a la protección con claves personalizadas.
- [5]

## **2.5. HERRAMIENTAS OPEN SOURCE PARA LA IMPLEMENTACIÓN DE BOTÓN DE PAGO**

### **2.5.1. SOFTWARE LIBRE**

El software libre está identificado por 4 principios básicos de libertad:

1. La libertad de ejecutar el programa, con cualquier propósito.
2. La libertad de estudiar cómo funciona el programa, y adaptarlo a tus necesidades. El acceso al código fuente es una condición previa para esto.
3. La libertad de redistribuir copias para que pueda ayudar a tu vecino.
4. La libertad de mejorar el programa, y publicar sus mejoras al público, de manera que beneficie toda la comunidad. El acceso al código fuente es una condición previa para esto.

El software no libre también es llamado software propietario. El software libre no debe confundirse con el software gratuito.

### **2.5.2. CÓDIGO ABIERTO**

Open Source (Código abierto) se conoce de esta manera al software distribuido y desarrollado libremente, orientado a compartir el código.

Los usuarios lo adaptan a sus necesidades, corrigen sus errores a una velocidad impresionante, aplicada en el desarrollo de software convencional o cerrado, dando como resultado la producción de un mejor software. [6]

### **2.5.3. HERRAMIENTAS PARA UTILIZAR**

El requerimiento del portal web de la empresa Powerlinks para la implementación del aplicativo botón de pagos necesita de las siguientes tecnologías:

1. Versión de ASP.NET Framework 4.0 o superior.
2. Contar con una base de datos SQL Server 2008 R2 o superior.
3. Una cuenta administradora con privilegios sobre IIS.
4. Servidor Web Microsoft 7.5 o Superior.
5. Espacio requerido para instalación es de 100MB.
6. Comunicación vía POST desde el Portal y genera vía GET la comunicación de respuesta hacia la página de respuesta de transacción del Portal E-Commerce.
7. Consola “REST” que permite a manera de “PlayGround” realizar pruebas de las API disponibles para integración con aplicaciones.

8. Test de Notificaciones con autenticación que permite verificar las respuestas y permite volver a enviarlas para verificación de integraciones.

#### 1. MICROSOFT .NET FRAMEWORK 4.5

.NET Framework es un entorno de ejecución administrado que proporciona diversos servicios a las aplicaciones en ejecución. Consta de dos componentes principales: Common Language Runtime (CLR), que es el motor de ejecución que controla las aplicaciones en ejecución, y la biblioteca de clases de .NET Framework, que proporciona una biblioteca de código probado y reutilizable al que pueden llamar los desarrolladores desde sus propias aplicaciones.

Los servicios que ofrece .NET Framework a las aplicaciones en ejecución son los siguientes:

**Administración de la memoria.** En muchos lenguajes de programación, los programadores son responsables de asignar y liberar memoria y de administrar la vida útil de los objetos. En las aplicaciones de .NET Framework, CLR proporciona estos servicios en nombre de la aplicación.

**Sistema de tipos comunes.** En los lenguajes de programación tradicionales, el compilador define los tipos básicos, lo que complica la interoperabilidad entre lenguajes. En .NET Framework,

los tipos básicos los define el sistema de tipos de .NET Framework y son comunes a todos los lenguajes que tienen como destino .NET Framework.

**Biblioteca de clases extensa.** En lugar de tener que escribir cantidades extensas de código para controlar operaciones comunes de programación de bajo nivel, los programadores pueden usar una biblioteca de tipos accesible en todo momento y sus miembros desde la biblioteca de clases de .NET Framework.

**Marcos y tecnologías de desarrollo.** .NET Framework incluye bibliotecas para determinadas áreas de desarrollo de aplicaciones, como ASP.NET para aplicaciones web, ADO.NET para el acceso a los datos y Windows Communication Foundation para las aplicaciones orientadas a servicios.

**Interoperabilidad de lenguajes.** Los compiladores de lenguajes cuya plataforma de destino es .NET Framework emiten un código intermedio denominado Lenguaje intermedio común (CIL), que, a su vez, se compila en tiempo de ejecución a través de Common Language Runtime. Con esta característica, las rutinas escritas en un lenguaje están accesibles a otros lenguajes, y los programadores pueden centrarse en crear aplicaciones en su lenguaje o lenguajes preferidos.

**Compatibilidad de versiones.** Con raras excepciones, las aplicaciones que se desarrollan con una versión determinada de

.NET Framework se pueden ejecutar sin modificaciones en una versión posterior.

**Ejecución en paralelo.** .NET Framework ayuda a resolver conflictos entre versiones y permite que varias versiones de Common Language Runtime coexistan en el mismo equipo. Esto significa que también pueden coexistir varias versiones de las aplicaciones, y que una aplicación se puede ejecutar en la versión de .NET Framework con la que se compiló.

**Compatibilidad con múltiples versiones (multi-targeting).** Al usar la Biblioteca de clases portable de .NET Framework, los desarrolladores pueden crear ensamblados que funcionen en varias plataformas, como Windows 7, Windows 8, Windows 8.1, Windows 10, Windows Phone y Xbox 360. [7]

## 2. MICROSOFT SQL SERVER 2008 R2

Microsoft SQL Server 2008 R2 Management Studio Express (SSMSE) es un entorno integrado para obtener acceso, configurar, controlar, administrar y desarrollar todos los componentes de SQL Server Express. Esta edición combina un amplio grupo de herramientas gráficas con una serie de editores de script enriquecidos que permiten a desarrolladores y administradores de todos los niveles obtener acceso a SQL Server.

Sistema operativo compatible Windows 7; Windows Server 2003; Windows Server 2008; Windows Server 2008 R2; Windows Vista; Windows XP. [8]

### **3. SERVIDOR DE APLICACIONES**

Los servidores de aplicación típicamente incluyen también middleware (o software de conectividad) que les permite intercomunicarse con variados servicios, para efectos de confiabilidad, seguridad, no-repudio, etc. Los servidores de aplicación también brindan a los desarrolladores una Interfaz para Programación de Aplicaciones (API), de tal manera que no tengan que preocuparse por el sistema operativo o por la gran cantidad de interfaces requeridas en una aplicación web moderna.

Los servidores de aplicación también brindan soporte a una gran variedad de estándares, tales como HTML, XML, IIOP, JDBC, SSL, etc., que les permiten su funcionamiento en ambientes web (como Internet) y la conexión a una gran variedad de fuentes de datos, sistemas y dispositivos.

Un ejemplo común del uso de servidores de aplicación (y de sus componentes) son los portales de internet, que permiten a las empresas la gestión y divulgación de su información, y un punto único de entrada a los usuarios internos y externos. Teniendo como base un servidor de aplicación, dichos portales permiten tener acceso a información y servicios (como servicios web) de manera segura y transparente, desde cualquier dispositivo. [9]

#### **4. SERVIDOR DE APLICACIONES WINDOWS SERVER 2012**

Se pueden implementar los servidores con el sistema operativo Windows Server 2012 R2 mediante la instalación y configuración de funciones de servidor, servicios de función y características de función.

Una función de servidor de Windows Server 2012 R2 es un conjunto relacionado de componentes de software que permite a un servidor realizar una función específica para los usuarios y otros equipos en una red.

Un servidor con Windows Server 2012 R2 puede estar dedicado a una sola función, o por el contrario proporcionar múltiples roles.

Un servicio de función de Windows Server 2012 R2 es un componente de software que proporciona la funcionalidad para una función de servidor. Cada función de Windows Server 2012 R2, puede tener uno o varios servicios de función relacionados. [10]

#### **5. INTERNET INFORMATION SERVER**

El rol servidor web (IIS) en Windows Server 2012 proporciona una plataforma segura, fácil de administrar, modular y extensible donde hospedar sitios web, servicios y aplicaciones de manera confiable. Con IIS 8, puede compartir información con usuarios en Internet, en una intranet o en una extranet. IIS 8 es una plataforma web unificada que integra IIS, ASP.NET, servicios de FTP, PHP y Windows Communication Foundation (WCF).

En la lista siguiente, se indican solo algunas de las ventajas que obtendrá al usar IIS 8:

- La seguridad web se refuerza gracias a una superficie reducida de servidor y al aislamiento automático de aplicaciones.
- Podrá implementar y ejecutar aplicaciones web de ASP.NET, ASP clásico y PHP en el mismo servidor de forma sencilla.
- Se logra el aislamiento de aplicaciones al proporcionar a los procesos de trabajo una identidad única y una configuración en espacio aislado de manera predeterminada, lo que reduce aún más los riesgos de seguridad.
- Podrá agregar y eliminar componentes IIS integrados e incluso reemplazarlos fácilmente por módulos personalizados que se adapten a las necesidades del cliente.
- Aumenta la velocidad del sitio web mediante el almacenamiento en caché dinámico integrado y la compresión mejorada. [11]

## 6. MÉTODOS GET VS POST

Tanto el método GET como POST son protocolo HTTP el cual envía al servidor como petición (request) y recibe una respuesta a dicha solicitud (response).

## **Petición HTTP**

El concepto **GET** es obtener información del servidor. Traer datos que están almacenadas en el servidor, ya sea una base de datos o archivo al cliente. Independientemente de que para eso tengamos que enviar (request) algún dato que será procesado para luego devolver la respuesta (response) que esperamos, un ejemplo sería recibir un identificador para obtener un artículo de la base de datos.

El concepto **POST** en cambio es enviar información desde el cliente para que sea procesada y actualice o agregue información en el servidor, como sería la carga o actualización en si de un artículo. Cuando enviamos (request) datos a través de un formulario, estos son procesados y luego a través de una redirección, ejemplo devolvemos (response) alguna página con información.

Tanto GET como POST solicitan una respuesta del servidor y ahí donde parece que los conceptos son iguales ya que con ambos se podría lograr los mismos objetivos. Yo podría, aunque no es correcto, enviar por GET ciertos datos en una URL u actualizar o insertar dicha información en mi base de datos, pero realmente eso le corresponde al método POST. De la misma manera podría solicitar una página diferente por medio de POST y simplemente mostrarla como respuesta, aunque eso debería ser a través de una llamada GET.

Las llamadas por método GET pueden ser cacheadas (historial navegador), indexadas por buscadores, agregar los enlaces a

nuestros favoritos o hasta pasar una URL completa a otra persona para que ingresa a dicha página. Con el método POST sin embargo no se puede hacer esto.

Generalmente usamos links para ejecutar llamadas GET ya que la idea del link es simplemente "solicitar" una información (pagina) al servidor y que sea devuelta como una respuesta. Mientras usamos formularios para actualizar datos, como artículos, usuarios, etc. también en cuenta que por el método POST también se puede enviar más cantidad de datos que por GET. [12]

## **2.6. METODOLOGÍAS DE PRUEBAS DE SOFTWARE**

### **2.6.1. CICLO DE VIDA DEL SOFTWARE TESTING**

El Ciclo de Vida de las Pruebas del Software (STLC) es la secuencia de actividades que lleva a cabo el equipo de control de calidad desde el inicio hasta el final del proyecto.

#### **1. PLANIFICACIÓN DE PRUEBAS**

El equipo de pruebas revisa los documentos de diseño y prepara el plan de pruebas. Se realiza el cronograma de las pruebas y posteriormente el plan de pruebas.

En esta fase se define qué tipo de pruebas se van a realizar sobre el software, las prioridades de las pruebas, los criterios de salida y el ambiente de pruebas.

## 2. CONTROL

Se ejecuta durante todo el ciclo de pruebas. Compara la situación actual contra el plan y se ajusta el plan en caso de ser necesario.

## 3. ANÁLISIS Y DISEÑO DE PRUEBAS

Luego de planificar las pruebas debemos identificar las posibles condiciones de pruebas, las tareas a realizar en esta fase son las siguientes:

- Revisar requerimientos, arquitectura del sistema, diseño e interfaces.
- Diseño de casos de pruebas (basados en «test basic»)
- Creación de scripts de pruebas (en caso de ser requerido).
- Verificación de casos de pruebas y scripts de automatización.
- Preparar datos y ambiente de pruebas.

## 4. IMPLEMENTACIÓN Y EJECUCIÓN DE PRUEBAS

En esta etapa se desarrollan y ejecutan los casos de prueba y se reportan los bugs encontrados. Se recomienda ejecutar al inicio los casos de pruebas más importantes. Usualmente los casos de prueba se mejoran después de las primeras ejecuciones.

Durante la ejecución de pruebas realizamos: «Retesting» y regresión:

- Retesting: Se ejecutan casos de pruebas para asegurar que el defecto ha sido corregido.

- Regresión: Se ejecutan casos de pruebas para asegurar que en otros módulos no se ha afectado alguna funcionalidad.

## 5. EVALUACIÓN DE CRITERIOS DE SALIDA Y REPORTE

Se verifican los resultados obtenidos contra los criterios definidos en el plan de pruebas. Evaluar registros de pruebas.

Se redacta el «Test Summary» para los stakeholders con el objetivo de indicar el resumen de los resultados de las pruebas. Evaluar si es necesario realizar pruebas adicionales.

## 6. ACTIVIDADES DE CIERRE DE PRUEBAS

Asegurarse que toda la documentación está en orden, que los defectos en su mayoría estén cerrados. Tener en cuenta las lecciones aprendidas del proyecto; para aplicarlas en futuros proyectos de testing. [13]

### 2.6.2. PRUEBAS A TRAVÉS DEL CICLO DE VIDA DEL SOFTWARE - MODELO V

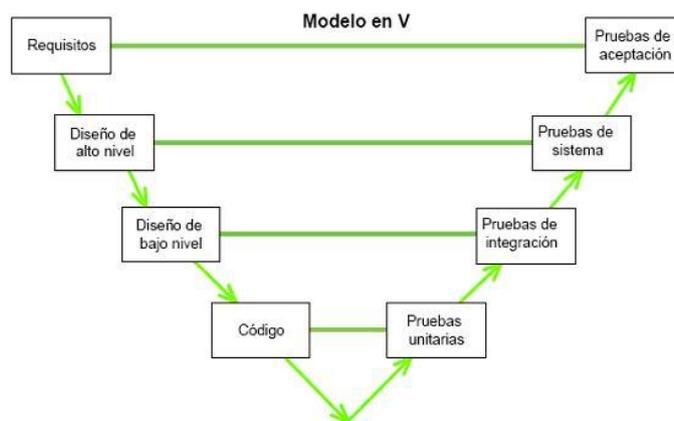


Figura 2.1 Modelo ciclo de vida de Desarrollo – V

En el modelo V permite ejecutar tareas de validación y verificación, se planifican en paralelo con las tareas de desarrollo.

Esta estructura obedece al principio que para cada fase del desarrollo debe existir un resultado verificable.

En la misma estructura se advierte también que la proximidad entre una fase del desarrollo y su fase de verificación correspondiente va decreciendo a medida que aumenta el nivel dentro de la V.

**Verificación** compara el producto con la especificación.

**Validación** confirma que el usuario está satisfecho.

## **1. REQUISITOS - PRUEBAS DE ACEPTACIÓN**

El nivel 1 está orientado al "cliente". El inicio del proyecto y el fin del proyecto constituyen los dos extremos del ciclo. Se compone del análisis de requisitos y especificaciones.

Estas pruebas se deben ejecutar en el sistema operativo del ambiente del usuario. El usuario selecciona los casos de pruebas a realizar en estas pruebas. Evaluar si el sistema satisface las necesidades del usuario. Validar procesos de negocio en ambientes integrados.

## **2. DISEÑO DE ALTO NIVEL - PRUEBAS DE SISTEMAS**

El nivel 2 se dedica a las características funcionales del sistema propuesto. Puede considerarse el sistema como una caja negra, y caracterizarla únicamente con aquellas funciones que son directa o indirectamente visibles por el usuario final.

Evalúa la funcionalidad, rendimiento, confiabilidad y seguridad del sistema. Descubrir errores en el comportamiento general del sistema. Ganar confianza con el funcionamiento del sistema. Validar requerimientos no funcionales. Verificar que se cumplan especificaciones. El entorno de pruebas debería coincidir con el entorno real.

### **3. DISEÑO DE BAJO NIVEL - PRUEBAS DE INTEGRACIÓN**

El nivel 3 define los componentes hardware y software del sistema final, a cuyo conjunto se denomina arquitectura del sistema. Detectar errores a nivel de interfaces. Asegurar que la funcionalidad de las unidades combinadas se cumpla de acuerdo con la especificación.

Ejecutar pruebas para dos o más componentes relacionados.

Se realiza la integración de componentes individuales en niveles inferiores hasta “crear” el sistema completo, utilizando drivers en donde sea apropiado.

Integrar los componentes de arriba hacia abajo, reemplazando componentes individuales con stubs en donde sea necesario.

### **4. CÓDIGO-PRUEBAS UNITARIAS**

El nivel 4 es la fase de implementación, en la que se desarrollan los elementos unitarios o módulos del programa.

Descubrir defectos. Asegurar que cada componente responde al comportamiento funcional esperado. Programas o unidades más

pequeñas. Componentes aislados. Funcionalidad (con solo un caso de prueba básico).

La ejecución de pruebas de componentes puede requerir el uso de Drivers y stubs.

**Driver**: Un componente que invoca al módulo a probar (TestedUnit). Simula datos de entrada.

**Stub**: Un componente que depende del módulo a probar, simula un componente que aún no está disponible o no es parte del objeto de prueba. [14] [13]

### 2.6.3. TIPOS DE PRUEBA

#### 1. PRUEBAS FUNCIONALES

Se ejecutan sobre el sistema para verificar los requisitos funcionales.

#### 2. PRUEBAS NO FUNCIONALES

Se seleccionan requisitos funcionales específicos para evaluar requisitos no funcionales. Entre estas tenemos las pruebas de carga, concurrencia y rendimiento.

Aplican para todos los niveles de pruebas.

#### 3. PRUEBA DE CARGA

Las pruebas que evalúan la respuesta de un sistema complejo o módulo bajo una pesada carga de los datos, la repetición de ciertas acciones de los datos de entrada, los grandes valores numéricos. Consultas grandes a una base de datos o un gran número

recurrente para comprobar el nivel de los usuarios de escalabilidad, es decir, el momento en que el tiempo de respuesta se empieza a degradar, es decir los sistemas / módulos comienzan a fallar.

#### **4. PRUEBA DE CONCURRENCIA**

Corresponde a la prueba que pretende verificar la capacidad del sistema o para manejar un gran volumen de datos como entrada y de salida o residente en la base de datos del módulo.

Este tipo de pruebas tiene como objetivo determinar los límites de la transformación y la carga del sistema o módulo.

#### **5. PRUEBAS DE RENDIMIENTO**

Corresponde a la prueba asegurar que el sistema o módulo cumple con los niveles de rendimiento y los tiempos de respuesta acordados con los usuarios definidos en los requisitos.

Este tipo de prueba es determinar si el rendimiento en las situaciones previstas acceso y pico transacción cumple con los requisitos de rendimiento no funcionales. [15]

### **2.7. METODOLOGÍAS PARA DISEÑOS DE API**

La mayoría de las soluciones modernas basadas en API's, hospedados en servidores web, para proporcionar funcionalidad a las aplicaciones cliente remotas. Las operaciones que expone un servicio constituyen una API web. Una API bien diseñada debe ser capaz de admitir:

**1.- Independencia de la plataforma:** Las aplicaciones cliente deben ser capaces de usar la API que proporciona el servicio sin necesidad de saber

cómo los datos o las operaciones que expone la API se implementan físicamente. Para ello, se requiere que la API se base en normas comunes que permiten que una aplicación cliente y un servicio acuerden formatos de datos se usan y la estructura de los datos que se intercambian entre las aplicaciones cliente y el servicio API.

**2.- Evolución del servicio:** El servicio API debe ser capaz de evolucionar y agregar (o quitar) funcionalidad con independencia de las aplicaciones cliente. Las aplicaciones cliente existentes deben ser capaces de continuar funcionando sin cambios a pesar de que las características que proporcione el servicio cambien.

Toda la funcionalidad debe poderse detectar para que las aplicaciones cliente la puedan usar completamente.

### **2.7.1. ESQUEMA API DE LA TRANSFERENCIA DE ESTADO REPRESENTACIONAL (REST)**

Con el fin de estructurar las operaciones expuestas por los servicios web se expuso una arquitectura basada en REST.

REST es un estilo de arquitectura para la creación de sistemas distribuidos basados en hipermedia. Una de las principales ventajas del modelo REST es que se basa en estándares abiertos y no vincula la implementación del modelo o las aplicaciones cliente que tienen acceso a él con ninguna implementación específica. Por ejemplo, un servicio web REST podría implementarse mediante la API web de Microsoft ASP.NET y las aplicaciones cliente podrían desarrollarse

usando cualquier lenguaje y conjunto de herramientas que pueda generar solicitudes HTTP y analizar las respuestas HTTP.

La arquitectura REST usa un esquema de navegación para representar objetos y servicios denominados recursos.

Los sistemas que implementan REST normalmente usan el protocolo HTTP para transmitir solicitudes de acceso a estos recursos. En estos sistemas, una aplicación cliente envía una solicitud en forma de un URI que identifica un recurso y un método HTTP el más común GET, POST, PUT o DELETE que indica la operación que se realizará en ese recurso. El cuerpo de la solicitud HTTP contiene los datos necesarios para realizar la operación. Es importante que comprenda que REST define un modelo de solicitud sin estado. Las solicitudes HTTP deben ser independientes y pueden producirse en cualquier orden, por lo que no es factible intentar conservar la información de estado transitorio entre solicitudes. El único lugar donde se almacena la información es en los propios recursos y cada solicitud debe ser una operación atómica. De hecho, un modelo REST implementa una máquina de estado finito en la que una solicitud pasa a un recurso desde un estado no transitorio bien definido a otro.

### **2.7.2. DEFINICIÓN DE OPERACIONES EN TÉRMINOS DE MÉTODOS HTTP**

El protocolo HTTP define una serie de métodos que asignan significado semántico a una solicitud. Los métodos HTTP comunes que usan la mayoría de las API web RESTful son:

GET, para recuperar una copia del recurso en el URI especificado. El cuerpo del mensaje de respuesta contiene los detalles del recurso solicitado.

POST, para crear un nuevo recurso en el URI especificado. El cuerpo del mensaje de solicitud proporciona los detalles del nuevo recurso. Tenga en cuenta que POST también puede usarse para desencadenar operaciones que en realidad no crean recursos.

PUT, para reemplazar o actualizar el recurso en el URI especificado. El cuerpo del mensaje de solicitud especifica el recurso que se va a modificar y los valores que se aplican.

DELETE, para quitar el recurso en el URI especificado. [16]

De forma resumida, una buena metodología de API, en este caso para Dinero Electrónico, nuestra API es un Componente HTTP que debe respetar el Uso correcto HTTP con los verbos GET, POST, UPDATE y DELETE.

Recordar que la Validación del Diseño de la API, debe regirse a las normas expuestas por la W3C usando Markup Validation Service.

Para los esquemas de Validación, nos regimos con el SOAP 1.2.

## **2.8. METODOLOGÍAS PARA DISEÑOS DE WEB SERVICES**

Un servicio Web o Web Services es un servicio que expone su lógica a clientes de cualquier plataforma mediante una interfaz accesible a través de la red utilizando tecnologías y protocolos estándar de Internet.

Las ideas de los servicios son accedidas a través de un servidor Web y de un modo independiente de la plataforma, utilizando protocolos estándar (HTTP, SOAP, WSDL, UDDI).

### **2.8.1. VENTAJA DE LOS SERVICIOS WEB**

Entre las ventajas más importantes que ofrecen los Servicios WEB se pueden indicar:

1. Ofrecen una “tecnología distribuida de componentes”.
2. Evitan los problemas inherentes a la existencia de firewalls, ya que SOAP utiliza HTTP como protocolo de comunicación.
3. Permiten una invocación sencilla de métodos, mediante SOAP.
4. Los clientes o “consumidores de servicios” pueden estar en cualquier plataforma (basta con que soporten XML/SOAP, incluso puede sustituirse SOAP por HTTP).
5. Permiten centralizar los datos, independientemente de si los Servicios WEB están distribuidos o no.

### **2.8.2. ESTRUCTURA DE UN SERVICIO WEB**

1. Los Servicios Web se implementan mediante una clase derivada de la clase Web Services, que pertenece al namespace System.Web.Services.
2. Ninguno de los métodos de esta clase, ya sean privados, protegidos o públicos será accesible para un cliente o consumidor del servicio. Si se desea que un método sea accesible desde un cliente, debe ser definido con el atributo [WebMethod].

3. Igual que los WebForms, un Web Services o servicio Web se compone de dos ficheros (los más importantes, aunque realmente son más:

4. Un Archivo con extensión .asmx: equivale al fichero .aspx de las páginas ASP.NET (WebForms). Es la página que se pedirá desde el navegador del cliente para acceder al servicio.

5. Un Archivo .cs: Contiene el código del servicio (equivale al .cs de las páginas ASP.NET).

### **2.8.3. OBJETIVOS DE LOS WEB SERVICES**

1. Interoperabilidad por medio de la utilización de estándares: XML, SOAP, WSDL y UDDI entre otros.

2. Independencia tanto del lenguaje como de la plataforma, por medio de la separación de especificación de la implementación

3. Acoplamiento débil debido al manejo de sistemas basados en mensajes en los cuales existen interacciones síncronas y asíncronas

4. Acceso a través de Internet lo cual permite un control no centralizado, por otro lado, se tiene la utilización de protocolos establecidos y las consideraciones de seguridad.

5. Modularidad y Reusabilidad de servicios debido al uso de estándares.

6. Escalabilidad teniendo en cuenta que se trata de aplicaciones uno-a uno frente a uno-a-muchos.

#### **2.8.4. WEB SERVICE DESCRIPTION LANGUAGE (WSDL)**

WSDL es un estándar planteado por la W3C, el cual se define como un formato XML para describir servicios en la red como un conjunto de puntos finales, los cuales operan en mensajes que contienen cualquier información ya sea orientada a procedimiento o a documento. En WSDL se describen de manera abstracta las operaciones y los mensajes, los cuales a su vez se unen para un protocolo de red y formato de mensaje específicos, con el objetivo de definir un punto final. Por su parte, estos puntos finales se combinan en puntos finales abstractos, los cuales se denominan servicios. Este estándar fue creado con el propósito de describir la sintaxis para la invocación técnica de un Web Services, y con ello permitir a las aplicaciones describir a otras aplicaciones las reglas para interactuar y dialogar entre ellas. Además, WSDL permite automatizar la generación de servidores proxy para los Web Services de forma independiente de lenguaje y de plataforma.

Una descripción WSDL básicamente describe las tres propiedades fundamentales de un Web Services:

Qué hace un servicio: Las operaciones o métodos que el servicio provee. Cómo es accedido un servicio: Los detalles de formatos de datos y de los protocolos necesarios para acceder las operaciones del servicio.

Dónde es localizado un servicio: Los detalles de la dirección de red del protocolo específico, tal como un URL.

Un documento WSDL es el contrato que existe entre el cliente y el servidor, en el cual se establecen las reglas de comunicación, se definen servicios como colecciones de puntos finales o puertos de una red, por lo cual dicha definición abstracta de puntos finales y mensajes es separada de su concreto despliegue de red o ligamentos de los formatos de datos. [17]

## CAPÍTULO 3

### LEVANTAMIENTO DE INFORMACIÓN Y DEFINICIÓN DE LA SITUACIÓN ACTUAL

#### 3.1. LEVANTAMIENTO DE REQUERIMIENTOS

La elaboración del levantamiento de requerimiento se realizó mediante la recopilación de los siguientes requisitos redactados como historias de usuario:

##### **#001 Inserción de datos de cliente.**

Como un **usuario del comercio**, quisiera realizar el **pago** para poder **confirmar la compra** en línea.

1. Seleccionar el método de pago con dinero electrónico, dando clic en el botón de pago **Efectivo**.

2. Ingresar datos de la **cuenta de usuario** de Banco Central del Ecuador.
  - a. **Número identificación** validar Formato.
  - b. **Número celular** validar Formato.
3. Confirmar la acción de pago, dando clic en el botón **Pagar**. Posteriormente se realizará lo siguiente:
  - Se verificará la cuenta de usuario de dinero electrónico en Banco Central del Ecuador.
  - Se enviará una notificación para la confirmación de pago vía SMS al usuario.
  - Se recibirá la respuesta de autorización del cobro de la transacción.
  - Se mostrará al usuario el mensaje de proceso satisfactorio en pantalla.

### **#002 Ejecución de proceso de pago-cobro con dinero electrónico de la transacción.**

Como un **usuario del comercio**, quisiera confirmar el pago para poder **completar la transacción de compra**.

1. Recibir notificación vía SMS de **confirmación de pago** desde su cuenta de dinero electrónico.
2. Ingresar **Clave**.
3. Seleccionar **opción Aceptar**. Posteriormente se realizará lo siguiente:
  - Se confirma el pago de la transacción.

- Se ejecuta el proceso de débito.
- Se autoriza el cobro de la transacción del usuario.
- Se acredita el dinero electrónico a la cuenta del comercio.
- Se enviará la notificación de acreditación de pago de la transacción vía SMS al dueño del comercio.
- Se mostrará al usuario el mensaje de transacción realizada correctamente.

**#003 Recepción de notificación de saldo acreditado en la cuenta del comercio.**

Como el **dueño del comercio**, quisiera recibir notificación del valor de la compra acreditado en la cuenta de dinero electrónico para poder realizar la **entrega del pedido**.

1. Recibir notificación vía SMS de acreditación del pago a su cuenta de dinero electrónico.

**#004 Consulta de transacciones realizadas información administrativa.**

Como el **dueño del comercio**, quisiera realizar consultas de las transacciones pagadas con dinero electrónico para poder revisar información.

1. Realizar **consulta** seleccionando cualquiera de los siguientes **filtros**:
  - Fecha de Inicio
  - Fecha de Fin

- Estado
  - Número celular
  - ID de Transacción
2. Realizar la acción de consulta dando clic en el botón **Consultar**, posteriormente se mostrarán los resultados en una vista de pantalla de la siguiente manera y dependiendo de la selección de los filtros de búsqueda:
- a. Datos administrativos de transacciones por Rango de Fechas y por Estados.
  - b. Datos Individuales de transacción de Usuario por Número Celular e ID de Transacción.
  - c. Datos generales de transacciones de Usuario por Rango de Fechas.

### **3.2. LEVANTAMIENTO DE INFORMACIÓN**

En el levantamiento de información se solicitó lo siguiente, se requiere de un Botón de pago con dinero electrónico que permitirá realizar el pago de transacciones en línea como la compra y venta de bienes o servicios de los diferentes comercios en la web, para que los compradores puedan realizar pagos mediante la transferencia de dinero electrónico desde la cuenta del comprador hacia la cuenta del vendedor. Este botón estará incorporado en la página web del comercio, para lo cual el proceso de compraventa en línea deberá realizar los siguientes pasos:

1. El comprador ingresa a la página web del comercio y realiza la selección de Productos a comprar, agregando cada ITEM seleccionado al Carrito de Compras.
2. El comprador confirma que ha completado su pedido y realiza el CheckOut, realizando el ingreso de todos los datos requeridos para la compra.
3. El comprador selecciona la forma de pago con dinero electrónico, dando clic en el Botón de pago.
4. El comprador visualiza la Interfaz API y realiza el ingreso de los siguientes datos solicitados, Número de Identificación y el Número Celular dando clic en el botón de Pagar. La cuenta del comprador y vendedor han sido creadas previamente para poder realizar el envío y recepción de datos al servicio de pago del Banco Central del Ecuador.
5. El pago se realiza mediante la conexión entre el API de servicio de pago con el servicio web del Banco Central del Ecuador, verificando la cuenta asociada de dinero electrónico del comprador.
6. El proceso de débito se realiza mediante a petición del Servicio de Cobro con los datos del comprador ingresados en el paso anterior. El sistema solicita la confirmación del servicio de pago enviando al Cliente un SMS mensaje de texto a su celular. El comprador confirma la transacción con su Clave, realizando la autorización del pago.
7. En la página web el API de servicio de Pago está censando la respuesta de la transacción si ha sido confirmado el pago por parte del servicio web del Banco Central, verificando el registro de la compra.

8. El vendedor recibe la notificación del valor acreditado por la transacción.
9. Se genera la entrega del producto a su lugar de destino.

### **3.3. DEFINICIÓN DE LA SITUACIÓN ACTUAL**

En cuanto a la situación actual tenemos que la solución dada deberá ser interoperable con cualquier plataforma web, permitirá realizar el pago de las transacciones de compra –venta de productos o servicios, a través de una forma de pago de pago fácil y confiable como lo es el dinero electrónico, ya que se encuentra disponible para los ciudadanos del Ecuador a nivel nacional.

El dinero electrónico es un medio de pago, implementado por el Banco Central del Ecuador, cuyo respaldo es el dinero físico. Los compradores que deseen acceder a este servicio deberán abrir una cuenta de efectivo a través de un mensaje de texto desde su teléfono móvil. Este servicio es rápido y seguro, las transacciones están protegidas por claves personales, se conoce con claridad el origen y el destino de los recursos. La transacción no consume el saldo de su celular, no es necesario tener un teléfono inteligente.

### **3.4. ALCANCE DEL PROYECTO**

El comercio electrónico desde su página web permitirá al cliente elegir la forma de pago efectivo, Dinero Electrónico.

1. Es requerida la información del cliente para realizar el pago desde la página Web, en conjunto con la interacción de su dispositivo móvil, podrá

completar su transacción de compra. El número de celular del cliente debe estar previamente registrado en la cuenta de dinero electrónico del Banco Central.

2. El proceso de Pago inicia en la página Web, dependerá de la respuesta a la transacción que genere el dispositivo móvil del cliente autorizando o negando la transacción, de esta forma el cliente tiene el control total de la transferencia desde su celular.
3. El API realizará las peticiones de los servicios para el pago y cobro de transacciones al dar clic sobre el Botón de Pagos con Dinero Electrónico en la página web. El mismo que utilizará los métodos POST – GET para sus solicitudes.
4. Adicionalmente utilizará un Servicio Web para realizar consultas de la Información Administrativa en la página web del comercio electrónico. Expone las siguientes Transacciones Vía Web Services para los Administradores del sitio del comercio.
  - a. Transacciones por Rango de Fechas y por Estado.
  - b. Transacción de Usuario por Número Celular e ID de Transacción.
  - c. Transacciones de Un Usuario en un Rango de Fechas.

## **CAPÍTULO 4**

### **ANÁLISIS Y DISEÑO DE LA SOLUCIÓN**

#### **4.1. FASE I: ANÁLISIS Y DISEÑO DEL BOTÓN DE PAGO CON DINERO ELECTRÓNICO**

##### **4.1.1. ANÁLISIS DE REQUERIMIENTOS**

El análisis de la solución requerida para un mejor entendimiento del usuario ha sido diseñado de la siguiente manera:

##### **DIAGRAMA DE OBJETOS**

El portal web del sistema de comercio electrónico envía el requerimiento al servicio web expuesto vía WCF, el cual se conecta con el motor de base de datos SQL Server para llevar el registro de las transacciones solicitadas por los usuarios desde el portal.

El portal web se comunica con el componente integrador de pagos y este a su vez con el servicio de dinero electrónico del Banco Central del Ecuador. Para acceder a los servicios de Banco Central en Desarrollo se accede a través de una URL con un Certificado Digital y una clave.

El servicio del Banco Central realiza el envío de las notificaciones al teléfono celular del usuario que autoriza el pago de la transacción.

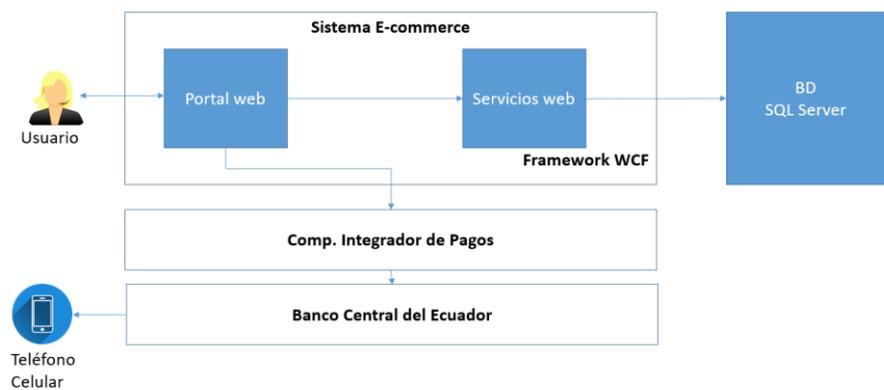


Figura 4.1 Diagrama de objetos API Botón de Pagos

**Nota:** Para las conexiones del servicio en productivo el acceso se hace vía VPN. Para ello se debe de llenar un formulario donde se indica las DIRECCIONES IP que consumirán los servicios del Banco Central del Ecuador.

## DIAGRAMA UML COMPRA EN LÍNEA Y PAGO CON DINERO ELECTRÓNICO

El portal web del sistema comercio electrónico podrá realizar los siguientes casos de uso por la transacción de compra en línea y pago con dinero electrónico.

### DIAGRAMA UML DE LA TRANSACCIÓN

En esta transacción realizada por los siguientes actores se presentan los siguientes casos:

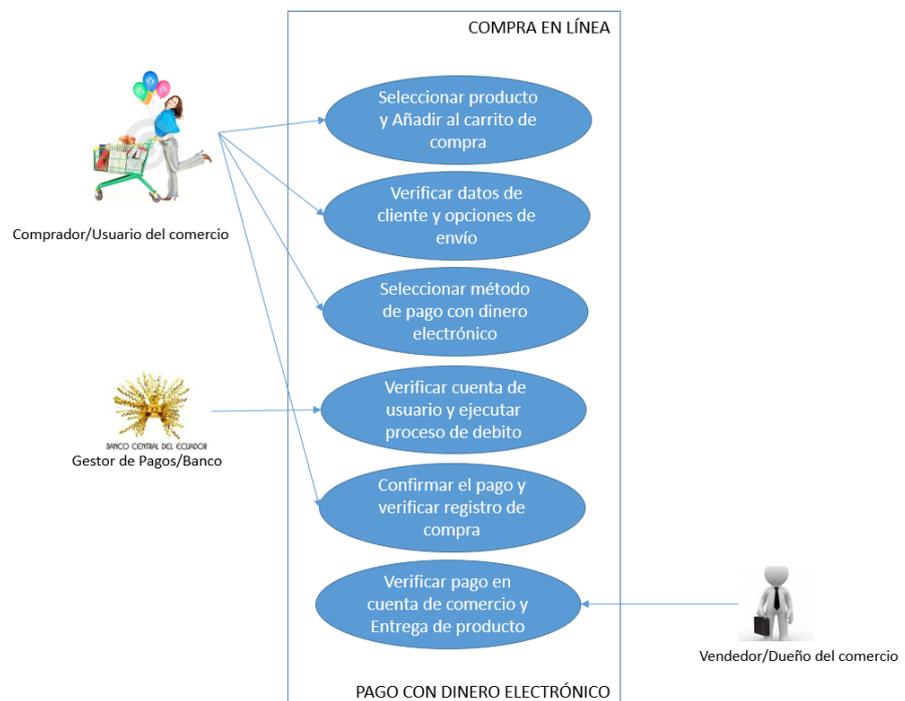


Figura 4.2 Diagrama UML Compra en línea y pago con dinero electrónico

## ACTORES, ROLES Y CASOS DE USO DE LA TRANSACCIÓN

Actor	Rol	Caso de uso
Comprador/Usuario del comercio	Adquiere productos o servicios	1. Seleccionar producto y Añadir al carrito de compra. 2. Verificar datos de cliente y opciones de envío. 3. Seleccionar método de pago con dinero electrónico. 5. Confirmar el pago y verificar registro de compra.
Gestor de pago/Banco	Permite establecer el mecanismo de pago	4. Verificar cuenta de cliente y Ejecutar proceso de débito.
Vendedor/Dueño del Comercio	Ofrece productos o servicios	6. Verificar pago en cuenta de comercio y Entrega de producto

*Tabla 1 Actores, Roles y Caso de uso de la transacción*

## CASOS DE USO DE LA TRANSACCIÓN

Seleccionar producto y Añadir al carrito de compra	
<b>Usuario del comercio</b>	Marcar producto seleccionado. Añadir producto al carrito. Ver productos seleccionados en carrito.

*Tabla 2 Caso de uso 1*

Verificar datos de cliente y opciones de envío	
<b>Usuario del comercio</b>	Ver datos de usuario nombre, número de identificación y dirección. Ver opciones de envío correo, Courier o recoger en tienda.

*Tabla 3 Caso de uso 2*

<b>Seleccionar método de pago con dinero electrónico</b>	
<b>Usuario del comercio</b>	<p>Marcar forma de pago dinero electrónico.</p> <p>Ingresar datos de usuario número de identificación y número de celular.</p>

*Tabla 4 Caso de uso 3*

<b>Verificar cuenta de cliente y ejecutar proceso de débito</b>	
<b>Banco Central del Ecuador</b>	<p>Verifica usuario de cuenta de dinero electrónico.</p> <p>Envía notificación de confirmación de pago vía SMS.</p> <p>Recibe autorización de cobro de la transacción.</p> <p>Ejecuta proceso de débito.</p> <p>Acredita dinero electrónico a la cuenta del comercio.</p> <p>Envía notificación de acreditación de pago vía SMS.</p>

*Tabla 5 Caso de uso 4*

<b>Confirmar el pago y verificar registro de compra</b>	
<b>Usuario del comercio</b>	<p>Confirmar pago vía SMS.</p> <p>Ver mensaje de proceso satisfactorio.</p> <p>Ver registro de compra.</p>

*Tabla 6 Caso de uso 5*

<b>Verificar pago en cuenta de comercio y Entrega de producto</b>	
<b>Dueño del comercio</b>	<p>Recibe notificación vía SMS de valor acreditado en la cuenta.</p> <p>Procede a realizar el envío y entrega del producto al usuario del comercio.</p>

*Tabla 7 Caso de uso 6*

#### 4.1.2. DISEÑO DE COMPONENTES

El portal web del sistema de comercio electrónico tiene los siguientes componentes:

- Componentes para la capa de presentación y Componentes del proceso. Estos componentes consumen 2 servicios, un servicio propietario del Web Services que se comunica con la base de datos propietaria al portal web. El otro servicio es para el botón de pago que se comunica con el componente de botón de pagos de dinero electrónico.
- El componente de botón de pago se comunica con el componente de acceso a datos para registrar las transacciones de forma local. También se comunica con el servicio web del Banco Central que se encarga de generar la petición.
- El servicio web del Banco Central se encarga de notificar y verificar hacia el celular del usuario la acción generando la respuesta del Web Services hacia el componente botón de pago el cual se encarga de actualizar en la base de datos de datos del portal la aprobación o negación de la transacción.
- El componente del botón de pago de dinero electrónico devuelve a la capa de presentación el mensaje respectivo para conocimiento del usuario.

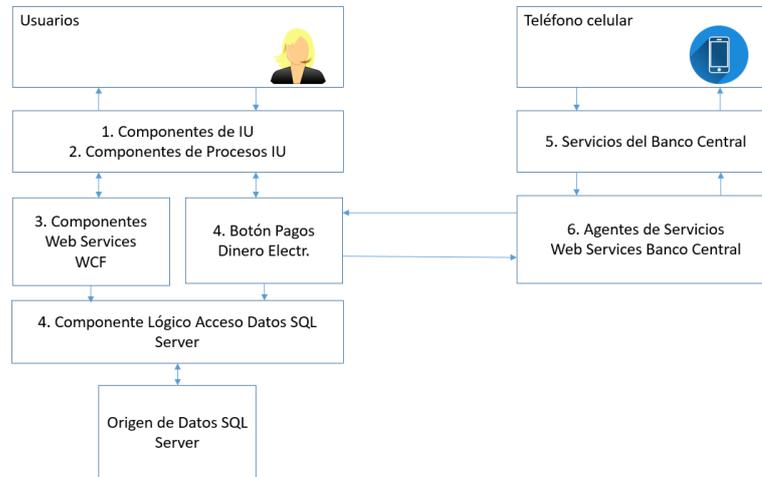


Figura 4.3 Diagrama de Componente API Botón de pago

### 4.1.3. DISEÑO DE INTERFACES DE USUARIOS

El portal presentara las siguientes pantallas y opciones para el usuario del comercio:

Pantalla de compras en línea que contendrá en la sección de forma de pago el API Botón de pagos con dinero electrónico.

La imagen muestra una captura de pantalla de un navegador Google Chrome en la URL <http://powerlink/compraenlinea/Demo.com>. La interfaz está dividida en tres secciones principales:

- Seleccionar Producto:** Muestra dos opciones: 'Productos' (desseleccionado) y 'Portatil' (seleccionado). A la derecha, se muestra un resumen de precios: Subtotal 1000,00, IVA 12% 120,00, y Total a pagar 1120,00.
- Forma de pago:** Presenta cuatro opciones de pago con botones circulares: un botón con un teléfono (seleccionado), el logo de MasterCard, el logo de PagarCaro y el logo de Dinero Club.
- Datos de factura:** Incluye campos para:
  - No. Identificación: 'Cedula' (menú desplegable) y '0922919733' (campo de texto).
  - Nombre: 'ANA KATIUSKA CRIOLLO CASSAGNE' (campo de texto).
  - Telefono: '0998108684' (campo de texto).
  - Direccion: 'FLORESTA 1 MZ 11 V17' (campo de texto).

Figura 4.4 Pantalla de compras en línea diseño Botón de pago

Pantalla de datos de usuario para realizar pago con dinero electrónico.

Google Chrome  
<http://powerlink/pagodineroelectronico/Demo.com>

Efectivo

No. Identificacion

No. celular

Figura 4.5 Pantalla ingreso de datos para pago

El portal presentara las siguientes pantallas y opciones para el dueño del comercio:

Pantalla de consulta de transacciones realizadas con dinero electrónico.

Google Chrome  
<http://powerlink/consultacuentadineroelectronico/Demo.com>

Fecha Inicio  Fecha Fin

Estado  Id Transaccion

No celular

▼ Fecha	▼ ID Tran.	▼ Estado	
16/09/2017	1	PENDIENTE	<input type="checkbox"/>
16/09/2017	2	ENVIADO	<input checked="" type="checkbox"/>

Figura 4.6 Pantalla de consulta de transacciones con dinero electrónico

#### 4.1.4. ARQUITECTURA DE APLICACIÓN

La aplicación botón de pago se estructurará de la siguiente manera en la red interna y extranet:

- Servidor para base de datos de transacciones con dinero electrónico, sitio web y WCF se encuentra en la red interna.
- Servicio web del Banco Central del Ecuador se encuentra en la extranet.

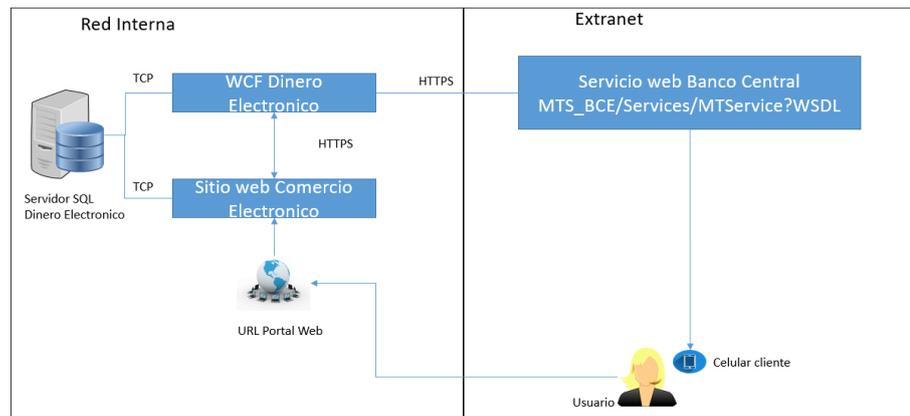


Figura 4.7 Diagrama de Arquitectura API Botón de pago dinero electrónico

#### 4.1.5. DESARROLLO DE LA SOLUCIÓN PARA BOTÓN DE PAGO CON DINERO ELECTRÓNICO

La solución para botón de pago con dinero electrónico tiene los siguientes Objetos de Aplicación Web de Dinero Electrónico.

Objeto	Funcionalidad
clibWebRequest	Librería de Conexión vía HTTP hacia las URL del Servicio WCF Dinero

	Electrónico y la conexión URL hacia el Servicio WEB de Banco Central.
WcfDineroElectronico servicioDineroElectronico. svc	<p>Servicio WCF de Portal Powerlinks que se comunica con el Servicio WEB de Banco Central y gestiona las peticiones de Dinero Electrónico. Expone los métodos:</p> <pre> &lt;OperationContract(&gt;   Function de_Cobro_Pre_Ws(PI_Datos As String) As String    &lt;OperationContract(&gt;   Function de_GetBalance_Ws(PI_Datos As String) As String    &lt;OperationContract(&gt;   Function de_CashIn_Pre_WS(PI_Datos As String) As String    &lt;OperationContract(&gt;   Function de_CashIn_Confirm_WS(PI_Datos As String) As String    &lt;OperationContract(&gt;   Function de_ReportTransaction(PI_Datos As String) As String    &lt;OperationContract(&gt;   Function de_ReportTransaction_Msisdn(PI_Dat os As String) As String    &lt;OperationContract(&gt;   Function de_Cobro_Confirm_Ws(PI_Datos As String) As String </pre>

<a href="http://www.powerlinks.com.ec">www.powerlinks.com.ec</a>	<ul style="list-style-type: none"> <li>▶ Account</li> <li>▶ Administracion</li> <li>▶ App_Code</li> <li>▶ App_Data</li> <li>▶ App_WebReferences</li> <li>▶ bin</li> <li>▶ BuzonUnico</li> <li>▶ CargaArchivosCedulaMasivo</li> <li>▶ CargaArchivosProductoMasivo</li> <li>▶ Clientes</li> <li>▶ Comprobantes</li> <li>▶ Configuracion</li> <li>▶ Content</li> <li>▶ css</li> <li>▶ development-bundle</li> <li>▶ Documentos</li> <li>▶ Esquemas</li> <li>▶ fonts</li> <li>▶ GestionAcademica</li> <li>▶ Images</li> <li>▶ js</li> <li>▶ Maestro</li> <li>▶ Mail</li> <li>▶ Procesos</li> <li>▶ Publico</li> <li>▶ puntoVenta</li> <li>▶ Reportes</li> <li>▶ Scripts</li> <li>▶ Seguridad</li> <li>▶ SeguridadEmpresarial</li> <li>▶ Soap</li> <li>▶ Styles</li> <li>▶ Temp</li> <li>▶ Tramas</li> <li>▶ About.aspx</li> <li>▶ aplicaciones.aspx</li> <li>▶ Bundle.config</li> <li>▶ BusquedaIdentificacion.aspx</li> <li>▶ Buzon.master</li> <li>▶ Clientes.master</li> <li>▶ Contact.aspx</li> <li>▶ DatosComprobanteRetencion.ascx</li> <li>▶ DatosEmisor.ascx</li> <li>▶ Default.aspx</li> <li>▶ favicon.ico</li> </ul>
--	--

Tabla 8 Objetos de API Botón de Pagos dinero electrónico

## **4.2. FASE II: PLAN DE PRUEBAS**

### **4.2.1. INTRODUCCIÓN**

El plan de pruebas de sistemas permite especificar lo que se va probar y cómo se ejecutan las pruebas del proyecto. Este Puede contener un conjunto de pruebas para sus casos de prueba o puede crear una jerarquía de conjuntos de pruebas. La configuración del plan de pruebas permitirá informar al evaluador el resultado de las pruebas. Cada vez que se ejecuta el plan de prueba se graban los resultados, de esta manera el plan permite medir el progreso de las pruebas.

El plan de pruebas se debe revisar por el equipo de trabajo, para comprobar que todos los casos creados cubren las funcionalidades de la aplicación. Los casos de prueba son un conjunto de condiciones o variables mediante las cuáles un evaluador determinará si una aplicación, un sistema software, o una característica, cumplen lo requerido por el usuario, si está parcialmente o completamente satisfactorio el requerimiento. Se asigna el conjunto de prueba al evaluador correspondiente.

Cuando se ejecuta el software desarrollado, la fase de las pruebas es la más crítica, existen diferentes interesados que requieren información sobre el estado de la calidad del software que se está desarrollando. Se debe manejar un informe del avance de las pruebas, el cual puede ser solicitado una o varias veces durante la ejecución de

las pruebas. La intención es comunicar a todos los miembros del equipo el estado de las pruebas realizadas, que defectos críticos se están reportando y cuantos casos faltan por ejecutar.

#### 4.2.2. ARTEFACTOS DE PRUEBA

Los artefactos de prueba utilizados para el desarrollo de la aplicación son los siguientes:

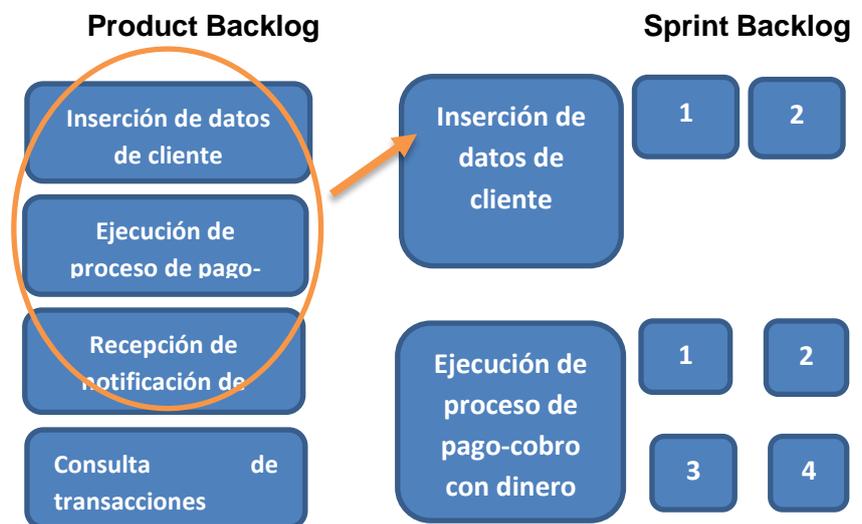


Figura 4.8 Product Backlog y Sprint Backlog del Proyecto

**Visión del producto:** Cliente elige forma de pago efectivo, es decir Dinero Electrónico.

**Objetivo del sprint 1:** Como un **usuario del comercio**, quisiera realizar el **pago** para poder **confirmar la compra** en línea.

**Listado de Tareas:** El tablero mostrará las tareas en estado pendiente, en progreso, por verificar y completadas de acuerdo a la planificación del proyecto.



Figura 4.9 Taskboard del proyecto

**Burndown Chart** Este gráfico muestra la cantidad de trabajo restante en el sprint.

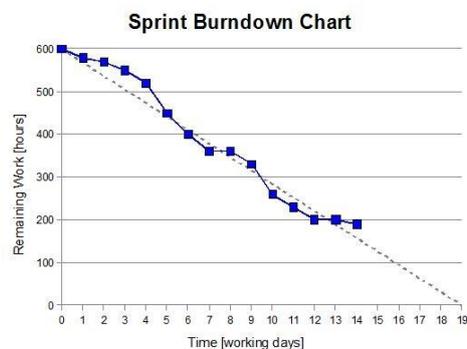


Figura 4.10 Burndown Chart del proyecto

### 4.2.3. CARACTERÍSTICAS A SER PROBADAS

#### Portal de clientes

- Visualización y ejecución de forma de pago efectivo (Botón de pagos dinero electrónico).

- Registro de Datos de usuario para pagos efectivo (Usuario con cuenta dinero electrónico).

#### **Portal Administrativo**

- Consulta de transacciones de compra realizadas con dinero electrónico.

#### **Servicios dinero electrónico**

- Envío y recepción de tramas de los métodos de cobro de dinero electrónico.

#### **Consulta de estados de transacciones**

- Envío y recepción de solicitudes de cobro de dinero electrónico.

#### **Carga, Concurrencia y Rendimiento del API**

- SET de datos para evaluar resultados por Pruebas de carga, concurrencia y rendimiento del API Botón de pagos.

### **4.2.4. CARACTERÍSTICAS QUE NO SERÁN PROBADAS**

No aplica.

### **4.2.5. TIPOS DE PRUEBAS**

**Pruebas Unitarias** Se realizarán pruebas al código a medida que avanza el desarrollo del software.

**Pruebas de integración** Se realizarán pruebas de integración con los servicios de cobro del Banco Central del Ecuador.

**Pruebas de sistemas** Se realizarán pruebas funcionales y no funcionales para verificación de lo requerido por el usuario del comercio.

**Pruebas de aceptación** Se realizarán pruebas para validación del requerimiento con el dueño del comercio.

#### 4.2.6. PROCESOS DE PRUEBA

La certificación del sprint consistirá en la verificación de la funcionalidad especificada en las historias de usuarios del proyecto.

El Servicio Web WCF **ServicioDineroElectronico.svc** que recibe la invocación desde el Portal Web, el cual contiene los métodos que envían las tramas que generan el cobro de la transacción de compra con dinero electrónico, será probado con la herramienta **WCF Test Client**.

Métodos del servicio	
1	Cobro_Pre
2	GetBalance
3	CashIn_Pre
4	CashIn_Confirm
5	ReportTransaction
6	ReportTransaction_Msisdn
7	Cobro_Confirm

*Tabla 9 Métodos del servicio de pago de la solución*

Adicionalmente se realizará un SET de Pruebas en el ambiente de pruebas de Dinero electrónico del Banco Central del Ecuador con la herramienta **REST Console**, la cual verificará el estado de las transacciones ejecutadas con las solicitudes.

**Envío → POST**

**Respuesta → GET**

Se ejecutarán las pruebas de funcionalidad sobre el Portal Web del comercio electrónico que contendrá el API Botón de pagos con dinero electrónico.

Las pruebas de carga, concurrencia y rendimiento se realizarán en la herramienta JMETER.

# CAPÍTULO 5

## EJECUCIÓN DE PRUEBAS

### 5.1 DESARROLLO E IMPLEMENTACIÓN DE CLIENTE DE BOTÓN DE PAGO

1.- Generar Sitio Web en formato Binario “powerlinks.com.ec” implementar la publicación usando la URL: <https://web804.discountasp.net:8172/MsDeploy.axd> o realizar la Transferencia de Fuentes hacia <ftp.ecualinkSCO.web804.discountasp.net>.

2.- Actualizar el archivo de configuración con la URL del Servicio de Dinero Electrónico de BCE. En ambiente TEST se usa el atributo UsuarioWS\_BCE y claveWS\_BCE los cual en ambiente producción deben ir en Blanco debido a que la conexión en producción es por medio de la VPN.

```

15 |<id key="ProviderExcel" value="Provider=Microsoft.Jet.OLEDB.4.0;Data Source=@PI_Archivo;Extended Properties=Excel 8.0;"/>
16 |<id key="FileLengthUpArchivosProducto" value="99000"/>
17 |<id key="RutaCorreoNotificacion" value="C:\Proyectos\FE\FE_WEB\Mail\"/>
18 |<id key="ValorIVA" value="0.12"/>
19 |<id key="Url" value="186.101.243.246"/>
20 |<id key="Path_Certificate" value="C:\Proyectos\FE\bcecertdig.pfx"/>
21 |<id key="Key_Certificate" value="bce1"/>
22 |<id key="UsuarioWs_BCE" value="" />
23 |<id key="ClaveWs_BCE" value="" />
24 |<id key="Lenguaje_BCE" value="ES"/>
25 |<id key="WS_mts_bce_WebReference.MTSService" value="https://test.dineroelectronico.ec:8443/mts_bce/services/MTSService.MTSServiceHttpsSoap11Endpoint/">
26 |</Settings>
27 |</SectionStrings>

```

3.-La URL de Dinero Electrónico debe actualizarse de la siguiente forma:

```

<add key="WS_mts_bce_WebReference.MTSService"
value="https://test.dineroelectronico.ec:8443/mts_bce/services/MTSService.MTSService
HttpsSoap11Endpoint"/>

```

Cambiar a:

```

<add key="WS_mts_bce_WebReference.MTSService" value="https://
dineroelectronico.ec:8443/mts_bce/services/MTSService.MTSServiceHttpsSoap11Endpoi
nt"/>

```

4.- Configurar los avisos por correo electrónico del Portal Web de Dinero Electrónico usando el SMTP: [smtp.powerlinks.com.ec](mailto:smtp.powerlinks.com.ec).

5.- Realizar la configuración del Lenguaje que se usará para los mensajes devueltos por Dinero Electrónico. Este mensaje se incluye en las respuestas a las transacciones:

```

<add key="Lenguaje_BCE" value="ES"/>

```

6.- Service Broker y SQL Server Agent, ambos servicios se deben configurar con una cuenta de dominio que tenga permisos de Administrador en el Servidor

Windows de BDD, que también será utilizada para el procesamiento en background por el Servicio del SQL Broker, y que debe ser incluido en las Base SQL2008R2\_953209\_ecualinks como “dbowner” (independiente de que tenga o no rol de “sysadmin”).

## 5.2 DESCRIPCIÓN DEL ENTORNO DE PRUEBAS

### 5.2.1 HARDWARE

Computador	Detalles de Procesador	Memoria	Especificaciones de Almacenamiento
<b>Servidor de BDD</b> SQL Server 2012 Estándar	Quad Core - 64 bits	16 GB	250 GB
<b>Servidor de Aplicación</b> Servidor Aplicativo WEB en Nube de Discountasp.net	Quad Core - 64 bits	16 GB	1250 GB

*Tabla 10 Hardware del ambiente de pruebas*

### 5.2.2 SOFTWARE DEL SISTEMA

Software / Componente	Versión	Sistema Operativo	Service Pack / Patch
Windows Server 2012	R2	Windows Server 2012	SP3
SQL Server 2012	Estándar	Windows Server 2012	SP3

Internet Information Server	8.0	Windows Server 2012	N/A
Message Queue Server	R2	Windows Server 2012	N/A

*Tabla 11 Software del sistema*

### 5.2.3 OTRAS CONFIGURACIONES

Tipo	Versión
.NET Framework	4.5
Windows Communication Foundation	N/A
SQL Server Management Studio	N/A
SQL Server Profiler	N/A
Service Broker SQL	N/A
SQL Server Agent	N/A

*Tabla 12 Otras configuraciones del sistema*

### 5.2.4 HERRAMIENTAS DE PRUEBAS

Herramienta	Versión	Tipo de Prueba
WCF Test Client	9.0	Pruebas de Integración
JMeter	2.10	Pruebas de Sistemas

*Tabla 13 Herramientas de pruebas*





Figura 5.2 Trama de ejemplo

3.- Se verifica el Esquema de SOAP de Entrada en Ejecución del método de cobro.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:bce="http://bce.webservices.mts.inswitch.us">
  <soapenv:Header/>
  <soapenv:Body>
    <bce:cobro>
      <!--Optional:-->
      <bce:dtoRequestCobro>
        <!--Optional:-->
        <xsd:amount@amount</xsd:amount>
        <!--Optional:-->
        <xsd:brandId</xsd:brandId>
        <!--Optional:-->
        <xsd:currency</xsd:currency>
        <!--Optional:-->
        <xsd:document</xsd:document>
        <!--Optional:-->
        <xsd:language@language</xsd:language>
        <!--Optional:-->
        <xsd:msisdnSource</xsd:msisdnSource>
        <!--Optional:-->
        <xsd:msisdnTarget@msisdnTarget</xsd:msisdnTarget>
        <!--Optional:-->
        <xsd:password@password</xsd:password>
        <!--Optional:-->
        <xsd:pin</xsd:pin>
        <!--Optional:-->
        <xsd:user@user</xsd:user>
        <!--Optional:-->
        <xsd:utfi@utfi</xsd:utfi>
      </bce:dtoRequestCobro>
    </bce:cobro>
  </soapenv:Body>
</soapenv:Envelope>
```

Figura 5.3 Esquema de método de cobro

4.- Se prepara un SET de Pruebas con la herramienta **REST Console** para verificar estados de Transacciones ejecutadas.

Envío POST	Respuesta GET
<pre> Array (   [idNegocio] =&gt; 00001   [idTransaccion] =&gt; 1234567890   [referencia] =&gt; "Valor Referencial al Cliente Ejemplo una Observacion"   [codigoAutorizacion] =&gt; 1234567890123456789 0123456789012345678 90   [cedula] =&gt; 0920388000   [celular] =&gt; 0912345678   [valor] =&gt; 1000   [fechaProcesamiento] =&gt; 03-12-2015 12:00:00 AM ) </pre>	<pre> Array (   [idNegocio] =&gt; 00001   [idTransaccion] =&gt; 1234567890   [idEstado] =&gt; 1   [nombreEstado] =&gt; Aprobada   [referencia] =&gt; TEST_767   [codigoRespuesta] =&gt; 1   [codigoAutorizacion] =&gt; 12345678901234567890 12345678901234567890   [valor] =&gt; 1000   [fechaProcesamiento] =&gt; 03-12-2015 12:00:00 AM   [mensaje] =&gt; ) </pre>

Tabla 14 Tramas de ejemplo REST Console

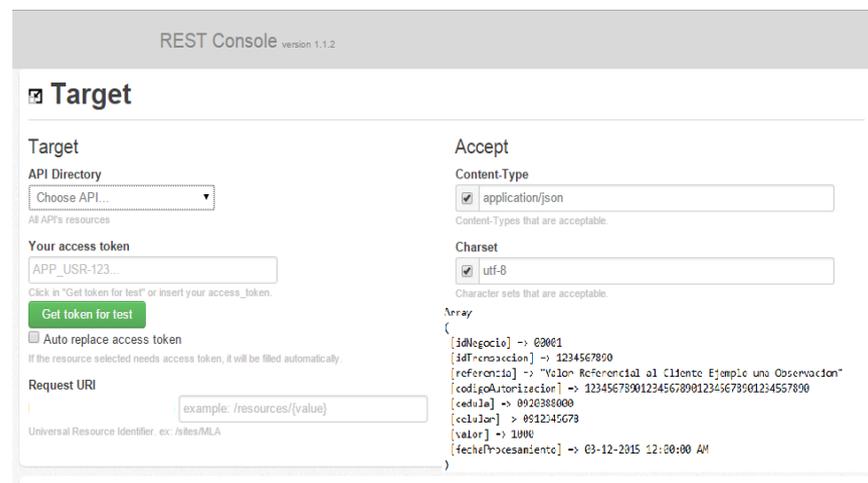


Figura 5.3 Envío de transacciones en REST Console

5.- Se realiza la carga de los datos para verificar el resultado de las transacciones realizadas. A continuación la especificación de los posibles mensajes de respuestas de acuerdo a las tramas enviadas.

Código de Respuesta	Mensaje	Mensaje Técnico
2	No se puede realizar la transacción. Inténtelo más tarde. Gracias por utilizar Efectivo desde mi c	Falla en el Servicio de Dinero Electrónico por indisponibilidad.
2	Transacción USSD no ha sido procesada	Falla cuando el Usuario no confirma la transacción.
1	La operación se completó con éxito. Ref.: 225450	Cuando se ejecutó con éxito Y genera un ID de Referencia .

Tabla 15 Código de respuesta de transacciones

The screenshot shows a SQL query result in a database management tool. The query is: `SELECT * FROM [SQL2008R2_953209_ecuainlinks].[DineroE].[DE_Transaccion_Cobro]`. The result is a table with columns: `idTransaccion`, `idReceptor`, `DocumentoOrigen`, `NumeroOrigen`, `Valor`, `CodError`, `MgsError`, `FechaPeticon`, `FechaRespuesta`, `FechaRegistro`, and `Referencia`. The table contains 29 rows of data, including transaction IDs, values, error codes, and messages. For example, row 1 shows a successful transaction with a value of 1.00 and a reference of 2016-12-18 22:40:10.083. Row 20 shows a successful transaction with a value of 1.00 and a reference of 2016-12-29 21:37:24.237. Row 21 shows a transaction that failed with the message 'Transacción USSD no ha sido procesada' and a reference of 2016-12-28 22:18:37.760.

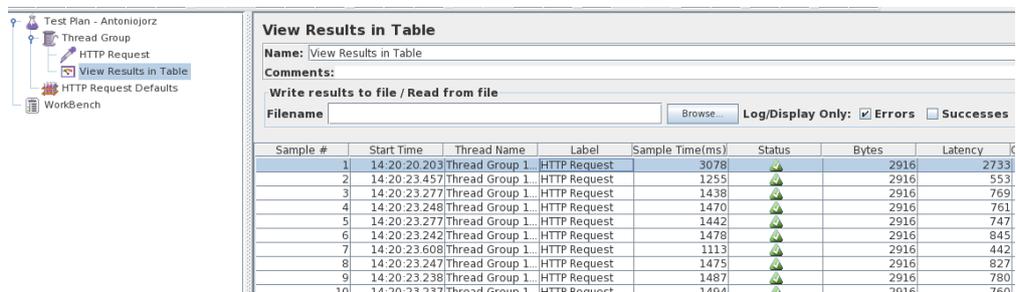
Figura 5.4 Consulta de transacciones en Base de datos

## 5.4 PRUEBAS INTERNAS DE CONCURRENCIA

El comportamiento de la aplicación dependerá de la concurrencia que exista en un momento determinado sobre la aplicación. La prueba nos indicará el número de operaciones que se realizan en un determinado periodo de

tiempo. Adicional el número de operaciones que se realizan de manera simultánea.

1. Ejecutar el programa JMETER. Crear el plan de pruebas.
2. Lanzar un conjunto de peticiones para ver cómo responde el API. Añadir al plan un grupo de Hilos que se encargará de lanzar las peticiones.
3. Configurar los elementos del grupo de hilos. Seleccionar opción para una petición web (Http Request).



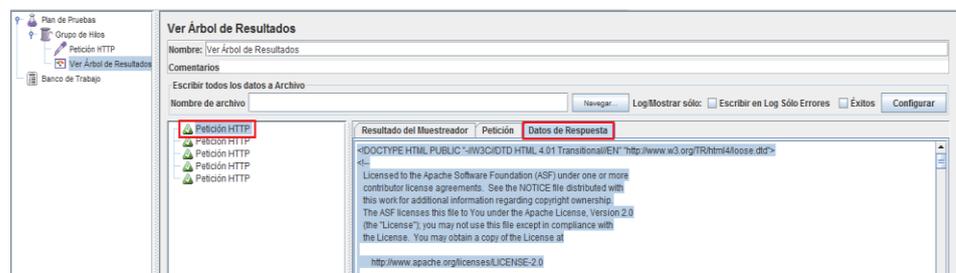
The screenshot shows the 'View Results in Table' window in JMeter. It displays a table with columns: Sample #, Start Time, Thread Name, Label, Sample Time(ms), Status, Bytes, and Latency. The table contains 10 rows of data for HTTP requests, with sample times ranging from 1113 to 3078 ms and latencies from 442 to 2733 ms. All requests show a green status icon, indicating success.

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Latency
1	14:20:20.203	Thread Group 1	HTTP Request	3078	🟢	2916	2733
2	14:20:23.457	Thread Group 1	HTTP Request	1255	🟢	2916	553
3	14:20:23.277	Thread Group 1	HTTP Request	1438	🟢	2916	769
4	14:20:23.248	Thread Group 1	HTTP Request	1470	🟢	2916	761
5	14:20:23.277	Thread Group 1	HTTP Request	1442	🟢	2916	747
6	14:20:23.242	Thread Group 1	HTTP Request	1478	🟢	2916	845
7	14:20:23.608	Thread Group 1	HTTP Request	1113	🟢	2916	442
8	14:20:23.247	Thread Group 1	HTTP Request	1475	🟢	2916	927
9	14:20:23.238	Thread Group 1	HTTP Request	1487	🟢	2916	780
10	14:20:23.237	Thread Group 1	HTTP Request	1464	🟢	2916	761

Figura 5.5 Vista de resultados JMETER

4. Verificar si las peticiones que lanzamos se ejecutan correctamente. Para ello usaremos el concepto de Listener el cual se encarga de almacenar los resultados.

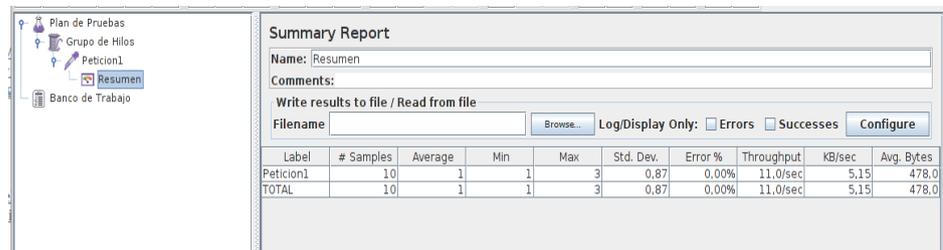
- 4.1. Listener para verificar el correcto funcionamiento. Seleccionar opción Árbol de resultados.



The screenshot shows the 'Ver Árbol de Resultados' window in JMeter. The left pane shows a tree view with 'Petición HTTP' selected. The right pane shows the 'Resultado del Muestreador' for the selected request, displaying the response body which is an Apache License notice. The 'Datos de Respuesta' tab is active, showing the raw response text.

Figura 5.6 Árbol de resultados JMETER

- 4.2. Listener para visualizar el resumen de resultados de las peticiones realizadas. Seleccionar opción Reporte Sumarizado.



The screenshot shows the JMeter Summary Report window. The 'Name' field is 'Resumen'. Below it, there are fields for 'Write results to file / Read from file' and 'Filename'. There are also checkboxes for 'Log/Display Only: Errors' and 'Successes', and a 'Configure' button. A table displays the test results:

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	KB/sec	Avg. Bytes
Peticion1	10	1	1	3	0,87	0,00%	11,0/sec	5,15	478,0
TOTAL	10	1	1	3	0,87	0,00%	11,0/sec	5,15	478,0

Figura 5.7 Reporte sumariada JMETER

## 5.5 PRUEBAS INTERNAS DE CARGA

Las pruebas internas de carga inicial permiten verificar la disponibilidad del Servicio Web WCF que recibe la invocación desde el Portal.

- 1.- Abrir herramienta WCF Test Client. Anadir el servicio **ServicioDineroElectronico.SVC**.
2. Seleccionar método del servicio para probar. Ingresar los parámetros de entrada y verificar respuesta que envía el servicio en cada uno de los métodos invocados.

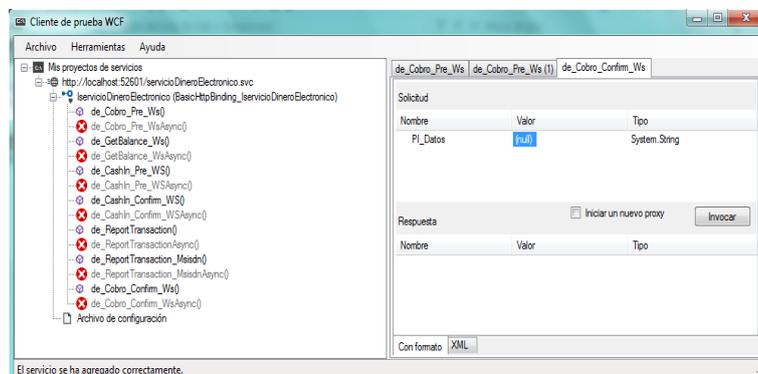


Figura 5.8 Vista de servicio de Dinero electrónico en WCF Test Client

## 2.1 Para obtener el Balance de una cuenta

**TRAMA de Entrada:** <de numero="0983072109" pin="0123"  
language="ES" sesion="AODA9WF9" />

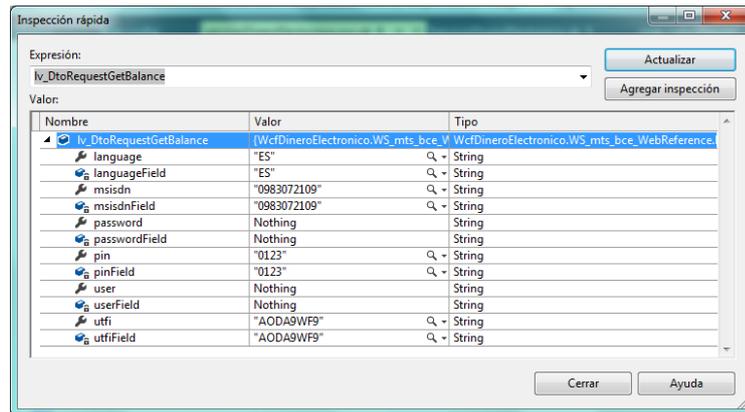


Figura 5.9 Trama de ejemplo 1

## 2.2. Para realizar una Transacción de cobro

**Trama de Entrada:** <de language="ES" sesion="Y3L5OWMY"  
document="0920377041" pin="0123" amount="0.10"  
numeroDestino="0999874131" numeroOrigen="0983072109"/>

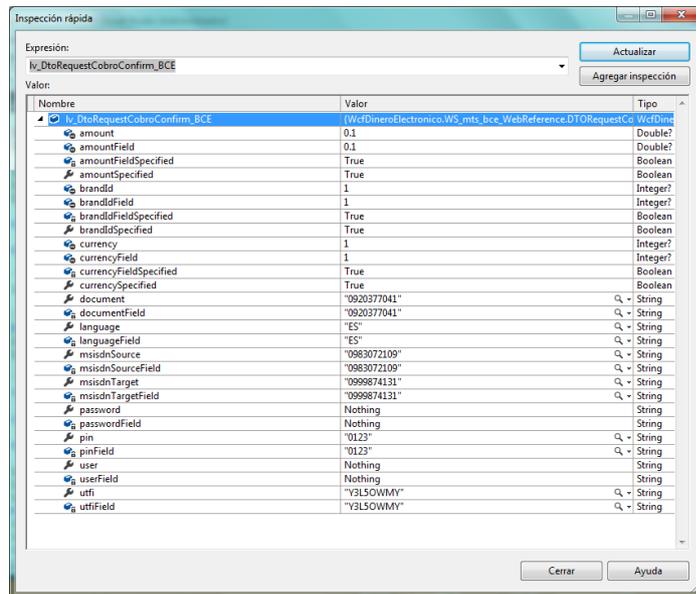


Figura 5.10 Trama de ejemplo 2

### 2.3. Visualizar Reporte de transacciones

**Trama de Entrada:** <de dateFrom="2017-09-06" dateTo="2017-10-06"  
language="ES" sesion="J0XMI1DK" />

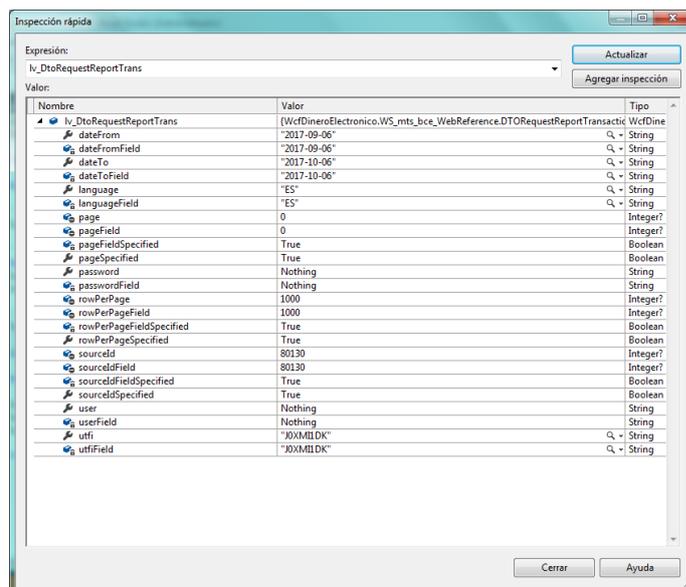


Figura 5.11 Trama de ejemplo 3

## 5.6 PRUEBAS DE ACEPTACIÓN CON LOS USUARIOS

Las pruebas de aceptación se realizaron con el dueño del comercio que ejecutó las validaciones necesarias para aprobar el requerimiento implementado en su Portal WEB.

1.- Ingresar a la página web del comercio. Realizar la selección de productos para compra en línea y datos de factura para la entrega del producto.

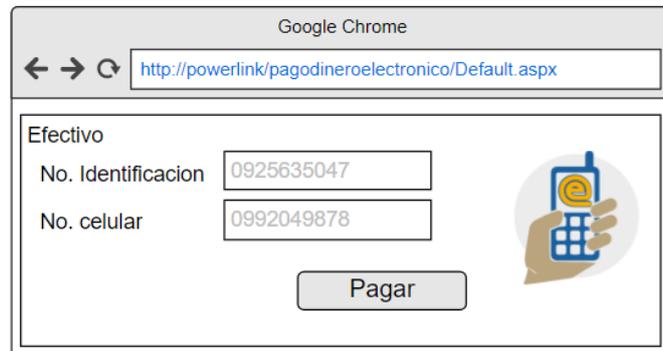
The screenshot shows a Google Chrome browser window with the URL <http://powerlink/compraenlinea/Default.aspx>. The page is divided into three main sections:

- Seleccionar Producto:** Contains two radio button options: "Celular Samsung" (unchecked) and "Laptop DELL" (checked). To the right, a summary table shows:
 

Subtotal	1800,00
IVA 12%	216,00
<b>Total a pagar</b>	<b>2016,00</b>
- Forma de pago:** Displays five payment method icons: a blue circle, a Mastercard logo, a blue circle, a PacificCard logo, and a blue circle with a Diners Club logo.
- Datos de factura:** Contains four input fields:
  - No. Identificacion: A dropdown menu set to "Cedula" and a text box containing "0925635047".
  - Nombre: A text box containing "JOSE DAVID LOPEZ COBENIA".
  - Telefono: A text box containing "0992049878".
  - Direccion: A text box containing "GUASMO SUR COOP 9 DE ABRIL MZ8 SL2".

Figura 5.12 Pantalla de compra en línea y botón de pago

2.- Seleccionar forma de pago con dinero electrónico. Ingresar datos del usuario del comercio para realizar el pago de la transacción.



Google Chrome

← → ↻ <http://powerlink/pagodineroelectronico/Default.aspx>

Efectivo

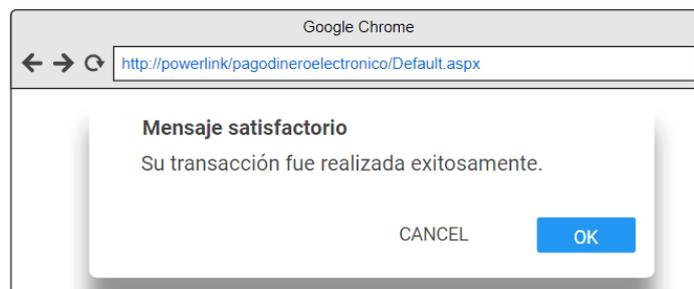
No. Identificacion

No. celular



Figura 5.13 Pantalla de datos de pago de transacción

- 3.- Confirmar el pago con SMS de aprobación. Visualizar mensaje de proceso satisfactorio.
- 4.- Visualizar SMS del valor de la transacción acreditado en la cuenta del dueño del comercio.



Google Chrome

← → ↻ <http://powerlink/pagodineroelectronico/Default.aspx>

**Mensaje satisfactorio**

Su transacción fue realizada exitosamente.

Figura 5.14 Pantalla de mensaje satisfactorio

## **CAPÍTULO 6**

### **ANÁLISIS DE RESULTADOS**

#### **6.1. EVALUACIÓN Y ANÁLISIS DE RESULTADOS**

La evaluación de los resultados de las pruebas de rendimiento, concurrencia y carga ejecutadas en el aplicativo botón de pago con dinero electrónico, fueron analizados y visualizados en la herramienta de prueba JMETER.

El servicio que procesa el cobro de las transacciones con dinero electrónico en el portal del comercio electrónico obtuvo una velocidad de respuesta inmediata al ejecutar los diferentes escenarios de prueba. Este análisis indica que el servicio respondió satisfactoriamente a las pruebas de carga y rendimiento.

El aplicativo botón de pago con dinero electrónico obtuvo un comportamiento esperado al realizar las pruebas de concurrencia. Este análisis demostró que puede realizar múltiples operaciones en simultáneo durante un lapso de tiempo determinado.

La muestra tomada fue de 300 transacciones, las mismas que fueron enviadas y que obtuvieron como resultado 235 respuestas de transacciones exitosas (aprobadas) y 65 respuestas de transacciones fallidas (rechazadas) con sus respectivos códigos.

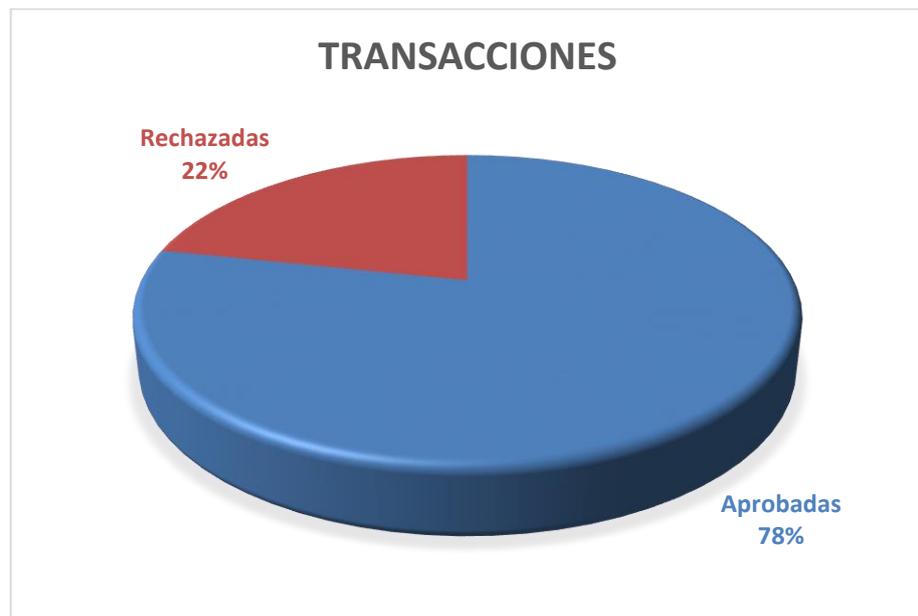


Figura 6.1 Gráfico de transacciones realizadas con dinero electrónico

## 6.2. PROPUESTAS INICIALES DE ALTERNATIVAS DE MEJORA

Los usuarios del comercio electrónico adoptarán dinero electrónico como otro medio de pago debido a los beneficios que este servicio ofrece mediante el uso de dispositivos móviles.



*Figura 6.2 Beneficios de dinero electrónico*

El uso de dinero electrónico no se verá afectado porque el servicio de dinero electrónico del Banco Central del Ecuador provee las seguridades necesarias porque no existe exposición de datos personales en la red y requiere de una clave única para cada cuenta asociada del usuario.



*Figura 6.3 Seguridad de sistema*

La inclusión financiera no será un obstáculo para los usuarios del comercio electrónico debido a que no necesitan realizar una serie de trámites para

crear su cuenta asociada en el Banco Central del Ecuador esto permite que dinero electrónico sea aceptado como medio de pago.



*Figura 6.4 Inclusión financiera*

## **CONCLUSIONES Y RECOMENDACIONES**

Al finalizar la tesis, llegamos a las siguientes conclusiones:

1. El uso del dinero electrónico a través del comercio electrónico incrementaría la velocidad de circulación del dinero y la masa monetaria, esto daría lugar a variaciones en los efectos de inflación del País.
2. La falta de información es porque el sistema de dinero electrónico no ha sido lo suficientemente impartido por los medios de comunicación, por tal motivo el desconocimiento del sistema genera que las personas desconfíen y sientan temor de utilizarlo.
3. El uso del dinero electrónico incrementaría siempre que a los usuarios del comercio electrónico se instruyan correctamente de las funcionalidades y operatividad del sistema.

Como recomendación podríamos señalar:

1. Dar a conocer los motivos por los que fue implementado por el Banco Central del Ecuador el sistema de dinero electrónico en los medios de comunicación.
2. Brindar capacitaciones a los negocios, comerciantes y a los usuarios, para incrementar la confiabilidad y credibilidad para que opten por la aplicación del sistema, como un medio regular en las transacciones que efectúen.
3. Asesorar los usuarios del comercio en términos que forman parte de la funcionalidad y operatividad del dinero electrónico.

## BIBLIOGRAFÍA

- [1] J. Pérez Porto, «Definicion .de,» 2016. [En línea]. Available: <https://definicion.de/comercio-electronico/>.
- [2] A. d. Molina, «conexionesan,» 1652. [En línea]. Available: <https://www.esan.edu.pe/apuntes-empresariales/2015/04/e-commerce-que-tipos-comercio-electronico-existen/>.
- [3] P. V. Mojica, «Human Level Communication,» 2016. [En línea]. Available: <https://www.humanlevel.com/articulos/comercio-electronico/medios-de-pago-de-un-comercio-electronico.html>.
- [4] Gobierno Nacional de la Republica del Ecuador, «SRI,» 2015. [En línea]. Available: <http://www.sri.gob.ec/web/guest/dinero-electronico>.
- [5] Banco Central del Ecuador, «efectivo desde mi celular,» 2016. [En línea]. Available: <https://www.efectivo.ec/beneficios-de-efectivo-2/>.
- [6] SUSE LLC and others, «heinlein,» 2001. [En línea]. Available: [https://es.opensuse.org/Software\\_libre\\_y\\_de\\_c%C3%B3digo\\_abierto](https://es.opensuse.org/Software_libre_y_de_c%C3%B3digo_abierto).
- [7] Microsoft , «Microsoft Developer Network,» 2016. [En línea]. Available: [https://msdn.microsoft.com/es-es/library/hh425099\(v=vs.110\).aspx](https://msdn.microsoft.com/es-es/library/hh425099(v=vs.110).aspx).
- [8] Microsoft, «Microsoft Centro de Descarga,» 2010. [En línea]. Available: <https://www.microsoft.com/es-es/download/details.aspx?id=22985>.
- [9] Ecured, «Ecured Conocimiento con todos y para todos,» 2004. [En línea]. Available: [https://www.ecured.cu/Servidor\\_de\\_Aplicaciones](https://www.ecured.cu/Servidor_de_Aplicaciones).
- [10] Microsoft, «Microsoft Centro de Descarga,» 2014. [En línea]. Available: <https://www.microsoft.com/es-es/download/details.aspx?id=22985>.
- [11] Microsoft, «Microsoft Developer Network,» 2016. [En línea]. Available: [https://msdn.microsoft.com/es-es/library/hh831725\(v=ws.11\).aspx](https://msdn.microsoft.com/es-es/library/hh831725(v=ws.11).aspx).
- [12] Stack Overflow Business, «Stack overflow en Español,» 2016. [En línea]. Available: <https://es.stackoverflow.com/questions/34904/cuando-debo-usar-los-m%C3%A9todos-post-y-get>.

- [13] International Software Testing Qualifications Board, «International Software Testing Qualifications Board,» 2011. [En línea]. Available: <http://www.istqb.org/downloads/send/2-foundation-level-documents/3-foundation-level-syllabus-2011.html>.
- [14] Unidad de Desarrollo Tecnológico en Inteligencia Artificial , «Unidad de Desarrollo Tecnológico en Inteligencia Artificial IIIAS-CSIC,» 2002. [En línea]. Available: <http://www.iiia.csic.es/udt/es/blog/jrodriguez/2008/metodologia-desarrollo-sotware-modelo-en-v-o-cuatro-niveles>.
- [15] Grupo HDI, «Grupo HDI Un Nuevo Paradigma en la Calidad del Software,» 2014. [En línea]. Available: <http://www.grupohdi.com/webnew/nonfunctional3.html>.
- [16] UN POCO DE JAVA Y +, «UN POCO DE JAVA Y + Otra forma de hablar de nuestro día a día,» 2013. [En línea]. Available: <https://unpocodejava.wordpress.com/2013/11/04/buenas-practicas-para-disenar-una-api-restful-pragmatic-parte-i-3/>.
- [17] M. B. y. M. Rodriguez, «Universidad del País de Vasco Euskal Herriko Unibertsitatea,» 2016. [En línea]. Available: <http://www.ehu.eus/mrodriguez/archivos/csharp/pdf/ServiciosWeb/Webservices.pdf>.

## GLOSARIO

**Comercio electrónico** más conocido como e-commerce (electronic commerce) o negocios online, consiste en la compra y venta de productos o de servicios a través de medios electrónicos como Internet.

**Dinero electrónico** es un medio de pago electrónico, implementado por el Banco Central del Ecuador, cuyo respaldo es el dinero físico.

**Dispositivo móvil** es un tipo de computadora de tamaño pequeño, con capacidades de procesamiento, con conexión a Internet, con memoria, diseñado específicamente para una función.

**SCRUM** es el nombre con el que se denomina a los marcos de desarrollo ágiles caracterizados por adoptar una estrategia de desarrollo incremental, en lugar de la planificación y ejecución completa del producto.

**Web Services** es una tecnología que utiliza un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones.

**IIOB** es un protocolo de comunicación de CORBA. Define el modo en que los bits se envían a través de una conexión entre clientes y los servidores CORBA. CORBA es una arquitectura de objetos distribuidos desarrollada por el grupo de gestión de objetos (OMG).

**Interfaz de programación de aplicaciones**, es un conjunto de subrutinas, funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

**Marco de trabajo de aplicaciones orientadas a servicios**, envía datos como mensajes asincrónicos de un extremo de servicio a otro.

**Red Privada Virtual**, es una tecnología de red de computadoras que permite una extensión segura de la red de área local (LAN) sobre una red pública o no controlada como Internet.

**GET** lleva los datos de forma "visible" al cliente (navegador web). El medio de envío es la URL. Los datos los puede ver cualquiera.

**POST** consiste en datos "ocultos" (porque el cliente no los ve) enviados por un formulario cuyo método de envío es post. Es adecuado para formularios. Los datos no son visibles.

**RESTful Representational state transfer (REST)**, hace referencia a un servicio web que implementa la arquitectura REST.

**Protocolo simple de acceso a objetos**, es un protocolo estándar que define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML

**Lenguaje de descripción del servicio web**, es un formato del Extensible Markup Language (XML) que se utiliza para describir servicios web (WS).

**La Asociación GSM**, es una organización de operadores móviles y compañías relacionadas, dedicada al apoyo de la normalización, la implementación y promoción del sistema de telefonía móvil GSM.

# ANEXOS

## 1. Cronograma Modelo de empresas 2016- Formato de Banco Central del Ecuador.

CRONOGRAMA INTEGRACIÓN						
Actividades	Fecha Inicio	Fecha Fin	Responsable	Contacto	Estado	
<b>1 CRONOGRAMA INTEGRACIÓN PARA SERVICIOS WEB</b>	17 días					
2 Reunión de arranque	1 día					
<b>3 Desarrollo Integración</b>						
4 Envío de URLs en test	1 día					
5 Envío de documentación de Web services	1 día					
6 Creación de usuarios en test	1 día					
7 Adición de móviles en lista blanca	1 día					
8 Adición de IP en test	1 día					
9 Envío de certificado digital de pruebas	1 día					
10 Consumo de web services en test con certificado	5 días					
11 Certificación de integración	1 día					
<b>12 VPNs</b>						
13 Envío de formularios para VPNs	1 día					
14 Reunión preliminar previo entrega de Formularios	1 día					
15 Recepción de formularios (BCE)	1 día					
16 Levantar conexión VPN con sitio principal	2 días					
17 Pruebas de conectividad sitio principal	1 día					
18 Levantar conexión VPN con sitio alternativo	2 días					
19 Pruebas de conectividad sitio alternativo	1 día					
20 Pruebas de contingencia	1 día					
21 Certificación VPNs	1 día					
<b>22 Certificado Digital</b>						
23 Reunión preliminar definición del Certificado Digital						
24 Entrega de documentación habilitante	1 día					
25 Recepción de documentación lleno	1 día					
26 Pago de certificado	1 día					
27 Generación de llave privada y archivo request	1 día					
28 Firma y entrega del certificado	1 día					
29 Consumo de Web services con certificado final en ambiente de test	1 día					
<b>30 PRUEBAS FINALES PREPRODUCCION</b>						
31 Adición de IP en lista blanca en preproducción						
32 Adición de números móviles para pruebas						
33 Envío de URLs en PreProducción	1 día					
34 Creación de estructura y entrega de usuarios en PreProducción	1 día					
35 Pruebas de consumo de web services con VPN y Certificado en ambiente de preproducción	1 día					
36 Informe de certificación del Agente y resultado de las pruebas	1 día					
<b>37 PRUEBAS FINALES PRODUCCION</b>						
38 Adición de IP en lista blanca en producción	1 día					
39 Envío de URLs en Producción	1 día					
40 Envío del archivo Registro de Participantes para la creación de usuarios en Producción PDE	1 día					
41 Creación de estructura y entrega de usuarios en Producción	1 día					
42 Pruebas de consumo de web services con VPN y Certificado en ambiente de producción	1 día					
43 Informe de certificación del Agente y resultado de las pruebas	1 día					
44 Pruebas en Producción Piloto						