



UNIVERSIDAD
PRIVADA
DEL NORTE

FACULTAD DE INGENIERÍA

CARRERA DE INGENIERÍA DE SISTEMAS COMPUTACIONALES

“IMPLEMENTACIÓN DE UN SISTEMA WEB PARA
MEJORAR LA GESTIÓN DE PRÉSTAMOS EN UNA
EMPRESA FINANCIERA”

Tesis para optar el título profesional de:

Ingeniero de Sistemas Computacionales

Autor(es):

Chávez Domínguez, Miguel Ángel
Troncozo Centurión, Graly Giovanni

Asesor:

Mg. Ing. Lourdes Roxana Díaz Amaya

Trujillo – Perú
2018

APROBACIÓN DE LA TESIS

La asesora y los miembros del jurado evaluador asignados, **APRUEBAN** la tesis desarrollada por el bachiller Chávez Domínguez Miguel Ángel y la bachiller Troncozo Centurión Graly Giovanni, denominada:

“IMPLEMENTACIÓN DE UN SISTEMA WEB PARA MEJORAR LA GESTIÓN DE PRÉSTAMOS EN UNA EMPRESA FINANCIERA”

Mg. Ing. Lourdes Roxana Díaz Amaya
ASESOR

Mg. Ing. Víctor Enemesio Dávila Rodríguez
JURADO
PRESIDENTE

Mg. Ing. Rolando Javier Berrú Beltrán
JURADO

Ing. Luis Mauricio Gutiérrez Magán
JURADO

DEDICATORIA

Dedicamos la presente tesis a nuestros padres que siempre confiaron en nosotros y que con su trabajo constante y enseñanzas nos ayudan día a día a alcanzar nuestras metas y a superar todas las brechas que se nos presentan en el camino.

También a todos aquellos quienes han estado a nuestro lado siempre, impulsándonos en todo momento a salir adelante.

AGRADECIMIENTO

Agradecemos a nuestra asesora por todo el tiempo y esfuerzo que dedicó apoyándonos a desarrollar la tesis. A nuestros profesores de carrera quien lograron llenarnos de los conocimientos necesarios para poder desarrollar este trabajo de investigación y que nunca dudaron en darnos consejos y apoyarnos cuando se lo solicitamos. A nuestros padres que siempre nos apoyaron moralmente y psicológicamente, para poder llegar a esta instancia de nuestros estudios.

ÍNDICE DE CONTENIDOS

APROBACIÓN DE LA TESIS.....	ii
DEDICATORIA.....	iii
AGRADECIMIENTO	iv
ÍNDICE DE CONTENIDOS	v
ÍNDICE DE TABLAS.....	vii
ÍNDICE DE FIGURAS	viii
RESUMEN.....	ix
ABSTRACT	x
CAPÍTULO 1. INTRODUCCIÓN.....	11
1.1. Realidad problemática	11
1.2. Formulación del problema.....	13
1.3. Justificación.....	13
1.4. Limitaciones	13
1.5. Objetivos	13
1.5.1. <i>Objetivo general</i>	13
1.5.2. <i>Objetivos específicos</i>	13
CAPÍTULO 2. MARCO TEÓRICO.....	14
2.1. Antecedentes	14
2.2. Bases teóricas.....	17
2.2.1. <i>Gestión de procesos</i>	17
2.2.2. <i>Servicio de préstamo</i>	17
2.2.3. <i>Automated Clearing House (ACH)</i>	17
2.2.4. <i>Declaraciones de pago</i>	18
2.2.5. <i>Metodologías de desarrollo de Software</i>	18
2.2.6. <i>Tecnología aplicada</i>	22
CAPÍTULO 3. HIPÓTESIS.....	34
3.1. Formulación de la Hipótesis.....	34
3.2. Operacionalización de variables	34
CAPÍTULO 4. DESARROLLO.....	35
CAPÍTULO 5. METODOLOGÍA.....	68
5.1. Diseño de investigación	68
5.2. Unidad de estudio	68
5.3. Población	68
5.4. Muestra (muestreo o selección).....	68
5.5. Técnicas, instrumentos y procedimientos de recolección de datos	68

5.6. Métodos, instrumentos y procedimientos de análisis de datos	69
CAPÍTULO 6. RESULTADOS	70
CAPÍTULO 7. DISCUSIÓN	85
CONCLUSIONES.....	88
RECOMENDACIONES	88
REFERENCIAS.....	89
ANEXOS	91

ÍNDICE DE TABLAS

Tabla 1: Comparación de metodologías: XP - Scrum.....	22
Tabla 2: Cuadro comparativo de Visual Studio y Eclipse	23
Tabla 3: Cuadro comparativo de Razor y ASPX.....	24
Tabla 4: Cuadro comparativo de MySQL y SqlServer	26
Tabla 5: Cuadro comparativo de Entity Framework y Linq to Sql.....	31
Tabla 6: Cuadro comparativo de XML y JSON	33
Tabla 7: Operacionalización de variable independiente	34
Tabla 8: Operacionalización de variable dependiente	34
Tabla 9: Equipo de trabajo	35
Tabla 10: Historia de usuario "Realización de un préstamo"	35
Tabla 11: Historia de usuario "Realización del pago de letras de un préstamo"	36
Tabla 12: Historia de usuario "Emisión de declaraciones de pago"	36
Tabla 13: Historia de usuario "Emisión del ACH Payment"	37
Tabla 14: Planificación Inicial de Historias de Usuario	38
Tabla 15: Estimación de tiempo de Historias de Usuario	38
Tabla 16: Plan de entrega	39
Tabla 17: Resumen de tareas de la historia de usuario "Realización de un préstamo"	39
Tabla 18: Tarjeta CRC del mantenedor prestatario	42
Tabla 19: Prueba funcional al mantenedor prestatario	43
Tabla 20 Tarjeta CRC del mantenedor prestamista:.....	47
Tabla 21: Prueba funcional del mantenedor prestamista	49
Tabla 22: Tarjeta CRC de la realización de un préstamo	51
Tabla 23: Prueba funcional de la realización de un préstamo	53
Tabla 24: Resumen de tareas de la historia de usuario "Realización del pago de letras de un préstamo"	54
Tabla 25: Tarjeta CRC de la realización del pago de letras de un préstamo	56
Tabla 26: Prueba funcional de la realización del pago de letras de un préstamo	58
Tabla 27: Resumen de tareas de la historia de usuario "Emisión de declaraciones de pago".....	59
Tabla 28: Tarjeta CRC de la emisión de declaraciones de pago.....	61
Tabla 29: Resumen de tareas de la historia de usuario "Emisión del ACH Payment"	63
Tabla 30: Tarjeta CRC de la realización de la emisión del ACH Payment	65
Tabla 31: Recolección de datos	68
Tabla 32: Análisis de datos	69
Tabla 33: Prueba de hipótesis.....	70
Tabla 34: Resultados del indicador tiempo promedio de la realización de un préstamo.....	71
Tabla 35: Resultados del indicador tiempo promedio de la realización del pago de letras de un préstamo.....	74
Tabla 36: Resultados del indicador tiempo promedio de la emisión de la declaración de pago	77
Tabla 37: Resultados del indicador tiempo promedio de la emisión del ACH Payment	80
Tabla 38: Resultados del indicador tiempo promedio de la emisión del ACH Payment.....	83
Tabla 39: Discusión Indicador Tiempo promedio de la realización del préstamo.....	86
Tabla 40: Discusión Indicador Tiempo promedio de la realización del pago de letras.....	86
Tabla 41: Discusión Indicador Tiempo promedio de la emisión de declaraciones de pago	86
Tabla 42: Discusión Indicador Tiempo promedio de la emisión del ACH Payment.....	87
Tabla 43: Discusión Indicador Valoración promedio de la gestión de préstamos	87
Tabla 44: Dimensión: Usabilidad - Indicador: Atracción	92
Tabla 45: Dimensión: Usabilidad - Indicador: Entendimiento	92
Tabla 46: Dimensión: Funcionalidad - Indicador: Adecuidad.....	93
Tabla 47: Dimensión: Confiabilidad - Indicador: Madurez	93

ÍNDICE DE FIGURAS

Figura 1: Iteraciones eXtreme Programming	19
Figura 2: El ciclo de Scrum	21
Figura 3: Arquitectura SignalR	28
Figura 4: Arquitectura de ADO Entity Framework.....	31
Figura 5: Esquema de funcionamiento XML	32
Figura 6: Diseño de interfaz principal del mantenedor prestatario	40
Figura 7: Diseño de interfaces del mantenedor prestatario	41
Figura 8: Diagrama de clases del mantenedor prestatario	42
Figura 9: Prueba unitaria de verificación de cuenta del mantenedor prestatario	43
Figura 10: Diseño de interfaz principal del mantenedor prestamista.....	45
Figura 11: Diseño de interfaces del mantenedor de prestamista	46
Figura 12: Diagrama de clases del mantenedor prestamista	47
Figura 13: Prueba unitaria de validación de apellido del mantenedor prestamista	48
Figura 14: Diseño de interfaz de la realización de un préstamo.....	50
Figura 15: Diagrama de clases de la realización de un préstamo	51
Figura 16: Prueba unitaria de verificación de cuenta en la realización de un préstamo	52
Figura 17: Diseño de interfaz de la realización del pago de letras de un préstamo	55
Figura 18: Diagrama de clases de la realización de pago de letras de un préstamo	56
Figura 19: Prueba unitaria de verificación de la tarifa del prestamista en la realización del pago de letras de un préstamo.....	58
Figura 20: Diseño de interfaz de la emisión de declaraciones de pago	60
Figura 21: Diagrama de clases de la emisión de declaraciones de pago	61
Figura 22: Prueba unitaria de validación de envío de declaraciones de pago	62
Figura 23: Diseño de interfaz de la emisión del ACH Payment	64
Figura 24: Diagrama de clases de la emisión del ACH Payment	65
Figura 25: Prueba unitaria de la emisión del ACH Payment.....	66
Figura 26: Diagrama de componentes.....	67
Figura 27: Diagrama de despliegue	67
Figura 28: Gráfico Resultado Indicador Tiempo promedio de la realización de un préstamo	73
Figura 29: Gráfico Resultado Indicador Tiempo promedio de realización del pago de letras de un préstamo.....	76
Figura 30: Gráfico Resultados Indicador del tiempo promedio de declaraciones de pago	79
Figura 31: Gráfico Resultados Indicador Tiempo promedio de la emisión del ACH Payment	82
Figura 32: Gráfico Resultados Indicador Valoración promedio de la gestión de préstamos.....	85
Figura 33: Declaración de pago de un prestamista	95
Figura 34: ACH Payment	96

RESUMEN

En el presente trabajo de investigación se tiene como finalidad mejorar la gestión de préstamos que se da entre dos o más entes dentro de una empresa financiera. Para ello, se propone el desarrollo de un sistema web, que reemplace al sistema de escritorio con el que cuentan actualmente, demostrando ser más eficiente, pues cumplimos satisfactoriamente los objetivos específicos de la siguiente forma:

Se logró determinar el tiempo promedio utilizado en la emisión de declaraciones de pago, reduciéndolo en un 25.58%.

Se logró determinar el tiempo promedio utilizado en la emisión del ACH Payment, reduciéndolo en un 46.15%.

Se logró determinar el tiempo promedio utilizado en la realización de un préstamo, reduciéndolo en un 17.33%.

Se logró determinar el tiempo promedio utilizado en la realización del pago de letras de un reduciéndolo reduciendo en un 21.43%.

Se logró determinar la valoración promedio de la gestión de préstamos, en un incremento del 85.94% con el sistema web propuesto.

ABSTRACT

The purpose of this research work is to improve the management of loans that occurs between two or more entities within a financial company. To do this, we propose the development of a web system that replaces the desktop system that currently has, proving to be more efficient, because we satisfactorily meet the specific objectives as follows:

It was possible to determine the average time used in the issuance of payment statements, reducing it by 25.58%.

It was possible to determine the average time used in the issuance of ACH Payment, reducing it by 46.15%.

It was possible to determine the average time used in the realization of a loan, reducing it by 17.33%.

It was possible to determine the average time used in the realization of the payment of letters of a reducing it reducing by 21.43%.

It was possible to determine the average valuation of the loan management, in an increase of 85.94% with the proposed web system.

CAPÍTULO 1. INTRODUCCIÓN

1.1. Realidad problemática

Hoy en día los préstamos informales entre personas naturales es algo que se ha venido aplicando en muchos países como solución a la falta de dinero de personas de bajos recursos o pequeñas empresas. Según (Maldonado, 2015) las personas que se dedican a esta clase de negocio lo hacen bajo su propio riesgo porque el deudor, en algún momento, puede decidir no pagar, pues no existe una ley que obligue a hacerlo, debido a que el préstamo se está haciendo de forma informal.

(Hernández & Oviedo, 2016) afirman que los créditos informales no requieren documentos o garantías como codeudores o activos de respaldo, porque la relación se basa en la confianza. Sin embargo, el retraso o mora de las cuotas y del dinero prestado si genera consecuencias graves para los deudores. Los llamados préstamos “gota a gota” son los que te dan el dinero rápido y te dan la opción de devolverlo en cuotas muy pequeñas, pero con intereses muy altos. Esta forma de préstamos ha conllevado a que se formen grupos de extorsionadores en diferentes países de Latinoamérica como Colombia, Perú, Bolivia, Argentina, entre otros que ponen en riesgo la vida de muchas personas, según informó la (BBC, 2016). (Cordero , 2017) afirma que, en Nueva York, si se presta dinero con un alto interés –más del 16% mensual– se cae en el terreno de la ilegalidad lo cual trae como consecuencia que muchos prestamistas operen en la clandestinidad. La otra razón es que no hacen negocio con cualquiera, solo suelen prestar a conocidos o porque son recomendados por gente de confianza.

En California - Estados Unidos existe una empresa financiera que brinda una solución parcial al problema de los préstamos informales al brindar un servicio que se encarga de gestionar los préstamos entre uno o más entes (personas naturales y/o instituciones) que cumplen los roles de prestamista y prestatario. La empresa financiera cuenta con un sistema de escritorio para la gestión de los préstamos registrados, pero este negocio al ser aceptado por muchas personas y empezar a crecer exponencialmente empezó a mostrar muchas deficiencias en los procesos del servicio de préstamo. Tal motivo, ha generado un malestar tanto en los clientes, por no recibir un buen servicio; en los empleados que usan el sistema actual, porque no les permite laborar eficientemente en su totalidad con una carga adicional de trabajo; y en la empresa misma, al verse perjudicada económicamente, no cumplir con su visión y misión, asimismo, perder la buena imagen que se requiere para poder ser una empresa competitiva y estar al nivel que el mercado exige.

La empresa cuenta con dos departamentos: El Standard y Specialty. El primero es donde se lleva un control de los préstamos diarios y el segundo se encarga de hacer seguimiento a los prestamistas morosos. En una entrevista realizada al jefe del área de sistemas, él dio a conocer que el sistema actual almacena la información en dos bases de datos diferentes lo

cual genera un problema al querer combinar o fusionar la información. (Jefe área de sistemas, entrevista vía email, 12 de enero del 2017, ANEXO 1)

Según la entrevista realizada se detectaron los siguientes problemas de alta importancia mencionados a continuación. El gobierno de Estados Unidos exige que las declaraciones de pago se tienen que enviar 21 días antes de la fecha de vencimiento del préstamo para poder informarle estados de cuenta sobre lo que debe y así evitar un cargo por un pago atrasado. La empresa financiera al contar una amplia clientela, el sistema actual ha llegado a colapsar, pues no puede procesar decenas de miles de declaraciones de pago que suele hacer la empresa dos veces al mes (Quincena y fin de mes) y muchas veces los propios empleados se ven perjudicados, pues el sistema se ralentiza o congela con la ejecución de este proceso. También debemos considerar que la empresa consume muchos recursos, como hojas, impresoras, tinta, personal de entrega de correo, etc., al tener que imprimir, doblar, almacenar y enviar la correspondencia de las declaraciones de pago a cada prestatario o prestador. Del mismo modo, ha sido un problema la poca adaptabilidad a los cambios de legislación, pues cuando una ley se modifica la empresa tiene que esperar que la nueva versión de su sistema sea ofertada para adecuarse a esas nuevas leyes quedando inhabilitado el uso de esos procesos de manera parcial, pues tienen que generar las declaraciones de pago y modificarlas manualmente, trayendo consigo también sanciones económicas por incumplimiento de leyes, pérdida de clientes y mala imagen a la empresa. (Jefe área de sistemas, entrevista vía email, 12 de enero del 2017)

Otro problema que se ha originado a medida que los clientes fueron aumentando fue el proceso de generación de archivos (ACH Payment) emitidos por el sistema ACH que contiene cada una de las cuentas de los prestatarios con su respectivo pago, pues el sistema de escritorio tiene un límite de cuentas adjuntas al ACH Payment, lo cual conlleva a que la empresa tenga que hacer la emisión del ACH Payment en cortos periodos de tiempo (cada 5 o 6 días), que del mismo modo ralentiza el sistema. En algunos casos, el banco ha rechazado el ACH Payment emitido, generando un gran problema a la empresa, pues los prestatarios se acercan a la organización presentando quejas con respecto a la falta de su pago en la fecha acordada. Como solución tienen que emitir un nuevo ACH Payment y realizar el pago a esa persona o grupos de personas rápidamente, reportar el problema y volver a intentar la emisión del ACH Payment con las personas restantes, lo cual conlleva a que tengan que estar siempre muy pendientes de este proceso y reportar en seguida cada falla que se presente. (Jefe área de sistemas, entrevista vía email, 12 de enero del 2017)

Ante estos problemas, para acelerar el proceso de solución de errores la empresa realizó la compra ineficiente de recursos como herramientas de software de apoyo generando la elevación de costos y aumentando la complejidad en las tareas diarias. (Jefe área de sistemas, entrevista vía email, 12 de enero del 2017)

1.2. Formulación del problema

¿De qué manera influye la implementación de un sistema web en la mejora de la gestión de préstamos en una empresa financiera?

1.3. Justificación

Para fundamentar la necesidad de realizar este trabajo de investigación, orientado al desarrollo de una aplicación web para la gestión de préstamos para una empresa financiera consideramos muchos elementos. Uno de los principales motivos es porque la empresa financiera les está ofreciendo a sus clientes la asistencia adecuada. Pues, el sistema de escritorio que es una de sus herramientas principales de trabajo para gestionar los préstamos registrados presenta muchos problemas expuestos en la realidad problemática.

Además, la empresa financiera se ha visto perjudicada económicamente, con la pérdida de clientes y creación de una mala imagen por las deficiencias del servicio que brinda.

Como justificación ambiental, la aplicación web reducirá el uso de millares de hojas que usa la empresa financiera mensualmente para la impresión de declaraciones de pago.

Finalmente, como justificación académica, cabe destacar que, para el desarrollo de esta aplicación web ofrecida como solución nos lleva a la investigación de nuevas herramientas tecnológicas que no han tenido un amplio uso en el campo de la enseñanza.

1.4. Limitaciones

La empresa financiera para la que se desarrolló el sistema web se encuentra en California – Estados Unidos, por lo cual no se pudo tener una comunicación personal con los usuarios finales. Esta limitación fue solucionada mediante la comunicación vía online, como el uso de videollamadas, mensajería instantánea y e-mails.

1.5. Objetivos

1.5.1. Objetivo general

Determinar la mejora de la gestión de préstamos en una empresa financiera a través de la implementación de un sistema web.

1.5.2. Objetivos específicos

- Determinar y evaluar el tiempo promedio utilizado en la emisión de declaraciones de pago.
- Determinar y evaluar el tiempo promedio utilizado en la emisión del ACH Payment.
- Determinar y evaluar el tiempo promedio utilizado en la realización del préstamo.
- Determinar y evaluar el tiempo promedio utilizado en la realización del pago de letras del préstamo.
- Determinar y evaluar la valoración promedio de la gestión de préstamos.

CAPÍTULO 2. MARCO TEÓRICO

2.1. Antecedentes

(Calapiña, 2016) en su tesis de investigación titulada “Sistema Web para la gestión de procesos de crédito y recuperación de cartera en la cooperativa de ahorro y crédito Santa Lucía Ltda. De la ciudad de Ambato”: Caso de estudio realizado en la Universidad Técnica de Ambato - Ecuador informó que La cooperativa Santa Lucía no cuenta con una aplicación que permita la automatización de los procesos de créditos de forma online, brindar información y seguridad a sus clientes, ni la comunicación correcta entre el personal de las diferentes sedes de la institución. Actualmente la información de los socios generada por el sistema de escritorio se guarda en FoxPro en un servidor de la agencia Quito, pero también existe información que se almacena en archivos de Word y Excel generada por el personal, el cual varía de agencia cada cierto tiempo, por lo que cuando el socio pide información sobre sus estados de cuenta el personal tiene dificultades en la toma de decisiones. De tal manera, debido al poco desarrollo tecnológico con el que cuentan no han podido posicionarse en grandes lugares y ha permitido dar ventaja a la competencia. El autor utilizó técnicas de investigación (Entrevista y observación) para encontrar los procesos críticos de la cooperativa para luego realizar un análisis de la información obtenida utilizando diagramas de flujo. Posteriormente, estudió y seleccionó las tecnologías y herramientas para el desarrollo del sistema, haciendo comparaciones entre ellas. Seleccionó el lenguaje de programación ASP.NET, el gestor de base de datos MySQL y realizó un programa replicación de los datos de FoxPro a MySQL. Utilizó la metodología de programación extrema. Elaboró y analizó las historias de usuarios y luego las asignó a una iteración con la estimación de tiempo de cada una de ellas. Diseñó el sistema y base de datos. Creó los diagramas y especificaciones de casos de uso. Realizó pruebas. Y finalmente colocó el sistema en producción y se capacitó al personal administrativo de la cooperativa. Se logró el uso del sistema web en diferentes navegadores permitiendo el correcto manejo de los procesos e información a los usuarios. Se logró tener un mejor control de la información en la base de datos sin afectar el origen de los datos mediante la construcción del programa replicador. La información se encuentra segura y mediante los roles de usuario ya no se tiene conflicto de operaciones. El desarrollo de esta tesis nos muestra bases teóricas para el análisis de las herramientas y técnicas que podemos utilizar para el desarrollo del sistema web de gestión de préstamos entre personas naturales.

(Pillajo & Tipán, 2015) en su tesis de investigación titulada “Desarrollo del sistema de planificación de créditos para la cooperativa de ahorro y crédito Santa Ana de Nayón”: Caso de estudio realizado en la Escuela Politécnica Nacional en Quito - Ecuador nos informan que la cooperativa de Ahorro y crédito Santa Ana no cuenta con una aplicación dedicada a la

planificación de créditos, lo cual conlleva a una descoordinación total en el proceso debido a que dichas actividades se realizan manualmente en Excel para luego ser impresas en un documento físico que se adjunta a la carpeta de crédito del socio. Finalmente, el oficial de crédito debe ingresar al programa “COPE” para realizar el ingreso de la información, pero esta no es siempre almacenada por algunos de los oficiales de crédito debido a que, al finalizar las actividades del proceso de planificación, dicha información no es guardada o simplemente es eliminada. Los autores utilizaron técnicas de investigación como encuestas y entrevistas para evaluar la situación actual de la cooperativa. Del mismo modo desarrollaron un diagrama de flujo del análisis de créditos para un mejor entendimiento de la normativa y el proceso. Realizaron una comparación entre metodologías y herramientas de desarrollo optando por la metodología XP, la base de datos PostgreSQL y el lenguaje de programación JAVA. Posteriormente, definieron los requerimientos, las historias de usuario y el plan de entrega. En el diseño definieron tarjetas CRC, el diseño arquitectónico y de interfaces y la estructura jerárquica del sistema. En la implementación describieron los estándares de codificación que se utilizarán en el desarrollo y la implementación de cada historia de usuario, realizaron pruebas unitarias del código para posteriormente llevarlo a producción sometiendo el producto a pruebas con los usuarios finales. Lograron un nivel de aceptación del 100% de cada historia de usuario, usabilidad y del control de acceso y seguridad por parte de los usuarios finales al finalizar todas las iteraciones. Se logró un sistema escalable y que logrará optimizar el tiempo y el uso de recursos, puesto que el sistema reduce el uso de personal en la configuración y adaptación de nuevos reglamentos institucionales y leyes de la Superintendencia de Economía Popular y Solidaria.

(Cuervo, 2008) en su tesis de investigación titulada “Análisis y desarrollo de un sistema de información para el cálculo y administración de créditos, en el sector financiero”: Caso de estudio realizado en la Fundación universitaria Konrad Lorenz en Bogotá dio a conocer que las exigencias del mercado y los nuevos modelos económicos que nacen a diario obligan a las empresas financieras a utilizar software que les permita un mejor manejo de los intereses a cobrar y pagos que se realizan de la deuda. Sin embargo, por la investigación realizada por el autor, se evidencia que en Colombia que son muy pocos los softwares que ofrecen estas facilidades de forma integral y a un precio aceptable, lo cual implica que busquen diferentes herramientas para cumplir sus necesidades en Internet como distintas clases de calculadoras de intereses y plantillas en Excel. No obstante, estos sistemas no contribuyen a la elaboración, manejo de datos y clasificación de resultados, necesarios y vitales para administración eficiente de la información. Frente a esta situación, el autor desarrolló un sistema que ofrece múltiples soluciones para las entidades financieras, que van desde administrar eficientemente la lista de clientes, hasta calcular las tasas de interés a las que están sometidos los créditos. Para su desarrollo aplicó la metodología BPM con el apoyo del

Lenguaje Unificado de Modelado (UML). Además, la plataforma de desarrollo fue Visual Studio.NET, como gestor de base datos utilizo Microsoft SQL Server y como patrón de arquitectura usó Modelo-Vista-Controlador (MVC). Finalmente, el autor logró implementar un sistema que puede suplir las necesidades principales del proceso de cálculo y amortización de créditos a nivel financiero utilizando un modelo metodológico de Ingeniería de Software que hace más óptimo los procesos organizacionales en las empresas financieras.

(Vásquez, 2013) en su tesis de investigación titulada “Análisis, diseño e implementación de un sistema de recaudación de deudas”: Caso de estudio realizado en la Pontificia Universidad Católica Del Perú informó que A raíz de los diferentes tipos de clientes y variedad de términos y condiciones, las empresas financieras están obligadas a realizar cambios constantes, los cuales están asociados con el rendimiento, actualización y mejora de atención para poder ofrecer un servicio de calidad a los prestamistas y de esta manera mantener fidelizados a los clientes. Con la necesidad de cubrir estos cambios y el consumo de recursos que implica, las empresas se ven afectadas en cuanto a su crecimiento como organización, de tal manera que las empresas financieras que no están preparadas para escalar y poder realizar cambios se encuentran en desventaja por las empresas competidoras que son capaces cambiar su flujo de trabajo y aumentar su nivel de productividad. El autor empezó por definir cuáles son los cambios más frecuentes que afectan significativamente al flujo a las empresas, principalmente encontró que era la pérdida de información y desactualización de cartera de clientes. Todo esto conlleva a que los créditos no se lleven a cabo de manera regular y muchas a veces no recuperar el total del crédito. Posteriormente eligió una metodología para el desarrollo de software, la metodología aplicada fue Rational Unified Process (RUP) debido a su naturaleza iterativa, también se apoyó de Lenguaje Unificado de Modelado (UML) para el modelamiento de los principales procesos. Del mismo modo, consideró que la aplicación deberá compilar sobre navegador Mozilla Firefox 21.0 y navegador Google Chrome 26.0, la plataforma de desarrollo fue .NET 4.5, como gestor de base datos utilizo Microsoft SQL Server 2008 y finalmente para la carga masiva de archivos será en el formato .xlsx. Finalmente se logró implementar un sistema de información que mejoró el proceso de cobranza y que es capaz de desplegar en más de un navegador web y facilitar el acceso a los usuarios

2.2. Bases teóricas

2.2.1. Gestión de procesos

(Bravo, 2013) lo define así: “La gestión de procesos es una disciplina de gestión que ayuda a la dirección de la empresa a identificar, representar, diseñar, formalizar, controlar, mejorar y hacer más productivos los procesos de la organización para lograr la confianza del cliente.”

Uno de sus objetivos principales es aumentar la productividad de las organizaciones lo cual incluye mejorar la eficiencia y eficacia. Eficiencia para optimizar el uso de recursos y eficacia para lograr objetivos hacia el exterior de la organización: cumplir las necesidades de los clientes para agregarles valor.

2.2.2. Servicio de préstamo

(Investopedia, Loan Servicing, 2015) lo define así: “El servicio de préstamo se refiere a los aspectos administrativos de un préstamo desde el momento en que los ingresos se dispersan hasta que se cancela el préstamo. El servicio de préstamos incluye el envío de estados de cuenta mensuales y la recolección de pagos mensuales, el mantenimiento de registros de pagos y saldos, la recaudación y el pago de impuestos y seguros (y el manejo de fondos en custodia y confiscación), el envío de fondos al titular del pagaré y el seguimiento de morosos”. El servicio de préstamos a lo largo del tiempo se ha vuelto un negocio rentable donde las ganancias se generan de un porcentaje relativamente pequeño de cada pago periódico del préstamo y otras fuentes auxiliares de ingresos, tales como cargos por flotación y cargos por mora. (Investopedia, Loan Servicing, 2015) afirma que esto es usualmente de 0.25% a 0.5% del pago de interés periódico.

2.2.3. Automated Clearing House (ACH)

(Investopedia, Automated Clearing House - ACH, 2015) lo define así: “ACH es una red electrónica de instituciones financieras que facilita las transacciones financieras en los Estados Unidos. Esta red ACH actúa como un centro financiero y ayuda a personas y organizaciones a mover dinero de una cuenta bancaria a otra. Las transacciones ACH consisten en depósitos directos y pagos directos, que incluyen transacciones B2B, transacciones gubernamentales y transacciones con consumidores.”

ACH es un sistema electrónico de transferencia de fondos administrado por la Asociación Nacional de Cámaras de Compensación (National Automated Clearing House Association, NACHA). Se encarga de agrupar las transacciones financieras y procesarlas en intervalos específicos a lo largo del día. Esto hace que las transacciones en línea sean extremadamente rápidas y fáciles. El sistema ACH supervisa más del 90% del valor total de todas las transacciones de pago electrónico en los Estados Unidos. (Investopedia, Automated Clearing House - ACH, 2015)

Este sistema de pagos es utilizado por las entidades financieras que desde sus sistemas generan un archivo, llamado ACH Payment, es enviado a los bancos y que contienen la descripción de la transacción a realizar. Estos deben seguir las reglas y regulaciones dictadas por la asociación NACHA.

2.2.4. Declaraciones de pago

De acuerdo a (Investopedia, Billing Statement, 2016) “los estados de cuenta son una comunicación esencial que proporciona al prestatario el pago mensual que debe pagar para mantener su cuenta al día”. Las declaraciones de pago generalmente se dividen en varias secciones donde se le proporciona al prestatario el pago mínimo adeudado y la fecha de vencimiento para evitar un cargo por pago atrasado. Además, también informa de sus actividades transaccionales, historial de pagos, alertas y avisos de morosidad, información de la cuenta, y más.

La ley exige que los estados de cuenta de los préstamos y/o tarjetas de crédito se envíen al menos 21 días antes de la fecha de vencimiento para que tenga tiempo de cancelar su deuda a tiempo y evitar cargos financieros.

2.2.5. Metodologías de desarrollo de Software

Existen dos tipos principales de metodologías, las Ligeras y las Pesadas. Las primeras son metodologías prácticas que representan una solución a los problemas que requieren una respuesta rápida en un ambiente flexible, obvian gran parte de la documentación, y están más preparadas para utilizarse en proyectos cuyos requisitos cambiarán constantemente durante todo el proceso (Wingu, 2016). Las segundas, son metodologías donde todo está mucho más controlado y se genera muchísima documentación antes de proceder a implementar el proyecto, con mucho mayor peso del análisis y el diseño sobre el proyecto. Estas también siguen un marco de disciplina estricto, un riguroso proceso estandarizado de aplicación y donde la participación del grupo de desarrollo es grande.

Para el desarrollo de esta tesis, nosotros hemos elegido orientar la elección entre las metodologías ágiles, teniendo en cuenta el corto período de tiempo en el que necesitamos que el sistema ya se encuentre en producción y que nos encontramos en un ambiente flexible de comunicación constante para el desarrollo de este sistema.

Entonces, dentro de las metodologías de desarrollo de software más utilizadas encontramos a Extreme Programming y Scrum. A continuación, una breve explicación de cada una de ellas y por qué se ha seleccionado para el desarrollo de esta tesis la metodología Extreme Programming (XP).

eXtreme Programming (XP)

Como bien señala (Beck, 2000) , es una metodología ágil, eficiente, de bajo riesgo, flexible, predecible y una manera más divertida de desarrollar software. Es diseñada para trabajar con proyectos realizados por un equipo de software compuesto de 2 hasta 10 programadores.

(Joskowicz, 2008) Nos indica que sus fases son las siguientes:

1. **Fase de exploración:** Es la fase en la que se define el alcance general del proyecto. En esta fase, el cliente define lo que necesita mediante la redacción de sencillas “historias de usuarios” y que podrían variar cuando se analicen más en detalle en la fase de iteraciones. El resultado de esta fase es una visión general del sistema.
2. **Fase de planificación:** La planificación es una fase corta, en la que el cliente y el grupo de desarrolladores acuerdan el orden en que deberán implementarse las historias de usuario, y las entregas. El resultado de esta fase es un Plan de Entregas.
3. **Fase de iteraciones:** Esta es la fase principal en el ciclo de desarrollo de XP. Las funcionalidades son desarrolladas en esta fase, generando al final de cada una un entregable funcional que implementa las historias de usuario asignadas a la iteración. Como las historias de usuario no tienen suficiente detalle como para permitir su análisis y desarrollo, al principio de cada iteración se realizan las tareas necesarias de análisis, recabando con el cliente todos los datos que sean necesarios.

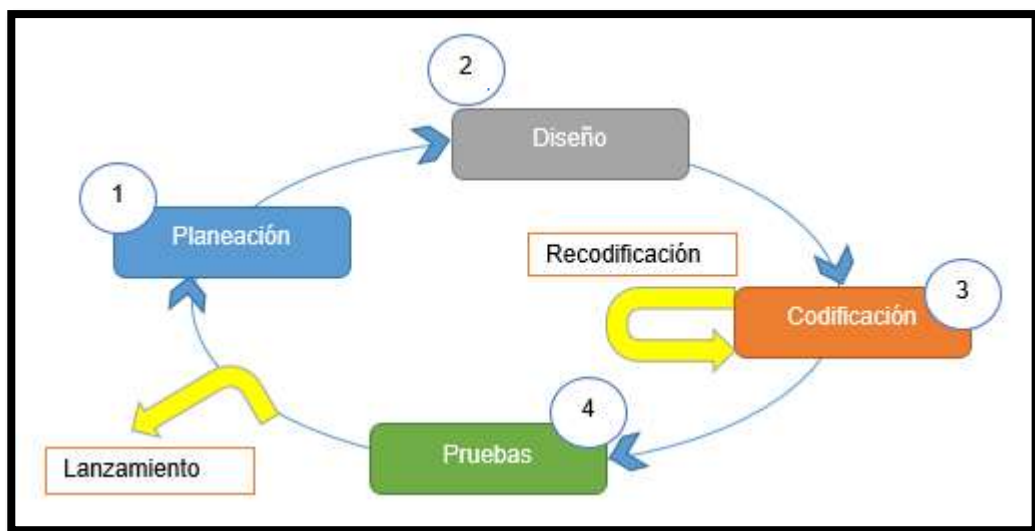


Figura 1: Iteraciones eXtreme Programming

4. **Fase de puesta en producción:** Si bien al final de cada iteración se entregan módulos funcionales y sin errores, puede ser deseable por parte del cliente no poner el sistema en producción hasta tanto no se tenga la funcionalidad completa. En esta fase no se realizan más desarrollos funcionales, pero pueden ser necesarias tareas de ajuste (“fine tuning”).

Lo que ha sido determinante para su utilización en esta tesis es lo siguiente:

- Realiza ciclos de desarrollo cortos (llamados iteraciones), con entregables funcionales al finalizar cada ciclo, lo cual necesita la empresa financiera actualmente. En cada iteración se realiza un ciclo completo de análisis, diseño, desarrollo y pruebas.
- Tiene una concreta, temprano y continua retroalimentación, ya que permite verificar con el cliente, en nuestro caso con el jefe del área de sistemas de la empresa financiera, si lo que se va realizando se ajusta a sus requerimientos.
- Promueve publicar lo antes posible las nuevas versiones, siempre que estén libres de errores. Usualmente las publicaciones se realizan cada una a tres semanas.
- Dependencia de pruebas automatizadas escritas por programadores y clientes para monitorear el progreso del desarrollo, lo cual permitirá identificar defectos en una etapa temprana y su solución inmediata.
- Comunicación oral, pruebas y código fuente para comunicar la estructura del sistema con su propósito.
- Las tareas que se van terminando en las diferentes entregas al cliente son susceptibles a modificaciones durante el transcurso de todo el proyecto, incluso después de que funcione correctamente.

La metodología propuesta en XP está diseñada para entregar el software que los clientes necesitan en el momento en que lo necesitan y alienta a los desarrolladores a responder a los requerimientos cambiantes de los clientes, aún en fases tardías del ciclo de vida del desarrollo. (Joskowicz, 2008)

SCRUM

(Dimes, 2015) nos dice que Scrum es “un marco de referencia para crear software complejo y entregarlo a tiempo de una forma mucho más sencilla”. Este marco de referencia ágil se basa en los principios de adaptación, inspección continua, autogestión e innovación.

Scrum adopta plenamente los principios de los métodos ágiles de desarrollo y los incorpora a la gestión de proyectos. Primero y ante todo, abarca la filosofía de que todos los requisitos están inicialmente sin perfeccionar y son pocos claros. Por tal, Scrum se centra en la mejora de la capacidad del equipo de desarrollo para observar y adaptarse a las nuevas exigencias.

Mientras que algunas metodologías ágiles cubren la totalidad del ciclo de vida del software de desarrollo, Scrum, en particular, se centra en la gestión de proyectos en el desarrollo de software. (The blokehead, 2016)

Scrum propone una división de roles al interior de cada equipo de trabajo. Tenemos al equipo, Product owner, Stakeholders y Scrum master. El equipo no debe ser de más de 7 personas, pero, tampoco de menos de 3 y es el encargado de pasar de la idea al producto. El Product owner es quien dirige al equipo, centraliza las nuevas tareas que llegan, tiene la visión del producto, define funciones y establece prioridades. Los Stakeholders son quienes presentan los proyectos sobre los cuales se va a trabajar. Finalmente, el Scrum master sirve como guía, soporte y es un experto en equipos que elimina las trabas que se les presentan y les da apoyo. (Wingu, 2016)

Scrum nos propone dividir el tiempo de trabajo en períodos cortos (iteraciones o Sprints). Un aspecto importante a destacar es que una vez comenzado el Sprint no se puede ingresar más trabajo. Es decir que, todas las tareas nuevas deberán esperar hasta que el ciclo llegue a su fin. Esta regla tiene la ventaja de que permite que ninguna nueva tarea nos distraiga pues una vez que el ciclo empieza sólo hay que concentrarse en lo que se planificó para el mismo. Pero la desventaja sería que si tomamos la prescripción de Scrum al pie de la letra no nos permite ser flexibles pues una vez comenzado el ciclo no podríamos ocuparnos de eventualidades urgentes. (Wingu, 2016)

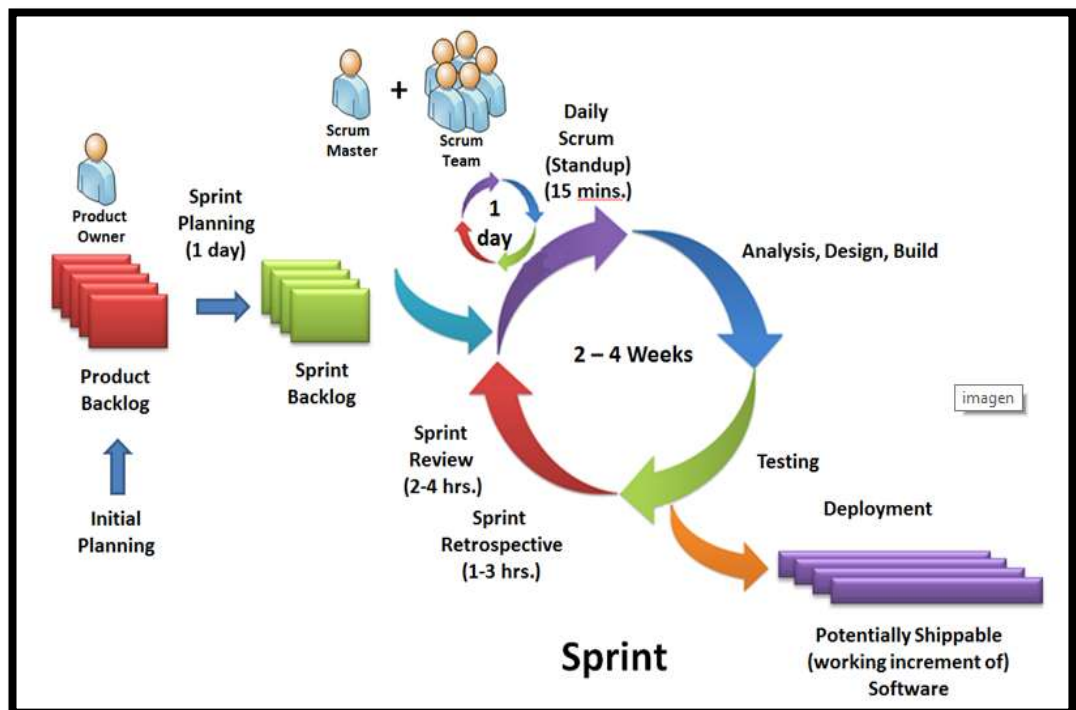


Figura 2: El ciclo de Scrum

Fuente: Página web de Microsoft, "Branching for Scrum", (Heys, 2011)

A continuación, se presenta una tabla comparativa entre las mencionadas metodologías de desarrollo, de entre las cuales se seleccionará una para el desarrollo del presente proyecto de investigación.

Tabla 1: Comparación de metodologías: XP - Scrum

EXtreme Programming	Scrum
Usado en el desarrollo de software con un grupo pequeño de desarrolladores. (2 - 10)	Usado en desarrollo de software con pocos programadores (Mínimo 3)
Permite cambios en todo momento.	Permite cambios hasta antes del desarrollo del Sprint.
Testeo a lo largo del proyecto	Testeo al final de cada Sprint
Iteraciones cada 1 - 3 semanas	Iteraciones cada 2 - 4 semanas

2.2.6. Tecnología aplicada

2.2.6.1. Visual Studio

(Microsoft, Visual Studio 2015, 2015) define a Visual Studio (IDE) como “conjunto de herramientas y otras tecnologías de desarrollo de software basado en componentes para crear aplicaciones eficaces y de alto rendimiento, permitiendo a los desarrolladores crear sitios y aplicaciones web, así como otros servicios web en cualquier entorno que soporte la plataforma”.

En palabras más específicas, Visual Studio es un conjunto completo de herramientas de desarrollo para la generación de aplicaciones web ASP.NET, Servicios Web XML, aplicaciones de escritorio y aplicaciones móviles. Visual Basic, Visual C# y Visual C++ utilizan el mismo entorno de desarrollo integrado (IDE), que habilita el uso compartido de herramientas y facilita la creación de soluciones en varios lenguajes. Asimismo, dichos lenguajes utilizan las funciones de .NET Framework, las cuales ofrecen acceso a tecnologías clave para simplificar el desarrollo de aplicaciones web ASP y Servicios Web XML.

Adicional se le ha dado un enfoque renovado para mejorar la eficiencia de las tareas fundamentales que los desarrolladores desempeñan diariamente. Desde una nueva instalación más ligera y modular adaptada a las necesidades del desarrollador, un IDE más rápido desde el arranque al apagado, una nueva manera de ver, editar y depurar cualquier código sin proyectos ni soluciones.

Cuadro comparativo para la elección de la herramienta

Tabla 2: Cuadro comparativo de Visual Studio y Eclipse

Recursos/Característica	Visual Studio	Eclipse
¿La empresa cuenta con licencias?	Si	Software libre
¿El equipo de desarrollo tiene conocimiento sobre la herramienta?	Si	Si
Intellisense, ui	Si, en todos los editores.	No completamente en todos los editores.
Integración con gestores de base datos.	Cuenta con drivers propios para conexión.	Mediante conectores extras, previa instalación
Documentación	Si	Si
Support	Personalizado, exacto y a tiempo.	Demanda de tiempo ante una consulta.

- La elección de la herramienta de visual studio se dio principalmente por las licencias adquiridas por parte la empresa financiera.

2.2.6.2. ASP.Net MVC

(Microsoft, Visión general de ASP.NET MVC, 2010). “Es una forma poderosa y basada en patrones para crear sitios web dinámicos que permite una separación clara de las preocupaciones y que le da un control total sobre el marcado para un desarrollo agradable y ágil. ASP.NET MVC incluye muchas características que permiten un desarrollo rápido y compatible con TDD para crear aplicaciones sofisticadas que utilizan los últimos estándares web”.

Las páginas web ASP.NET MVC y la sintaxis Razor proporcionan una forma rápida, accesible y ligera de combinar código de servidor con HTML para crear contenido web dinámico.

Los tres marcos de ASP.NET MVC están basados en .NET Framework y comparten la funcionalidad principal de .NET y de ASP.NET. Por ejemplo, los tres marcos ofrecen un modelo de seguridad de inicio de sesión basado en la membresía, y los tres comparten las mismas funciones para administrar solicitudes, manejar sesiones, etc., que son parte de la funcionalidad central de ASP.NET. Los marcos no son completamente independientes, y elegir uno no impide usar otro. Como los marcos pueden coexistir en la misma aplicación web, no es raro ver componentes individuales de aplicaciones escritas usando diferentes marcos. Por

ejemplo, las porciones de una aplicación orientadas al cliente pueden desarrollarse en MVC para optimizar el marcado, mientras que el acceso a los datos y las partes administrativas se desarrollan en formularios web para aprovechar los controles de datos y el acceso simple a los datos.

Cuadro comparativo para la elección de la herramienta

Tabla 3: Cuadro comparativo de Razor y ASPX

Recursos/Característica	Razor	ASPX
Ensamblado	El espacio de nombre utilizado por Razor View Engine es System.Web.Razor	El espacio de nombres utilizado por ASPX View Engine es System.Web.Mvc.WebFormViewEngine
Motor de vista	Introdujo con MVC 3.0. Este no es un idioma nuevo, pero es un marcado.	Vista de formulario web es el motor de vista predeterminado y está disponible desde el comienzo de MVC
Sintaxis	Razor tiene una sintaxis que es muy compacta y nos ayuda a reducir el tipeo.	El motor de vista de formulario web tiene una sintaxis que es la misma que una aplicación de formularios ASP.Net.
Sintaxis para contenido servidor	Razor View Engine usa @ para mostrar el contenido del servidor.	El motor de vista de formulario ASPX / web utiliza "<% =%>" o "<%: %>" para mostrar el contenido del servidor.
Sintaxis en código de bloque	Razor no requiere que se cierre el bloque de código, Razor View Engine se analiza a sí mismo y puede decidir en el tiempo de ejecución que es un elemento de contenido y que es un elemento de código.	Un motor de vista de formulario web requiere que el bloque de código se cierre correctamente, de lo contrario genera una excepción de tiempo de ejecución.
Compatibilidad con TDD	Compatible con Test Driven Development (TDD).	No es compatible con Test Driven Development (TDD) porque depende de la clase System.Web.UI.Page para que la prueba sea compleja.

- La elección de ASP .NET MVC se dio principalmente por las características de la sintaxis razor que proporciona un nuevo motor de visualización con código optimizado para plantillas centradas. La sintaxis de Razor es muy compacta y mejora la legibilidad del código y el marcado. Por defecto, MVC admite ASPX (formularios web) y Razor View Engine.

2.2.6.3. SQL Server

(Microsoft, What's new in SQL Server 2016, 2017) "SQL Server es un sistema de gestión de bases de datos relacionales (RDBMS) de Microsoft que está diseñado para el entorno empresarial. SQL Server se ejecuta en T-SQL (Transact -SQL), un conjunto de

extensiones de programación de Sybase y Microsoft que añaden varias características a SQL estándar, incluyendo control de transacciones, excepción y manejo de errores, procesamiento fila, así como variables declaradas”.

SQL Server, construye aplicaciones inteligentes de misión crítica utilizando una plataforma de base de datos híbrida escalable que tiene todo integrado, desde el rendimiento en memoria y la seguridad avanzada hasta el análisis en la base de datos. SQL Server agrega nuevas características de seguridad, capacidades de consulta, Hadoop e integración en la nube, análisis de R y más.

Sus principales servicios son:

- **Database Engine**

El Motor de base de datos es el servicio principal para almacenar, procesar y proteger datos. Database Engine proporciona acceso controlado y procesamiento de transacciones rápido para cumplir con los requisitos de las aplicaciones de consumo de datos más exigentes dentro de su empresa. El motor de base de datos también proporciona una gran compatibilidad para mantener la alta disponibilidad.

- **Machine Learning Services**

Microsoft Machine Learning Services admite la integración del aprendizaje automático, utilizando los populares lenguajes R y Python, en flujos de trabajo empresariales. Machine Learning Services (en la base de datos) integra R y Python con SQL Server, lo que facilita la creación, el reciclaje y la puntuación de los modelos llamando a los procedimientos almacenados. Microsoft Machine Learning Server proporciona soporte a escala empresarial para R y Python, sin requerir SQL Server.

- **Integration Services**

Integration Services es una plataforma para crear soluciones de integración de datos de alto rendimiento, incluidos paquetes que proporcionan procesamiento de extracción, transformación y carga (ETL) para el almacenamiento de datos.

- **Analysis Services**

Microsoft Machine Learning Services admite la integración del aprendizaje automático, utilizando los populares lenguajes R y Python, en flujos de trabajo empresariales. Machine Learning Services (en la base de datos) integra R y Python con SQL Server, lo que facilita la creación, el reciclaje y la puntuación de los modelos llamando a los procedimientos almacenados. Microsoft Machine Learning Server proporciona soporte a escala empresarial para R y Python, sin requerir SQL Server.

- **Reporting Services**

Reporting Services ofrece funcionalidad de informes empresarial y habilitada para la web. Puede crear informes que extraigan contenido de una variedad de orígenes de datos, publiquen informes en varios formatos y administren de forma centralizada la seguridad y las suscripciones.

- **Replication**

La replicación es un conjunto de tecnologías para copiar y distribuir datos y objetos de bases de datos de una base de datos a otra, y luego sincronizar entre bases de datos para mantener la coherencia. Al usar la replicación, puede distribuir datos a diferentes ubicaciones y a usuarios remotos o móviles a través de redes de área local y amplia, conexiones de acceso telefónico, conexiones inalámbricas e Internet.

- **Data Quality Services**

Los servicios de calidad de datos de SQL Server (DQS) le proporcionan una solución de limpieza de datos basada en el conocimiento. DQS le permite crear una base de conocimiento y luego usar esa base de conocimientos para realizar la corrección de datos y la deduplicación de sus datos, utilizando medios interactivos y asistidos por computadora. Puede usar servicios de datos de referencia basados en la nube y puede construir una solución de administración de datos que integre DQS con SQL Server Integration Services y Master Data Services.

- **Master Data Services**

Master Data Services es la solución de SQL Server para la gestión de datos maestros. Una solución basada en Master Data Services ayuda a garantizar que los informes y análisis se basen en la información correcta. Al utilizar Master Data Services, usted crea un repositorio central para sus datos maestros y mantiene un registro auditable y asegurable de esos datos a medida que cambia con el tiempo.

Cuadro comparativo para la elección de la herramienta

Tabla 4: Cuadro comparativo de MySQL y SqlServer

Recursos/Característica	MYSQL	SQL Server
Costo	Libre y suscripción con soporte	La empresa cuenta con licencias.
Open Source	Si	No
Plataformas	Linux, Windows y muchas otras	Sólo Windows
Límite de tamaño de la base de datos	Limitado por el sistema operativo	Limitado por el sistema operativo
Servicio de reportes	No	Si

Vistas	Si	Si
Procedimientos almacenados	Si	Si
Triggers	Si	Si
Cursores	Si	si
Sub consultas	Si	Si
Replicación	Si	Si

- La elección de la herramienta de SQL Server se dio principalmente por las licencias adquiridas por parte la empresa financiera.

2.2.6.4. ASP.NET Core Signal R

(Microsoft, Introduction to SignalR, 2014). ASP.NET Core SignalR es una biblioteca que simplifica la adición de funcionalidad web en tiempo real a las aplicaciones. La funcionalidad web en tiempo real permite que el código del lado del servidor envíe contenido a los clientes al instante. Buenos candidatos para SignalR: Aplicaciones que requieren actualizaciones de alta frecuencia del servidor. Los ejemplos son juegos, redes sociales, votación, subastas, mapas y aplicaciones de GPS. Los ejemplos incluyen actualizaciones de ventas instantáneas o alertas de viaje. Las redes sociales, correo electrónico, chat, juegos, y muchas otras aplicaciones usan notificaciones.

WebSockets es el transporte óptimo, pero otras técnicas como eventos enviados por el servidor y Long Polling se pueden usar cuando no están disponibles. SignalR detectará e inicializará automáticamente el transporte apropiado en función de las características admitidas en el servidor y el cliente, maneja la administración de conexiones automáticamente y le permite transmitir mensajes a todos los clientes conectados simultáneamente, como una sala de chat. También puede enviar mensajes a clientes específicos. La conexión entre el cliente y el servidor es persistente, a diferencia de una conexión HTTP clásica, que se restablece para cada comunicación.

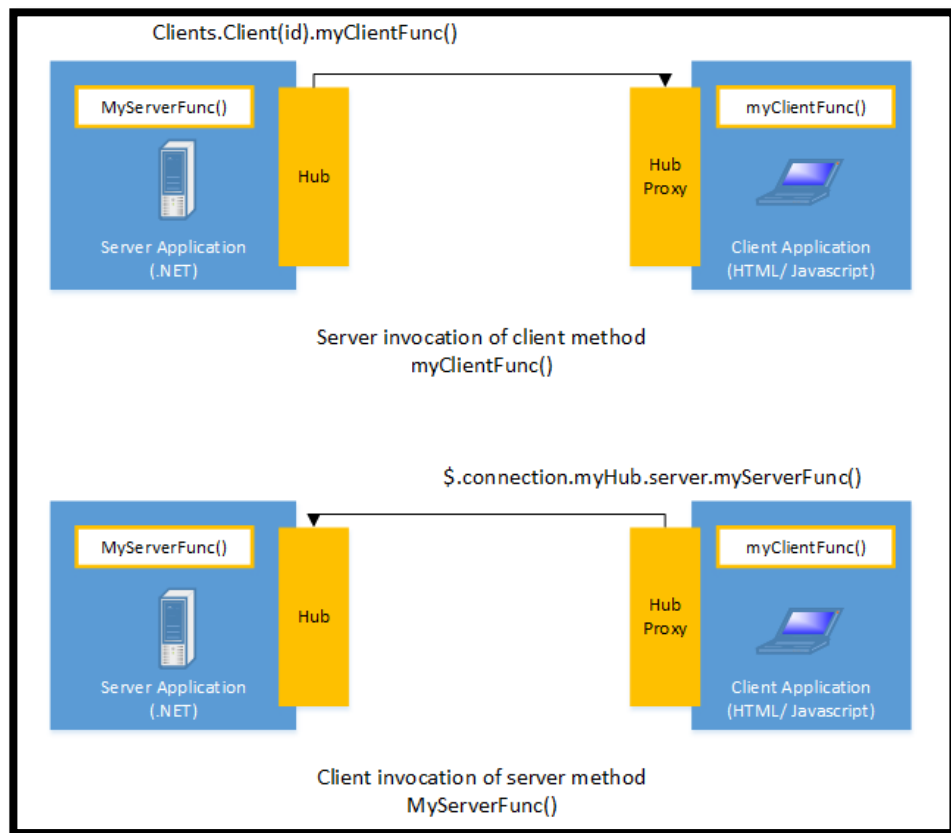


Figura 3: Arquitectura SignalR

Fuente: Página web de Microsoft, "ASP.NET Core SignalR", Microsoft. (2018)

Aplicación en el desarrollo del sistema web

El sistema web que se está implementando debido al rubro del trabajo realiza muchos procesos en masa, lo cuales deben estar sujetos a poderse cancelar durante su ejecución. Es fundamental que todos estos procesos sean controlados mediante transacciones con el fin de poder habilitar la opción de cancelar a los usuarios del sistema web y también poder realizar rollback de lado servidor. Los sistemas web por su naturaleza de comunicación son mediante Cliente-Servidor, cuando los usuarios ejecutan procesos que implican demasiados flujos de datos es necesario mantener un WebSocket abierto y en espera, con el fin de que el servidor pueda responder ante el llamado de cancelación desde el lado cliente. Para esto, se usa Signal R, ya que puede mantener WebSockets abiertos y totalmente independientes entre cliente y servidor. Por cada proceso donde se aplica Signal R se crearán hubs de comunicación con funciones predeterminadas con fin de poder controlar las transacciones desde el lado cliente y mantener la integridad de datos del lado del servidor.

2.2.6.5. Crystal Report

(EcuRed, 2018) Es una herramienta para el diseño y generación de informes a partir de datos almacenados en una base de datos u otra fuente de información. Es, con diferencia, la herramienta más popular en su categoría, se usa cuando se necesita obtener periódicamente información para la toma de decisiones a partir de los datos de la empresa, se ha convertido en la herramienta favorita a la hora de embeber capacidades de generación de informes dentro de las aplicaciones a medida.

Características

- **El más completo acceso a datos**
Incluye más de 30 drivers para acceso a bases de datos relacionales, fuentes de datos XML y cubos OLAP (Incluyendo sistemas ERP, CRM, Oracle, IBM DB2 y Microsoft SQL Server). También puede acceder a datos personalizados a través de JavaBeans y objetos COM (ADO record sets) para una conectividad más flexible.
- **Diseño integral y opciones de formato**
Posee más de 100 opciones de formato, incluyendo parámetros, mapas, tablas cruzadas, gráficos e hipervínculos, para incrementar el impacto de los reportes. También se incluyen más de 160 fórmulas, funciones y operadores para un control completo de la presentación de los datos
- El nuevo Repositorio Crystal permite almacenar elementos clave de los reportes, tales como objetos de texto, imágenes, sentencias SQL y funciones personalizadas. Gracias a este repositorio central se puede reutilizar estos objetos en múltiples reportes. Este almacén centralizado de objetos permite minimizar los esfuerzos de mantenimiento de los reportes y al mismo tiempo ser más productivo en el diseño de reportes nuevos.
- **La interacción y acceso a los datos que sus usuarios demandan**
Crystal Reports habilita la visualización e interacción con los reportes a través de una amplia variedad de dispositivos y entornos.
- Permite editar las sentencias SQL directamente para un control ilimitado sobre la conectividad con la base de datos.
- Permite almacenar objetos de reporte clave en una librería centralizada, para ser reutilizados en otros reportes.
- Servicio de procesamiento de reportes basado en Web, para realizar integración de contenido dinámico en aplicaciones Web empresariales.

- Permite crear aplicaciones utilizando cualquier plataforma de desarrollo y controlando la visualización, interacción y modificación de los reportes en tiempo de ejecución.

Ventajas

- Integrar estrechamente capacidades de diseño, modificación y visualización en aplicaciones .NET, Java o COM.
- Permitir a los usuarios finales acceder e interactuar con los reportes a través de portales Web, dispositivos móviles y documentos de Microsoft Office.

2.2.6.6. ADO Entity Framework

(Microsoft, ADO.NET Entity Framework, 2017) “Es un conjunto de tecnologías de ADO.NET que permiten el desarrollo de aplicaciones de software orientadas a datos. Las aplicaciones orientadas a datos se han enfrentado a la necesidad de lograr dos objetivos muy diferentes”. Deben modelar las entidades, las relaciones y la lógica de los problemas empresariales que resuelven, y también deben trabajar con los motores de datos que se usan para almacenar y recuperar los datos. Los datos pueden abarcar varios sistemas de almacenamiento, cada uno con sus propios protocolos; incluso las aplicaciones que funcionan con un único sistema de almacenamiento deben equilibrar los requisitos del sistema de almacenamiento con respecto a los requisitos de escribir un código de aplicación eficaz y fácil de mantener, permite trabajar con datos en forma de objetos y propiedades específicos del dominio, por ejemplo, con clientes y direcciones, sin tener que pensar en las tablas de las bases de datos subyacentes y en las columnas en las que se almacenan estos datos. Con Entity Framework, se puede trabajar en un nivel más alto de abstracción cuando tratan con datos, y puede crear y mantener aplicaciones orientadas a datos con menos código que en las aplicaciones tradicionales. Dado que Entity Framework es un componente de .NET Framework, las aplicaciones de Entity Framework se pueden ejecutar en cualquier equipo en el que esté instalado .NET Framework a partir de la versión 3.5 SP1.

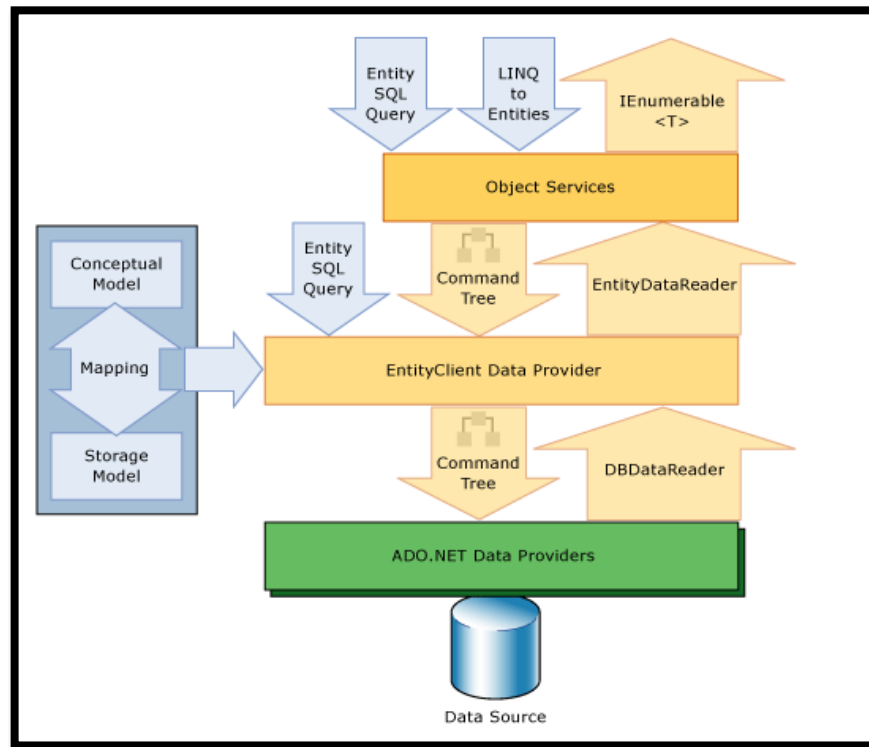


Figura 4: Arquitectura de ADO Entity Framework

Fuente: Página web de Microsoft, "ADO Entity Framework", Microsoft. (2018)

Se optó por utilizar ADO Entity Framework por la capacidad de utilizar el modelo conceptual para separar las estructuras de datos que usa el sistema del esquema en el origen de datos. De esta forma, más adelante, se pueden realizar cambios con facilidad en la base de datos sin modificar manualmente el código fuente del sistema.

Cuadro comparativo para la elección de la herramienta

Tabla 5: Cuadro comparativo de Entity Framework y Linq to Sql

Recursos/Característica	Entity Framework	Linq to Sql
Conectividad a gestores de bases de datos	Puede funcionar con varias bases de datos como Oracle, DB2, MYSQL, SQL Server, etc.	Solo funciona con la base de datos SQL Server.
Estructura	Genera un archivo .edmx inicialmente. La relación se mantiene usando 3 archivos diferentes .csdl, .msl y .ssdl	Genera un .dbml para mantener la relación
Acceso a datos	Permite consultar datos utilizando EntitySQL,ObjectContext, DbContext.	permite consultar datos usando DataContext.

Enfoque en proyectos	Ideal para usar para el desarrollo rápido de aplicaciones solo con SQL Server.	Se puede usar para el desarrollo rápido de aplicaciones con RDBMS como SQL Server, Oracle, DB2 y MySQL, etc.
----------------------	--	--

- La elección de la herramienta de ADO Entitie Framwork se dio principalmente por las características de conectividad a diferentes gestores de bases de datos.

2.2.6.7. SGML (Estandar Generalised Mark-up Language) XML

(IBM, 2016) “Es un metalenguaje de definición de documentos, estructurado mediante etiquetas o marcas. Su nombre viene de las palabras inglesas eXtensible Markup Lenguaje (Lenguaje de marcado ampliable o extensible)”. Su objetivo es lograr páginas web mucho más semánticas, separando la estructura del contenido web y ofreciéndole al desarrollador la capacidad de crear vocabularios modulares personalizados es utilizado en programación web, que optimiza los documentos utilizando las normativas de la W3C. Gracias a esta normativa, los desarrolladores pueden crear sus propias etiquetas, lo que facilita el proceso de programación orientada a web y a Internet, así como también la interpretación, transmisión y organización de datos.

XML cuenta con una sintaxis y una estructura mucho más rígida que el HTML, lo cual lo hace mucho más complejo, pero a su vez, permite crear documentos mucho más estables, que cumplen a cabalidad con la normativa internacional.

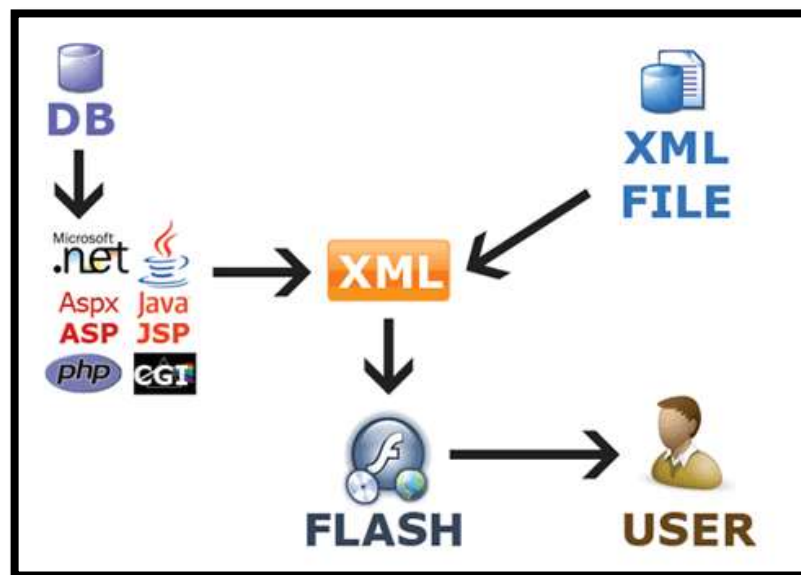


Figura 5: Esquema de funcionamiento XML

Fuente: Página web de Culturación, “XML”, Culturación. (2018)

Ventajas

- Al utilizar XML, se facilita ampliamente la transmisión de datos, ya que, al trabajar con texto plano, es más sencillo el envío de información a través de la red.
- XML simplifica los procesos de migración de datos de una base de datos a otra, siendo un formato ideal para realizar las migraciones entre BD.
- XML ahorra tiempo y trabajo a los desarrolladores, ya que no es necesario realizar varios códigos fuentes para crear el mismo efecto o aplicación, con un solo programa basta.

El uso del formato XML en la implementación del sistema web se aplicará al momento de enviar todas las declaraciones de pago al servicio tercerizado de oficina postal llamado ImailTracking.

Cuadro comparativo para la elección de la herramienta

Tabla 6: Cuadro comparativo de XML y JSON

Recursos/Característica	XML	JSON
Tipo	Lenguaje de marcado extensible , es un formato independiente de software y hardware para el intercambio de datos.	(JavaScript Object Notation) es un estándar abierto basado en texto para el intercambio de datos.
Complejidad	Es más complicado de leer	Es simple y fácil de leer.
Orientación	Orientado a documentos.	Orientado a datos.
Extensión	El archivo XML termina con la extensión .xml.	El archivo JSON termina con la extensión .json.

- La elección de la herramienta XML se dio principalmente por la orientación a la que está enfocada, que es principalmente es a documentos, también porque el servicio web para reparto de documentos solo admite XML como formato de para intercambio de datos.

CAPÍTULO 3. HIPÓTESIS

3.1. Formulación de la Hipótesis

La implementación de una aplicación web influye en la mejora significativa de la gestión de préstamos en una empresa financiera.

3.2. Operacionalización de variables

Tabla 7: Operacionalización de variable independiente

VARIABLE	DEFINICIÓN CONCEPTUAL	DIMENSIONES	INDICADORES
Aplicación web	Según Tahuiton J. (2011) las aplicaciones Web son grandes sistemas distribuidos complejos que utilizan una combinación de plataformas tecnológicas para su funcionamiento y que pueden atender a millones de usuarios de forma simultánea	Usabilidad	Atracción ¿Es atractivo el diseño del software?
			Entendimiento ¿Es fácil de entender su aplicabilidad?
		Funcionalidad	Adecuidad ¿Tiene el conjunto de funciones apropiadas para las tareas especificadas?
		Confiability	Madurez ¿Con qué frecuencia presenta fallas?

Tabla 8: Operacionalización de variable dependiente

VARIABLE	DEFINICIÓN CONCEPTUAL	DIMENSIONES	INDICADORES
Gestión de préstamos de una empresa financiera	La gestión del préstamo se realiza desde el momento que se realiza la solicitud de préstamo hasta que se cancela. Lo cual ayuda a la dirección de la empresa a identificar, representar, diseñar, formalizar, controlar, mejorar y hacer más productivos sus procesos. (Bravo, 2013)	Préstamo	Tiempo promedio de la realización de un préstamo
			Valoración promedio de la gestión de préstamos
		Pago de letras del préstamo	Tiempo promedio de la realización del pago de letras de un préstamo
		Declaraciones de pago	Tiempo promedio de la emisión de declaraciones de pago
		ACH Payment	Tiempo promedio de la emisión del ACH Payment

CAPÍTULO 4. DESARROLLO

METODOLOGÍA DE DESARROLLO: XP

1. FASE DE EXPLORACIÓN

1.1. Alcance del proyecto

1.1.1. Equipo de trabajo

Tabla 9: Equipo de trabajo

Actores	ROLES			
	Programador	Tester	Cliente	Guía
Chávez Domínguez, Miguel	X	X		
Troncozo Centurión, Graly	X	X		
Jefe área de sistemas				X
Empresa financiera			X	

1.1.2. Solución Propuesta

Este sistema web es concebido como una herramienta que apoyará a los empleados de la empresa financiera durante sus actividades diarias, con lo que respecta la gestión de los préstamos. En base a ellos se obtendrán datos para su análisis y determinar si existe una mejora en sus tareas. El desarrollo se hará con el uso del IDE Visual Studio 2016, con el lenguaje de programación C#, y el motor de base de datos SQL Server Management Studio.

1.1.3. Usuarios

Los principales interesados en el sistema web son los empleados de la empresa financiera en California – Estados Unidos. No obstante, el gerente también tendrá interacción con el sistema web.

1.2. Historial de Usuario

Tabla 10: Historia de usuario "Realización de un préstamo"

Historia de Usuario	
Número: 1	Usuario: Programador
Nombre historia: Realización de un préstamo	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Iteración asignada: 1	
Programador responsable: Graly Troncozo Centurión	
Descripción:	
Se requiere dividir el ingreso del préstamo en dos pasos:	
<ol style="list-style-type: none"> 1. Registrar al prestamista: su información personal, teniendo en cuenta que puede ser una persona natural o una institución. Además, ingresar 	

<p>información para contactarlo como direcciones, teléfonos, horario para contactarlo, etc.</p> <p>2. Registrar la información del préstamo, como la cuenta de préstamo, saldos, tasa, detalles del pago de las letras (como frecuencia, día de vencimiento, etc.), asignación de fechas, etc.</p> <p>Finalizado el registro, se necesita poder editar el préstamo para poder asignarle uno o más prestatarios, donde se definirá que porcentaje dará cada uno.</p> <p>Además, se necesita poder visualizar la información del préstamo detallada y un historial de pagos realizada por el prestamista.</p>
<p>Observaciones:</p> <ul style="list-style-type: none"> Al ser muchos los datos para ingresar, se requiere de una interfaz sencilla con criterios de usabilidad.

Tabla 11: Historia de usuario "Realización del pago de letras de un préstamo"

Historia de Usuario	
Número: 2	Usuario: Programador
Nombre historia: Realización del pago de letras de un préstamo	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Iteración asignada: 1	
Programador responsable: Miguel Chávez Domínguez	
<p>Descripción</p> <p>Se debe realizar la búsqueda del prestamista mediante su número de cuenta. A partir de ello debe poder mostrar la información del préstamo que ha solicitado y las letras que debe pagar (saldos, cargos, fechas de vencimiento, tasa, etc.) Con la opción de poder editar la distribución del pago.</p> <p>Una vez que se llenó la información en el formulario, debe haber un botón para guardar la información y realizar el pago al mismo tiempo.</p>	
<p>Observaciones:</p> <ul style="list-style-type: none"> Se debe usar la pasarela de pago PayPal para las transferencias del dinero. 	

Tabla 12: Historia de usuario "Emisión de declaraciones de pago"

Historia de Usuario	
Número: 3	Usuario: Programador
Nombre historia: Emisión de declaraciones de pago	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Iteración asignada: 2	

Programador responsable: Equipo de programación
<p>Descripción</p> <p>Se desea poder filtrar las declaraciones de pago que se enviarán por el tipo de estado de préstamo (Activo, pagado, transferido, etc.). Además, debe permitir poder definir cuantas declaraciones de pago se van a emitir y/o a quien, dentro de un rango de fechas.</p> <p>Antes de poder procesar las declaraciones de pago y enviarlas al servicio externo ImailTracking se desea poder mostrar un reporte en PDF y que permita exportarlo e imprimirlo.</p>
<p>Observaciones:</p> <ul style="list-style-type: none"> • Se generan declaraciones de pago de forma masiva 2 veces al mes (usualmente los días 15 y 30 de cada mes). • El sistema debe soportar la emisión de más 10000 declaraciones de pago.

Tabla 13: Historia de usuario "Emisión del ACH Payment"

Historia de Usuario	
Número: 4	Usuario: Programador
Nombre historia: Emisión del ACH Payment	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Iteración asignada: 3	
Programador responsable: Equipo de programación	
<p>Descripción</p> <p>Se debe mostrar una lista de todos los pagos pendientes a realizar por la empresa financiera hacia los prestatarios, indicando el nombre del prestatario, cantidad de pago, referencia, etc.</p> <p>El usuario deberá poder seleccionar uno, varios o todos los pagos pendientes, para posteriormente generar el archivo ACH Payment.</p> <p>Del mismo modo, el usuario debe poder tener las opciones de exportar el archivo ACH Payment en un bloc de notas, transmitir por fax, y mostrar el informe de los pagos que se realizarán de forma detallada, denominado "Informe de transmisión ACH".</p>	
<p>Observaciones:</p> <ul style="list-style-type: none"> • Se requiere que se genere el ACH Payment una vez al mes (usualmente al final de cada mes) • El código ACH Payment debe poder almacenar entre 5000 y 10000 cuentas aproximadamente. 	

2. FASE DE PLANIFICACIÓN

2.1. Planificación Inicial

En la planificación inicial listaremos las historias de usuario de acuerdo a su **PRIORIDAD** (Bajo, Media o Alta según la importancia y relevancia que tenga), **RIESGO** (Bajo, Medio o Alto según la probabilidad de fallo de cada historia de usuario en el desarrollo), **ESFUERZO** (calificando 1, 2 o 3 según el tiempo y trabajo que demandará en desarrollar la historia de usuario) e **ITERACIÓN** (momento de implementación de cada historia), las cuales se acordaron con el jefe del área de sistemas.

Tabla 14: Planificación Inicial de Historias de Usuario

Historio de Usuario		Prioridad	Riesgo	Esfuerzo	Iteración
Nro.	Nombre	del Negocio			
001	Realización de un préstamo	Alta	Media	2	1
002	Realización del pago de letras de un préstamo	Alta	Media	2	1
003	Emisión de declaraciones de pago	Alta	Alto	3	2
004	Emisión del ACH Payment	Alta	Alto	3	3

2.2. Estimación de tiempos

De acuerdo a las ponderaciones de la prioridad, riesgo y esfuerzo se ha estimado el tiempo de desarrollo de cada historia.

Tabla 15: Estimación de tiempo de Historias de Usuario

Historia de Usuario		Tiempo Estimado
Nro.	Nombre	
001	Control del préstamo	20 días
002	Control del pago	20 días
003	Emisión de declaraciones de pago	25 días
004	Emisión del ACH Payment	25 días

Estimación de la velocidad del proyecto Inicial

- Tiempo del calendario: 05 días por la semana, de lunes a viernes.
- Tiempo total estimado para la elaboración de las historias de usuario: 90 días.

2.3. Plan de Entrega

Tabla 16: Plan de entrega

Entregable	Historias	Fecha Inicio	Fecha Término	Fecha Entrega
Entregable 1	01	10/10/2016	04/11/2016	07/11/2016
Entregable 2	02	10/10/2016	04/11/2016	07/11/2016
Entregable 3	03	08/11/2016	12/12/2016	13/11/2016
Entregable 4	04	13/11/2016	16/01/2017	17/01/2017

3. FASE ITERACIONES

3.1. Iteración 1

3.1.1. Historia de Usuario N°1

Tabla 17: Resumen de tareas de la historia de usuario "Realización de un préstamo"

Historia 001: Realización del préstamo			Iteración 01
Sub historia	Fase	Tarea por Sub Historia	
Mantenedor prestatario	Diseño	3.1.1.1	Diseño de Interfaz del mantenedor prestatario
	Diseño	3.1.1.2	Diagrama de clases del mantenedor prestatario
	Implementación	3.1.1.3	Tarjeta CRC del mantenedor prestatario
	Pruebas	3.1.1.4	Prueba Unitaria y Funcional del mantenedor prestatario
Mantenedor prestamista	Diseño	3.1.1.5	Diseño de Interfaz del mantenedor prestamista
	Diseño	3.1.1.6	Diagrama de clases del mantenedor prestamista
	Implementación	3.1.1.7	Tarjeta CRC del mantenedor prestamista
	Pruebas	3.1.1.8	Prueba Unitaria y Funcional del mantenedor prestamista
Realización de un préstamo	Diseño	3.1.1.9	Diseño de Interfaz de la realización de un préstamo
	Diseño	3.1.1.10	Diagrama de clases de la realización de un préstamo
	Implementación	3.1.1.11	Tarjeta CRC de la realización de un préstamo
	Pruebas	3.1.1.12	Prueba Unitaria y Funcional de la realización de un préstamo

3.1.1.1. Diseño de interfaz del mantenedor prestatario

← All Investors Profile ▾

Investor Information :

Account Information

Account*: Alternative Acc.: Active On Hold

Vendor:

Company Information

Company:

Vesting:

Contact Name:

Tin Type: Tin:

Personal Information

First Name:

Vesting:

Salutation: Middle Name:

Last Name:

Birthday:

Tin Type: Tin:

Full Name

Terms

Sold Rate: Prin. Bal. %: Plus Fee:

Minim. Fee: Generate Historical Reports

Billing & Accounts

Trust Balance: Last Invoice Paid: **N/A**

Figura 6: Diseño de interfaz principal del mantenedor prestatario

Vendors ×

← All Vendors Profile History Portfolio Billing Charges Vendor Charges Notes & Attachments Vendor Reports

Vendor Additional : INS-000601

Contact Phone

Home Phone: Home Phn Work Phone: Work Phon

Mobile Phone: Mobile Phc Fax: Fax

Email: Email

Url: Url

Localization

Street:

City: City

State: Zip:

Options

Delivery Op.:

Send 1099

Receive Late Letters

Freeze all outgoing disbursements to Lender/Vendor

Send Check to Payee

Address:

All Lenders

Additional Lenders:

Save
Refresh

[← Back](#) / [Edit Charge](#) | Save

Charge Detail

Loan:

Original Amount:

Description*:

Comments:

Pay Automatically

Deferred

Creation Date:

Category:

Interest Rate:

Reference:

Interest From:

Distribution

Vendor/Investor:

Amount:

For Benefit of:

Amount:

Total:

Figura 7: Diseño de interfaces del mantenedor prestatario

3.1.1.2. Diagrama de clases del mantenedor prestatario

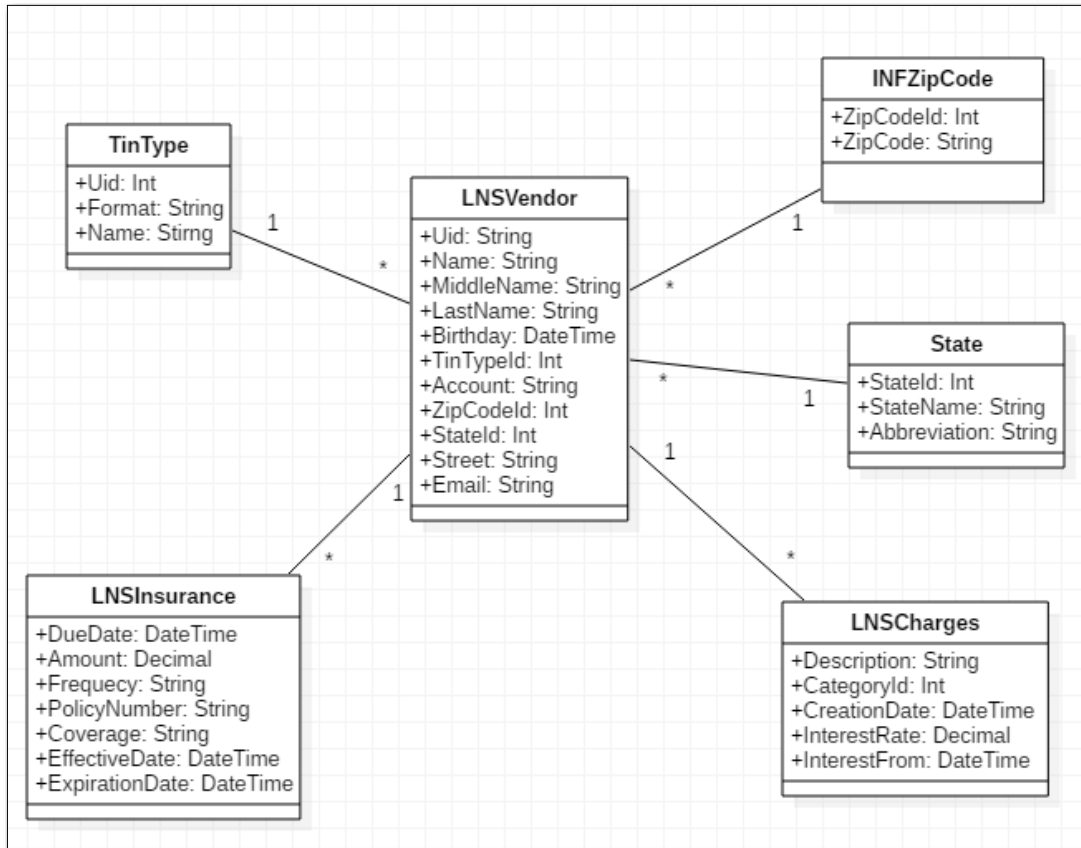


Figura 8: Diagrama de clases del mantenedor prestatario

3.1.1.3. Tarjeta CRC del mantenedor prestatario

Tabla 18: Tarjeta CRC del mantenedor prestatario

Mantenedor prestatario	
Responsabilidad	Colaboradores
<p>Vista prestatario</p> <ul style="list-style-type: none"> • GetAll(\$optionalParams) • Insert(\$LNSVendor) • Update(\$LNSVendor) • Delete(\$LNSVendor) • GetByID(\$ID) 	<ul style="list-style-type: none"> • UtilitiesController • LNSVendor • LNSVendorFacade • LNSVendorController • LNSInsurance • LNSInsuranceFacade • LNSInsuranceController • LNSCharges • LNSChargesFacade • LNSChargesController
<p>Modelo prestatario</p> <ul style="list-style-type: none"> • GetAll(\$optionalParams) • Insert(\$LNSVendor) • Update(\$LNSVendor) 	

<ul style="list-style-type: none"> • Delete(\$LNSVendor) • GetByID(\$ID) <p>Control prestatario</p> <ul style="list-style-type: none"> • GetALL(\$optionalParams) • Insert(\$LNSVendor) • Update(\$LNSVendor) • Delete(\$LNSVendor) • GetByID(\$ID) 	
---	--

3.1.1.4. Prueba Unitaria del mantenedor prestatario

Prueba Unitaria

```

24 vendor.Street = loan.BrokerStreet ?? string.Empty;
25 vendor.City = loan.BrokerCity ?? string.Empty;
26 vendor.State = loan.BrokerState ?? string.Empty;
27 vendor.ZipCode = loan.BrokerZipCode ?? string.Empty;
28 vendor.HomePhone = loan.BrokerHomePhone ?? string.Empty;
29 vendor.Fax = loan.BrokerFax ?? string.Empty;
30 vendor.Email = loan.BrokerEmail ?? string.Empty;
31 vendor.ACHBankAccountType = (loan.BrokerAccountType && loan.BrokerAccountType) ? 1 : 0;
32 vendor.ACHReceivingDFI = loan.BrokerRoutingNumber ?? string.Empty;
33 vendor.ACHAccountNumber = loan.BrokerACHAccountNumber ?? string.Empty;
34 vendor.FullName = loan.BrokerFullName ?? string.Empty;
35
36 vendor.Type = (int)LNSVendor.VendorTypeEnum.VENDOR;
37 vendor.IsActive = true;
38 string expected = "Success !";
39 string actual = LNSVendorController.ValidateVendor(vendor);
40 Assert.AreEqual(expected,actual,"Message Information");
41 }
42 }
43 }

```

Test Results

miguel@CENTURIONDEV04 2018 - Ejecutar - Depurar - Agrupar por: [None] [All Co

Test run failed Results: 0/1 passed; Item(s) checked: 1

Result	Test Name	Project	Error Message
Failed	ValidateVendor	UnitTest	Error de Assert.AreEqual. Se esperaba <Success !>, pero es <Account is required and should be unique.>.

Figura 9: Prueba unitaria de verificación de cuenta del mantenedor prestatario

Si el usuario no ingresa un número de cuenta como identificador (Account) no podrá registrar el préstamo.

Pruebas funcionales

Tabla 19: Prueba funcional al mantenedor prestatario

Dato	Valores	Resultado
Account	= Letra mayúscula o minúscula	Inválido
	= Vacío	Inválido
	= Numérico	Válido

First Name	= Letra mayúscula o minúscula	Válido
	= Vacío	Válido
	= Numérico	Inválido
Last Name	= Letra mayúscula o minúscula	Válido
	= Vacío	Inválido
	= Numérico	Inválido
Birthday	= (75 a más) años	Inválido
	= (22 – 75) años	Válido
	= (0 - 21) años	Inválido
	= Letra mayúscula o minúscula	Inválido
Tin	= Letra mayúscula o minúscula	Inválido
	= Formato NN-NNNNNNNN	Válido
	= Formato diferente a NN-NNNNNNNN	Inválido
	= Caracteres ASCII	Inválido
Company	= Letra mayúscula o minúscula	Válido
	= Vacío	Inválido
	= Numérico	Válido
Contact Name	= Letra mayúscula o minúscula	Válido
	= Vacío	Válido
	= Numérico	Válido
Sold Rate:	= Valor predeterminado (0.0000)	Válido
	= Letra mayúscula o minúscula	Inválido
	= Formato (N.NNNNNNN)	Válido – Parte decimal redondeada a 4 decimales
	= Caracteres ASCII	Inválido
Prin. Bal. %	= Valor predeterminado (0.0000)	Válido
	= Letra mayúscula o minúscula	Inválido
	= Formato (N.NNNNNNN)	Válido – Parte decimal redondeada a 4 decimales
	= Caracteres ASCII	Inválido
Plus Fee	= Valor predeterminado (0.00)	Válido
	= Letra mayúscula o minúscula	Inválido
	= Formato (N.NNNNNNN)	Válido – Parte decimal redondeada a 2 decimales
Minim. Fee	= Valor predeterminado (0.00)	Válido
	= Letra mayúscula o minúscula	Inválido
	= Formato (N.NNNNNNN)	Válido – Parte decimal redondeada a 2 decimales

***N: Valor numérico [0-9]**

3.1.1.5. Diseño de interfaz del mantenedor prestamista

Personal Information

First Name

Middle Name

Last Name*

Salutation

BirthDay None

TIN Type TIN

Company Information

Company*

Contact Name

TIN Type TIN

Full Name

Contact Information

Home Phone

Work Phone

Mobile Phone

Fax

Street

City

State Zip Code

Email

Delivery Op. Pref. Lang

Best Time to Call

Time Range

Days M T W T F S S

Figura 10: Diseño de interfaz principal del mantenedor prestamista

🏠 Properties : 003579 / [Refresh](#)

🏠 Primary Property

Description 1,945 SQ. FT.-PARTY SUPPLY STORE & NUTRITION STORE

Street 10231 A & 10231 B Long Beach Blvd.

City Lynwood

State CA

Zip Code 90626

✎ Edit
🗑 Delete

🏠 New Property

Description

Street

City

State

Zip Code

+ Add Property

General | Legal | REO | Liens | Insurance | Disasters | Valuations

General Information

Main Property

Description	<input type="text" value="Description"/>	Actual Rent Amount	<input type="text" value="\$0.00"/>
Street*	<input style="border: 1px solid red;" type="text" value="Street"/>	Market Rent Amount	<input type="text" value="\$0.00"/>
City	<input type="text" value="City"/>	Zip Code	<input type="text" value="Zip Code"/>
County	<input type="text" value="County"/>	State	<input type="text" value="California"/>
Beds	<input type="text" value="0.00"/>	Baths	<input type="text" value="0.00"/>
Purchase Price	<input type="text" value="\$0.00"/>	Purchase Date	<input type="text" value="Enter date"/>
		Vacancy Rate	<input type="text" value="\$0.00"/>
		MSC	<input type="text"/>
		Year Built	<input type="text" value="Enter date"/>
		Closing Cost	<input type="text" value="\$0.00"/>

Valuations

Valuation	<input type="text"/>	Current Type	<input type="text" value="Current Appraisal"/>	Valuation Date	<input type="text" value="07/05/2000"/>
Original	<input type="text"/>	Original Types	<input type="text" value="Original Appraisal"/>	Original Date	<input type="text" value="Enter date"/>

Appraiser Information

Occupancy	<input type="text" value="Primary Borrower"/>	Type	<input type="text" value="Commercial"/>	Number of units	<input type="text"/>
LTV	<input type="text" value="0.0000 %"/>	APN	<input type="text" value="APN"/>	SQFT	<input type="text"/>
Market Value	<input type="text"/>	Appraiser Date	<input type="text" value="Enter date"/>	Swimming Poll	<input type="text" value="No"/>
Pledged Equity	<input type="text" value="\$0.00"/>	Priority	<input type="text" value="Nine"/>		

Figura 11: Diseño de interfaces del mantenedor de prestamista

3.1.1.6. Modelo de datos del mantenedor prestamista

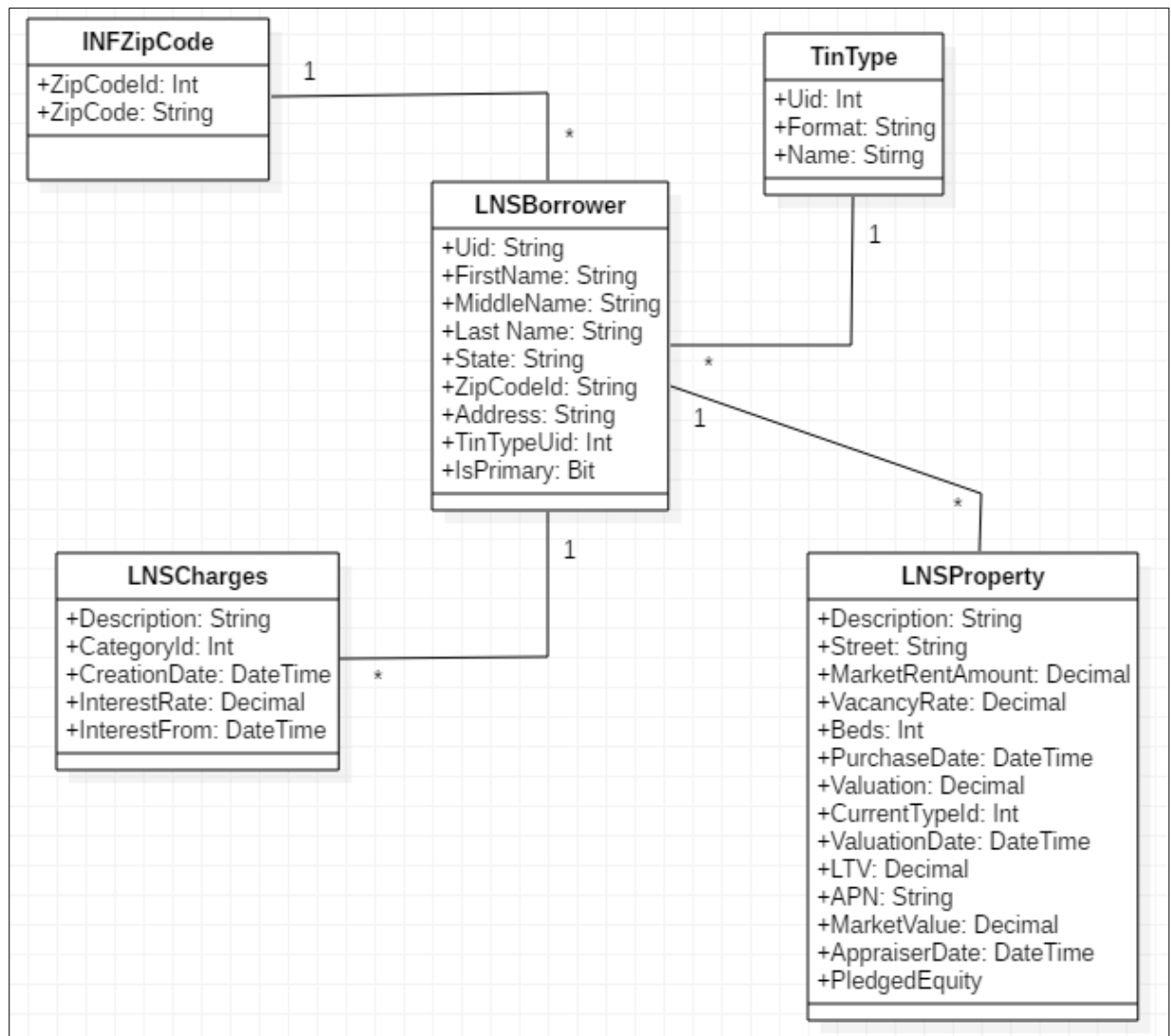


Figura 12: Diagrama de clases del mantenedor prestamista

3.1.1.7. Tarjeta CRC del mantenedor prestamista

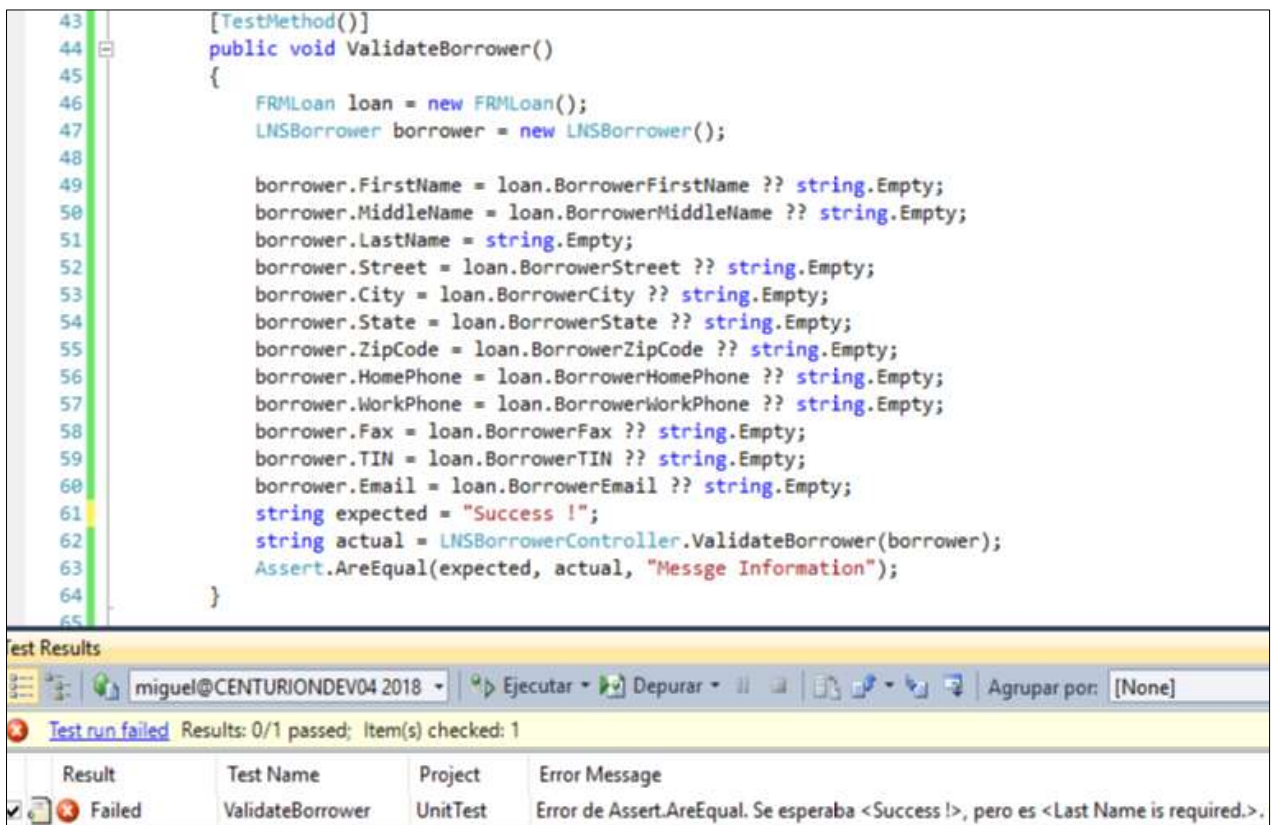
Tabla 20 Tarjeta CRC del mantenedor prestamista:

Mantenedor prestamista	
Responsabilidad	Colaboradores
<p><u>Vista prestamista</u></p> <ul style="list-style-type: none"> • GetAll(\$optionalParams) • Insert(\$LNSBorrower) • Update(\$LNSBorrower) • Delete(\$LNSBorrower) • GetByID(\$ID) 	<ul style="list-style-type: none"> • UtilitiesController • LNSBorrower • LNSBorrowerFacade • LNSBorrowerController

<p><u>Modelo prestamista</u></p> <ul style="list-style-type: none"> • GetAll(\$optionalParams) • Insert(\$LNSBorrower) • Update(\$LNSBorrower) • Delete(\$LNSBorrower) • GetByID(\$ID) <p><u>Control prestamista</u></p> <ul style="list-style-type: none"> • GetAll(\$optionalParams) • Insert(\$LNSBorrower) • Update(\$LNSBorrower) • Delete(\$LNSBorrower) • GetByID(\$ID) 	
--	--

3.1.1.8. Prueba unitaria y funcional del mantenedor prestamista

Prueba unitaria



```

43 [TestMethod()]
44 public void ValidateBorrower()
45 {
46     FRMLoan loan = new FRMLoan();
47     LNSBorrower borrower = new LNSBorrower();
48
49     borrower.FirstName = loan.BorrowerFirstName ?? string.Empty;
50     borrower.MiddleName = loan.BorrowerMiddleName ?? string.Empty;
51     borrower.LastName = string.Empty;
52     borrower.Street = loan.BorrowerStreet ?? string.Empty;
53     borrower.City = loan.BorrowerCity ?? string.Empty;
54     borrower.State = loan.BorrowerState ?? string.Empty;
55     borrower.ZipCode = loan.BorrowerZipCode ?? string.Empty;
56     borrower.HomePhone = loan.BorrowerHomePhone ?? string.Empty;
57     borrower.WorkPhone = loan.BorrowerWorkPhone ?? string.Empty;
58     borrower.Fax = loan.BorrowerFax ?? string.Empty;
59     borrower.TIN = loan.BorrowerTIN ?? string.Empty;
60     borrower.Email = loan.BorrowerEmail ?? string.Empty;
61     string expected = "Success !";
62     string actual = LNSBorrowerController.ValidateBorrower(borrower);
63     Assert.AreEqual(expected, actual, "Message Information");
64 }
65

```

Test Results

miguel@CENTURIONDEV04 2018 - Ejecutar - Depurar - Agrupar por: [None]

Test run failed Results: 0/1 passed; Item(s) checked: 1

Result	Test Name	Project	Error Message
Failed	ValidateBorrower	UnitTest	Error de Assert.AreEqual. Se esperaba <Success !>, pero es <Last Name is required.>.

Figura 13: Prueba unitaria de validacion de apellido del mantenedor prestamista

Si el usuario no ingresa el Apellido (Last Name) del prestamista no podrá registrar y se reportará un mensaje de error.

Prueba Funcional

Tabla 21: Prueba funcional del mantenedor prestamista

Dato	Valores	Resultado
First Name	= Letra mayúscula o minúscula	Válido
	= Vacío	Válido
	= Caracteres ASCII	Inválido
Middle Name	= Letra mayúscula o minúscula	Válido
	= Vacío	Válido
	= Caracteres ASCII	Inválido
Last Name	= Letra mayúscula o minúscula	Válido
	= Vacío	Inválido
	= Caracteres ASCII	Inválido
Birthday	= (65 a más) años	Inválido
	= (22 – 65) años	Válido
	= (0 - 21) años	Inválido
	= Letra mayúscula o minúscula	Inválido
Tin	= Letra mayúscula o minúscula	Inválido
	= Formato NN-NNNNNNN	Válido
	= Formato diferente a NN-NNNNNNN	Inválido
	= Caracteres ASCII	Inválido
Company	= Letra mayúscula o minúscula	Válido
	= Vacío	Inválido
	= Numérico	Válido
	= Caracteres ASCII	Válido
Contact Name	= Letra mayúscula o minúscula	Válido
	= Vacío	Válido
	= Caracteres ASCII	Válido
	= Numérico	Válido
Home Phone	= Letra mayúscula o minúscula	Inválido
	= Formato (NNN)NNN-NNNN	Válido
	= Formato diferente a (NNN)NNN-NNNN	Inválido
	= Vacío	Válido
Fax	= Letra mayúscula o minúscula	Inválido
	= Formato (NNN)NNN-NNNN	Válido
	= Formato diferente a (NNN)NNN-NNNN	Inválido
	= Vacío	Válido
Street	= Letra mayúscula o minúscula	Válido
	= Vacío	Válido
	= Caracteres ASCII	Válido
Zip Code	= Letra mayúscula o minúscula	Válido
	= Vacío	Válido
	= Caracteres ASCII	Inválido
State	= Valores predeterminados	Válido
	= Vacío	Inválido
Pref. Lang	= Valores predeterminados	Válido
	= Vacío	Inválido
Time Range	= Letra mayúscula o minúscula	Inválido
	= Vacío	Válido
	= Numérico	Válido

*N = Valor numérico [0-9]

3.1.1.9. Diseño de interfaz de la realización de un préstamo

Account Information

Loan Account * ?

Active (OnHold)

Trust Account ?

☰ Dates

Origination:

Assignment:

Boarding:

First Payment:

Payment Revision:

Int. Paid To:

Next Due:

PaidOff Date:

Maturity:

☰ Payments

Frequency:

Day Due*:

P&I*:

Suspense*:

Escrow*:

Other Payments

Total:

☰ General

Note Rate*:

Sold Rate:

Original Balance*:

Starting Balance*:

Principal Balance:

Total in trust:

Unpaid Interest:

Unpaid Fees:

Unpaid Late Charges:

Unpaid Loan Charges:

< Back
✔ Submit

Figura 14: Diseño de interfaz de la realización de un préstamo

3.1.1.10. Diagrama de clases de la realización de un préstamo

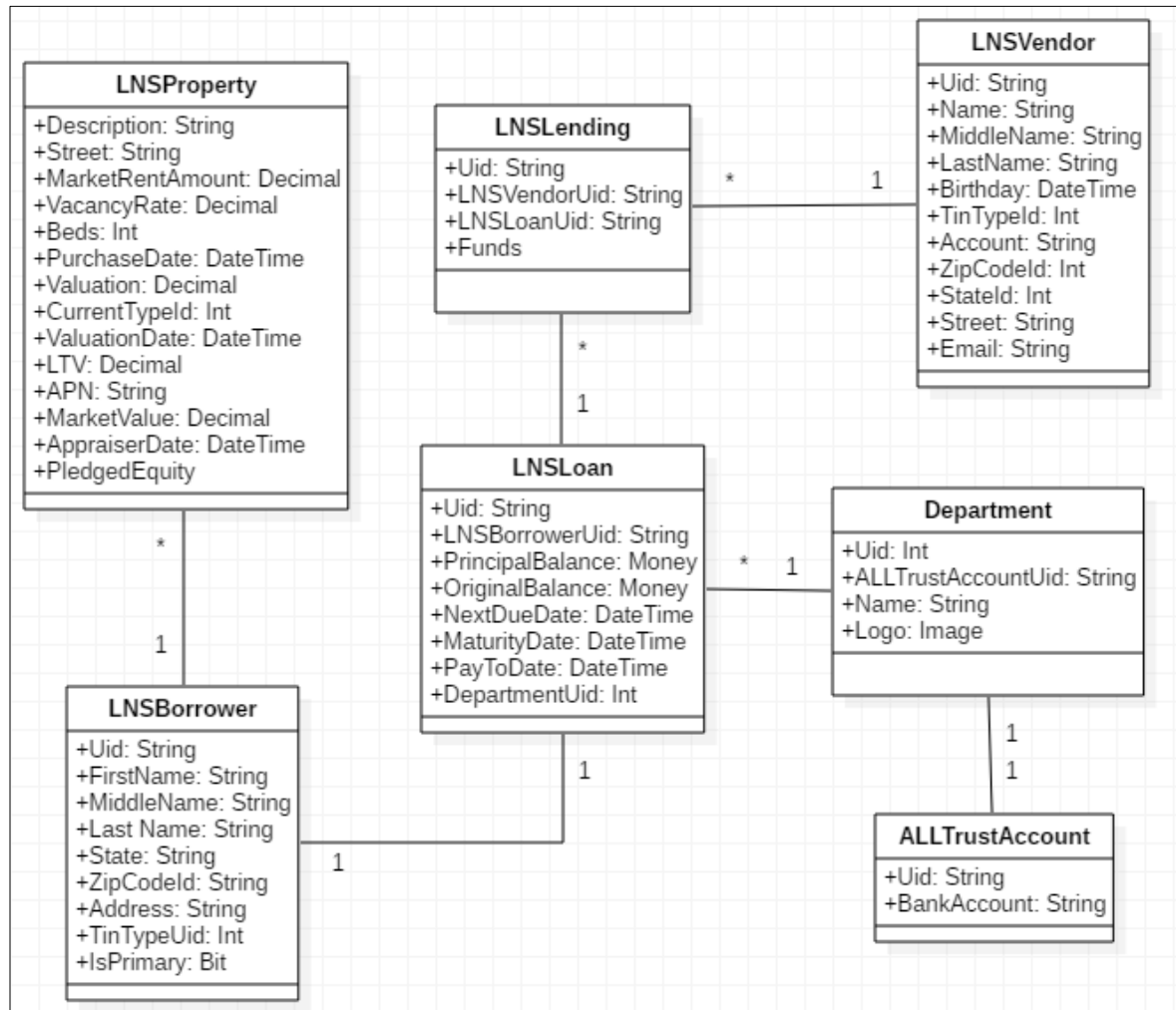


Figura 15: Diagrama de clases de la realización de un préstamo

3.1.1.11. Tarjeta CRC de la realización de un préstamo

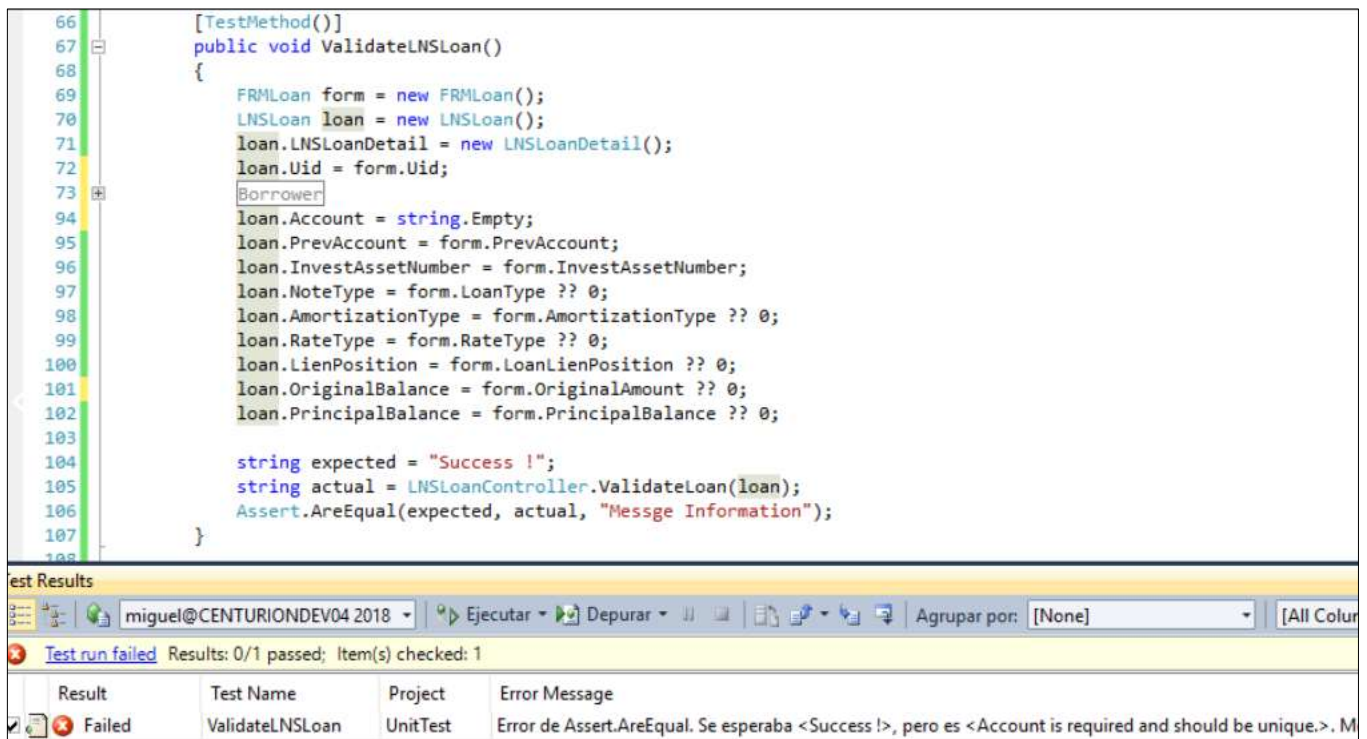
Tabla 22: Tarjeta CRC de la realización de un préstamo

Realización del préstamo	
Responsabilidad	Colaboradores
<u>Vista préstamo</u> <ul style="list-style-type: none"> • GetALL(\$OptionalParams) • Insert(\$LNSLoan) • Update(\$LNSLoan) • Delete(\$LNSLoan) • GetByAccount(\$Account) • GetAllBasicInformation() • GetResumeByLoanQueryable(\$IQueryable) • GetByID(\$ID) 	UtilitiesController LNSLoan LNSLoanFacade LNSLoanController

<p>Modelo préstamo</p> <ul style="list-style-type: none"> • GetAll(\$OptionalParams) • Insert(\$LNSLoan) • Update(\$LNSLoan) • Delete(\$LNSLoan) • GetByAccount(\$Account) • GetAllBasicInformation() • GetResumeByLoanQueryable(\$IQueryable) • GetByID(\$ID) <p>Control préstamo</p> <ul style="list-style-type: none"> • GetAll(\$OptionalParams) • Insert(\$LNSLoan) • Update(\$LNSLoan) • Delete(\$LNSLoan) • GetByAccount(\$Account) • GetAllBasicInformation() • GetResumeByLoanQueryable(\$IQueryable) • GetByID(\$ID) 	
--	--

3.1.1.12. Prueba unitaria y funcional de la realización de un préstamo

Prueba unitaria



```

66 [TestMethod()]
67 public void ValidateLNSLoan()
68 {
69     FRMLoan form = new FRMLoan();
70     LNSLoan loan = new LNSLoan();
71     loan.LNSLoanDetail = new LNSLoanDetail();
72     loan.Uid = form.Uid;
73     Borrower
74     loan.Account = string.Empty;
75     loan.PrevAccount = form.PrevAccount;
76     loan.InvestAssetNumber = form.InvestAssetNumber;
77     loan.NoteType = form.LoanType ?? 0;
78     loan.AmortizationType = form.AmortizationType ?? 0;
79     loan.RateType = form.RateType ?? 0;
80     loan.LienPosition = form.LoanLienPosition ?? 0;
81     loan.OriginalBalance = form.OriginalAmount ?? 0;
82     loan.PrincipalBalance = form.PrincipalBalance ?? 0;
83
84     string expected = "Success !";
85     string actual = LNSLoanController.ValidateLoan(loan);
86     Assert.AreEqual(expected, actual, "Message Information");
87 }
88

```

Test Results

miguel@CENTURIONDEV04 2018 | Ejecutar | Depurar | Agrupar por: [None] | [All Colors]

Test run failed Results: 0/1 passed; Item(s) checked: 1

Result	Test Name	Project	Error Message
Failed	ValidateLNSLoan	UnitTest	Error de Assert.AreEqual. Se esperaba <Success !>, pero es <Account is required and should be unique.>. M

Figura 16: Prueba unitaria de verificación de cuenta en la realización de un préstamo

Si el usuario no ingresa un número de cuenta como identificador (Account) no podrá registrar el préstamo.

Prueba funcional de caja negra

Tabla 23: Prueba funcional de la realización de un préstamo

Dato	Valores	Resultado
Loan Account	= Letra mayúscula o minúscula	Válido
	= Vacío	Inválido
	= Caracteres ASCII	Inválido
	= Numérico	Válido
Trust Account	= Valores predeterminados	Válido
	= Vacío	Inválido
First Payment	= Fecha ingresada < fecha actual	Inválido
	= Fecha ingresada > fecha actual	Válido
	= Vacío	Inválido
Payment Revision	= Fecha ingresada < fecha actual	Inválido
	= Fecha ingresada > fecha actual	Válido
	= Vacío	Inválido
Next Due	= Fecha ingresada < fecha actual	Inválido
	= Fecha ingresada > fecha actual	Válido
	= Vacío	Inválido
Maturity	= Fecha ingresada < fecha actual	Inválido
	= Fecha ingresada > fecha actual	Válido
	= Vacío	Válido
Day Due	= Letra mayúscula o minúscula	Inválido
	= Vacío	Inválido
	= Numérico	Válido
P&I	= Valor predeterminado (0.00)	Válido
	= Letra mayúscula o minúscula	Inválido
	= Formato (N.NNNNNNN)	Válido – Parte decimal redondeada a 2 decimales
	= Caracteres ASCII	Inválido
Suspense	= Valor predeterminado (0.00)	Válido
	= Letra mayúscula o minúscula	Inválido
	= Formato (N.NNNNNNN)	Válido – Parte decimal redondeada a 2 decimales
	= Caracteres ASCII	Inválido
Escrow	= Valor predeterminado (0.00)	Válido
	= Letra mayúscula o minúscula	Inválido
	= Formato (N.NNNNNNN)	Válido – Parte decimal redondeada a 2 decimales
	= Caracteres ASCII	Inválido
Original Balance	= Valor predeterminado (0.00)	Válido
	= Letra mayúscula o minúscula	Inválido
	= Formato (N.NNNNNNN)	Válido – Parte decimal redondeada a 2 decimales
	= Caracteres ASCII	Inválido
Starting Balance	= Valor predeterminado (0.00)	Válido
	= Letra mayúscula o minúscula	Inválido

	= Formato (N.NNNNNNN)	Válido – Parte decimal redondeada a 2 decimales
	= Valor predeterminado (0.00)	Válido
Note Rate	= Valor predeterminado (0.0000)	Válido
	= Letra mayúscula o minúscula	Inválido
	= Formato (N.NNNNNNN)	Válido – Parte decimal redondeada a 4 decimales
	= Caracteres ASCII	Inválido
Sold Rate	= Valor predeterminado (0.0000)	Válido
	= Letra mayúscula o minúscula	Inválido
	= Formato (N.NNNNNNN)	Válido – Parte decimal redondeada a 4 decimales
	= Caracteres ASCII	Inválido

***N: Valor Numérico [0-9]**

3.1.2. Historia de Usuario N°2

Tabla 24: Resumen de tareas de la historia de usuario "Realización del pago de letras de un préstamo"

Historia 002: Realización del pago de letras de un préstamo			Iteración 01
Sub historia	Fase	Tarea por Sub Historia	
Realización del pago de letras de un préstamo	Diseño	3.1.2.1	Diseño de Interfaz de la realización del pago de letras de un préstamo
	Diseño	3.1.2.2	Diagrama de clases de la realización del pago de letras de un préstamo
	Implementación	3.1.2.3	Tarjeta CRC de la realización del pago de letras de un préstamo
	Pruebas	3.1.2.4	Prueba Unitaria y Funcional de la realización del pago de letras de un préstamo

3.1.2.1. Diseño de Interfaz de la realización del pago de letras de un préstamo

Regular Payment x

Regular Payment

Account:

☰ Loan Information

Principal Balance	<input style="width: 90%;" type="text"/>
Suspense Balance	<input style="width: 90%;" type="text"/>
Escrow Balance	<input style="width: 90%;" type="text"/>
Unpaid Interest	<input style="width: 90%;" type="text"/>
Unpaid Fees	<input style="width: 90%;" type="text"/>
Unpaid Late Charges	<input style="width: 90%;" type="text"/>
Unpaid Charges	<input style="width: 90%;" type="text"/>
Regular Payment	<input style="width: 90%;" type="text"/>
Paid to Date	<input style="width: 90%;" type="text"/> <input type="button" value="Calendar"/>
Next Payment Due	<input style="width: 90%;" type="text"/> <input type="button" value="Calendar"/>
Next Revision	<input style="width: 90%;" type="text"/> <input type="button" value="Calendar"/>
Maturity Date	<input style="width: 90%;" type="text"/> <input type="button" value="Calendar"/>
Note Rate	<input style="width: 90%;" type="text"/>
Total Waived <input style="width: 90%;" type="text"/>	

☰ Payments

Principal Due	<input style="width: 90%;" type="text"/>
Interest Due	<input style="width: 90%;" type="text"/>
Paid Interest	<input style="width: 90%;" type="text"/>
Paid Late Charges	<input style="width: 90%;" type="text"/>
Late Charges Due	<input style="width: 90%;" type="text"/>
To Suspense	<input style="width: 90%;" type="text"/>
To Escrow	<input style="width: 90%;" type="text"/>
Unpaid Interest	<input style="width: 90%;" type="text"/>
Unpaid Fees	<input style="width: 90%;" type="text"/>
Pay Charges	<input style="width: 90%;" type="text"/>
Other Payments	<input style="width: 90%;" type="text"/>
Investor Fees	<input style="width: 90%;" type="text"/>
Broker Fees	<input style="width: 90%;" type="text"/>
Total <input style="width: 90%;" type="text"/>	

☰ Payment Information

Pymt Source	<input style="width: 90%;" type="text" value="Other"/>
Reference*	<input style="width: 90%;" type="text"/>
Date Due	<input style="width: 90%;" type="text"/> <input type="checkbox"/> <input type="button" value="Calendar"/>
Date Received	<input style="width: 90%;" type="text"/> <input type="button" value="Calendar"/>
Release Date	<input style="width: 90%;" type="text"/> <input type="button" value="Calendar"/>
Release Status	<input style="width: 90%;" type="text" value="Release"/>

☰ Source of Funds

From Borrower	<input style="width: 90%;" type="text"/>
From Suspense	<input style="width: 90%;" type="text"/>
From Escrow	<input style="width: 90%;" type="text"/>
Total Received <input style="width: 90%;" type="text"/>	
<input type="checkbox"/> Auto Distribution Adjustment	

Edit Payment Distribution

Figura 17: Diseño de interfaz de la realización del pago de letras de un préstamo

3.1.2.2. Diagrama de clases de la realización del pago de letras de un préstamo

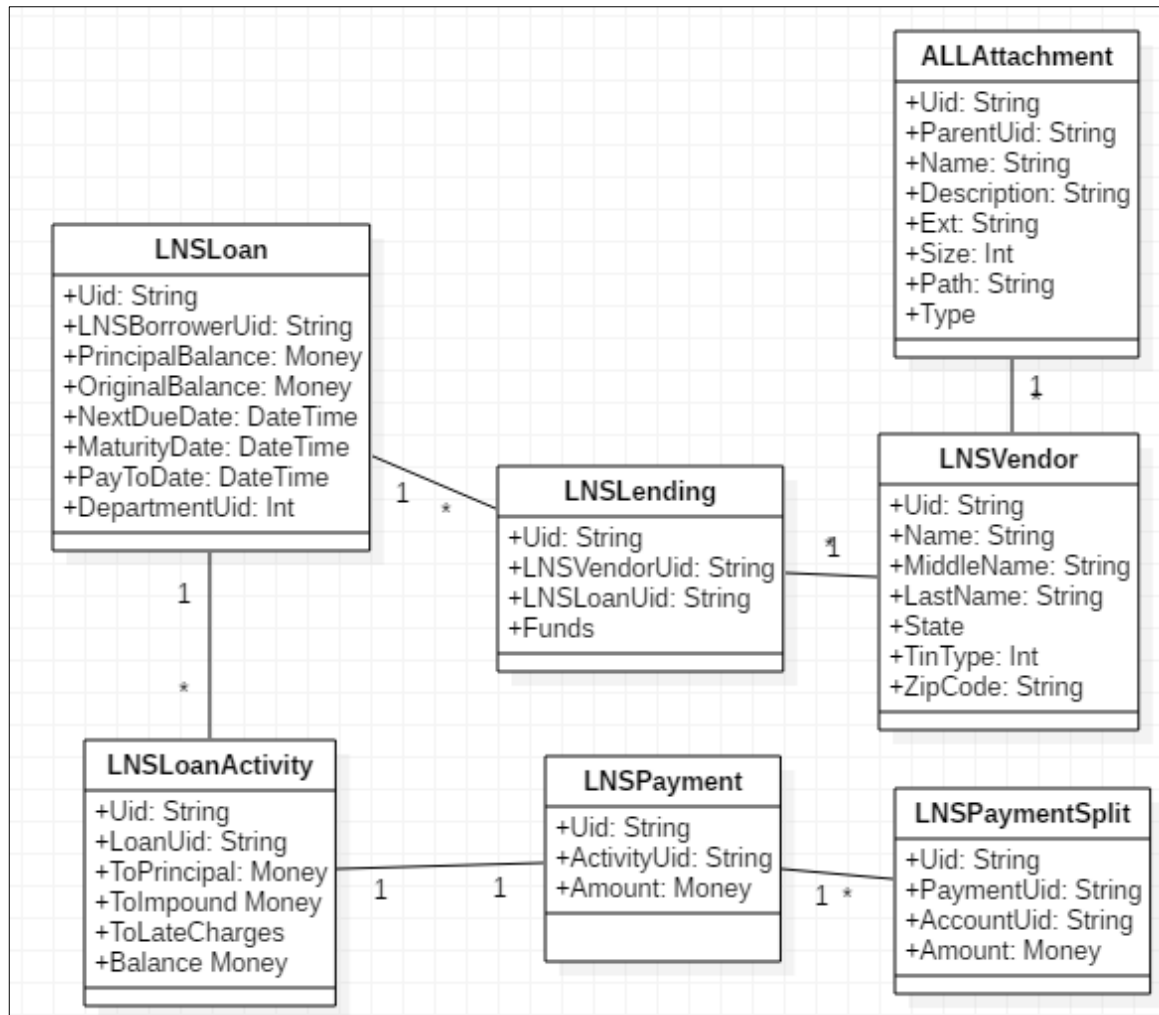


Figura 18: Diagrama de clases de la realización de pago de letras de un préstamo

3.1.2.3. Tarjeta CRC de la realización del pago de letras de un préstamo

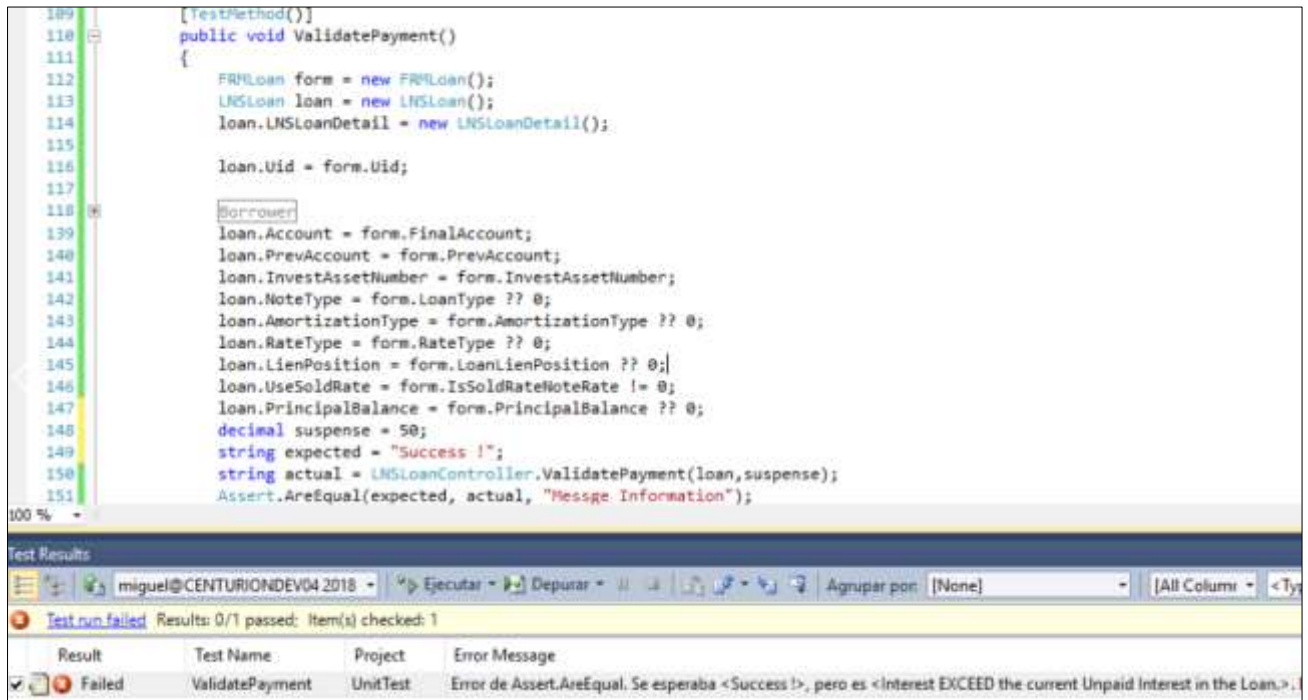
Tabla 25: Tarjeta CRC de la realización del pago de letras de un préstamo

Realización del pago de letras	
Responsabilidad	Colaboradores
<p>Vista pago</p> <ul style="list-style-type: none"> • GetAll(\$optionalParams) • Insert(\$LNSVendor) • Update(\$LNSVendor) • Delete(\$LNSVendor) • GetByID(\$ID) 	<ul style="list-style-type: none"> • UtilitiesController • LNSVendor • LNSLoan • LNSPayment • LNSPaymentSplit • LNSLoanActivity

<ul style="list-style-type: none"> • getTotalDistributionByRepaymentUid() • SavePaymentDistributions() • DeletePaymentWithSplits() • CalculateDepositDate() <p><u>Modelo pago</u></p> <ul style="list-style-type: none"> • GetAll(\$optionalParams) • Insert(\$LNSVendor) • Update(\$LNSVendor) • Delete(\$LNSVendor) • GetByID(\$ID) • getTotalDistributionByRepaymentUid() • SavePaymentDistributions() • DeletePaymentWithSplits() • CalculateDepositDate() <p><u>Control pago</u></p> <ul style="list-style-type: none"> • GetAll(\$optionalParams) • Insert(\$LNSVendor) • Update(\$LNSVendor) • Delete(\$LNSVendor) • GetByID(\$ID) • getTotalDistributionByRepaymentUid() • SavePaymentDistributions() • DeletePaymentWithSplits() • CalculateDepositDate() 	<ul style="list-style-type: none"> • LNSLenderActivity • LNSVendorFacade • LNSLoanFacade • LNSPaymentFacade • LNSPaymentController • LNSVendorController • LNSLoanController • LNSPaymentController
--	---

3.1.2.4. Prueba Unitaria y Funcional de la realización del pago de letras de un préstamo

Prueba Unitaria



```

109 [TestMethod()]
110 public void ValidatePayment()
111 {
112     FRMLoan form = new FRMLoan();
113     LNSLoan loan = new LNSLoan();
114     loan.LNSLoanDetail = new LNSLoanDetail();
115
116     loan.Uid = form.Uid;
117
118     Borrower
119     loan.Account = form.FinalAccount;
120     loan.PrevAccount = form.PrevAccount;
121     loan.InvestAssetNumber = form.InvestAssetNumber;
122     loan.NoteType = form.LoanType ?? 0;
123     loan.AmortizationType = form.AmortizationType ?? 0;
124     loan.RateType = form.RateType ?? 0;
125     loan.LienPosition = form.LoanLienPosition ?? 0;
126     loan.UseSoldRate = form.IsSoldRateNoteRate != 0;
127     loan.PrincipalBalance = form.PrincipalBalance ?? 0;
128     decimal suspense = 50;
129     string expected = "Success !";
130     string actual = LNSLoanController.ValidatePayment(loan, suspense);
131     Assert.AreEqual(expected, actual, "Message: Information");

```

Test Results

miguel@CENTURIONDEV04 2018 - Ejecutar - Depurar - Agnirar por [None] - [All Column] - <Ty

Test run failed. Results: 0/1 passed; Item(s) checked: 1

Result	Test Name	Project	Error Message
Failed	ValidatePayment	UnitTest	Error de Assert.AreEqual. Se esperaba <Success !>, pero es <Interest EXCEED the current Unpaid Interest in the Loan.>.

Figura 19: Prueba unitaria de verificación de la tarifa del prestamista en la realización del pago de letras de un préstamo

Si la tarifa del prestamista (LNSBorrower) no está configurada, no se podrá realizar el pago.

Prueba funcional de caja negra

Tabla 26: Prueba funcional de la realización del pago de letras de un préstamo

Dato	Valores	Resultado
Account	= Letra mayúscula o minúscula	Válido
	= Vacío	Inválido
	= Caracteres ASCII	Inválido
	= Numérico	Válido
Interest Due	= N <= 0	Inválido
	= Letra mayúscula o minúscula	Inválido
	= Formato (N.NNNNNNN)	Válido – Parte decimal redondeada a 2 decimales
	= Caracteres ASCII	Inválido
Paid Late Charges	= N <= 0	Inválido
	= Letra mayúscula o minúscula	Inválido
	= Formato (N.NNNNNNN)	Válido – Parte decimal redondeada a 2 decimales
	= Caracteres ASCII	Inválido

To Suspense	= N <= Suspense Balance	Válido
	= N > Suspense Balance	Inválido
	= Letra mayúscula o minúscula	Inválido
	= Formato (N.NNNNNNN)	Válido – Parte decimal redondeada a 2 decimales
	= Caracteres ASCII	Inválido
To Escrow	= N <= Escrow Balance	Válido
	= N > Escrow Balance	Inválido
	= Letra mayúscula o minúscula	Inválido
	= Formato (N.NNNNNNN)	Válido – Parte decimal redondeada a 2 decimales
	= Caracteres ASCII	Inválido
Investor Fees	= N <= 0	Inválido
	= Letra mayúscula o minúscula	Inválido
	= Formato (N.NNNNNNN)	Válido – Parte decimal redondeada a 2 decimales
	= Caracteres ASCII	Inválido
Broker Fees	= N <= 0	Inválido
	= Letra mayúscula o minúscula	Inválido
	= Formato (N.NNNNNNN)	Válido – Parte decimal redondeada a 2 decimales
	= Caracteres ASCII	Inválido

*N: Valor numérico [0-9]

3.1.3. Historia de usuario N°3

Tabla 27: Resumen de tareas de la historia de usuario "Emisión de declaraciones de pago"

Historia 003: Emisión de la declaración de pago			Iteración 02
Sub historia	Fase	Tarea por Sub Historia	
Emisión de la declaración de pago	Diseño	3.1.3.1	Diseño de Interfaz de la emisión de declaraciones de pago
	Diseño	3.1.3.2	Diagrama de clases de la emisión de declaraciones de pago
	Implementación	3.1.3.3	Tarjeta CRC de la emisión de declaraciones de pago
	Pruebas	3.1.3.4	Prueba Unitaria de la emisión de declaraciones de pago

3.1.3.1. Diseño de Interfaz de la emisión de declaraciones de pago

Payment Statement Report

Payment Information

Specify Range

From

To

Interval 1 17711

Print Statement for loans with payment due between

1 and 31

Choose Flags

Search in a flag (Type and Press Enter to Search)

All Flags

- Paid [Description]
- Special Transfer [Description]
- Foreclosure [Description]
- Complete Charge Off [Description]
- Transferred Out [Description]
- Payoff Demand [Description]
- Final Boarding [Description]
- RESPA [Description]

Include Flags

- Active [Description]

Exclude Flags

- Bankruptcy [Description]
- REO [Description]
- Charge Off [Description]
- Pre Boarding [Description]

Additional

Statement Date

Activity Since

[Include Privacy Police](#)

Send Parameters

SMTP to send

Release Date

Show Report

Process & Send

Figura 20: Diseño de interfaz de la emisión de declaraciones de pago

3.1.3.2. Diagrama de clases de la emisión de la declaración de pago

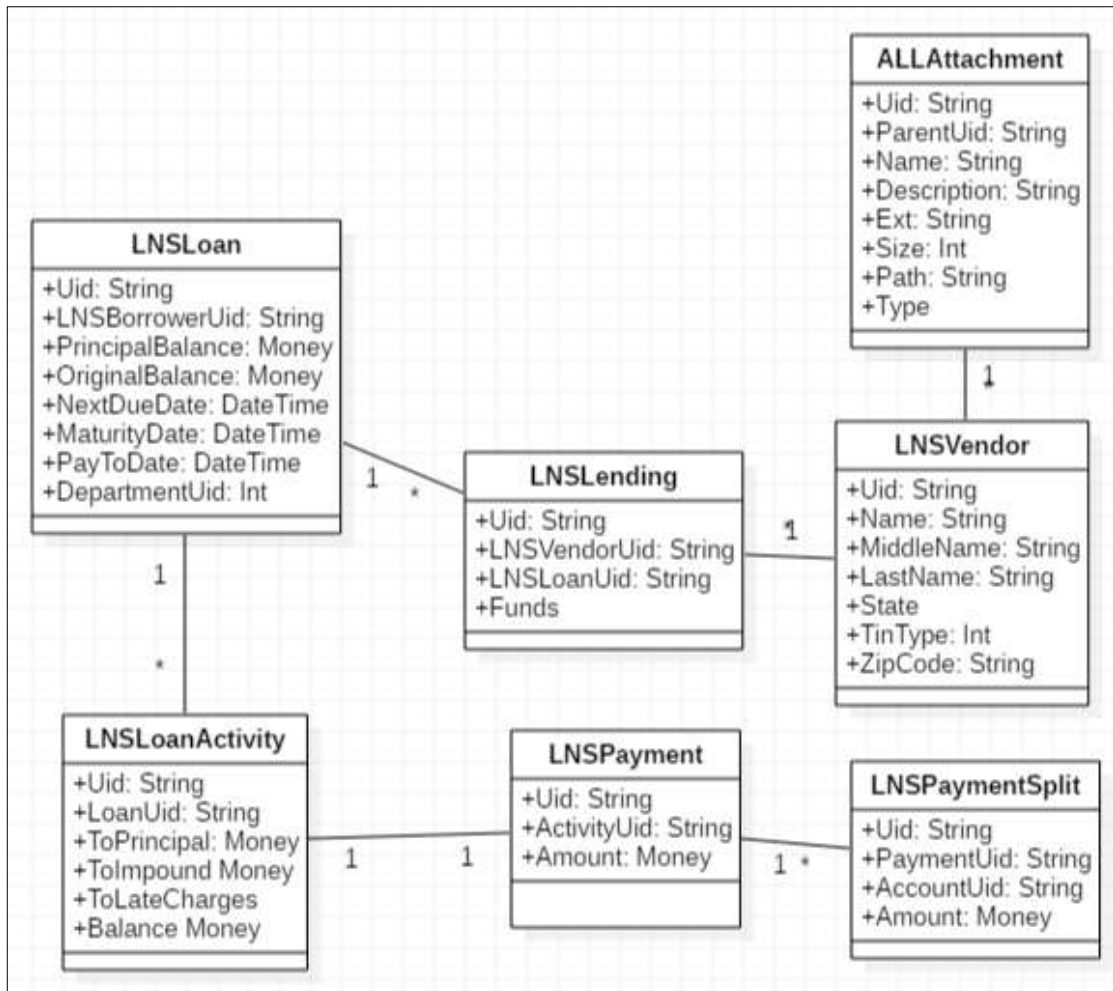


Figura 21: Diagrama de clases de la emisión de declaraciones de pago

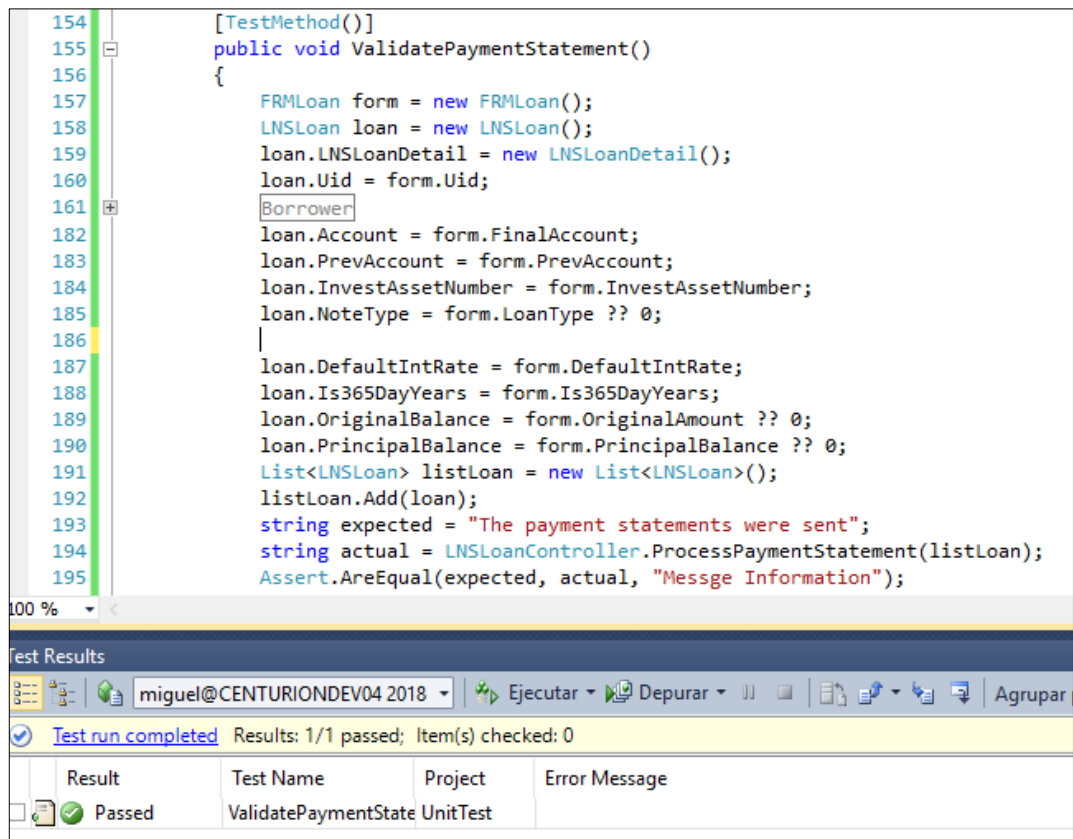
3.1.3.3. Tarjeta CRC de la emisión de declaraciones de pago

Tabla 28: Tarjeta CRC de la emisión de declaraciones de pago

Emisión de la declaración de pago	
Responsabilidad	Colaboradores
<p><u>Vista declaración de pago</u></p> <ul style="list-style-type: none"> • GetAll(\$optionalParams) • Insert(\$LNSVendor) • Update(\$LNSVendor) • Delete(\$LNSVendor) • GetById(\$ID) • getTotalDistributionByRepaymentUid() • SavePaymentDistributions() • DeletePaymentWithSplits() • CalculateDepositDate() 	<ul style="list-style-type: none"> • UtilitiesController • LNSVendor • LNSLoan • LNSPayment • LNSPaymentSplit • LNSLoanActivity • LNSLenderActivity • LNSVendorFacade • LNSLoanFacade • LNSPaymentFacade • LNSPaymentController
<p><u>Modelo declaración de pago</u></p>	

<ul style="list-style-type: none"> • GetAll(\$OptionalParams) • Insert(\$LNSVendor) • Update(\$LNSVendor) • Delete(\$LNSVendor) • GetByID(\$ID) • getTotalDistributionByRepaymentUid() • SavePaymentDistributions() • DeletePaymentWithSplits() • CalculateDepositDate() <p><u>Control declaración de pago</u></p> <ul style="list-style-type: none"> • GetAll(\$OptionalParams) • Insert(\$LNSVendor) • Update(\$LNSVendor) • Delete(\$LNSVendor) • GetByID(\$ID) • getTotalDistributionByRepaymentUid() • SavePaymentDistributions() • DeletePaymentWithSplits() • CalculateDepositDate() 	<ul style="list-style-type: none"> • LNSVendorController • LNSLoanController • LNSPaymentController
--	--

3.1.3.4. Prueba Unitaria de la emisión de la declaración de pago



```

154 [TestMethod()]
155 public void ValidatePaymentStatement()
156 {
157     FRMLoan form = new FRMLoan();
158     LNSLoan loan = new LNSLoan();
159     loan.LNSLoanDetail = new LNSLoanDetail();
160     loan.Uid = form.Uid;
161     Borrower
162     loan.Account = form.FinalAccount;
163     loan.PrevAccount = form.PrevAccount;
164     loan.InvestAssetNumber = form.InvestAssetNumber;
165     loan.NoteType = form.LoanType ?? 0;
166     |
167     loan.DefaultIntRate = form.DefaultIntRate;
168     loan.Is365DayYears = form.Is365DayYears;
169     loan.OriginalBalance = form.OriginalAmount ?? 0;
170     loan.PrincipalBalance = form.PrincipalBalance ?? 0;
171     List<LNSLoan> listLoan = new List<LNSLoan>();
172     listLoan.Add(loan);
173     string expected = "The payment statements were sent";
174     string actual = LNSLoanController.ProcessPaymentStatement(listLoan);
175     Assert.AreEqual(expected, actual, "Messge Information");

```

Test Results

miguel@CENTURIONDEV04 2018 Ejecutar Depurar Agrupar

Test run completed Results: 1/1 passed; Item(s) checked: 0

Result	Test Name	Project	Error Message
Passed	ValidatePaymentState	UnitTest	

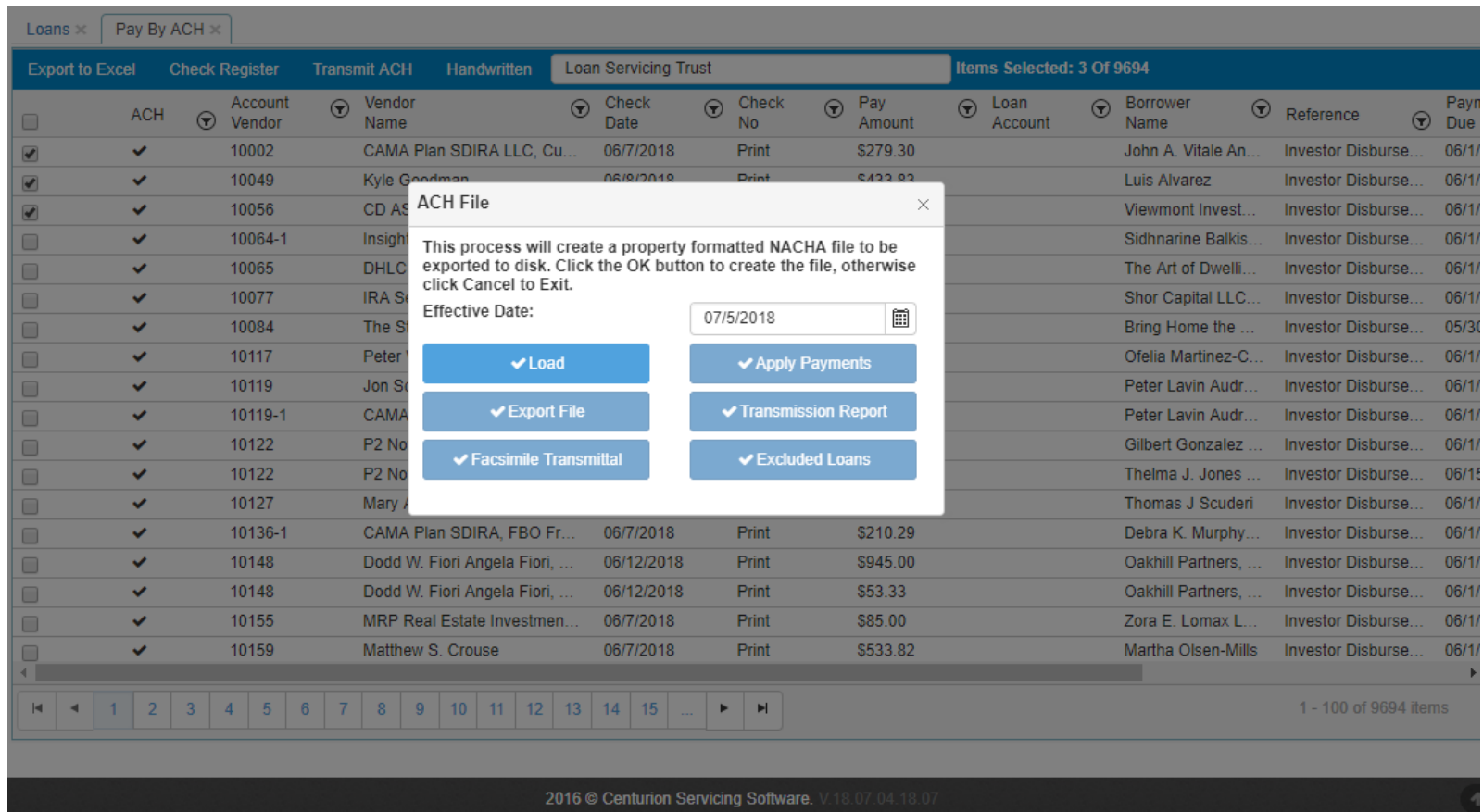
Figura 22: Prueba unitaria de validación de envío de declaraciones de pago

3.1.4. Historia de usuario N°4

Tabla 29: Resumen de tareas de la historia de usuario "Emisión del ACH Payment"

Historia 004: Emisión del ACH Payment			Iteración 03
Sub historia	Fase	Tarea por Sub Historia	
Emisión del ACH Payment	Diseño	3.1.4.1	Diseño de Interfaz de la emisión del ACH Payment
	Diseño	3.1.4.2	Diagrama de clases de la emisión del ACH Payment
	Implementación	3.1.4.3	Tarjeta CRC de la realización de la emisión del ACH Payment
	Pruebas	3.1.4.4	Prueba Unitaria de la emisión del ACH Payment

3.1.4.1. Diseño de Interfaz de la emisión del ACH Payment



Loans x Pay By ACH x

Export to Excel Check Register Transmit ACH Handwritten Loan Servicing Trust Items Selected: 3 Of 9694

<input type="checkbox"/>	ACH	Account Vendor	Vendor Name	Check Date	Check No	Pay Amount	Loan Account	Borrower Name	Reference	Pay Due
<input checked="" type="checkbox"/>	✓	10002	CAMA Plan SDIRA LLC, Cu...	06/7/2018	Print	\$279.30		John A. Vitale An...	Investor Disburse...	06/17
<input checked="" type="checkbox"/>	✓	10049	Kyle Goodman	06/8/2018	Print	\$423.83		Luis Alvarez	Investor Disburse...	06/17
<input checked="" type="checkbox"/>	✓	10056	CD AS					Viewmont Invest...	Investor Disburse...	06/17
<input type="checkbox"/>	✓	10064-1	Insigh					Sidhnarine Balkis...	Investor Disburse...	06/17
<input type="checkbox"/>	✓	10065	DHLC					The Art of Dwelli...	Investor Disburse...	06/17
<input type="checkbox"/>	✓	10077	IRA S					Shor Capital LLC...	Investor Disburse...	06/17
<input type="checkbox"/>	✓	10084	The S					Bring Home the ...	Investor Disburse...	05/30
<input type="checkbox"/>	✓	10117	Peter					Ofelia Martinez-C...	Investor Disburse...	06/17
<input type="checkbox"/>	✓	10119	Jon S					Peter Lavin Audr...	Investor Disburse...	06/17
<input type="checkbox"/>	✓	10119-1	CAMA					Peter Lavin Audr...	Investor Disburse...	06/17
<input type="checkbox"/>	✓	10122	P2 No					Gilbert Gonzalez ...	Investor Disburse...	06/17
<input type="checkbox"/>	✓	10122	P2 No					Thelma J. Jones ...	Investor Disburse...	06/15
<input type="checkbox"/>	✓	10127	Mary J					Thomas J Scuderi	Investor Disburse...	06/17
<input type="checkbox"/>	✓	10136-1	CAMA Plan SDIRA, FBO Fr...	06/7/2018	Print	\$210.29		Debra K. Murphy...	Investor Disburse...	06/17
<input type="checkbox"/>	✓	10148	Dodd W. Fiori Angela Fiori, ...	06/12/2018	Print	\$945.00		Oakhill Partners, ...	Investor Disburse...	06/17
<input type="checkbox"/>	✓	10148	Dodd W. Fiori Angela Fiori, ...	06/12/2018	Print	\$53.33		Oakhill Partners, ...	Investor Disburse...	06/17
<input type="checkbox"/>	✓	10155	MRP Real Estate Investmen...	06/7/2018	Print	\$85.00		Zora E. Lomax L...	Investor Disburse...	06/17
<input type="checkbox"/>	✓	10159	Matthew S. Crouse	06/7/2018	Print	\$533.82		Martha Olsen-Mills	Investor Disburse...	06/17

1 - 100 of 9694 items

2016 © Centurion Servicing Software. V.18.07.04.18.07

Figura 23: Diseño de interfaz de la emisión del ACH Payment

3.1.4.2. Diagrama de clases de la emisión del ACH Payment

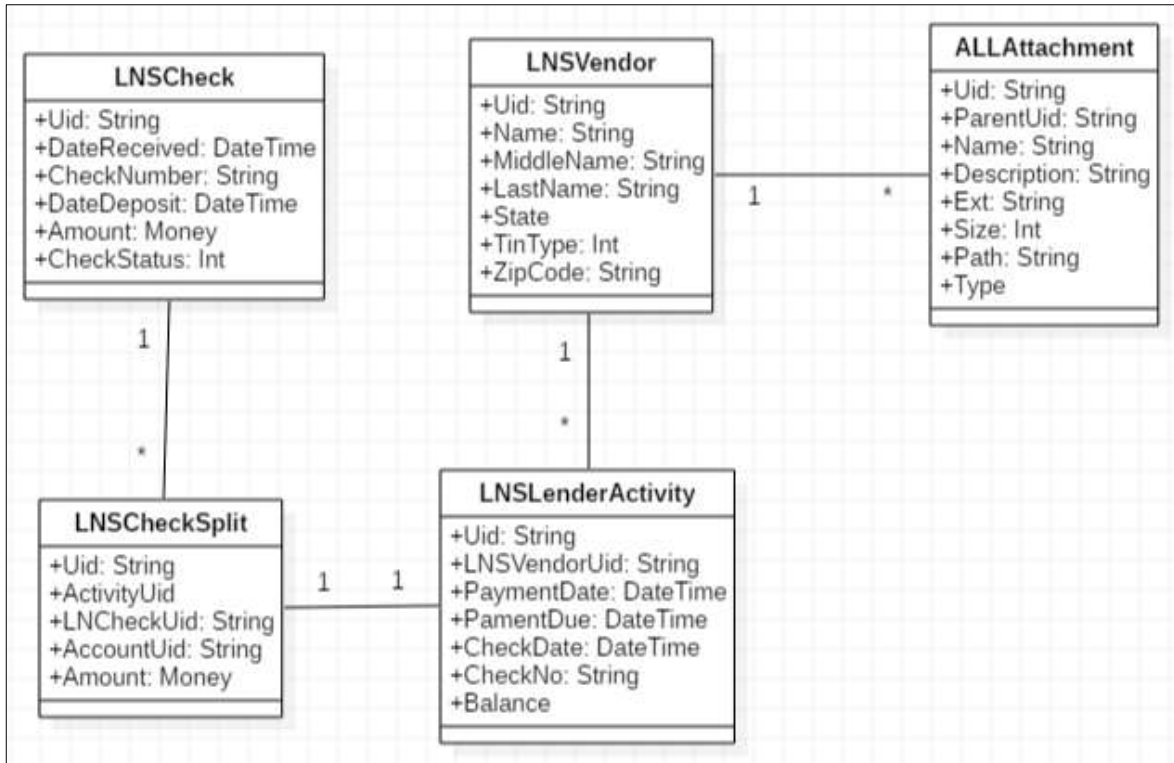


Figura 24: Diagrama de clases de la emisión del ACH Payment

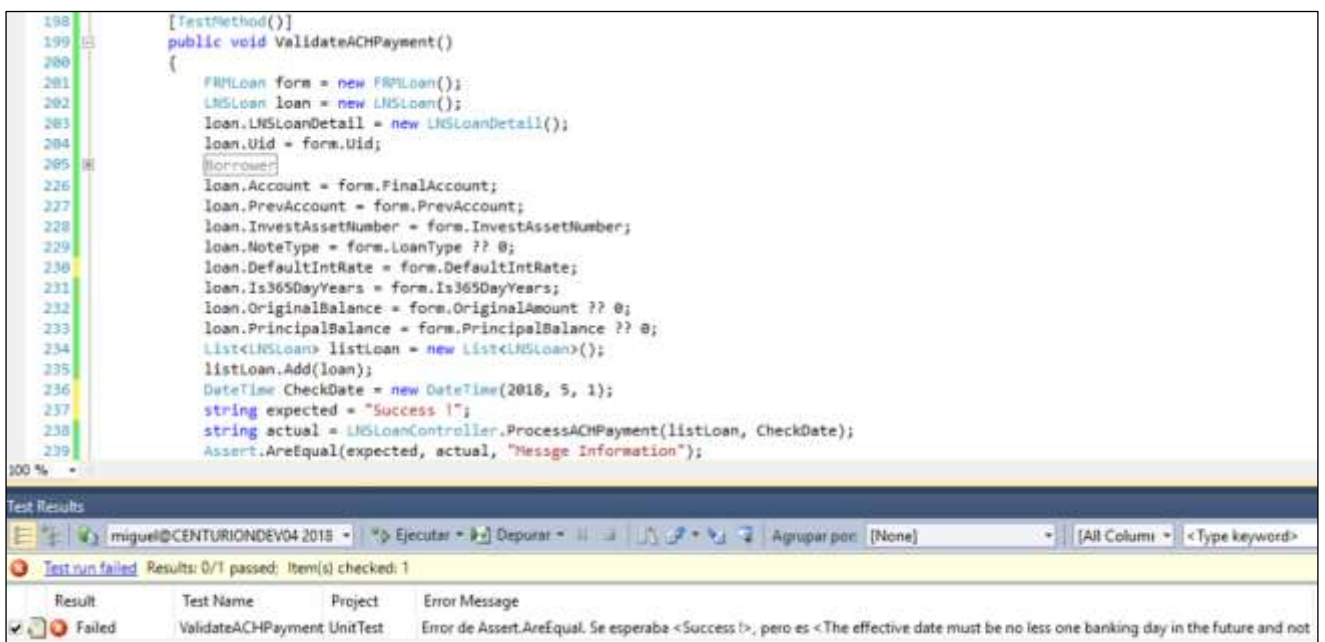
3.1.4.3. Tarjeta CRC de la realización de la emisión del ACH Payment

Tabla 30: Tarjeta CRC de la realización de la emisión del ACH Payment

Emisión del ACH Payment	
Responsabilidad	Colaboradores
<p>Vista ACH Payment</p> <ul style="list-style-type: none"> • GetAll(\$OptionalParams) • Insert(\$LNSVendor) • Update(\$LNSVendor) • Delete(\$LNSVendor) • GetById(\$ID) • getTotalDistributionByRepaymentUid() • SavePaymentDistributions() • DeletePaymentWithSplits() • CalculateDepositDate() <p>Modelo ACH Payment</p> <ul style="list-style-type: none"> • GetAll(\$OptionalParams) 	<ul style="list-style-type: none"> • UtilitiesController • LNSVendor • LNSLoan • LNSPayment • LNSPaymentSplit • LNSLoanActivity • LNSLenderActivity • LNSVendorFacade • LNSLoanFacade • LNSPaymentFacade • LNSPaymentController

<ul style="list-style-type: none"> • Insert(\$LNSVendor) • Update(\$LNSVendor) • Delete(\$LNSVendor) • GetByID(\$ID) • getTotalDistributionByRepaymentUid() • SavePaymentDistributions() • DeletePaymentWithSplits() • CalculateDepositDate() <p><u>Control ACH Payment</u></p> <ul style="list-style-type: none"> • GetALL(\$optionalParams) • Insert(\$LNSVendor) • Update(\$LNSVendor) • Delete(\$LNSVendor) • GetByID(\$ID) • getTotalDistributionByRepaymentUid() • SavePaymentDistributions() • DeletePaymentWithSplits() • CalculateDepositDate() 	<ul style="list-style-type: none"> • LNSVendorController • LNSLoanController • LNSPaymentController
--	--

3.1.4.4. Prueba Unitaria de la emisión del ACH Payment



```

198 [TestMethod()]
199 public void ValidateACHPayment()
200 {
201     FRMLoan form = new FRMLoan();
202     LNSLoan loan = new LNSLoan();
203     loan.LNSLoanDetail = new LNSLoanDetail();
204     loan.Uid = form.Uid;
205     Borrower
226     loan.Account = form.FinalAccount;
227     loan.PrevAccount = form.PrevAccount;
228     loan.InvestAssetNumber = form.InvestAssetNumber;
229     loan.NoteType = form.LoanType ?? 0;
230     loan.DefaultIntrRate = form.DefaultIntrRate;
231     loan.Is365DayYears = form.Is365DayYears;
232     loan.OriginalBalance = form.OriginalAmount ?? 0;
233     loan.PrincipalBalance = form.PrincipalBalance ?? 0;
234     List<LNSLoan> listLoan = new List<LNSLoan>();
235     listLoan.Add(loan);
236     DateTime CheckDate = new DateTime(2018, 5, 1);
237     string expected = "Success !";
238     string actual = LNSLoanController.ProcessACHPayment(listLoan, CheckDate);
239     Assert.AreEqual(expected, actual, "Message Information");

```

Test Results

miguel@CENTURIONDEV04 2018 - Ejecutar - Depurar - Agrupar por: [None] - [All Columns] - <Type keyword>

Test run failed Results: 0/1 passed: Item(s) checked: 1

Result	Test Name	Project	Error Message
Failed	ValidateACHPayment	UnitTest	Error de Assert.AreEqual. Se esperaba <Success !>, pero es <The effective date must be no less one banking day in the future and not

Figura 25: Prueba unitaria de la emisión del ACH Payment

Diagrama de componentes

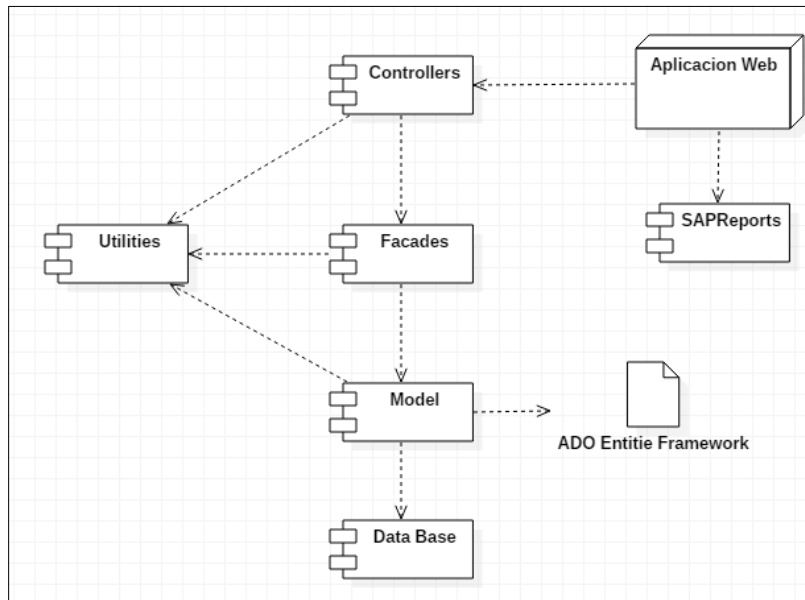


Figura 26: Diagrama de componentes

Diagrama de despliegue

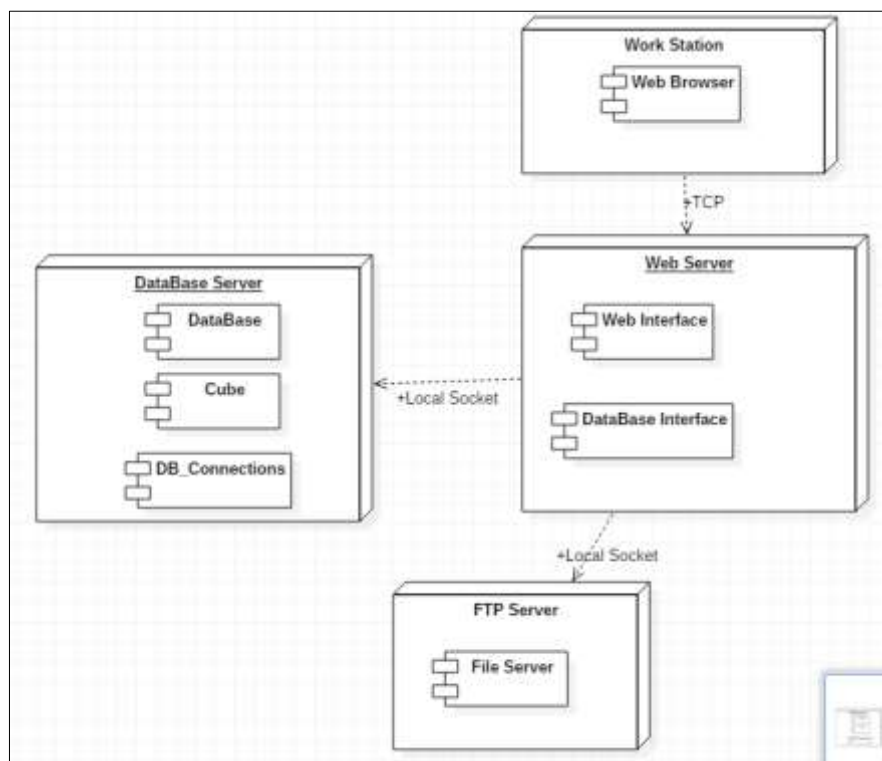


Figura 27: Diagrama de despliegue

CAPÍTULO 5. METODOLOGÍA

5.1. Diseño de investigación

Se seguirá un diseño de investigación pre experimental. El diagrama a continuación muestra el diseño propuesto, pre test – post test con un solo grupo.

$$O_1 \times O_2$$

Donde:

O_1 : Observación del grupo antes de aplicar la solución

X: Aplicación de la solución

O_2 : Observación del grupo luego de aplicar la solución

5.2. Unidad de estudio

El préstamo

5.3. Población

Cantidad de procesos realizados en un 1 mes:

8 procesos de emisión de declaraciones de pago

12 procesos de emisión de ACH Payment

550 procesos de la realización de pago

76 procesos de la realización de préstamo

5.4. Muestra (muestreo o selección)

5 procesos de emisión de declaraciones de pago

5 procesos de emisión de ACH Payment

10 procesos de la realización de pago

10 procesos de la realización de préstamo

5.5. Técnicas, instrumentos y procedimientos de recolección de datos

Tabla 31: Recolección de datos

Análisis de realidad problemática	INSTRUMENTOS	FUENTES
Causas y consecuencias de la realidad problemática	Cuestionario	Jefe del área de sistemas de la empresa financiera
	Observación	Comunicación constante con el jefe del área de sistemas de la empresa financiera.

Para poder obtener información con respecto a la variable de gestión de préstamos, se realizó una encuesta al jefe del área de sistemas de la empresa financiera para poder obtener las

debilidades que tienen actualmente con el software de escritorio que poseen, y poder de esta manera desarrollar un sistema que cumpla sus necesidades completamente **(ANEXO 1)**.

5.6. Métodos, instrumentos y procedimientos de análisis de datos

Tabla 32: Análisis de datos

TÉCNICAS	INSTRUMENTOS	FUENTES	INFORMANTES
Análisis de la calidad	Métricas ISO 9126: Se evalúan algunas características del software según la norma 9126 en base a las métricas	Software educativo	Programadores
Estabilidad (Test - retest)	Coefficiente de correlación	Jefe del área de sistemas	Usuarios finales

Procedimiento de análisis de datos:

Para el Análisis de la calidad del sistema de software, se tendrán en cuenta algunas métricas con sus respectivas fórmulas **(ANEXO 2)**.

Para verificar si el sistema web mejora el servicio de préstamos, se compararán los diferentes tiempos obtenidos del sistema de escritorio actual y el sistema web propuesto en cada uno de los indicadores. En el caso de obtención del nivel aceptación (Valoración) de los usuarios finales, se utilizó una encuesta realizada antes y después del sistema web propuesto. **(ANEXO 3)**.

CAPÍTULO 6. RESULTADOS

PRUEBA DE HIPÓTESIS

Para la contratación de hipótesis se ha realizado el método Pre Test - Post Test, lo cual permitirá evaluar la eficacia de poner en práctica la hipótesis, y determinar su aceptación o rechazo. Así mismo, para la realización de este diseño se identificaron indicadores cuantitativos y cualitativos, los cuales se describen a continuación:

Tabla 33: Prueba de hipótesis

	INDICADOR	TIPO DE INDICADOR
01	Tiempo promedio de la realización de un préstamo	Cuantitativo
02	Tiempo promedio de la realización del pago de letras de un préstamo	Cuantitativo
03	Tiempo promedio de la emisión de declaraciones de pago	Cuantitativo
04	Tiempo promedio de la emisión del ACH Payment	Cuantitativo
05	Valoración promedio de la gestión de préstamos	Cualitativo

PRUEBA DE HIPÓTESIS INDICADOR DEPENDIENTE

Prueba de hipótesis para indicador 01 (cuantitativo)

Tiempo promedio de la realización de un préstamo:

a) Definición de variables

TPPa: Tiempo promedio de la realización de un préstamo con el sistema escritorio actual.

TPPs: Tiempo promedio de la realización de un préstamo con el sistema web propuesto.

b) Hipótesis estadística

Hipótesis H_0 : El tiempo promedio de la realización de un préstamo con el sistema de escritorio actual es menor o igual que el tiempo promedio de la realización de un préstamo con el sistema web propuesto. (Segundos)

$$H_0 = TPPa - TPPs \leq 0$$

Hipótesis H_a : El tiempo promedio de la realización de un préstamo con el sistema de escritorio actual es mayor que el tiempo promedio de la realización de un préstamo con el sistema web propuesto. (Segundos)

$$H_a = TPPa - TPPs > 0$$

c) Nivel de significancia (α)

Para tomar la decisión de rechazar la hipótesis nula (H_0) cuando ésta sea verdadera (falso positivo). El nivel de confianza es del 95%, por tanto, el nivel de significancia será de $\alpha = 0.05$

d) Estadígrafo de contraste

Debido a que la muestra es menor a 30, se usará la distribución t de Student.

$$t = \frac{\bar{d} - \bar{D}}{S_d / \sqrt{n}}$$

e) Resultados

Debido a que se está realizando una comparación de una característica de la muestra es que se usará la t de Student para dos muestras apareadas.

Tabla 34: Resultados del indicador tiempo promedio de la realización de un préstamo

	TPPa	TPPs	Diferencia (d)	$d_1 - \bar{d}$	$(d_i - \bar{d})^2$
1	4347.00	3447.00	900.00	110.10	12122.01
2	4503.00	3555.00	948.00	158.10	24995.61
3	4119.00	3927.00	192.00	-597.90	357484.41
4	4395.00	3795.00	600.00	-189.90	36062.01
5	4887.00	4080.00	807.00	17.10	292.41
6	4572.00	3495.00	1077.00	287.10	82426.41
7	4467.00	3795.00	672.00	-117.90	13900.41
8	4638.00	3855.00	783.00	-6.90	47.61
9	4878.00	3744.00	1134.00	344.10	118404.81
10	4350.00	3564.00	786.00	-3.90	15.21
SUMATORIA			7899.00		645750.90
MEDIA MUESTRAL (\bar{d})			789.90	Varianza (S)	267.862

Primero, se calcula el promedio de las diferencias obtenidas:

$$\bar{d} = \frac{\sum_i^n d_i}{n}$$

$$\bar{d} = \frac{7899}{10}$$

$$\bar{d} = 789.90$$

Luego se calcula la desviación estándar muestral de la diferencia:

$$S_d = \sqrt{\frac{1}{n-1} * \sum_{i=1}^n (d_i - \bar{d})^2}$$

$$S_d = \sqrt{\frac{1}{10-1} * \sum_{i=1}^n (d_i - 789.9)^2}$$

$$S_d = \sqrt{\frac{645750.9}{9}}$$

$$S_d = 267.862$$

Para calcular t:

$$t = \frac{\bar{d}}{S_d / \sqrt{n}}$$

$$t = \frac{789.9}{267.862 / \sqrt{10}}$$

$$t = 9.33$$

f) Región crítica

Para $\alpha = 0.05$ y $n - 1$ grado de libertad = 9 encontramos $t_{\alpha} = 1.833$. Lo cual significa que

$$P(T < 1.833) = 0.05$$

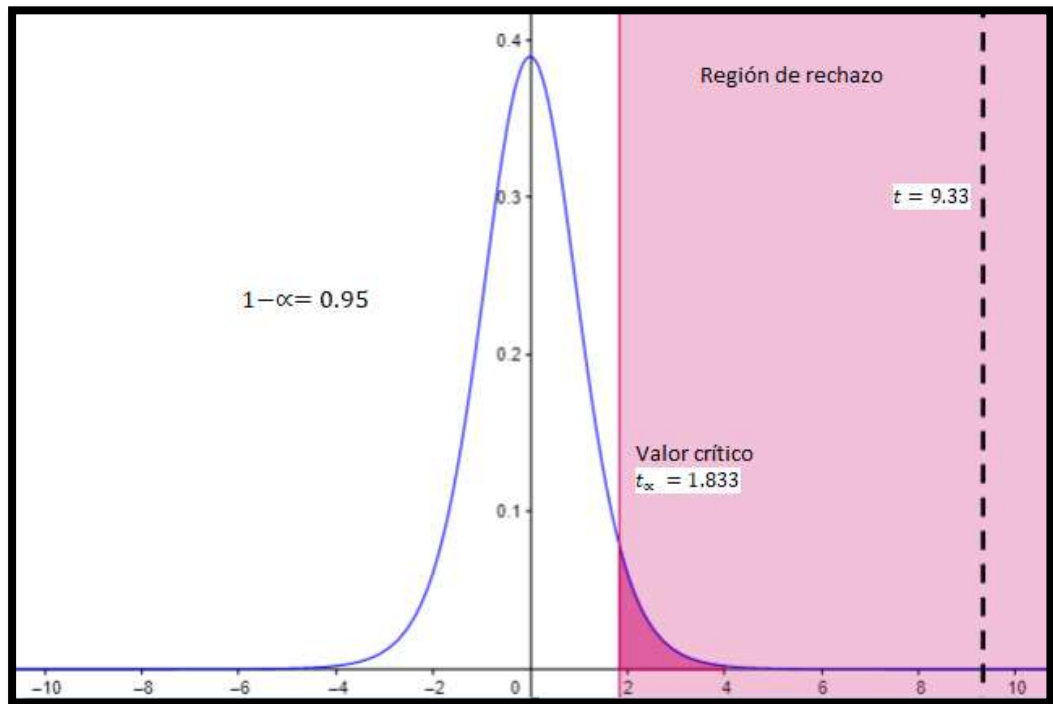


Figura 28: Gráfico Resultado Indicador Tiempo promedio de la realización de un préstamo

g) Conclusión

Puesto que nuestro valor calculado de t es 9.33 y es mayor que el valor de la tabla en un nivel de significancia de 0.05 ($9.33 > 1.833$) se da por aceptada H_a y rechazamos H_0 . Entonces, el tiempo promedio de la realización de un préstamo es menor con el sistema web propuesto que con el sistema de escritorio actual con un nivel de error del 5% y un nivel de confianza del 95%.

Prueba de hipótesis para indicador 02 (cuantitativo)

Tiempo promedio de la realización del pago de letras de un préstamo:

a) Definición de variables

TPRP_a: Tiempo promedio de la realización del pago de letras de un préstamo con el sistema escritorio actual.

TPRP_s: Tiempo promedio de la realización del pago de letras de un préstamo con el sistema web propuesto.

b) Hipótesis estadística

Hipótesis H_0 : El tiempo promedio de la realización del pago de letras de un préstamo con el sistema de escritorio actual es menor o igual que el tiempo promedio de la realización del pago de letras de un préstamo con el sistema web propuesto. (Segundos)

$$H_0 = TPRP_a - TPRP_s \leq 0$$

Hipótesis H_a : El tiempo promedio de la realización del pago de letras de un préstamo con el sistema de escritorio actual es mayor que el tiempo promedio de la realización del pago de letras de un préstamo con el sistema web propuesto. (Segundos)

$$H_0 = TPRPa - TPRPs > 0$$

c) Nivel de significancia (α)

Para tomar la decisión de rechazar la hipótesis nula (H_0) cuando ésta sea verdadera (falso positivo). El nivel de confianza es del 95%, por tanto, el nivel de significancia será de $\alpha = 0.05$

d) Estadígrafo de contraste

Debido a que la muestra es menor a 30, se usará la distribución t de Student.

$$t = \frac{\bar{d} - \bar{D}}{S_d / \sqrt{n}}$$

e) Resultados

Debido a que se está realizando una comparación de una característica de la muestra es que se usará la t de Student para dos muestras apareadas.

Tabla 35: Resultados del indicador tiempo promedio de la realización del pago de letras de un préstamo

	TPRPa	TPRPs	Diferencia (d)	$d_1 - \bar{d}$	$(d_i - \bar{d})^2$
1	1527.00	1260.00	267	-67	4489
2	1560.00	1143.00	417	83	6889
3	1521.00	1287.00	234	-100	10000
4	1755.00	1540.00	215	-119	14161
5	1590.00	1311.00	279	-55	3025
6	1641.00	1284.00	357	23	529
7	1875.00	1327.00	548	214	45796
8	1752.00	1227.00	525	191	36481
9	1581.00	1395.00	186	-148	21904
10	1713.00	1401.00	312	-22	484
SUMATORIA			3340		143758
MEDIA MUESTRAL (\bar{d})			334	Varianza (S)	126.39

Primero, se calcula el promedio de las diferencias obtenidas:

$$\bar{d} = \frac{\sum_i^n d_i}{n}$$

$$\bar{d} = \frac{3340}{10}$$

$$\bar{d} = 334$$

Luego se calcula la desviación estándar muestral de la diferencia:

$$S_d = \sqrt{\frac{1}{n-1} * \sum_{i=1}^n (d_i - \bar{d})^2}$$

$$S_d = \sqrt{\frac{1}{10-1} * \sum_{i=1}^n (d_i - 334)^2}$$

$$S_d = \sqrt{\frac{143758}{9}}$$

$$S_d = 126.39$$

Para calcular t:

$$t = \frac{\bar{d}}{S_d / \sqrt{n}}$$

$$t = \frac{334}{126.39 / \sqrt{10}}$$

$$t = 8.36$$

f) Región crítica

Para $\alpha = 0.05$ y $n - 1$ grado de libertad = 9 encontramos $t_{\alpha} = 1.833$. Lo cual significa que $P(T < 1.833) = 0.05$

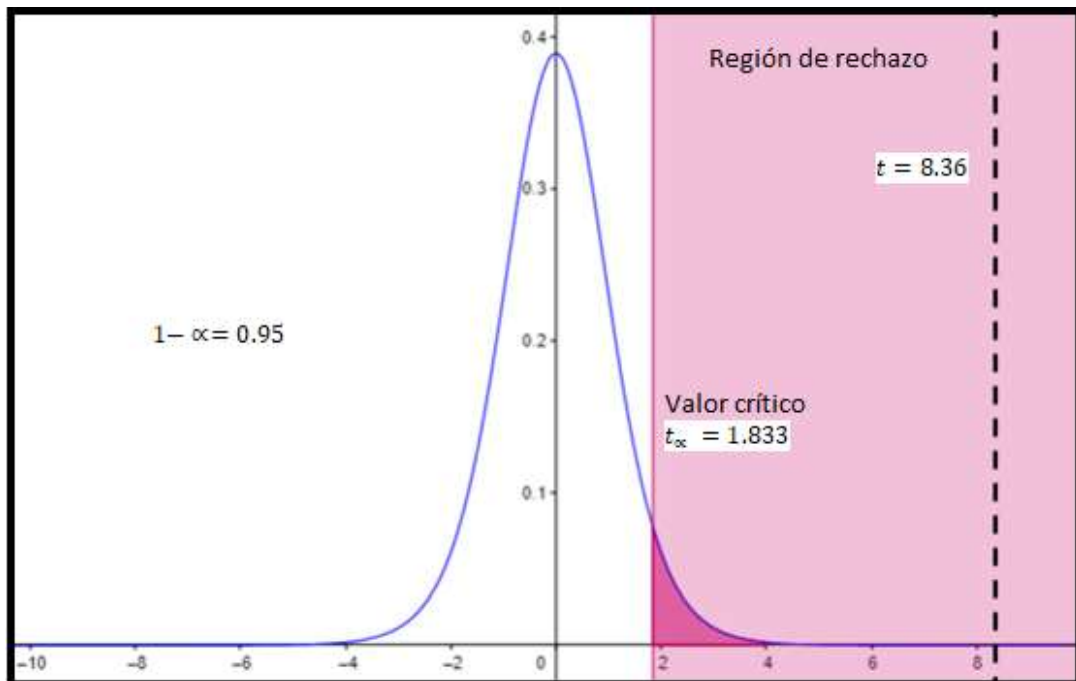


Figura 29: Gráfico Resultado Indicador Tiempo promedio de realización del pago de letras de un préstamo

g) Conclusión

Puesto que nuestro valor calculado de t es 8.36 y es mayor que el valor de la tabla en un nivel de significancia de 0.05 ($8.36 > 1.833$) se da por aceptada H_a y rechazamos H_0 . Entonces, el tiempo promedio de la realización del pago de letras de un préstamo es menor con el sistema web propuesto que con el sistema de escritorio actual con un nivel de error del 5% y un nivel de confianza del 95%.

Prueba de hipótesis para indicador 03 (cuantitativo)

Tiempo promedio de la emisión de declaraciones de pago

a) Definición de variables

TPDa: Tiempo promedio de la emisión de declaraciones de pago con el sistema escritorio actual.

TPDs: Tiempo promedio de la emisión de declaraciones de pago con el sistema web propuesto.

b) Hipótesis estadística

Hipótesis H_0 : El tiempo promedio de la emisión de declaraciones de pago con el sistema de escritorio actual es menor o igual que el tiempo promedio de la emisión de declaraciones de pago con el sistema web propuesto. (Minutos)

$$H_0 = TPDa - TPDs \leq 0$$

Hipótesis H_a : El tiempo promedio de la emisión de declaraciones de pago con el sistema de escritorio actual es mayor que el tiempo promedio de la emisión de declaraciones de pago con el sistema web propuesto. (Minutos)

$$H_0 = TPDa - TPDs > 0$$

c) Nivel de significancia (α)

Para tomar la decisión de rechazar la hipótesis nula (H_0) cuando ésta sea verdadera (falso positivo). El nivel de confianza es del 95%, por tanto, el nivel de significancia será de $\alpha = 0.05$

d) Estadígrafo de contraste

Debido a que la muestra es menor a 30, se usará la distribución t de Student.

$$t = \frac{\bar{d} - \bar{D}}{S_d / \sqrt{n}}$$

e) Resultados

Debido a que se está realizando una comparación de una característica de la muestra es que se usará la t de Student para dos muestras apareadas.

Tabla 36: Resultados del indicador tiempo promedio de la emisión de la declaración de pago

	Cantidad de declaraciones de Pago	TPDa	TPDs	Diferencia (d)	$d_1 - \bar{d}$	$(d_i - \bar{d})^2$
1	8845	494.40	345.00	149.40	18.18	330.51
2	7218	445.20	313.20	132.00	0.78	0.61
3	9112	521.50	396.12	125.38	-5.84	34.11
4	10103	565.40	439.20	126.20	-5.02	25.20
5	9855	551.52	428.42	123.10	-8.12	65.93
SUMATORIA				656.08		456.36
MEDIA MUESTRAL (\bar{d})				131.22	Varianza (S)	10.68

Primero, se calcula el promedio de las diferencias obtenidas:

$$\bar{d} = \frac{\sum_i^n d_i}{n}$$

$$\bar{d} = \frac{656.59}{5}$$

$$\bar{d} = 131.22$$

Luego se calcula la desviación estándar muestral de la diferencia:

$$S_d = \sqrt{\frac{1}{n-1} * \sum_{i=1}^n (d_i - \bar{d})^2}$$

$$S_d = \sqrt{\frac{1}{5-1} * \sum_{i=1}^n (d_i - 131.22)^2}$$

$$S_d = \sqrt{\frac{456.36}{4}}$$

$$S_d = 10.68$$

Para calcular t:

$$t = \frac{\bar{d}}{S_d / \sqrt{n}}$$

$$t = \frac{131.22}{10.68 / \sqrt{5}}$$

$$t = 27.47$$

f) Región crítica

Para $\alpha = 0.05$ y $n - 1$ grado de libertad = 4 encontramos $t_{\alpha} = 2.132$. Lo cual significa que $P(T < 2.132) = 0.05$

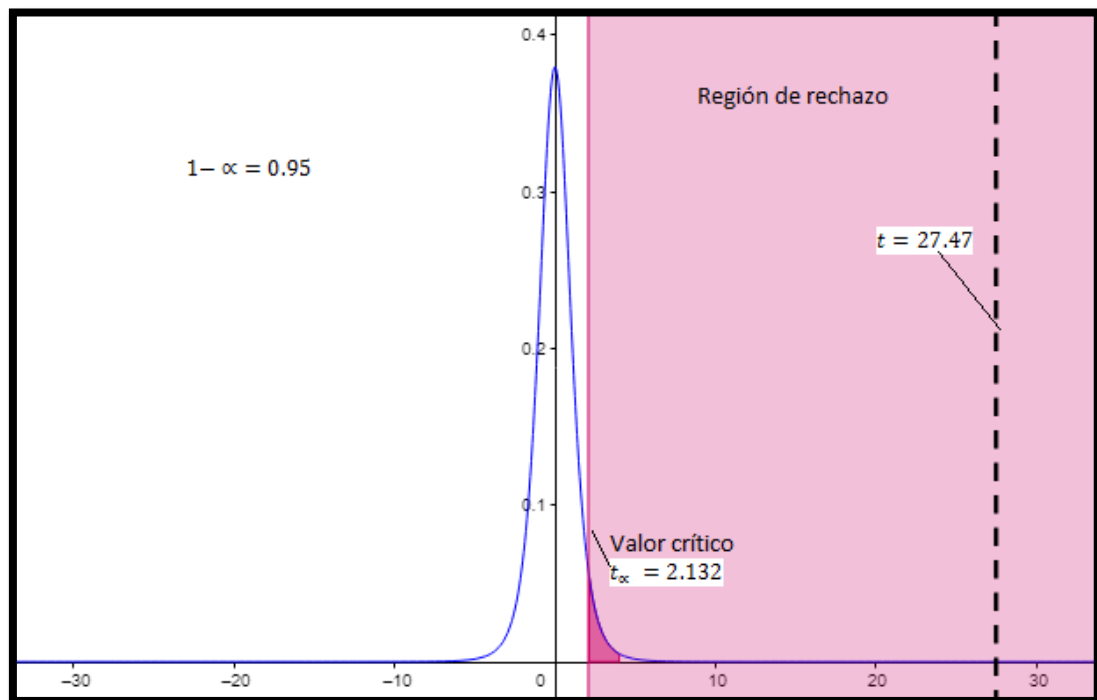


Figura 30: Gráfico Resultados Indicador del tiempo promedio de declaraciones de pago

g) Conclusión

Puesto que nuestro valor calculado de t es 27.47 y es mayor que el valor de la tabla en un nivel de significancia de 0.05 ($27.47 > 2.132$) se da por aceptada H_a y rechazamos H_0 . Entonces, el tiempo promedio de la emisión de declaraciones de pago es menor con el sistema web propuesto que con el sistema de escritorio actual con un nivel de error del 5% y un nivel de confianza del 95%.

Prueba de hipótesis para indicador 04 (cuantitativo)

Tiempo promedio de la emisión del ACH Payment

a) Definición de variables

TPAa: Tiempo promedio de la emisión del ACH Payment con el sistema escritorio actual.

TPAs: Tiempo promedio de la emisión del ACH Payment con el sistema web propuesto.

b) Hipótesis estadística

Hipótesis H_0 : El tiempo promedio de la emisión del ACH Payment con el sistema de escritorio actual es menor o igual que el tiempo promedio la emisión del ACH Payment con el sistema web propuesto. (Minutos)

$$H_0 = TPAa - TPAs \leq 0$$

Hipótesis H_a : El tiempo promedio de la emisión del ACH Payment con el sistema de escritorio actual es mayor que el tiempo promedio de la emisión del ACH Payment con el sistema web propuesto. (Minutos)

$$H_0 = TPAa - TPAs > 0$$

c) Nivel de significancia (α)

Para tomar la decisión de rechazar la hipótesis nula (H_0) cuando ésta sea verdadera (falso positivo). El nivel de confianza es del 95%, por tanto, el nivel de significancia será de $\alpha = 0.05$

d) Estadígrafo de contraste

Debido a que la muestra es menor a 30, se usará la distribución t de Student.

$$t = \frac{\bar{d} - \bar{D}}{S_d / \sqrt{n}}$$

e) Resultados

Debido a que se está realizando una comparación de una característica de la muestra es que se usará la t de Student para dos muestras apareadas.

Tabla 37: Resultados del indicador tiempo promedio de la emisión del ACH Payment

	Cuentas adjuntas	TPAa	TPAs	Diferencia (d)	$d_1 - \bar{d}$	$(d_i - \bar{d})^2$
1	1993	31.50	20.00	11.50	-29.73	883.87
2	5226	91.26	49.32	41.94	0.71	0.50
3	10003	210.10	110.50	99.60	58.37	3407.06
4	6613	115.10	63.20	51.90	10.67	113.85
5	412	5.25	4.05	1.20	-40.03	1602.40
SUMATORIA				206.14		6007.68
MEDIA MUESTRAL (\bar{d})				41.23	Varianza (S)	38.75

Primero, se calcula el promedio de las diferencias obtenidas:

$$\bar{d} = \frac{\sum_i^n d_i}{n}$$

$$\bar{d} = \frac{206.14}{5}$$

$$\bar{d} = 41.23$$

Luego se calcula la desviación estándar muestral de la diferencia:

$$S_d = \sqrt{\frac{1}{n-1} * \sum_{i=1}^n (d_i - \bar{d})^2}$$

$$S_d = \sqrt{\frac{1}{5-1} * \sum_{i=1}^n (d_i - 41.23)^2}$$

$$S_d = \sqrt{\frac{6007.68}{4}}$$

$$S_d = 38.75$$

Para calcular t:

$$t = \frac{\bar{d}}{S_d / \sqrt{n}}$$

$$t = \frac{41.23}{38.75 / \sqrt{5}}$$

$$t = 2.38$$

f) Región crítica

Para $\alpha = 0.05$ y $n - 1$ grado de libertad = 4 encontramos $t_{\alpha} = 2.132$. Lo cual significa que $P(T < 2.132) = 0.05$

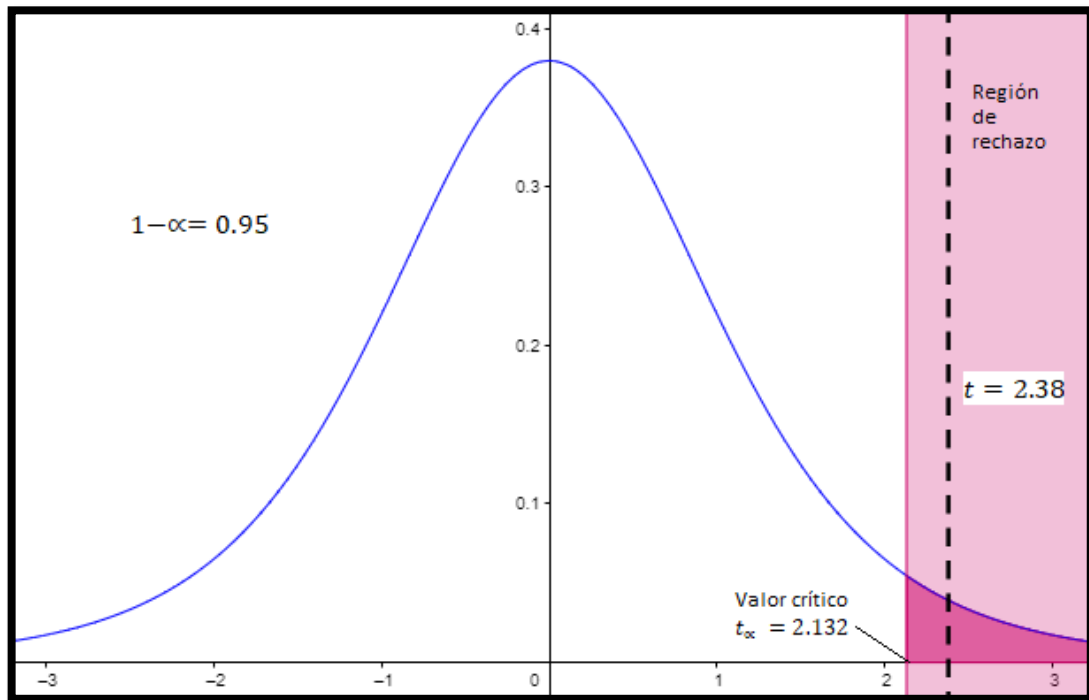


Figura 31: Gráfico Resultados Indicador Tiempo promedio de la emisión del ACH Payment

g) Conclusión

Puesto que nuestro valor calculado de t es 2.38 y es mayor que el valor de la tabla en un nivel de significancia de 0.05 ($2.38 > 2.132$) se da por aceptada H_a y rechazamos H_0 . Entonces, el tiempo promedio de la emisión del ACH Payment es menor con el sistema web propuesto que con el sistema de escritorio actual con un nivel de error del 5% y un nivel de confianza del 95%.

Prueba de hipótesis para indicador 05 (cualitativo)

Valoración promedio de la gestión de préstamos

a) Definición de variables

VPa: Valoración promedio de la gestión de préstamos con el sistema de escritorio actual.

VPs: Valoración promedio de la gestión de préstamos con el sistema de web propuesto.

b) Hipótesis estadística

Hipótesis H_0 : La valoración promedio de la gestión de préstamos con el sistema de escritorio actual es mayor o igual que la valoración promedio de la gestión de préstamos con el sistema de web propuesto.

$$H_0 = VP_a - VP_s \leq 0$$

Hipótesis H_a : La valoración promedio de la gestión de préstamos con el sistema de escritorio actual es menor que la valoración promedio de la gestión de préstamos con el sistema de web propuesto.

$$H_0 = VP_a - VP_s > 0$$

c) Nivel de significancia (α)

Para tomar la decisión de rechazar la hipótesis nula (H_0) cuando ésta sea verdadera (falso positivo). El nivel de confianza es del 95%, por tanto, el nivel de significancia será de $\alpha = 0.05$

d) Estadígrafo de contraste

Debido a que la muestra es menor a 30, se usará la distribución t de Student.

$$t = \frac{\bar{d} - \bar{D}}{S_d / \sqrt{n}}$$

e) Resultados

Debido a que se está realizando una comparación de una característica de la muestra es que se usará la t de Student para dos muestras apareadas.

Tabla 38: Resultados del indicador tiempo promedio de la emisión del ACH Payment

	VPa	VPs	Diferencia (d)	$d_i - \bar{d}$	$(d_i - \bar{d})^2$
1	2.40	4.60	-2.20	0.00	0.00
2	3.00	4.80	-1.80	0.40	0.16
3	2.60	5.00	-2.40	-0.20	0.04
4	2.60	4.80	-2.20	0.00	0.00
5	2.20	4.60	-2.40	-0.20	0.04
SUMATORIA			-11.00		0.24
MEDIA MUESTRAL (\bar{d})			-2.20	Varianza (S)	0.24

Primero, se calcula el promedio de las diferencias obtenidas:

$$\bar{d} = \frac{\sum_i^n d_i}{n}$$

$$\bar{d} = \frac{-11.00}{5}$$

$$\bar{d} = -2.20$$

Luego se calcula la desviación estándar muestral de la diferencia:

$$S_d = \sqrt{\frac{1}{n-1} * \sum_{i=1}^n (d_i - \bar{d})^2}$$

$$S_d = \sqrt{\frac{1}{5-1} * \sum_{i=1}^n (d_i - (-2.20))^2}$$

$$S_d = \sqrt{\frac{0.24}{4}}$$

$$S_d = 0.24$$

Para calcular t:

$$t = \frac{\bar{d}}{S_d / \sqrt{n}}$$

$$t = \frac{-2.20}{0.24 / \sqrt{5}}$$

$$t = -20.08$$

f) Región crítica

Para $\alpha = 0.05$ y $n - 1$ grado de libertad = 4 encontramos $t_{\alpha} = 2.132$. Lo cual significa que $P(T < 2.132) = 0.05$

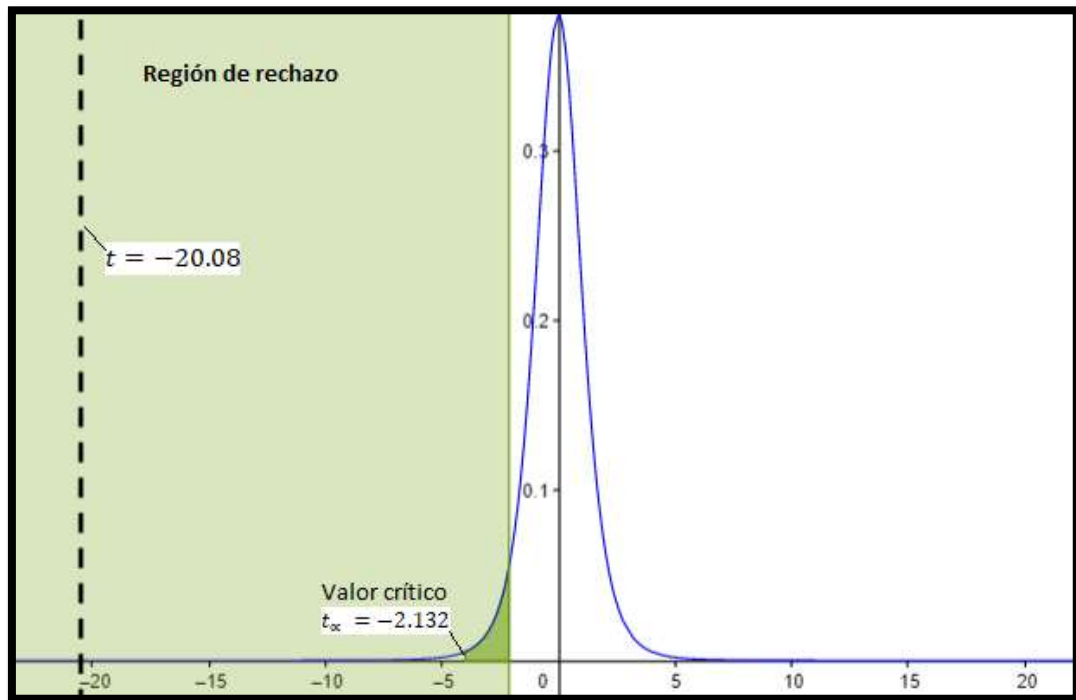


Figura 32: Gráfico Resultados Indicador Valoración promedio de la gestión de préstamos

g) Conclusión

Puesto que nuestro valor calculado de t es -20.08 , es menor que -2.132 y se encuentra en la región de rechazo se da por aceptada H_a y rechazamos H_0 . Entonces, la valoración promedio de la gestión de préstamos es mayor con el sistema web propuesto que con el sistema de escritorio actual con un nivel de error del 5% y un nivel de confianza del 95%.

CAPÍTULO 7. DISCUSIÓN

Para que las empresas financieras que ofrecen el servicio de poder gestionar préstamos, entre 2 o más entes, sea competente; es necesario, poder contar con las tecnologías adecuadas que den soporte a gran cantidad de clientes que buscan satisfacer esta necesidad. Un factor muy importante para el triunfo de este negocio, es cumplir las tareas en el tiempo previsto.

Ante ello, analizamos los procesos de una empresa financiera que se dedica a este negocio, y elegimos 4 procesos fundamentales.

Luego de haber efectuado el análisis de resultados para el indicador de tiempo promedio de la realización de un préstamo, se obtiene que mediante el uso del sistema de escritorio con el que la empresa financiera contaba, este proceso duraba aproximadamente 75 minutos. A comparación del sistema web propuesto que dura aproximadamente 62 minutos, obteniendo un decremento del 17.33%. Si bien es cierto, el proceso para realizar un préstamo sigue siendo el mismo, se da notar

que el uso de una buena interfaz, mejor distribuida, acelera el proceso y ayuda a los usuarios finales a realizar un trabajo un poco más rápido.

Tabla 39: Discusión Indicador Tiempo promedio de la realización del préstamo

TPPa (100%)	TPPs (TPPs %)	Decremento (%)
75 min.	62 min. (82.67%)	13 min. (17.33%)

Del mismo modo, el análisis de resultados para el indicador de tiempo promedio de la realización del pago de letras de un préstamo nos indica como el uso del sistema web promedio reduce en un 21.43% la realización de este proceso. Ya que con el sistema de escritorio se obtuvo un tiempo promedio de 28 minutos, a diferencia del sistema web, con un tiempo aproximado de 22 minutos.

Tabla 40: Discusión Indicador Tiempo promedio de la realización del pago de letras

TPRPa (100%)	TPRPs (TPPs %)	Decremento (%)
28 min.	22 min. (78.57 %)	6 min. (21.43%)

De similar forma, el análisis de resultados para el indicador de tiempo promedio de la emisión de declaraciones de pago, tiene una reducción de 70 minutos aproximadamente, lo cual equivale a un decremento del 25.58%. Con el uso del sistema de escritorio, el tiempo promedio de este proceso es de 516 minutos. A comparación del sistema web que baja a un tiempo promedio de 384 minutos. Ante esto, se ha de considerar también, que se soluciona un gran problema a la empresa financiera, puesto que, al realizarse este proceso, el sistema de escritorio con el que contaban bajaba considerablemente el rendimiento del este, afectando el resto de actividades que realizaban en paralelo los empleados de la empresa financiera.

Tabla 41: Discusión Indicador Tiempo promedio de la emisión de declaraciones de pago

TPDa (100%)	TPDs (TPPs %)	Decremento (%)
516 min.	384 min. (74.42%)	132 min. (25.58%)

De igual manera, el análisis de resultados para el indicador del tiempo promedio de la emisión del ACH Payment, tiene un decremento bastante significativo del 46.15%. Pues, se obtuvo que con el uso del sistema de escritorio este proceso demoraba aproximadamente 91 minutos, a gran diferencia del sistema web que tiene un tiempo promedio de 49 minutos. La interpretación de este resultado se debe a que el sistema escritorio tiene un límite de cuentas adjuntas al ACH Payment, lo cual obligaba al usuario final a ejecutar el proceso más de 1 vez en la mayoría de oportunidades. A diferencia del sistema web propuesto que solo lo hace 1 sola vez y adjunta todas las cuentas que el usuario final pueda seleccionar.

Tabla 42: Discusión Indicador Tiempo promedio de la emisión del ACH Payment

TPAa (100%)	TPAs (TPPs %)	Decremento (%)
91 min.	49 min. (53.85%)	42 min. (46.15%)

Asimismo, el análisis de resultados para el indicador de valoración promedio para la gestión de préstamos de acuerdo a los resultados de las encuestas (en base a puntaje 5) se obtiene que la diferencia entre la percepción de los usuarios finales (empleados) con el uso del sistema de escritorio era baja con un promedio de 2.56, en comparación de los resultados obtenidos después del uso del sistema web propuesto con 4.76, lo que significa que la satisfacción de los usuarios finales incrementó en un 85.94%. Esto se interpreta que el uso del sistema de escritorio era lento y el tiempo que los empleados usaban para sus tareas diarias era mucho a comparación del sistema web que les facilita el desarrollo de estas en un menor tiempo, aumentando su productividad.

Tabla 43: Discusión Indicador Valoración promedio de la gestión de préstamos

VPa (100%)	VPs (TPPs %)	Incremento (%)
2.56	4.76 (185.94%)	2.20 (85.94%)

Finalmente, en investigaciones como el “Análisis y desarrollo de un sistema de información para el cálculo y administración de créditos, en el sector financiero” presentado por Fabián Cuervo, donde afirmo que las empresas financieras deberían utilizar un software al nivel que lo exige el mercado que día a día sigue siendo más competitivo, logrando implementar un sistema web que suple las necesidades requeridas por este sector, logrando hacer más óptimo sus procesos organizacionales de las empresas orientadas a este rubro, tal y como se logró también en esta investigación. Ambas investigaciones utilizaron el patrón de diseño MVC para su desarrollo.

De igual manera, la investigación “Análisis, diseño e implementación de un sistema de recaudación de deudas”, desarrollada por César Vásquez Flores nos expone como ofrecer un servicio de calidad mantiene a los clientes fidelizados, mejorar su flujo de trabajo y aumentar su nivel de productiva. Y esto lo logró mediante la implementación de un sistema de información que mejoró el proceso de cobranzas, de tal manera como sucedió en este trabajo de investigación.

CONCLUSIONES

Se concluye que, se logró mejorar la gestión de préstamos de la empresa financiera a través de la implementación de un sistema web, cumpliendo los siguientes objetivos específicos:

- Se logró determinar el tiempo promedio utilizado en la emisión de declaraciones de pago, reduciéndolo en un 25.58% con el sistema web propuesto.
- Se logró determinar el tiempo promedio utilizado en la emisión del ACH Payment, reduciéndolo en un 46.15% con el sistema web propuesto.
- Se logró determinar el tiempo promedio utilizado en la realización de un préstamo, reduciéndolo en un 17.33% con el sistema web propuesto.
- Se logró determinar el tiempo promedio utilizado en la realización del pago de letras de un reduciéndolo reduciendo en un 21.43% con el sistema web propuesto.
- Se logró determinar la valoración promedio de la gestión de préstamos, en un incremento del 85.94% con el sistema web propuesto.

RECOMENDACIONES

- Se podría trabajar en una versión móvil para facilitar la visión de reportes, control de clientes, préstamos, etc. Por ejemplo, muchos de los empleados de la empresa financiera y gerente suelen ir a las diferentes sedes que tiene la empresa financiera en los diferentes estados de los Estados Unidos. Sería una herramienta óptima para no desconectarse del negocio.
- Actualmente el sistema web solo es usado por los empleados y gerente de la empresa financiera. Solo cuenta con una página informativa para que los prestamistas o prestatarios puedan enviar sus dudas o solicitudes. Una idea innovadora, sería crear diferentes tipos de cuenta para los clientes externos puedan ingresar al sistema para ver sus estados de cuenta y consultar la información necesaria.

REFERENCIAS

- BBC. (21 de Octubre de 2016). *Qué son los préstamos “gota a gota” que grupos criminales de Colombia exportan al resto de América Latina*. Obtenido de <http://www.bbc.com/mundo/noticias-america-latina-37708989>
- Beck, K. (2000). *Extreme Programming Explained: Embrace Change*. Obtenido de <https://books.google.com.pe/books?hl=es&lr=&id=G8EL4H4vf7UC&oi=fnd&pg=PR13&dq=extreme+programming&ots=j9wHAtfQAs&sig=exVRdIBxRPXsO9SICDeFr1uwLDg#v=onepage&q=extreme%20programming&f=false>
- Bravo, J. (25 de Febrero de 2013). *Gestión de procesos*. Obtenido de http://www.evolucion.cl/resumenes/Resumen_libro_Gestion_de_procesos_5_edicion_JBC_2013.pdf
- Calapiña, L. (Agosto de 2016). *Sistema Web para la gestión de procesos de crédito y recuperación de cartera en la cooperativa de ahorro y crédito “Santa Lucía” Ltda. De la ciudad de Ambato*. Obtenido de http://repo.uta.edu.ec/bitstream/123456789/23658/1/Tesis_t1147si.pdf
- Cordero, M. (05 de Junio de 2017). Obtenido de <https://www.univision.com/noticias/citylab-vida-urbana/los-prestamistas-de-barrio-el-costoso-ultimo-recurso-de-los-latinos>
- Cuervo, F. (2008). *Análisis y desarrollo de un sistema de información para el cálculo y administración de créditos, en el sector financiero*. Obtenido de http://www.konradlorenz.edu.co/images/stories/suma_digital_sistemas/2009_01/ProyectedeGrado_Fabian%20Cuervo_2008.pdf
- Culturación. (2018). *XML*. Obtenido de <http://culturacion.com/que-es-xml/>
- Dimes, T. (2015). *Conceptos básicos de Scrum: Desarrollo de software Agile y manejo de proyectos Agile*. Obtenido de <https://books.google.com.pe/books?id=ETuXBgAAQBAJ&printsec=frontcover&hl=es#v=onepage&q&f=false>
- EcuRed. (02 de Julio de 2018). *Crystal Reports*. Obtenido de https://www.ecured.cu/Crystal_Reports
- Hernández, E., & Oviedo, F. (2016). *Mercado del crédito informal en Colombia: una aproximación empírica*. Colombia.
- Heys, B. (2011). *Branching for Scrum*. Obtenido de <https://blogs.msdn.microsoft.com/billheys/2011/01/18/branching-for-scrum/>
- IBM. (2016). *¿Qué es XML?* Obtenido de https://www.ibm.com/support/knowledgecenter/es/SSEPGG_8.2.0/com.ibm.db2.ii.doc/opt/c0007799.htm
- Investopedia. (2015). *Automated Clearing House - ACH*. Obtenido de <https://www.investopedia.com/terms/a/ach.asp>
- Investopedia. (2015). *Loan Servicing*. Obtenido de https://www.investopedia.com/terms/l/loan_servicing.asp
- Investopedia. (2016). *Billing Statement*. Obtenido de <https://www.investopedia.com/terms/b/billing-statement.asp>
- Joskowicz, J. (10 de Febrero de 2008). *Reglas y Prácticas en eXtreme Programming*. Obtenido de <https://iie.fing.edu.uy/~josej/docs/XP%20-%20Jose%20Joskowicz.pdf>
- Maldonado, N. (20 de Febrero de 2015). *Préstamos ilegales, negocio de muchos*. Obtenido de <http://elsiglo.com.pa/economia/prestamos-ilegalesnegocio-muchos/23845039>
- Microsoft. (03 de Diciembre de 2010). *Visión general de ASP.NET MVC*. Obtenido de <https://docs.microsoft.com/en-us/aspnet/overview>
- Microsoft. (10 de Junio de 2014). *Introduction to SignalR*. Obtenido de <https://docs.microsoft.com/en-us/aspnet/signalr/overview/getting-started/introduction-to-signalr>

- Microsoft. (2015). *Visual Studio 2015*. Obtenido de <https://www.msn.com/es-cl/noticias/microsoftstore/%C2%BFqu%C3%A9-es-y-para-qu%C3%A9-sirve-visual-studio-2017/ar-AAAnLZL9>
- Microsoft. (2017). *ADO.NET Entity Framework*. Obtenido de [https://msdn.microsoft.com/es-es/library/bb399567\(v=vs.100\).aspx](https://msdn.microsoft.com/es-es/library/bb399567(v=vs.100).aspx)
- Microsoft. (21 de Julio de 2017). *What's new in SQL Server 2016*. Obtenido de <https://docs.microsoft.com/en-us/sql/sql-server/what-s-new-in-sql-server-2016?view=sql-server-2017>
- Microsoft. (25 de Abril de 2018). *Introduction to ASP.NET Core SignalR*. Obtenido de <https://docs.microsoft.com/en-us/aspnet/core/signalr/introduction?view=aspnetcore-2.1>
- Pillajo, D., & Tipán, D. (2015). *Desarrollo del sistema de planificación de créditos para la cooperativa de ahorro y crédito Santa Ana de Nayón*. Obtenido de <http://bibdigital.epn.edu.ec/bitstream/15000/10719/1/CD-6298.pdf>
- The blokehead. (2016). *Scrum, guía definitiva de prácticas ágiles esenciales de Scrum*. Obtenido de https://books.google.com.pe/books?id=T24eDQAAQBAJ&printsec=frontcover&dq=scrum&hl=es-419&sa=X&ved=0ahUKEwjo3bq83_HbAhVG7VMKHfz2BRkQ6AEIMDAB#v=onepage&q=scrum&f=false
- Vásquez, C. (Noviembre de 2013). *Análisis, diseño e implementación de un sistema de recaudación de deudas*. Obtenido de http://tesis.pucp.edu.pe/repositorio/bitstream/handle/123456789/5213/VASQUEZ_CESAR_ANALISIS_SISTEMA_RECAUDACION_DEUDAS.pdf?sequence=1
- Wingu. (Agosto de 2016). *Manual de metodologías ágiles*. Obtenido de https://www.winguweb.org/system/files/biblioteca/manual_de_metologias_agiles_final.pdf

ANEXOS

ANEXO 1

Cuestionario para la recolección de datos

1. ¿Cuáles son sus procesos más críticos y cuáles consumen la mayor cantidad de recursos?

El proceso más crítico de nuestro negocio es el Servicio de Préstamo, que incluye tareas difíciles como:

- Entrega de declaraciones mensuales de pago a tiempo a cada prestatario de un préstamo.
- Envío de cartas y declaraciones
- Hacer pagos diarios a los inversores a través de ACH o cheques.
- Tener todos los datos en 1 sola base de datos para una respuesta más rápida.

2. ¿Qué problemas tienen los procesos mencionados anteriormente?

El procesamiento de las declaraciones de pago mensuales es una tarea difícil debido a la fecha de pago establecida en la mayoría de los préstamos. Según el cumplimiento del gobierno, debemos imprimir y enviar por correo todas las declaraciones de pago 15 días antes de la fecha de vencimiento. La mayoría de los préstamos tienen una fecha de vencimiento del 30 o el 15 de un mes. Por lo tanto, necesitamos procesar decenas de miles de declaraciones de pago 2 veces al mes. Es tan intensivo en recursos que congela o ralentiza a otros usuarios del sistema.

La integración con Imail Tracking es una ventaja clave para nuestro negocio, enviar cartas y declaraciones sobre una base diaria consume mucho tiempo, al integrar con Imail ahora no necesitamos imprimir, doblar, almacenar y enviar la correspondencia a cada prestatario o prestador. Hacer pagos diarios a los inversores a través de ACH o cheques es muy difícil porque necesitamos verificar si todos los cheques pendientes han sido aprobados por los bancos, también se lee a través de la base de datos si hay algún cargo pendiente que debemos deducir o si hay alguno Pagos adicionales han llegado. Todo esto provoca retrasos en el software y, a veces se congela.

Tener todos los datos en 1 base de datos única para que los informes maestros puedan ejecutarse al mismo tiempo es muy crítico, antes de tener a cada compañía en su propia base de datos y fusionar o combinar datos era muy difícil, y algunas veces incluso imposible.

3. ¿Cuáles son los recursos consumidos por estos procesos?

Base de datos, memoria Ram, hojas de papel, personal de entrega y tinta de impresora.

4. ¿Cuánto tiempo lleva completar estos procesos?

Entre 4 a 8 horas.

5. ¿Qué limitaciones tiene el sistema ABS con respecto a la usabilidad?

Tiene 1 base de datos por departamento, entonces de ninguna manera puede procesar decenas de miles de declaraciones al mismo tiempo. Además, no hay soporte continuo para mantener el software actualizado con la nueva regulación.

ANEXO 2

ISO 9126 para la calidad del software – Métricas

Tabla 44: Dimensión: Usabilidad - Indicador: Atracción

Nombre	Buen diseño de las interfaces
Propósito	Determinar si la interfaz cumple con patrones de diseño para que sea bien vista por los usuarios.
Método de aplicación	Contar los elementos y controles que hacen siguen los patrones de diseño (percepción, cognición, color, memoria, metáforas bien elaboradas, etc.)
Medición, fórmula	$X = A/B$ A = número de elementos que siguen los patrones de diseño B = número total de elementos en la interfaz
Interpretación	$0 < X \leq 1$ Entre más cercano a 1, más completa
Tipo de escala	absoluta
Tipo de medida	$X = \text{entero}/\text{entero}$ A = números enteros mayores que 0 B = números enteros mayores que 0
Fuente de medición	Especificación de requisitos, Diseño, Informe de revisión y Patrones de diseño establecidos

Tabla 45: Dimensión: Usabilidad - Indicador: Entendimiento

Nombre	Comprensión de las funciones del sistema
Propósito	Medir el número de funciones que el usuario es capaz de realizar al usar el software.
Método de aplicación	Contar las funciones que se utilizaron con éxito durante las pruebas y comparar con el número de funciones descritas en la especificación de requisitos. La resta de ambos nos dará el número de funciones que no se entendieron.
Medición, fórmula	$X = A/B$ A = número de funciones comprendidas con éxito B = número total de funciones
Interpretación	$0 < X \leq 1$ Entre más cercano a 1, más completa
Tipo de escala	absoluta
Tipo de medida	$X = A/B$ A = números enteros mayores que 0 B = números enteros mayores que 0

Fuente de medición	Especificación de requisitos, Diseño, Código fuente, Pruebas del software e Informe de revisión
---------------------------	---

Tabla 46: Dimensión: Funcionalidad - Indicador: Adecuidad

Nombre	Compleitud de implementación funcional
Propósito	Mide el nivel de completitud de la implementación funcional
Método de aplicación	Contar las funciones no implementadas durante el uso del sistema y comparar con el número de funciones descritas en la especificación de requisitos.
Medición, fórmula	$X = A/B$ A = número de funciones faltantes detectadas en la evaluación B = Número de funciones descritas en la especificación de requisitos
Interpretación	$0 < X \leq 1$ Entre más cercano a 1, más completa
Tipo de escala	absoluta
Tipo de medida	$X = A/B$ A = números enteros mayores que 0 B = números enteros mayores que 0
Fuente de medición	Especificación de requisitos, diseño, código fuente, informe de revisión

Tabla 47: Dimensión: Confiabilidad - Indicador: Madurez

Nombre	Suficiencia de pruebas
Propósito	Determinar el porcentaje de casos de prueba que son cubiertos.
Método de aplicación	Contar las pruebas que se realizaron con éxito y compararlas con el número total de pruebas que se realizaron. La medición se hará una vez realizado la capacitación al personal de la empresa financiera.
Medición, fórmula	$X = A/B$ A = Número de pruebas con éxito. B = Número de total de pruebas.
Interpretación	$0 < X \leq 1$ Entre más cercano a 1, más completa
Tipo de escala	absoluta
Tipo de medida	$X = A/B$ A = números enteros mayores que 0 B = números enteros mayores que 0
Fuente de medición	Código Fuente y pruebas del software

ANEXO 3

Encuesta para obtener valoración de cómo se realiza la gestión de préstamos (Utilizada para evaluar el sistema de escritorio y el Sistema web propuesto)

Responder en el espacio en blanco con un número entre 1 a 5, donde cada número significa lo siguiente:

- (1) Muy malo
- (2) Malo
- (3) Normal
- (4) Bueno
- (5) Muy bueno

Cuestionario	Respuesta
¿Qué tan clara es la estructura y organización de la interfaz de usuario?	
¿Qué tan entendibles son los mensajes informativos que presenta el sistema?	
¿Es adecuado el tiempo que demora en cumplir con sus tareas diarias con el uso del sistema?	
¿Qué tan eficiente es la velocidad del sistema?	
¿Qué tanto considera que el sistema actual facilita el cumplimiento de sus tareas?	

ANEXO 4

Declaración de pago

Account Information			
Outstanding Principal Balance:	\$55,865.13	Prepayment Penalty:	NO
Deferred Amounts:	\$0.00	Loan Maturity Date:	11/01/1999
Current Interest Rate:	13.000%	Payment Type:	Other/ Other
Next Interest Rate Change Date:	-	Next Due Date:	11/01/1999
Remaining Loan Term (Months):	0		

Account Number:	003579
Payment Due Date:	08/01/2018
AMOUNT DUE:	\$82,570.16
If Payment is not received by 08/13/2018, \$0.00 Late Fee will be charged	

Explanation of Payment Due	
Principal:	\$55,865.13
Est Interest:	\$616.81
Other Amounts Due:	\$0.00
Escrow (Taxes and/or Insurance):	\$0.00
Current Payment:	\$56,481.94
Total Fees and Charges:	\$19,442.56
Overdue Payments:	\$6,645.65
Total Amount Due:	\$82,570.16

Transaction Activity Since (05/16/2018 - 07/16/2018)							
Date	Description	Charges	Payments	Date	Description	Charges	Payments

Past Payments Breakdown		
	Paid Since Last Statement	Paid year to Date
Principal:	\$0.00	\$1,562.53
Interest:	\$0.00	\$2,463.37
Escrow (Taxes and/or Insurance):	\$0.00	\$0.00
Fees:	\$0.00	\$200.66
*Partial Payment (Unapplied):	\$0.00	\$0.00
Others:	\$0.00	\$60.00
TOTAL:	\$0.00	\$4,286.56

Delinquency Notice
You are late on your mortgage payments. Failure to bring your loan current may result in fees and foreclosure - the loss of your home. As of 07/16/2018, you are 318 days delinquent on your mortgage loan.
Recent Payment Account History:
* Payment 03/01/2018 - Unpaid balance of \$/23,660.76
* Payment 04/01/2018 - Unpaid balance of \$/616.81
* Payment 05/01/2018 - Unpaid balance of \$/596.92
* Payment 06/01/2018 - Unpaid balance of \$/616.81
* Payment 07/01/2018 - Unpaid balance of \$/596.92
* Current Payment 08/01/2018: \$/56,481.94
Total: \$82,570.16 due. You must pay this amount to keep your loan current.

This loan has Matured and is all Due and Payable. Please contact this office for full Payoff amounts, as it may be different from this statement

----- PLEASE DETACH THE BOTTOM PORTION OF THIS STATEMENT, RETURN IT WITH YOUR PAYMENT AND RETAIN THE TOP PORTION FOR YOUR RECORDS -----

Amount Due	
Account	003579
Current Payment Due By 08/01/2018	\$56,481.94
Total Payment(s) Due	\$82,570.16
\$0.00 Late Fee will be charged after 08/13/2018	
Additional Principal:	_____
Additional Escrow:	_____
Total Amount Enclosed:	_____

Check box if your address or phone number has changed
And write your new information on the bottom portion of this statement.

Property Address: 10231 A & 10231 B Long Beach Blvd.
Lynwood, CA 90626

Figura 33: Declaración de pago de un prestamista

