



**UNIVERSIDAD NACIONAL PEDRO RUIZ GALLO**

**ESCUELA DE POSTGRADO**

**MAESTRÍA EN CIENCIAS CON MENCIÓN EN  
INFORMÁTICA Y SISTEMAS**

**SOFTWARE PARA LA GESTIÓN DE CRÉDITOS  
EN LA COOPERATIVA DE AHORRO Y CRÉDITO  
CREDISERVIS LTDA**

**TESIS PRESENTADA PARA OPTAR EL GRADO  
ACADÉMICO DE MAESTRO EN CIENCIAS CON  
MENCIÓN EN INFORMÁTICA Y SISTEMAS**

**PRESENTADO POR:**

**ING. ELIO ENALDO FLOREANO QUIPUSCOA**

**MG ING. REGIS JORGE ALBERTO DIAZ PLAZA**

**ASESOR**

**LAMBAYEQUE – PERÚ – 2017**



# **Software para la gestión de créditos en la Cooperativa de Ahorro y Crédito CREDISERVIS LTDA.**

---

**ING. ELIO ENALDO FLOREANO QUIPUSCOA**  
**AUTOR**

**MA. ING. REGIS JORGE ALBERTO DIAZ PLAZA**  
**ASESOR**

**Presentada a la Escuela de Postgrado de la Universidad Nacional Pedro Ruiz Gallo para optar el grado de MAESTRO EN CIENCIAS CON MENCIÓN EN INFORMÁTICA Y SISTEMAS**

**APROBADO POR:**

**DR. GILLERMO GUERRERO OJEDA**

---

**PRESIDENTE DEL JURADO**

**MG. ERNESTO CELI AREVALO**

---

**SECRETARIO DEL JURADO**

**MG. SANTOS GUEVARA QUILICHE**

---

**VOCAL DEL JURADO**

**Julio, 2017**

## TABLA DE CONTENIDOS

TABLA DE CONTENIDOS.....	2
RESUMEN.....	3
ABSTRACT.....	4
INTRODUCCION .....	5
CAP. I ANÁLISIS DEL OBJETO DE ESTUDIO.....	6
1.1 Ubicación .....	6
1.2 Surgimiento del problema .....	6
1.3 Características y formulación del problema.....	7
1.5 Justificación e importancia de la investigación.....	8
1.5 Metodología .....	8
CAP. II MARCO TEORICO .....	10
2.1 Antecedentes .....	10
2.2 Teorías.....	11
CAP III. MODELO CONCEPTUAL Y DESARROLLO .....	17
3.1 Modelo conceptual .....	17
3.2 Desarrollo del software .....	19
3.2.1 Modelado del Negocio .....	19
3.2.2 Requerimientos Funcionales .....	21
3.2.3 Análisis.....	23
3.2.4 Diseño .....	26
3.2.5 Implementación.....	33
CAP. IV ANÁLISIS Y DISCUSIÓN DE LOS RESULTADOS.....	35
CONCLUSIONES .....	41
RECOMENDACIONES .....	43

## RESUMEN

La investigación analiza el problema de mejorar el proceso de gestión de créditos, tiene como objetivo mejorar el proceso de gestión de créditos con la implementación de un software.

El modelo conceptual que se presenta permite verificar el cumplimiento de los objetivos, esta propuesta se presenta a partir de las teorías relacionadas con desarrollo de software.

El alcance logrado es que el software mejora la gestión de créditos, la dimensión funcionalidad indicaron que se aceptaron todos los requerimientos funcionales.

La investigación contribuye a proponer un modelo conceptual para evaluar software para el soporte de procesos, a través de los requerimientos funcionales y facilidad uso.

La investigación no evalúa la correlación o relación entre el software y el proceso, pero plantea un inicio para futuras investigaciones.

## **ABSTRACT**

The research analyzes the problem of improving the credit management process, aims to improve the credit management process with the implementation of software.

The conceptual model presented allows to verify the fulfillment of the objectives, this proposal is presented from the theories related to software development.

The achieved reach is that the software improves the credit management, the functionality dimension indicated that all the functional requirements were accepted

The research contributes to propose a conceptual model to evaluate software for the support of processes, through the functional requirements and ease of use.

The research does not evaluate the correlation or relationship between the software and the process, but it poses a start for future research.

## INTRODUCCION

La gestión de créditos está influenciada directamente por las tecnologías de información y sistemas de información. El trabajo de investigación plantea medir la mejora de un proceso de gestión de crédito por el desarrollo de un software que brinde el soporte tecnológico. El software es dimensionado en base la funcionalidad y facilidad de uso y el proceso de crédito en base a tiempo.

En el capítulo I: Análisis del objeto de estudio, se presenta la caracterización del problema, logrando diseñar una investigación bajo un tipo no experimental, aplicada, descriptiva.

En el capítulo II: Marco teórico, se presenta las investigaciones relacionadas sobre desarrollo de software financiero, teoría que sirvió para diseñar el proceso de desarrollo de software.

En el capítulo III: Modelo Conceptual y Desarrollo, se presenta el diseño conceptual de la investigación diseñado en base a los antecedentes y teorías, y el desarrollo de cada una de las actividades del desarrollo del software

En el capítulo IV: Análisis y discusión de los resultados, se justifica el cumplimiento de cada uno de los objetivos específicos y el objetivo general.

La investigación presenta las conclusiones sobre el modelo conceptual y recomendaciones sobre futuras investigaciones.

## **CAP. I ANÁLISIS DEL OBJETO DE ESTUDIO**

### **1.1 Ubicación**

La investigación tiene como lugar de desarrollo en la Cooperativa de Ahorro y Crédito CREDISERVIS LTDA. Distrito de Chiclayo, Provincia de Chiclayo, Departamento de Lambayeque

### **1.2 Surgimiento del problema**

La Cooperativa de Ahorro y Crédito CREDISERVIS promueve el desarrollo económico y social de sus socios a través de la promoción del ahorro y otorgamiento de crédito para lo cual debe operar en tiempo oportuno y garantizar la transparencia de la gestión de los aportes de los socios.

Estas operaciones se desarrollan a través del canal de otorgamiento de crédito de forma manual, convirtiéndose en una barrera que genera retrasos y es observado como informalidad ante los socios que desean ahorrar. Por otra parte los socios que solicitan crédito se ha incrementado logrando sobrepasar el control minucioso que la cooperativa debe tener.

El estado a promueve el cooperativismo y ha establecido un marco jurídico a través de la Ley de Cooperativas N° 15260 (Presidencia de la República, 1985) y perfeccionada a través de Decreto Legislativo N° 85, donde declara de necesidad nacional y utilidad pública la promoción y la protección del Cooperativismo, como un sistema eficaz para contribuir el desarrollo económico.

A partir del año 1959 empieza el funcionamiento al FENACREP, Federación Nacional de Cooperativas de Ahorro y Crédito y se le encarga desde el año 1993 la supervisión de las Cooperativas de Ahorro y Crédito (COOPAC), poniéndola bajo la supervisión de la Superintendencia de Banca, Seguros y AFP (SBS) dispuesta en la Ley N° 26702 (Presidencia de la República, 2011). La adhesión a la FENACREP de las cooperativas de ahorro y crédito y debido

a las exigencias de cumplimiento de políticas de seguridad financiera garantiza la transparencia y seguridad para los socios y estado, esto obliga al cumplimiento de estas políticas entre ellas las referidas a las operaciones de ahorro y crédito que realice deban utilizar en forma obligatoria tecnologías de información y sistemas de información

La cooperativa ha demostrado un crecimiento de cartera, que trae buenos resultados pero que a la vez exigencias por la transparencia y seguridad de los ahorros de los socios, por esta razón el Consejo de Administración ha decidido realizar el esfuerzo para alinear los procesos de acuerdo a los requerimientos de la FENACREP y poder lograr pasar con éxito la supervisión cuando se realice.

### **1.3 Características y formulación del problema**

Se ha caracterizado la problemática de la siguiente manera:

Se identificaron deficiencias en el registro de ahorros de los socios.

Se identificaron deficiencias en el registro de las solicitudes de crédito de los socios.

Se identificaron demoras en realizar los informes de cierre mensual.

Se posibilita la transparencia en la ejecución del proceso de otorgamiento de créditos.

Estas razones son argumentos para tomar como desafío el desarrollo de un proyecto que permita mejorar el proceso de otorgamiento créditos.

### **1.4 Formulación del problema**

En el presente proyecto de tesis se plantea el problema de la siguiente manera:

¿El proceso de gestión de créditos mejorará con el apoyo de un software para su gestión?

## **1.5 Justificación e importancia de la investigación**

### Conveniencia

Es conveniente para la Cooperativa de Ahorro y Crédito porque le permitirá mejorar el proceso de otorgamiento de crédito.

### Relevancia Social

Los beneficiados de este proyecto son la cooperativa porque mejorará su imagen a través de un modelo de gestión de crédito ágil y transparente, el socio porque le permitirá estar informado de las operaciones realizadas por la cooperativa, el estado porque mantiene el control sobre el sistema financiero.

### Implicancia Práctica

Aumentará la eficiencia a través de la reducción de los tiempos de elaboración de informes sobre la gestión de créditos y la cooperativa.

### Viabilidad

Existen recursos financieros, humanos y materiales para la ejecución del proyecto.

## **1.5 Metodología**

### **Objetivos**

Para el desarrollo de la investigación se establecieron los objetivos siguientes:

#### **Objetivo general**

Mejorar el proceso de créditos con la implementación de un software de gestión de crédito en la cooperativa de ahorro y crédito CREDISERVIS LTDA.

#### **Objetivo específicos**

Determinar la funcionalidad del software de gestión de crédito

Determinar la facilidad de uso del software de gestión de crédito.

### **Del diseño de la investigación**

Para la presente investigación no se ha encontrado fuentes de invalidación interna para descartar el diseño de la investigación.

Considerando a las diferentes perspectivas de la investigación se tipificó a la investigación según:

El tipo de estudio: se procedió como una investigación no experimental, no se interviene en el objeto de estudio.

El fin que se persiguió: se procedió como investigación aplicada, porque está orientada a la solución de un problema práctico.

El alcance de la investigación: se procedió como una investigación descriptiva, para determinar la mejora del proceso de gestión de créditos utilizando un software como soporte tecnológico.

## **CAP. II MARCO TEORICO**

### **2.1 Antecedentes**

En la investigación exploratoria se identificó las siguientes investigaciones:

En la tesis “Diseño e implementación de un sistema de control del proceso de crédito y cobranza por medio de indicadores de gestión para una sociedad financiera ubicada en la ciudad de Guayaquil” (Guijarro Riera & Guijarro Riera, 2010) fue aplicada en una Sociedad Financiera autorizada a proporcionar productos y servicios financieros, que la ley de Instituciones Financieras determina. La organización decidió considerar la aplicación de un Sistema de Control de Procesos mediante Indicadores de Gestión, la entidad realiza reportes gerenciales en forma manual en Excel y se desea alcanzar una mayor eficacia y efectividad con la implementación del sistema que utilizará mediciones e indicadores que describa el desarrollo y progreso de los procesos seleccionados para análisis. Se aprecia la recolección de funciones del sistema, el uso de técnicas de diseño de base de datos y diseño de las interfaces.

Sobre el análisis de proceso de cobranza existen investigaciones como “Análisis, diseño e implementación del sistema para la gestión de cobranzas en la Cooperativa de Ahorro y Crédito Alfonso Jaramillo León” (Carrasco Espinoza, 2012), la finalidad del trabajo ha sido situar a la Cooperativa de Ahorro y Crédito Alfonso Jaramillo León, dentro de un escenario real para la toma de decisiones estratégicas, considerando su desenvolvimiento interno y entorno, en relación al crédito y posterior recuperación. Siendo la Cooperativa una entidad financiera, la cobranza constituye parte primordial de su diario accionar y la gestión desarrollada en torno a este sistema se vuelve fundamental para lo cual ha sido preciso efectuar una analogía con una entidad bancaria en cuanto al proceso desarrollado por Cobranzas y a partir de ello se estructura la propuesta para la Gestión de Recuperación de Cartera en la Cooperativa, que permita cumplir con una serie de objetivos primordiales. De esta investigación observamos que los procesos de crédito y cobranza son análogos a los de los sistemas bancarios y pueden ser adaptados a los procesos de la cooperativa.

La investigación “Desarrollo de una infraestructura de software para realizar pruebas automatizadas de sistemas de información desarrollados en lenguaje cobol en el contexto bancario” (Tasato Cánepa, 2013) permite observar la secuencia de desarrollo que implica la identificación de requerimientos funcionales y no funcionales, análisis de la solución, diagrama de paquetes, especificaciones de casos de uso, diseño, arquitectura del software y construcción.

La investigación “Análisis, diseño e implementación de un sistema de recaudación de deudas” (Vásquez Flores, 2013), presenta la secuencia de desarrollo con la identificación de requisitos funcionales y no funcionales, catálogo de actores, casos de uso, arquitectura de la solución: vista lógica, de despliegue, diagrama de clases, diagrama de base de datos, diseño de interfaces gráficas, construcción con patrones y pruebas.

Por lo tanto, se aprecia el avance de las investigaciones sobre el estudio de mejora de la eficiencia de los procesos en entidades financieras y la viabilidad de desarrollo de software.

## **2.2 Teorías**

En la investigación “Selección de un enfoque de Desarrollo” presenta descriptivamente las ventajas y desventajas de diversos modelos de ciclo de vida como cascada, prototipos, incremental, espiral, desarrollo rápido de aplicaciones y tres entornos de trabajo lineal, iterativo, combinación lineal e iterativo (Department of Health and Human Services, 2008) aunque no determina cual es el ciclo de vida apropiado para el desarrollo de sistemas, se resume en la siguiente tabla.

Tabla 001  
Modelo de ciclo de vida y entorno de trabajo

Modelo de Ciclo de Vida	Entorno de trabajo
Cascada	Lineal
Prototipo	Iterativo
Incremental	Lineal – iterativo
Espiral	Lineal – iterativo
Desarrollo rápido de aplicaciones (RAD)	Iterativo

Fuente: Selección de un enfoque de Desarrollo (Department of Health and Human Services, 2008)

La investigación “Comparative study on software development methodologies”, se centra en el estado actual de los conocimientos en el campo de desarrollo de software metodologías. Su objetivo es sentar las bases para la formalización de un desarrollo de software metodología dedicada a la innovación orientada proyectos de TI”. “...Metodologías de desarrollo de software siguen uno de dos caminos: peso pesado o ligero. Metodologías de peso pesado se derivan del modelo de cascada y hace hincapié en la planificación detallada, especificaciones exhaustivas y diseño detallado de aplicaciones. Metodologías ligeras se derivan del modelo Ágil y promover software de trabajo, las personas y las interacciones, aceptación de cambios en los requisitos y sugerencias de los usuarios” (Liviu, 2014)

Tabla 002  
Características de las metodologías de desarrollo de software

METODOLOGÍA	CARACTERÍSTICAS	FORTALEZAS	DEBILIDADES
<b>CASCADA</b>	<ul style="list-style-type: none"> <li>- Documentación comprensiva.</li> <li>- Planificación meticulosa.</li> <li>- Proceso lineal secuencial.</li> <li>- Cada fase tiene sus propias entregas.</li> </ul>	<ul style="list-style-type: none"> <li>- Fácil de dirigir.</li> <li>- Fácil de entender para el propietario del proyecto y el equipo.</li> </ul>	<ul style="list-style-type: none"> <li>- Se entrega el código de trabajo tarde en el proyecto.</li> <li>- No cooperar con cambios en los requisitos.</li> <li>- Baja tolerancia para el diseño y los errores de planificación.</li> </ul>
<b>PROTOTIPO</b>	<ul style="list-style-type: none"> <li>- Construir una o más versiones demo del software de producto.</li> <li>- Propietario del proyecto está activamente comprometido.</li> <li>- Los prototipos están destinados a ser descartado.</li> <li>- Código de escritura es valorado fuera de las especificaciones de escritura.</li> </ul>	<ul style="list-style-type: none"> <li>- Identificación precisa de requerimientos de aplicación.</li> <li>- Retroalimentación temprana del propietario del proyecto.</li> <li>- La mejora de la experiencia del usuario.</li> <li>- La identificación temprana de falta o redundante.</li> <li>- Funcionalidad.</li> </ul>	<ul style="list-style-type: none"> <li>- Lleva al innecesario aumento de la complejidad de la aplicación.</li> <li>- Aumenta el esfuerzo de la programación.</li> <li>- Los costes generados por la construcción del prototipo.</li> </ul>

<b>ITERATIVO Y GRADUAL</b>	<ul style="list-style-type: none"> <li>- Construir un modelo inicial que se extienda en sucesivas iteraciones.</li> <li>- Énfasis en el diseño antes que en la documentación.</li> <li>- El propietario del proyecto está activamente involucrado.</li> </ul>	<ul style="list-style-type: none"> <li>- Continúa retroalimentación del propietario del proyecto.</li> <li>- Múltiples revisiones en la aplicación entera y en funcionalidades específicas.</li> <li>- El código de trabajo se entrega rápido en el proyecto.</li> </ul>	<ul style="list-style-type: none"> <li>- Cada iteración es una estructura rígida que se parece a un proyecto de cascada en menor escala.</li> </ul>
<b>ESPIRAL</b>	<ul style="list-style-type: none"> <li>- Enfoque en los objetivos, alternativas y restricciones.</li> <li>- Tiene 4 fases mayores: planeamiento, análisis de riesgo, desarrollo y evaluación.</li> <li>- Énfasis en el análisis de riesgo.</li> <li>- Evalúa las alternativas múltiples antes de proceder a la fase de la planificación.</li> </ul>	<ul style="list-style-type: none"> <li>- El código de trabajo se entrega rápido en el proyecto.</li> <li>- Minimiza el riesgo.</li> <li>- Sólida documentación.</li> </ul>	<ul style="list-style-type: none"> <li>- Manejo de los costos generados por el riesgo.</li> <li>- Dependencia en la exactitud del análisis riesgo.</li> </ul>
<b>RÁPIDO DESARROLLO DE LA APLICACIÓN</b>	<ul style="list-style-type: none"> <li>- Menos énfasis en planificar las tareas y más enfoque en el desarrollo.</li> <li>- Cercanía del Timebox.</li> </ul>	<ul style="list-style-type: none"> <li>- Las aplicaciones son desarrolladas rápidamente.</li> <li>- El código puede reusarse fácilmente.</li> </ul>	<ul style="list-style-type: none"> <li>- Pobre documentación.</li> <li>- Altos costos de desarrollo.</li> <li>- Problemas de integración de código.</li> <li>- La aplicación tiene que ser irrupida en módulos.</li> </ul>
<b>PROGRAMACIÓN EXTREMA</b>	<ul style="list-style-type: none"> <li>- La programación par.</li> <li>- La comprobación de la unidad.</li> <li>- Lanzamientos consecutivos rápidos.</li> <li>- Propiedad colectiva.</li> <li>- Propietario del proyecto in situ.</li> <li>- Área de trabajo abierta.</li> <li>- El dueño del proyecto decide la prioridad de las tareas.</li> </ul>	<ul style="list-style-type: none"> <li>- La aplicación consigue estar rápidamente en el entorno de producción.</li> <li>- Frecuentes actualizaciones del código de trabajo.</li> <li>- Se reduce el número de errores.</li> <li>- La integración del código Smooth.</li> <li>- Continúa retroalimentación del dueño del proyecto.</li> </ul>	<ul style="list-style-type: none"> <li>- Falte de documentación.</li> <li>- Renuencia de los desarrolladores para programar de a par.</li> <li>- Renuencia de los desarrolladores para primero escribir pruebas y codificar después.</li> <li>- Requiere reuniones frecuentes.</li> <li>- Falte de compromiso hacia un bien definido liderazgo de producto para proyectar la renuencia del propietario.</li> </ul>
<b>MODELO-V</b>	<ul style="list-style-type: none"> <li>- Introduce la comprobación en cada fase de desarrollo.</li> <li>- Resalta la importancia de mantenimiento.</li> </ul>	<ul style="list-style-type: none"> <li>- Bajo rango de errores.</li> <li>- Fácil de entender y usar.</li> </ul>	<ul style="list-style-type: none"> <li>- Vulnerable al crecimiento excesivo.</li> <li>- Confiar excesivamente en el conjunto de especificaciones iniciales.</li> </ul>
<b>SCRUM</b>	<ul style="list-style-type: none"> <li>- El desarrollo iterativo.</li> <li>- Cercanía a los Timebox conocidos como sprints (periodos de tiempo dentro del proyecto).</li> <li>- Reuniones diarias para evaluar el progreso conocido como Scrum Diario.</li> <li>- El mismo equipo de desarrollo de organización.</li> <li>- Se manejan las tareas usando los atrasos; el atrasos del producto y atrasos de la carrera corta.</li> </ul>	<ul style="list-style-type: none"> <li>- Entrega de productos en ciclos cortos.</li> <li>- Retroalimentación rápida disponible.</li> <li>- Rápida adaptación al cambio.</li> </ul>	<ul style="list-style-type: none"> <li>- Falta de documentación.</li> <li>- Requiere de desarrolladores experimentados.</li> <li>- Difícil de estimar en el al comienzo el esfuerzo general requerido para implementar proyectos largos por lo tanto los costos no están muy precisos.</li> </ul>
<b>CLEANROOM</b>	<ul style="list-style-type: none"> <li>- Desarrollo iterativo.</li> <li>- El método de estructura de caja.</li> <li>- Uso de modelos matemáticos en el control de calidad.</li> </ul>	<ul style="list-style-type: none"> <li>- Reducción considerable en el rango de errores.</li> <li>- Alta calidad en productos de software.</li> </ul>	<ul style="list-style-type: none"> <li>- Aumentó de los costos de desarrollo.</li> <li>- Aumento del tiempo de promoción del producto.</li> <li>- Requiere desarrolladores experimentados de alto nivel.</li> </ul>

<p><b>PROCESO RACIONAL UNIFICADO</b></p>	<ul style="list-style-type: none"> <li>- Desarrollo iterativo.</li> <li>- Priorice el manejo de riesgo.</li> <li>- Modelo comercial adecuado.</li> <li>- Cambio de gestión.</li> <li>- Pruebas de desempeño.</li> </ul>	<ul style="list-style-type: none"> <li>- Documentación exacta y comprensiva.</li> <li>- Eficiente gestión en las solicitudes de cambio.</li> <li>- Eficiente integración de código nuevo.</li> <li>- Reuso disponible de componentes de código y software.</li> </ul>	<ul style="list-style-type: none"> <li>- Requiere a los profesionales altamente calificados.</li> <li>- El proceso de desarrollo es complejo y pobremente organizado.</li> </ul>
<p><b>SOLUCIONES DE ESTRUCTURA MICROSOFT</b></p>	<ul style="list-style-type: none"> <li>- Tiene versiones para la implementación ligera y pesada.</li> <li>- Promueve la comunicación.</li> <li>- Fortalece a los miembros del equipo y establece claramente la responsabilidad compartida.</li> </ul>	<ul style="list-style-type: none"> <li>- Soporte al progreso de múltiples procesos.</li> <li>- Políticas de manejo sólido del riesgo.</li> <li>- Construido para responder.</li> <li>- Reduce el tamaño del equipo.</li> </ul>	<ul style="list-style-type: none"> <li>- Difícil de ejecutar e instalar.</li> </ul>

Fuente: Comparative study on software development methodologies (Liviu, 2014)

En la investigación “Selection of Software Development Methodology (SDM): A Comparative Approach”, “...identificadas diferentes características del proyecto que influyen en la decisión para seleccionar a la metodología de desarrollo de software más adecuada para un proyecto específico son los siguientes basado en el análisis de requerimientos, basado en el estado del equipo de desarrollo, basado en la participación de los usuarios y basado en el tipo de proyecto y la asociación de riesgo”, “...Desde el estudio comparativo anterior queda claro que cada metodología tiene algunas características específicas a apoyar un proyecto específico. Supongamos que un proyecto de desarrollo de software requisito tiene usuarios claros elegimos prototipo, espiral o metodología XP. Una vez más, si el proyecto es altamente riesgoso, entonces elegimos la metodología en espiral para manejar técnicamente desafiantes productos de software que son propensos a varios tipos de riesgos. Si un proyecto tiene requisitos razonablemente bien conocidos y cronograma del proyecto vinculado en tiempo entonces elegimos RAD Metodología (Rapid Application Development) para la ejecución del proyecto. De nuevo, si un proyecto es simple (que tiene bajo riesgo) y la escala pequeña (que tiene el tamaño del equipo de 2 a 10 personas), entonces XP será la metodología apropiada” (Bhusan Mahapatra & Goswami, 2015).

**Tabla 003**  
**Basado en Análisis de Requerimiento**

ANÁLISIS DE REQUISITOS	CASCADA	PROTOTIPO	REITERATIVO	ESPIRAL	RAD	XP
¿Los requisitos son fácilmente entendibles y definidos?	Si	No	No	No	Si	No
¿Cambiamos los requisitos dejados a menudo?	No	Si	No	Si	No	Si
¿Nosotros podemos definir los requisitos de arranque de iteración?	Si	No	Si	No	Si	No
Los requisitos están indicando un sistema complejo a ser construido.	No	Si	Si	Si	No	Si

Fuente: Selection of Software Development Methodology (SDM): A Comparative Approach (Bhusan Mahapatra & Goswami, 2015)

**Tabla 004**  
**Basado en el estado del equipo de desarrollo**

EL EQUIPO DE DESARROLLO	CASCADA	PROTOTIPO	REITERATIVO	ESPIRAL	RAD	XP
Menos experiencia en proyectos similares	No	Si	No	Si	No	No
Menos dominio del conocimiento (nuevo en la tecnología)	Si	No	Si	Si	No	No
Menos experiencia en las herramientas que serán usadas	Si	No	No	Si	No	No
disponibilidad de entrenar si es necesario	No	No	Si	No	Si	Si

Fuente: Selection of Software Development Methodology (SDM): A Comparative Approach (Bhusan Mahapatra & Goswami, 2015)

**Tabla 005**  
**Basado en la participación de los usuarios**

PARTICIPACIONES DE USUARIOS	CASCADA	PROTOTIPO	REITERATIVO	ESPIRAL	RAD	XP
Participan los usuarios en todas las fases	No	Si	No	No	Si	Si
Participación limitada de los usuarios	Si	No	Si	Si	No	No
Los usuarios no tienen experiencia previa en la participación en similares proyectos	No	Si	Si	Si	No	No
Los usuarios son expertos en el dominio del problema	No	Si	Si	No	Si	Si

Fuente: Selection of Software Development Methodology (SDM): A Comparative Approach (Bhusan Mahapatra & Goswami, 2015)

**Tabla 006**  
**Basado en el tipo de proyecto y asociación al riesgo**

Tipo de proyecto y Riesgo	CASCADA	PROTOTIPO	REITERATIVO	ESPIRAL	RAD	XP
El proyecto es el perfeccionamiento del sistema existente	No	No	Si	No	Si	Si
El financiamiento es apropiado para el proyecto	Si	Si	No	No	Si	No
Alta fiabilidad en los requerimientos	No	No	Si	Si	No	Si
Ajustado plazo de proyecto	No	Si	Si	Si	Si	No
uso de componentes reusables	No	Si	No	Si	Si	No
Los recursos (tiempo, el dinero, el etc de las personas) desalientan.	No	Si	No	Si	No	No

Fuente: Selection of Software Development Methodology (SDM): A Comparative Approach (Bhusan Mahapatra & Goswami, 2015)

De lo revisado se concluye que no existe un ciclo de desarrollo de software o metodología de desarrollo aceptado universalmente, sino que el ingeniero de software debe evaluar las condiciones para elaborar su proceso de desarrollo de software.

Para la investigación, con base en los antecedentes y la teoría revisada se propone el siguiente proceso de desarrollo, diseñado a partir de las ventajas del modelo cascado:

Tabla 007  
Modelo de desarrollo que se utilizó en la presente investigación

ACTIVIDAD	TÉCNICA	ENTREGABLE
<b>Modelado del negocio</b>	Casos de uso del negocio Paquetes Modelo de proceso de negocio	Diagrama de casos de uso de negocio Diagrama de paquetes Diagrama de procesos de negocio.
<b>Requerimientos</b>	Requerimientos funcionales	Listado de requerimientos funcionales Listado de requerimientos no funcionales
<b>Análisis</b>	Casos de uso de sistema	Diagramas de casos de uso de sistema
<b>Diseño</b>	Entidad relación Maqueteado	Diagrama Entidad relación Maqueteado del sistema
<b>Implementación</b>	Programas	Programas

Fuente. Elaboración propia

### CAP III. MODELO CONCEPTUAL Y DESARROLLO

#### 3.1 Modelo conceptual

Tomando como referencia la base teórica y los objetivos, se consideró el siguiente modelo conceptual para la presente investigación:



Figura 001

Modelo conceptual propuesto

#### Definición de las variables

Para la investigación de definieron las siguientes variables:

Variable independiente: Software para la gestión de créditos.

Variable dependiente: Gestión de créditos.

Tabla 008  
Operacionalización de variables

VARIABLES	DIMENSIONES	INDICADORES	ESCALA	TIPO
Gestión de créditos	Crédito, desembolso y pago	Tiempo	Intervalo	Continuo
Software para gestión de créditos	Funcionalidad	Requerimientos funcionales	Intervalo	Discreta
	Facilidad de uso	Facilidad de uso	Intervalo	Discreta

Fuente: Elaboración propia



## 3.2 Desarrollo del software

El servicio de la Cooperativa de Ahorro y Crédito es denominado **Crédito Personal**, por lo que el desarrollo de software estará orientado a dar el soporte tecnológico a este proceso.

### 3.2.1 Modelado del Negocio

#### A. Casos de uso de Negocio

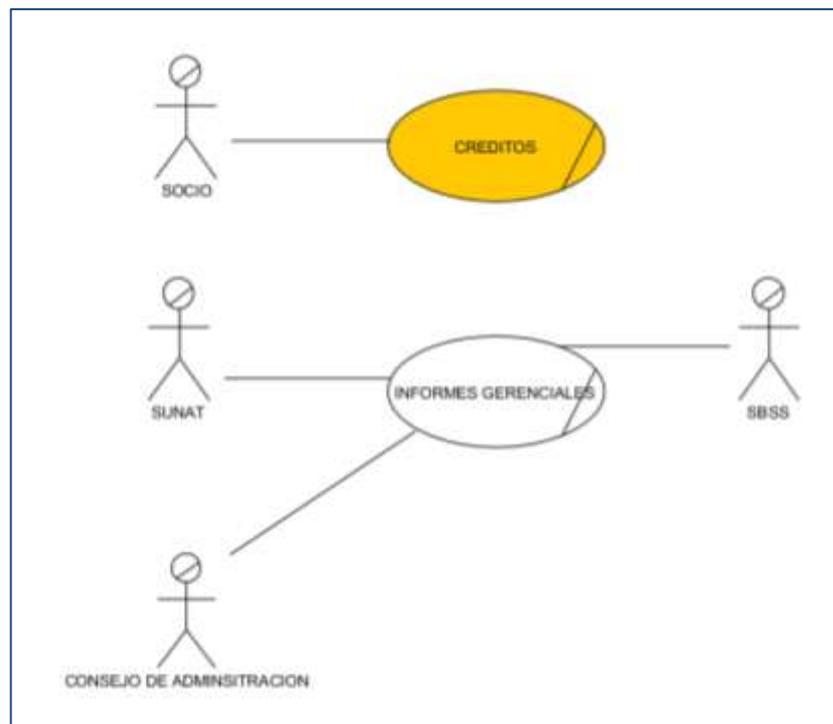


Figura 002  
Diagrama de caso de uso de negocio  
Fuente: Elaboración propia

## B. Modelo de proceso de negocio

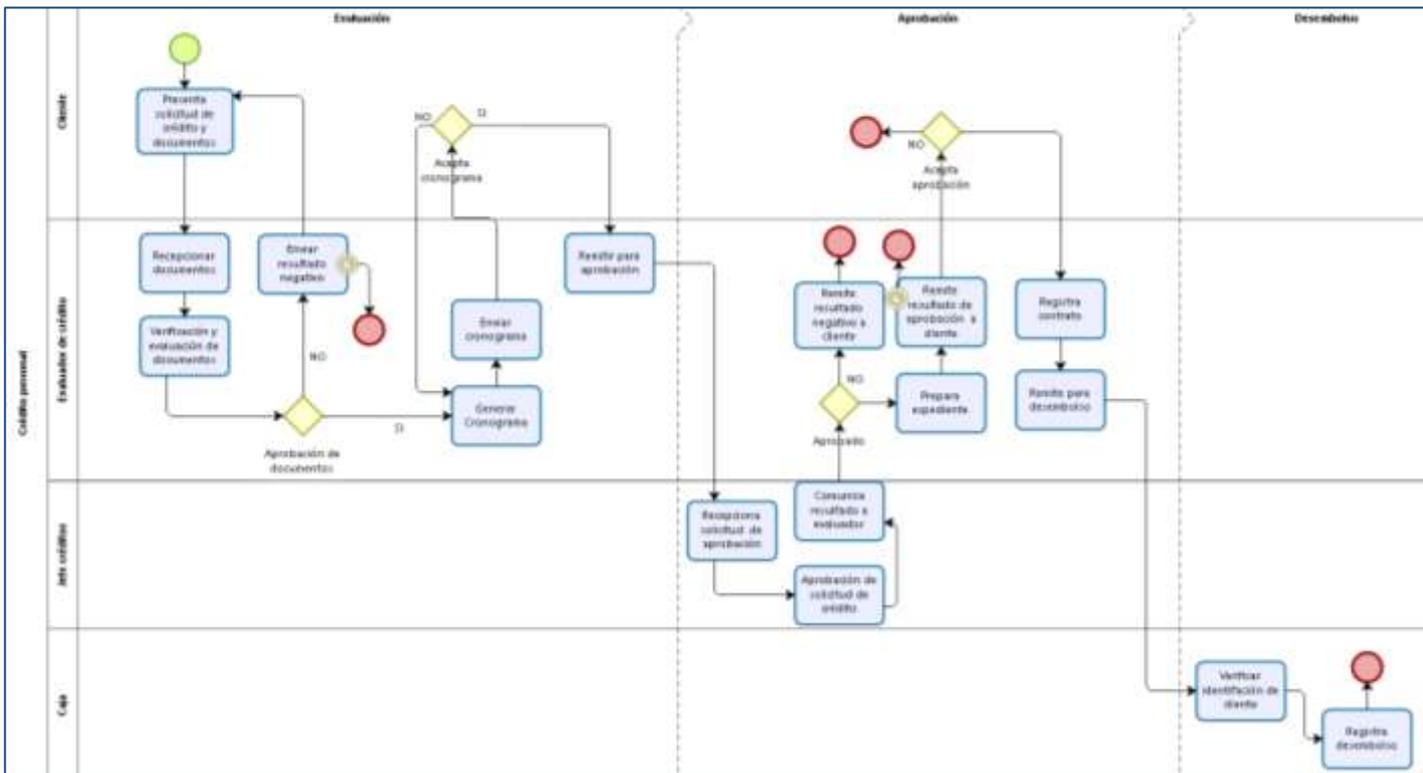


Figura 003  
Proceso para registrar crédito otorgado al cliente  
Fuente: Elaboración propia

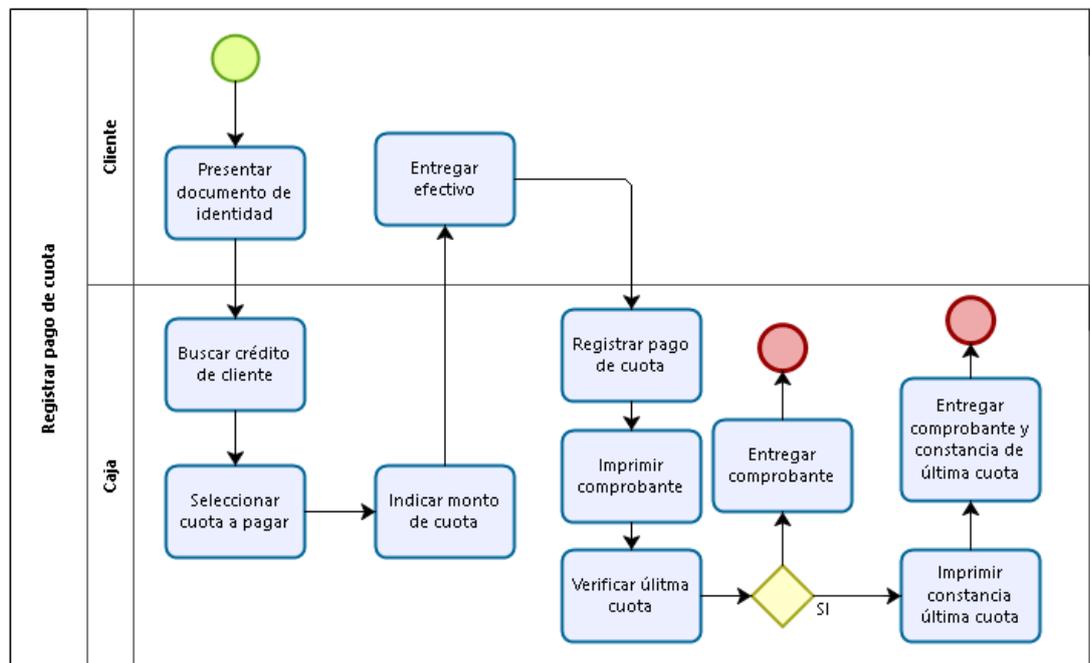


Figura 004  
Proceso para registrar los pagos de las cuotas del crédito  
Fuente: Elaboración propia

### C. Paquetes

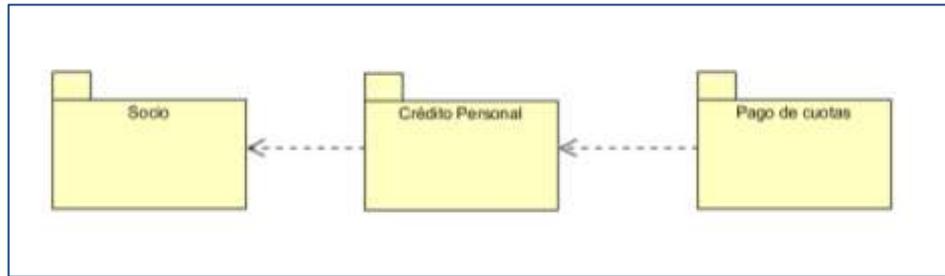


Figura 005  
Diagrama de paquetes  
Fuente: Elaboración propia

### 3.2.2 Requerimientos Funcionales

#### Paquete socio

Tabla 009  
Requisitos funcionales de paquete socio

Código	Actor de Sistema	Descripción
R01	Asistente Gerencia	Buscar socio: búsqueda en la base de datos de los socios registrados
R02	Asistente Gerencia	Nuevo Socio: registro de los datos de un nuevo socio en la base de datos.
R03	Asistente Gerencia	Modificar socio: modifica los datos de un socio en la base de datos.
R04	Asistente Gerencia	Eliminar socio: registra la salida de un socio de la cooperativa, en la base de datos.
R05	Asistente Gerencia	Generar PDF: imprime el resultado de la búsqueda.

Fuente: Elaboración propia

## Paquete Crédito Personal

Tabla 010  
Requisitos funcionales de paquete Crédito Personal

<b>Código</b>	<b>Actor de Sistema</b>	<b>Descripción</b>
R06	Evaluador de crédito	Buscar socio: búsqueda en la base de datos de los socios registrados
R07	Evaluador de crédito	Recepcionar documentos: escanear los documentos del cliente y los carga al sistema.
R08	Evaluador de crédito	Cronograma: ingresar los datos del crédito solicitado por el cliente
R09	Evaluador de crédito	Generar cronograma: genera el cronograma del futuro crédito.
R10	Evaluador de crédito	Imprimir cronograma
R11	Evaluador de crédito	Enviar cronograma a socio: se remite al correo del socio el cronograma.
R12	Evaluador de crédito	Aprobar cronograma: el evaluador una vez acordado con el cliente aprueba el cronograma.
R13	Evaluador de crédito	Remite aprobación: se envía la aprobación al Jefe de Créditos para su aprobación final.
R14	Evaluador de crédito	Preparación de desembolso: visualiza si el crédito personal fue aprobado por el Jefe de Créditos.
R15	Evaluador de crédito	Generar desembolso: registra la aprobación para desembolso.
R16	Evaluador de crédito	Anexar contrato: sube escaneado del contrato.
R17	Jefe de Créditos	Créditos para aprobación: visualiza los créditos para aprobación.
R18	Jefe de Créditos	Aprobar créditos: visualiza datos del crédito para aprobar
R19	Jefe de Créditos	Aprobar: aprueba o rechaza los créditos
R20	Jefe de Créditos	Remitir resultado: se remite el resultado por el sistema para el desembolso en Caja.
R21	Cajero	Desembolso: visualiza los desembolsos por realizar.
R22	Cajero	Buscar socio: busca a socio por DNI
R23	Cajero	Registrar desembolso
R24	Cajero	Imprimir desembolso: imprime el recibo del desembolso.

Fuente: Elaboración propia

### Paquete cuotas

Tabla 011  
Requisitos funcionales de paquete Pago Cuotas

Código	Actor de Sistema	Descripción
R25	Cajero	Buscar socio: búsqueda en la base de datos de los socios con créditos
R26	Cajero	Pago Cuota: visualiza los datos de la cuota a pagar
R27	Cajero	Pagar cuota: registra pago de cuota
R28	Cajero	Imprimir comprobante: imprime el recibo del pago de la cuota

Fuente: Elaboración propia

### 3.2.3 Análisis

#### Paquete socio

A. Diagrama de casos de uso de sistema Asistente Gerencia

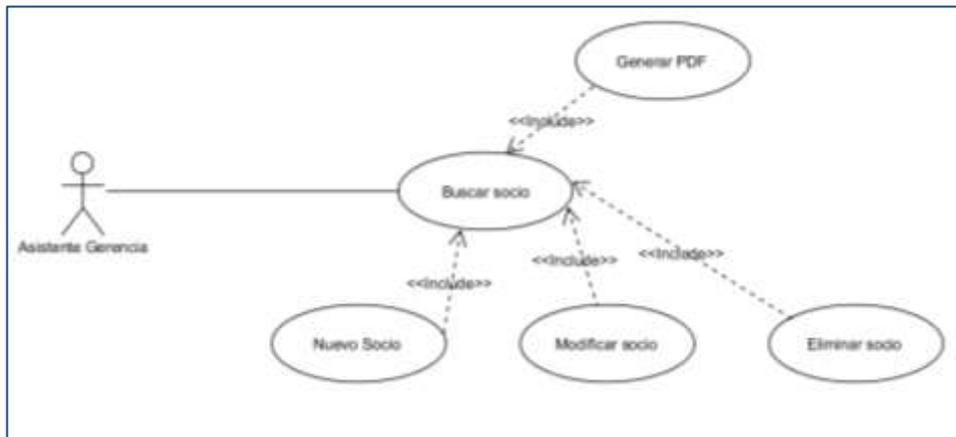


Figura 006  
Diagrama de caso de uso de Asistente Gerencia  
Fuente: Elaboración propia

**Paquete crédito personal**

A. Diagrama de casos de uso de sistema evaluador de crédito

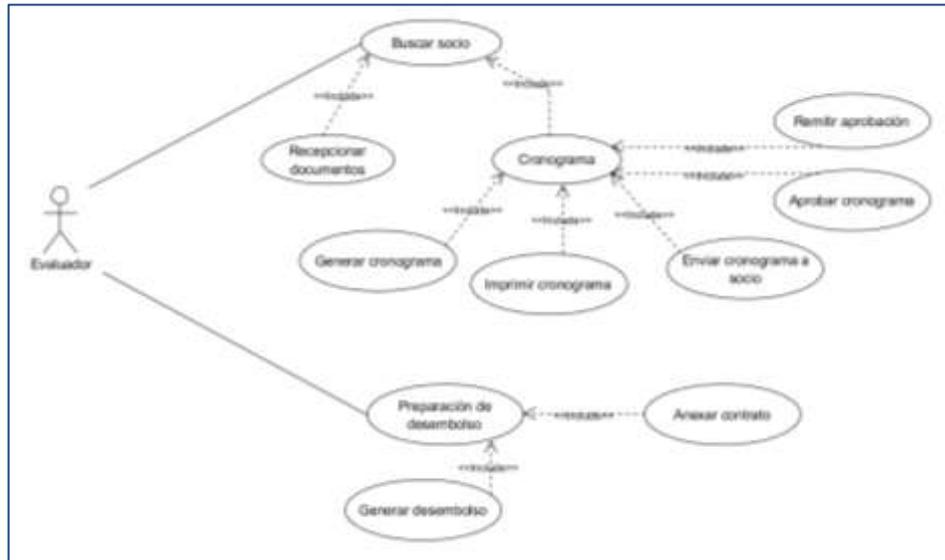


Figura 007

Diagrama de caso de uso de Evaluador de Crédito

Fuente: Elaboración propia

B. Diagrama de caso de uso de sistema Jefe Crédito

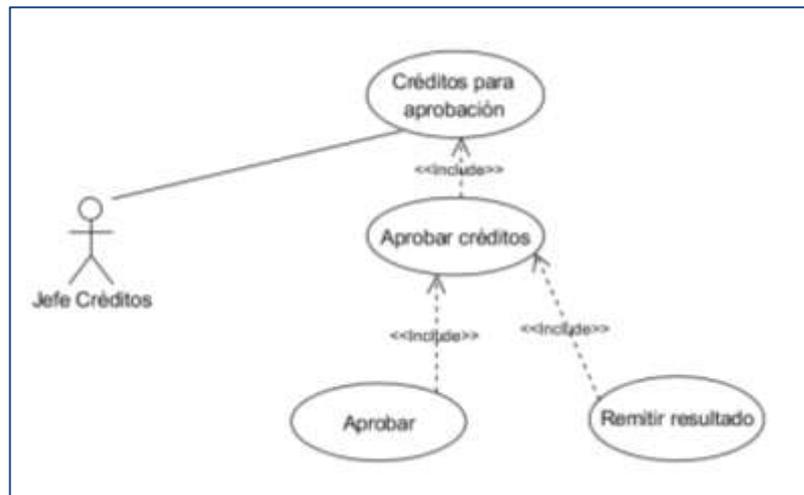


Figura 008

Diagrama de caso de uso de Jefe de Créditos

Fuente: Elaboración propia

C. Diagrama de caso de uso de sistema Caja paquete Crédito Personal

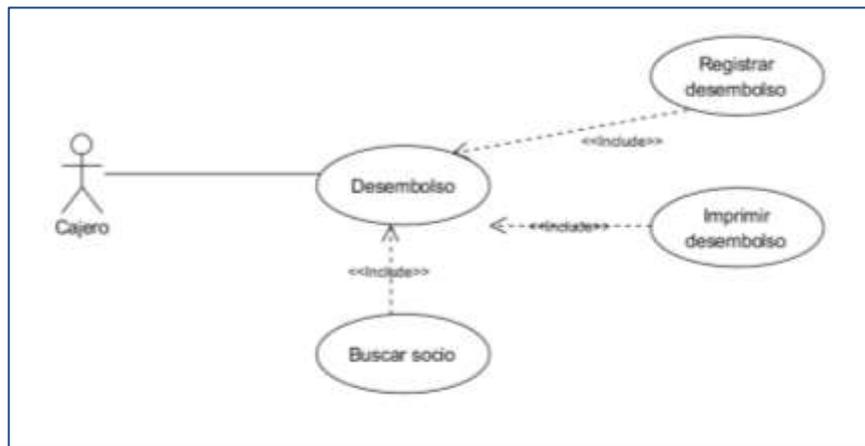


Figura 009  
Diagrama de caso de uso de Caja - Paquete Crédito Personal  
Fuente: Elaboración propia

**Paquete pago cuotas**

A. Diagrama de caso de uso de sistema Caja paquete Pago Cuotas

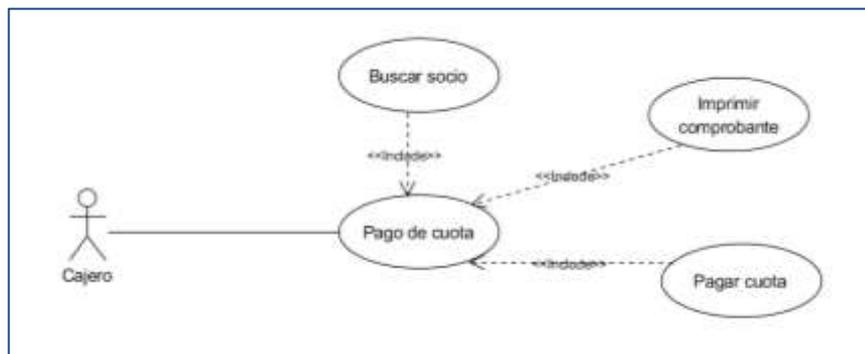


Figura 010  
Diagrama de caso de uso de Caja – Paquete cuotas  
Fuente: Elaboración propia

### 3.2.4 Diseño

#### A. Diagrama entidad relación

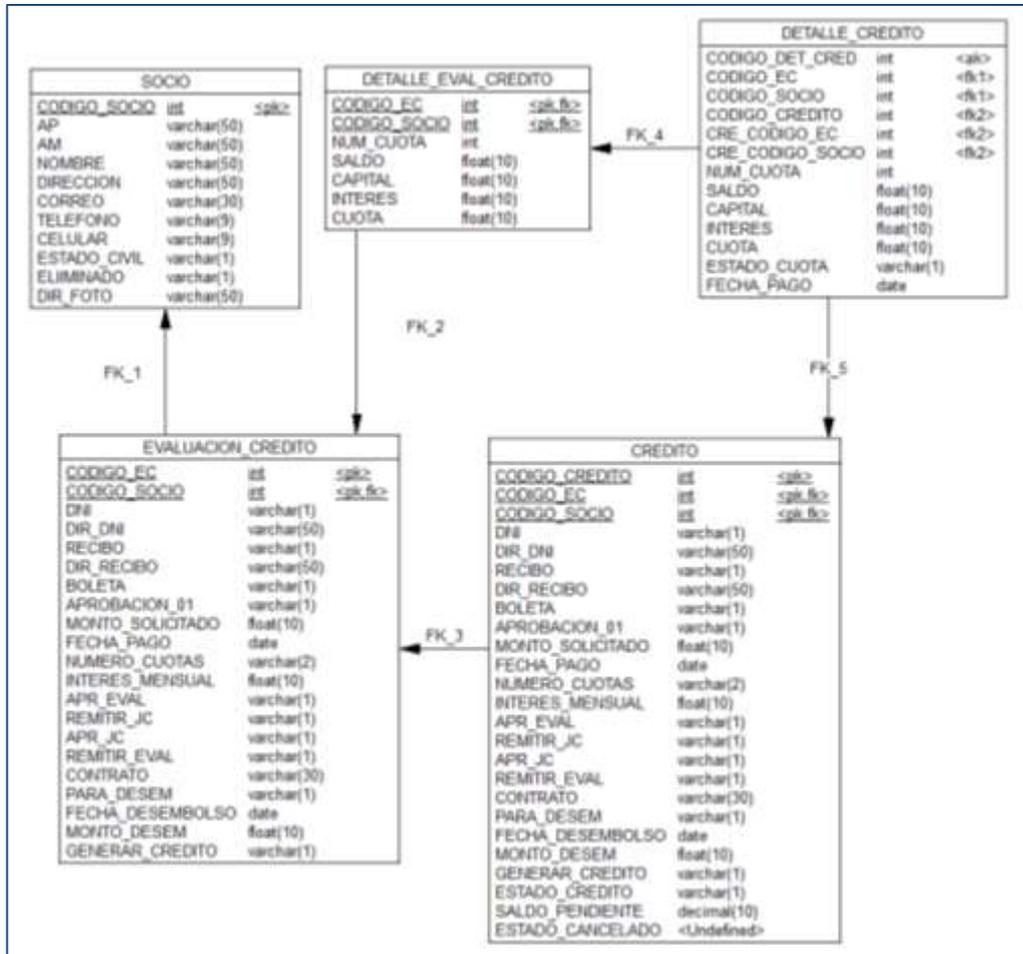


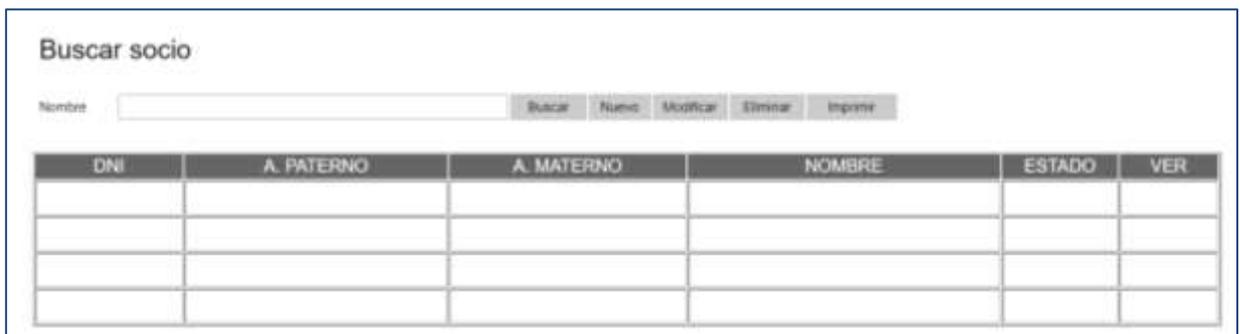
Figura 011  
Diagrama de Entidad Relación  
Fuente: Elaboración propia

## B. Maquetado del Paquete Mantenimiento de Socio



The screenshot shows the login interface for Crediservis, a cooperative for savings and credit. At the top, there is a logo with three stylized mountain peaks in red, green, and blue, followed by the text "Crediservis" and "Cooperativa de Ahorro y Crédito" below it. Below the logo, there are two input fields: "Usuario" (User) and "Clave" (Password). To the right of these fields are two buttons: "Ingresar" (Login) and "Cancelar" (Cancel).

Figura 012  
Ingreso a sistema  
frmIngreso.php  
Fuente: Elaboración propia



The screenshot shows a web form titled "Buscar socio" (Search member). It features a search input field labeled "Nombre" (Name) and a row of action buttons: "Buscar" (Search), "Nuevo" (New), "Modificar" (Modify), "Eliminar" (Delete), and "Imprimir" (Print). Below the search area is a table with the following columns: "DNI", "A. PATERNO" (Paternal Surname), "A. MATERNO" (Maternal Surname), "NOMBRE" (Name), "ESTADO" (Status), and "VER" (View). The table currently contains no data rows.

Figura 013  
Buscar socio  
frmBuscarSocio.php  
Fuente: Elaboración propia

**Nuevo Socio**

Código

DNI

Apellido Paterno

Apellido Materno

Nombre

Dirección

Correo

Teléfono

Celular

Subir foto  No se eligió archivo

Estado civil

Figura 014  
Nuevo socio  
frmNuevoSocio.php  
Fuente: Elaboración propia

**Modificar Socio**

Código

DNI

Apellido Paterno

Apellido Materno

Nombre

Dirección

Correo

Teléfono

Celular

Subir foto  No se eligió archivo

Estado civil

Figura 015  
Modificar socio  
frmModificarSocio.php  
Fuente: Elaboración propia

**Eliminar Socio**

Código

DNI

Apellido Paterno

Apellido Materno

Nombre

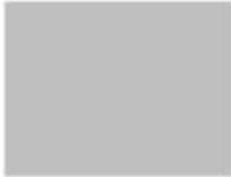
Foto 

Figura 016  
 Eliminar socio  
 frmEliminarSocio.php  
 Fuente: Elaboración propia

**Ficha Socio**

Código

DNI

Apellido Paterno

Apellido Materno

Nombre

Dirección

Comeo

Teléfono

Celular

Estado civil

Foto 

Figura 017  
 Eliminar socio  
 frmImprimirSocio.php  
 Fuente: Elaboración propia

### C. Maquetado del Paquete Crédito Personal

**Evaluación de crédito**

**1. Buscar socio**

DNI

AP  AM  NOMBRE

**2. Recepcionar documentos**

DNI   No se eligió archivo

Recibo agua / luz   No se eligió archivo

Boleta de pago   No se eligió archivo

**3. Generar cronograma**

Monto de crédito solicitado:

Fecha de pago:  Enero 2018

Número de cuotas:

Interés mensual:

Cuota	Saldo	Capital	Interés	Cuota
1	500	100	50	150
2	400	100	40	140
3	300	100	30	130
4	200	100	20	120
5	100	100	10	110

Figura 018  
Evaluación de crédito  
[frmEvaluacionCredito.php](#)  
Fuente: Elaboración propia

**Aprobación de Jefe de Créditos**

**1. Buscar socio**

DNI

AP  AM  NOMBRE

**2. Recepcionar documentos**

DNI

Recibo agua / luz

Boleta de pago

**3. Generar cronograma**

Monto de crédito solicitado:

Fecha de pago:  Enero 2018

Número de cuotas:

Interés mensual:

Cuota	Saldo	Capital	Interés	Cuota
1	500	100	50	150
2	400	100	40	140
3	300	100	30	130
4	200	100	20	120
5	100	100	10	110

**4. Aprobación de crédito**

Figura 019  
Aprobación de jefe de créditos  
[frmEvaluacionJefeCredito.php](#)  
Fuente: Elaboración propia

### Preparación de desembolso

**1. Buscar socio**

DNI    
AP  AM  NOMBRE 

**2. Recepcionar documentos**

DNI    
Recibo agua / luz    
Boleta de pago  

**3. Generar cronograma**

Monto de crédito solicitado:   
Fecha de pago:     
Número de cuotas:   
Interés mensual: 

Cuota	Saldo	Capital	Interés	Cuota
1	500	100	50	150
2	400	100	40	140
3	300	100	30	130
4	200	100	20	120
5	100	100	10	110

**4. Aprobación de crédito**

Resultado: 

**5. Preparación de expediente y contrato**

Anexar contrato:  No se eligió archivo

Figura 020  
Preparación de desembolso  
[frmPreparacionDesembolso.php](#)  
Fuente: Elaboración propia

### Registro de desembolso

**1. Buscar socio**

DNI    
AP  AM  NOMBRE 

**2. Registrar desembolso**

Monto de crédito para desembolso:   
Fecha de desembolso:

Figura 021  
Registro de desembolso  
[frmRegistroDesembolso.php](#)  
Fuente: Elaboración propia

### C. Maquetado del Paquete Registro Pago Cuota

**Pago de cuota**

**1. Buscar socio**

DNI

AP  AM  NOMBRE

**2. Seleccionar cuota**

Cuota	Saldo	Capital	Interés	Cuota	Estado	Fecha Pago
1	500	100	50	150	Pagada	
2	400	100	40	140	Pagada	
3	300	100	30	130		28/01/2018 <input type="button" value="Pagar"/>
4	200	100	20	120		
5	100	100	10	110		

**3. Comprobante**

DNI XXX  
AP XXX  
AM XXX  
NOMBRE XXX

N° Cuota XXX  
Saldo XXX  
Capital XXX  
Interés XXX  
Cuota XXX

Próxima fecha pago dd/mm/yyyy  
Alerta última cuota Si

Figura 022  
Registro de pago de cuota  
[frmRegistroPagoCuota.php](#)  
Fuente: Elaboración propia

### 3.2.5 Implementación

#### A. Despliegue paquete Mantenimiento Socio

Tabla 012  
Despliegue Manteamiento Socio

Formulario	Programa	Tabla
frmIngreso	ingreso	USUARIO
frmBuscarSocios	buscar	SOCIO
frmNuevoSocio	nuevo	SOCIO
frmModificarSocio	modificar	SOCIO
frmEliminarSocio	eliminar	SOCIO
frmImprimirSocio	imprimir	SOCIO

Fuente: Elaboración propia

#### B. Despliegue paquete Crédito Personal

Tabla 013  
Despliegue Crédito Personal

Formulario	Programa	Tabla
frmIngreso	ingreso	USUARIO
frmEvaluacionCredito.php	buscarSocio GenerarCronograma ImprimirCronograma EnviarCliente AprobarCronograma RemitirAprobacion	SOCIO EVALUACION_CREDITO DETALLE_EVAL_CREDITO
frmEvaluacionJefeCredito.php	AprobacioCredito RemitirResultado	SOCIO EVALUACION_CREDITO DETALLE_EVAL_CREDITO
frmPreparacionDesembolso.php	GenerarDesembolso	SOCIO EVALUACION_CREDITO DETALLE_EVAL_CREDITO
frmRegistroDesembolso.php	RegistroDesembolso	SOCIO EVALUACION_CREDITO DETALLE_EVAL_CREDITO CREDITO DETALLE_CREDITO

Fuente: Elaboración Propia

C. Despliegue paquete Pago de Cuotas

Tabla 014  
Despliegue Pago Cuota

Formulario	Programa	Tabla
frmRegistroPagoCuota.php	Pagar ImprimirComprobante	SOCIO EVALUACION_CREDITO DETALLE_EVAL_CREDITO CREDITO DETALLE_CREDITO

Fuente: Elaboración Propia

## CAP. IV ANÁLISIS Y DISCUSIÓN DE LOS RESULTADOS

### Objetivo Específico 01

Determinar la funcionalidad del software de gestión de crédito

Los resultados indicaron que los requerimientos funcionales cumplieron con las expectativas.

Se realizó prueba de funcionalidad con casos de prueba para determinar la aceptación de cada requerimiento.

Se determinó en que el valor del parámetro de aprobación del requerimiento funcional sea como mínimo 8

### Paquete Mantenimiento Socio

Tabla 015  
Requisitos funcionales de paquete socio

Código	Actor de Sistema	Descripción	Caso de prueba	Resultado	Nivel de Aprobación
R01	Asistente Gerencia	Buscar socio: búsqueda en la base de datos de los socios registrados	P01	Aprobado	9
R02	Asistente Gerencia	Nuevo Socio: registro de los datos de un nuevo socio en la base de datos.	P02	Aprobado	10
R03	Asistente Gerencia	Modificar socio: modifica los datos de un socio en la base de datos.	P03	Aprobado	10
R04	Asistente Gerencia	Eliminar socio: registra la salida de un socio de la cooperativa, en la base de datos.	P04	Aprobado	10
R05	Asistente Gerencia	Generar PDF: imprime el resultado de la búsqueda.	P05	Aprobado	9

Fuente: Elaboración propia

## Paquete Crédito Personal

Tabla 016  
Requisitos funcionales de paquete Crédito Personal

<b>Código</b>	<b>Actor de Sistema</b>	<b>Descripción</b>	<b>Caso de prueba</b>	<b>Resultado</b>	<b>Nivel de Aprobación</b>
R06	Evaluador de crédito	Buscar socio: búsqueda en la base de datos de los socios registrados	P06	Aprobado	9
R07	Evaluador de crédito	Recepcionar documentos: escanear los documentos del cliente y los carga al sistema.	P07	Aprobado	10
R08	Evaluador de crédito	Cronograma: ingresar los datos del crédito solicitado por el cliente	P08	Aprobado	10
R09	Evaluador de crédito	Generar cronograma: genera el cronograma del futuro crédito.	P09	Aprobado	10
R10	Evaluador de crédito	Imprimir cronograma	P10	Aprobado	9
R11	Evaluador de crédito	Enviar cronograma a socio: se remite al correo del socio el cronograma.	P11	Aprobado	10
R12	Evaluador de crédito	Aprobar cronograma: el evaluador una vez acordado con el cliente aprueba el cronograma.	P12	Aprobado	10
R13	Evaluador de crédito	Remite aprobación: se envía la aprobación al Jefe de Créditos para su aprobación final.	P13	Aprobado	10
R14	Evaluador de crédito	Preparación de desembolso: visualiza si el crédito personal fue aprobado por el Jefe de Créditos.	P14	Aprobado	10
R15	Evaluador de crédito	Generar desembolso: registra la aprobación para desembolso.	P15	Aprobado	10
R16	Evaluador de crédito	Anexar contrato: sube escaneado del contrato.	P16	Aprobado	10
R17	Jefe de	Créditos para	P17	Aprobado	9

	Créditos	aprobación: visualiza los créditos para aprobación.			
R18	Jefe de Créditos	Aprobar créditos: visualiza datos del crédito para aprobar	P18	Aprobado	10
R19	Jefe de Créditos	Aprobar: aprueba o rechaza los créditos	P19	Aprobado	10
R20	Jefe de Créditos	Remitir resultado: se remite el resultado por el sistema para el desembolso en Caja.	P20	Aprobado	10
R21	Cajero	Desembolso: visualiza los desembolsos por realizar.	P21	Aprobado	10
R22	Cajero	Buscar socio: busca a socio por DNI	P22	Aprobado	9
R23	Cajero	Registrar desembolso	P23	Aprobado	10
R24	Cajero	Imprimir desembolso: imprime el recibo del desembolso.	P24	Aprobado	10

Fuente: Elaboración propia

### **Paquete cuotas**

Tabla 017  
Requisitos funcionales de paquete Pago Cuotas

<b>Código</b>	<b>Actor de Sistema</b>	<b>Descripción</b>	<b>Caso de prueba</b>	<b>Resultado</b>	<b>Nivel de Aprobación</b>
R25	Cajero	Buscar socio: búsqueda en la base de datos de los socios con créditos	P25	Aprobado	9
R26	Cajero	Pago Cuota: visualiza los datos de la cuota a pagar	P26	Aprobado	10
R27	Cajero	Pagar cuota: registra pago de cuota	P27	Aprobado	10
R28	Cajero	Imprimir comprobante: imprime el recibo del pago de la cuota	P28	Aprobado	10

Fuente: Elaboración propia

Los resultados indicaron que se aceptaron todos los requerimientos funcionales con un nivel de aprobación superior al nivel ocho que se había planteado como meta.

Tabla 018  
Operacionalización de variables

Nivel de aprobación	Cantidad	Frecuencia
9	7	25%
10	21	75%
Total	28	100%

Fuente: Elaboración propia

Los resultados indicaron que el 25% de los requisitos fueron aprobados con nivel de aprobación 9 y 75% con nivel de aprobación 10.

### Objetivo Específico 02

Determinar la facilidad de uso del software de gestión de crédito.

Se diseñó un cuestionario para la evaluación de la facilidad de uso en base al Cuestionario para la Evaluación de la Usabilidad (Shneiderman, 98).

Tabla 019  
Reacciones globales al sistema

Mínimo	Evaluación										Máximo	Resultado:
Terrible	1	2	3	4	5	6	7	8	9	NA	Maravillo	8
Frustrante	1	2	3	4	5	6	7	8	9	NA	Satisfactorio	8
Aburrido	1	2	3	4	5	6	7	8	9	NA	Estimulante	7
Difícil	1	2	3	4	5	6	7	8	9	NA	Fácil	8
Potencia Inadecuada	1	2	3	4	5	6	7	8	9	NA	Potencia adecuada	8
Rígido	1	2	3	4	5	6	7	8	9	NA	Flexible	8

Fuente: Elaboración propia

Considerando valores superiores a 8, las reacciones globales al sistema indicaron que era maravilloso, satisfactorio, poco estimulante, fácil de utilizar, con la potencia adecuada y flexible.

Tabla 020  
Pantalla

Mínimo	Evaluación										Máximo	Resultado:
Caracteres sobre la pantalla												
Difícil de leer	1	2	3	4	5	6	7	8	9	NA	Fácil de leer	9
¿Resultaron de ayuda los aspectos que se destacaban sobre la pantalla?												
En absoluto	1	2	3	4	5	6	7	8	9	NA	Mucho	7
Secuencia de pantallas												
Confuso	1	2	3	4	5	6	7	8	9	NA	Claro	8

Fuente: Elaboración propia

Con respecto a la pantalla se determinó que es fácil de leer, los aspectos que se destacan en la pantalla ayudan con la consideración que son pocos los aspectos resaltantes, y la secuencia de pantalla al utilizarse modelo de proceso de negocio al principio resultó confuso, utilizando capacitaciones se logró que quede claro la secuencia de pantallas.

Tabla 021  
Capacidad del sistema

Mínimo	Evaluación										Máximo	Resultado:
Velocidad del sistema												
Demasiado lento	1	2	3	4	5	6	7	8	9	NA	Demasiado rápido	8
¿En qué medida el sistema es viable?												
Nada fiable	1	2	3	4	5	6	7	8	9	NA	Muy fiable	8
Corrección de errores												
Difícil	1	2	3	4	5	6	7	8	9	NA	Sencillo	7

Fuente: Elaboración propia

La capacidad del sistema es considerada como un sistema rápido y bastante fiable. Los errores cometidos por los usuarios del sistema un poco sencillo de corregir, se considera mejorar esta característica para las nuevas versiones del sistema.

### **Objetivo general**

Para determinar si hubo mejorar el proceso de créditos con la implementación de un software de gestión de crédito en la cooperativa de ahorro y crédito CREDISERVIS LTDA, se consideraron dos indicadores: tiempo y errores cometidos.

Para el indicador tiempo, se redujo el tiempo de evaluación, aprobación y desembolso de tres días a un solo día, el tiempo de respuesta para el cliente es bastante satisfactorio. Los tiempos que más disminuyeron fueron los tiempos de búsquedas.

Para el indicador errores cometidos, se eliminan los errores de transcripción, pérdidas de documentos en lo expedientes de solicitud de crédito

Shneiderman, B. Designing the User Interface: Strategies for Effective Human-Computer Interaction, Third Edition. Addison Wesley, Reading, MA, 1998.

## CONCLUSIONES

La presente investigación permitió concluir que:

- A. Se ha desarrollado el software para la gestión estratégica en la cooperativa de ahorro y crédito CREDISERVIS LTDA. Diseñando un modelo de desarrollo de software tomando como referencia las metodologías utilizadas en otras investigaciones.
- B. El modelo conceptual propuesto, para esta investigación de tipo aplicada, descriptiva y no experimental ha permitido verificar los resultados, donde se propone mejoras de procesos utilizando tecnologías de información y sistema de información.
- C. La combinación de dos metodologías: el Modelo de Proceso de Negocio (BPM) y el Modelo de Desarrollo de Software fue posible, la primera metodología ayudó en modelar el proceso identificando la secuencia de actividades a diferencia de los casos de usos de negocio, esta técnica no está incluida en alguna metodología de desarrollo de software. Posteriormente se diseñó el software para seguir la secuencia de actividades.
- D. Los requerimientos funcionales del software fueron implementados y probados logrando una aceptación de cada uno, este logro fue posible considerando una buena gestión de requisitos y una correcta secuencia de actividades de desarrollo del requerimiento.

- E. La facilidad de uso fue comprobada logrando una evaluación aceptable del software.
  
- F. Se ha logrado mejorado el proceso de gestión de crédito, dimensionado a través principalmente en base a tiempo y posteriormente a errores cometidos.

## RECOMENDACIONES

Con los resultados de la investigación se recomienda:

- A. Extender las investigaciones sobre metodologías de desarrollo de software para diferentes tipos de organizaciones, la contextualización de metodologías de desarrollo de software significa una alternativa para estas investigaciones.
- B. Existe poca investigación sobre de verificación de resultados de investigación relacionados con la influencia de tecnologías de información y sistemas de información sobre procesos de una organización. Los modelos conceptuales para este tipo de investigaciones serían importantes que permitiría un avance para el desarrollo de nuevas investigaciones aplicadas.
- C. En las combinaciones metodologías para el desarrollo de software, se recomienda incluir técnicas que permitan medir el proceso bajo diferentes dimensiones. Estas técnicas permitirán diseñar el proceso cuando es nuevo o realizar reingeniería de procesos cuando ya existe el proceso.
- D. Los requerimientos funcionales se convierten en el punto más importante del desarrollo del software, culminan con la comprobación a través de los casos de prueba. Existe investigaciones sobre gestión de requisitos, lo contrario sucede con la secuencia de actividades de desarrollo que relacionen los requerimientos funcionales y los casos de prueba.

- E. Diversas investigaciones sobre facilidad de uso o usabilidad se han desarrollado, bajo la perspectiva de entorno, sin embargo, no existe o no se ha determinado referencia de facilidad de uso bajo la perspectiva sobre el tipo de proceso sobre el cual se implementa.
- F. Es escasa las investigaciones sobre indicadores de procesos cuando debe ser dimensionado posteriormente de la implantación de tecnologías de información y sistemas de información. También investigaciones sobre la relación entre las tecnologías de información y sistemas de información con los procesos es escasa, el campo de estas nuevas investigaciones se presenta atractivo para nuevas investigaciones.

## REFERENCIAS BIBLIOGRÁFICAS

Bhusan Mahapatra, H., & Goswami, B. (2015). Selection of Software Development Methodology (SDM): A Comparative Approach. *International Journal of Advanced Research in Computer Science and Software Engineering*, 58-61.

Carrasco Espinoza, A. (2012). “Análisis, diseño e implementación del sistema para la gestión de cobranzas en la cooperativa de ahorro y crédito Alfonso Jaramillo León. Cuenca - Ecuador: Universidad de Cuenca.” Obtenido de <http://dspace.ucuenca.edu.ec/handle/123456789/2738>

Guijarro Riera, A., & Guijarro Riera, A. (2010). Diseño e implementación de un sistema de control del proceso de crédito y cobranza por medio de indicadores de gestión para una sociedad financiera ubicada en la ciudad de Guayaquil. Guayaquil - Ecuador: ESPO. Obtenido de <http://www.dspace.espol.edu.ec/xmlui/handle/123456789/16709>

Liviu, M. (2014). Comparative study on software development methodologies. *Database Systems Journal*, 37-56.

Presidencia de la República. (1985). Ley de cooperatias N° 15260. Lima.

Presidencia de la República. (2011). Ley 26702 General del Sistema Financiero y del Sistema de Seguros y Orgánica de la Superintendencia. Lima.

Tasato Cánepa, K. (2013). Desarrollo de una infraestructura de software para realizar pruebas automatizadas de sistemas de información desarrollados en lenguaje cobol en el contexto bancario. Lima-Peru: PUCP. Obtenido de <http://tesis.pucp.edu.pe/repositorio/handle/123456789/5424>

Vásquez Flores, C. (2013). Análisis, diseño e implementación de un sistema de recaudación de deudas. Lima-Perú: PUCP.