



Universidad
Inca Garcilaso de la Vega

Nuevos Tiempos. Nuevas Ideas

Facultad de Ingeniería de Sistemas, Cómputo y Telecomunicaciones

**Desarrollo de una aplicación web para la mejora de la aprobación
de arrendamientos financieros en el Banco Scotiabank Perú**

Tesis para optar el Título de Ingeniero de Sistemas y Cómputo

Bach. Fuertes Ramos, Jesús Joel

Asesor

MSc. Gonzales Sánchez, Santiago

Lima – Perú
Octubre de 2017

ÍNDICE

ÍNDICE DE FIGURAS	5
ÍNDICE DE TABLAS	8
RESUMEN.....	9
ABSTRACT.....	10
INTRODUCCIÓN.....	11
CAPÍTULO I: PLANTEAMIENTO DEL PROBLEMA	12
1.1 Situación Problemática	12
1.2 Problema de la investigación	12
1.2.1 Problema General.....	12
1.2.2 Problemas Específicos	12
1.3 Objetivos	13
1.3.1 Objetivo General.....	13
1.3.2 Objetivos Específicos.....	13
1.4 Justificación	13
1.5 Alcance	14
CAPÍTULO II: MARCO TEÓRICO	15
2.1 Antecedentes de la investigación.....	15
2.2 Bases teóricas.....	17
2.2.1 Aplicación web	17
2.2.2 Niveles de medición para el desarrollo de un software.....	21
2.2.3 Metodologías para el desarrollo de software	23
2.2.4 Arrendamiento (leasing) financiero	36
2.3 Glosario de términos	38
CAPÍTULO III: VARIABLES E HIPÓTESIS.....	40
3.1 Variables e Indicadores	40
3.1.1 Identificación de Variables	40
3.1.2 Operacionalización de Variables	40
3.2 Hipótesis.....	40
3.2.1 Hipótesis General.....	40
3.2.2 Hipótesis Específicas	40
3.3 Matriz de Consistencia.....	40
CAPÍTULO IV: METODOLOGÍA PARA EL DESARROLLO DE LA INVESTIGACIÓN	41
4.1 Metodología RUP - Rational Unified Process.....	41

4.2	RUP como un proceso personalizable	42
4.3	Fases de los ciclos de vida, objetivos e hitos.....	43
4.4	Buenas prácticas.....	44
4.5	Artefactos RUP.....	45
CAPÍTULO V: SOLUCIÓN TECNOLÓGICA.....		49
5.1	Flujo del negocio.....	49
5.1.1.	Modelo de casos de uso del negocio.....	49
5.1.2.	Actor del negocio.....	49
5.1.3.	Trabajadores del negocio.....	49
5.1.4.	Casos de uso del negocio.....	50
5.1.5.	Metas del negocio.....	51
5.1.6.	Entidades del negocio.....	52
5.1.7.	Diagrama de actividades.....	52
5.1.8.	Matriz de actividades y requerimientos.....	56
5.1.9.	Matriz de requerimientos adicionales.....	57
5.2.	Flujo de requerimientos.....	58
5.2.1.	Diagrama de casos de uso.....	58
5.2.2.	Especificaciones de caso de uso.....	59
a.	Especificación de caso de uso 01 - Gestionar cliente.....	59
b.	Especificación de caso de uso 02 - Gestionar solicitud.....	62
c.	Especificación de caso de uso 03 - Gestionar aprobación inicial.....	66
d.	Especificación de caso de uso 04 - Gestionar cotización.....	69
e.	Especificación de caso de uso 05 - Gestionar información de riesgos.....	75
f.	Especificación de caso de uso 06 - Gestionar aprobación de evaluación.....	81
g.	Especificación de caso de uso 07 - Gestionar aprobación final.....	85
h.	Especificación de caso de uso 08 - Buscar Cliente.....	89
i.	Especificación de caso de uso 09 - Buscar solicitud.....	91
j.	Especificación de caso de uso 10 - Exportar solicitud.....	93
k.	Especificación de caso de uso 11 - Exportar cotización.....	95
l.	Especificación de caso de uso 12 - Gestionar usuarios.....	97
m.	Especificación de caso de uso 13 - Gestionar perfiles de usuario.....	102
n.	Especificación de caso de uso 14 - Gestionar accesos por perfiles.....	106
o.	Especificación de caso de uso 15 - Ingresar al sistema.....	108
p.	Especificación de caso de uso 16 - Restablecer contraseña.....	111
5.3	Modelo de datos.....	113
5.4	Diagrama de componentes.....	115
5.5	Diagrama de despliegue.....	115
5.6	Diagrama de alto nivel.....	116

CAPÍTULO VI: RESULTADOS	117
6.1 Funcionalidad de la aplicación web	117
6.2 Usabilidad de la aplicación web	117
6.3 Eficiencia de la aplicación web	118
6.4 Mantenibilidad de la aplicación web	118
CONCLUSIONES	121
RECOMENDACIONES	122
REFERENCIAS BIBLIOGRÁFICAS	123
ANEXOS	125

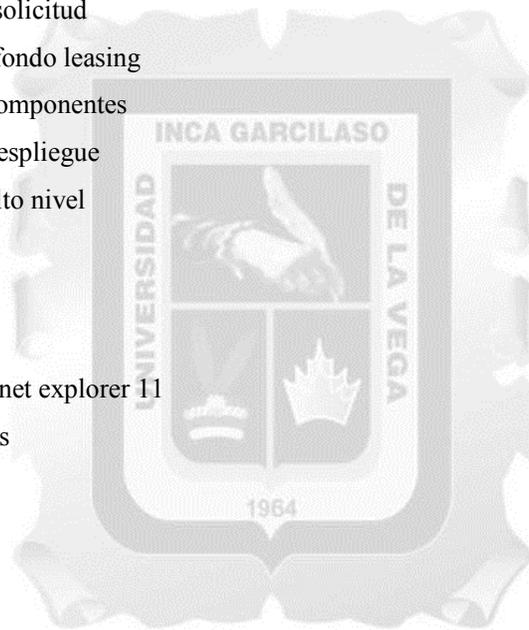


ÍNDICE DE FIGURAS

Figura 2.1: Arquitectura TCP/IP	19
Figura 2.2: Servidor web	20
Figura 2.3: Fases, iteraciones, disciplinas y organizaciones del RUP	30
Figura 2.4: Proceso Scrum	35
Figura 2.5: Elementos del contrato de arrendamiento financiero	38
Figura 4.1: Artefacto modelo de casos de uso del negocio	45
Figura 4.2: Artefacto actor del negocio	45
Figura 4.3: Artefacto caso de uso del negocio	46
Figura 4.4: Artefacto metas del negocio	46
Figura 4.5: Artefacto trabajadores del negocio	46
Figura 4.6: Artefacto realización de casos de uso del negocio	46
Figura 4.7: Artefacto entidades del negocio	46
Figura 4.8: Artefacto matriz de proceso, servicio y funcionalidad	47
Figura 4.9: Artefacto modelo de casos de uso	47
Figura 4.10: Artefacto actores	47
Figura 4.11: Artefacto casos de uso	47
Figura 4.12: Artefacto especificación de casos de uso	48
Figura 4.13: Artefacto modelo de datos	48
Figura 4.14: Artefacto diagrama de componentes	48
Figura 4.15: Artefacto diagrama de despliegue	48
Figura 5.1: Diagrama de casos de uso del negocio	49
Figura 5.2: Actor del negocio	49
Figura 5.3: Trabajadores del negocio	50
Figura 5.4: Casos de uso del negocio	51
Figura 5.5: Metas del negocio	51
Figura 5.6: Entidades del negocio	52
Figura 1.7: Diagrama de actividad - Proceso de solicitud de arrendamiento financiero	53
Figura 5.8: Diagrama de actividad - Proceso de evaluación de solicitud	54
Figura 5.9: Diagrama de actividad - Proceso de cierre de solicitud	55
Figura 5.10: Diagrama de casos de uso	58
Figura 5.11: Flujo básico - Buscar cliente	60
Figura 5.12: Sub flujo - Nuevo cliente	61
Figura 5.13: Flujo alternativo - Ingresar correo electrónico válido	61
Figura 5.14: Sub flujo - Nueva solicitud	64
Figura 5.15: Sub flujo - Modificar solicitud	64

Figura 5.16: Flujo alternativo - Ingresar descripción de empresa	65
Figura 5.17: Flujo alternativo - Ingresar descripción de activos	65
Figura 5.18: Flujo básico - Gestionar aprobación	67
Figura 5.19: Sub Flujo - Realizar aprobación inicial	68
Figura 5.20: Flujo alternativo - No se encontraron registros	68
Figura 5.21: Flujo básico - Cotizar solicitud	72
Figura 5.22: Sub flujo - Modificar cotización	72
Figura 5.23: Sub flujo - Aprobar cotización	73
Figura 5.24: Sub flujo - Registrar datos complementarios	73
Figura 5.25: Flujo alternativo - Ingresar cantidad para activos netos	74
Figura 5.26: Flujo alternativo - Especificar las consideraciones especiales	74
Figura 5.27: Flujo básico - Gestionar información de riesgos	77
Figura 5.28: Sub flujo - Registrar formulario de riesgos	78
Figura 5.29: Sub flujo - Registrar garantía	78
Figura 5.30: Sub flujo - Registrar covenant	79
Figura 5.31: Sub flujo - Enviar solicitud	79
Figura 5.32: Flujo alternativo - Especificar detalle de la garantía	80
Figura 5.33: Flujo alternativo – Especificar detalle del covenant	80
Figura 5.34: Flujo básico - Gestionar aprobación de evaluación	82
Figura 5.35: Sub flujo - Aprobar evaluación	83
Figura 5.36: Flujo alternativo - No se encontraron registros	83
Figura 5.37: Flujo alternativo - Especificar observación de desaprobación	84
Figura 5.38: Flujo básico - Gestionar aprobación final	86
Figura 5.39: Sub flujo - Realizar aprobación final	87
Figura 5.40: Flujo alternativo - No se encontraron registros	87
Figura 5.41: Flujo alternativo - Especificar observación de desaprobación	88
Figura 5.42: Flujo básico - Buscar cliente	90
Figura 5.43: Flujo alternativo - No se encontraron registros	90
Figura 5.44: Flujo básico - Buscar solicitud	92
Figura 5.45: Flujo alternativo - No se encontraron registros	92
Figura 5.46: Flujo básico - Exportar solicitud	94
Figura 5.47: Flujo alternativo - No se encontraron registros	94
Figura 5.48: Flujo básico - Exportar cotización	96
Figura 5.49: Flujo alternativo - No se encontraron registros	96
Figura 5.50: Flujo básico - Gestionar usuario	99
Figura 5.51: Sub flujo - Nuevo usuario	99
Figura 5.52: Sub flujo - Modificar usuario	100
Figura 5.53: Sub flujo - Eliminar usuario	100

Figura 5.54: Flujo alternativo - Ingresar el correo del usuario	101
Figura 5.55: Flujo básico - Nuevo perfil	103
Figura 5.56: Sub flujo - Modificar perfil	104
Figura 5.57: Sub flujo - Eliminar perfil	104
Figura 5.58: Flujo alternativo - Ingresar el nombre del perfil	105
Figura 5.59: Flujo básico - Gestionar accesos	107
Figura 5.60: Flujo alternativo - No se encontraron registros	107
Figura 5.63: Flujo básico: Ingresar al sistema	109
Figura 5.64: Flujo alternativo - Verificar si usuario y contraseña son correctos	109
Figura 5.65: Flujo alternativo - Especificar su contraseña	110
Figura 5.66: Flujo básico - Restablecer contraseña	112
Figura 5.67: Flujo alternativo - Clave anterior no coincide	112
Figura 5.68: Base de datos solicitud	117
Figura 5.69: Base de datos fondo leasing	118
Figura 5.70: Diagrama de componentes	119
Figura 5.71: Diagrama de despliegue	119
Figura 5.72: Diagrama de alto nivel	120
Figura 6.1: SQL 2012	123
Figura 6.2: Framework 3.5	123
Figura 6.3: Lenguaje C#.	123
Figura 6.4: Navegador internet explorer 11	124
Figura 6.5: Lista de registros	124



ÍNDICE DE TABLAS

Tabla 4.1: Las seis mejores prácticas RUP	40
Tabla 4.2: Artefactos RUP utilizados	44
Tabla 5.1: Matriz de proceso, servicio y funcionalidades	55
Tabla 5.2: Matriz de requerimientos adicionales	56



RESUMEN

Scotiabank Perú, con sede principal en Lima y agencias a nivel nacional, no cuenta con una plataforma tecnológica para la aprobación de arrendamientos financieros (leasing), este proceso se viene realizando de manera semiautomatizada mediante el uso de aplicaciones ofimáticas y archivos físicos como hojas de registro, además de, no contar con un servidor de base de datos para almacenar las solicitudes; lo que genera demora en el proceso, que conlleva a la insatisfacción en el cliente, que hace que se pierda la fidelidad y recomendación de este servicio del banco. En el presente trabajo de investigación desarrollamos una aplicación web para la mejora de la aprobación de arrendamientos financieros, para ello empleamos la metodología rational unified process (RUP), en su despliegue se implementa un servidor web con software privativo. Con esta solución se llega a mejorar el proceso actual, en temas de: fluidez, rapidez y almacenamiento centralizado de las solicitudes de arrendamiento financiero, obteniendo así la satisfacción final del cliente.

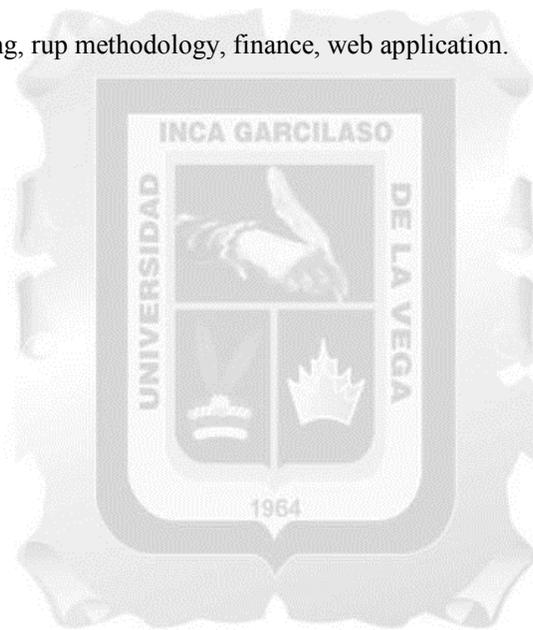
Palabras clave: Arrendamiento, leasing, metodología rup, finanzas, aplicación web.



ABSTRACT

Scotiabank Peru, headquartered in Lima and national agencies, does not have a technological platform for the approval of leasing, this process is carried out in a semi-automated way through the use of image applications and physical files such as sheets Registration, in addition to not having a database server to store the requests; What generates the delay in the process, which leads to dissatisfaction in the client, which causes that the loyalty and recommendation of this service of the bank is lost. In the present research work, a web application is developed for the improvement of the approval of financial leases, for the use of rational unified process methodology (RUP), in its deployment is implemented a web server with proprietary software. With this solution we can improve the current process, in terms of: fluidity, speed and centralized storage of financial lease requests, thus obtaining the final customer satisfaction.

Key words: Leasing, leasing, rup methodology, finance, web application.



INTRODUCCIÓN

El sector financiero en el país, ha ido creciendo a paso seguro, esto debido al auge de la economía, que, por ende, genera más comercio a nivel nacional e internacional. Las entidades financieras, como: bancos, cajas, etc; tratan de innovar sus servicios, para brindar opciones a los usuarios de unirse a su amplia lista de clientes. Uno de los servicios que actualmente se está haciendo más común como solicitud de los usuarios son los arrendamientos financieros o también conocidos como leasings financieros. Se trata de un contrato crediticio a partir de la adquisición de un bien por parte del banco para otorgárselo a un plazo acordado al cliente, en este plazo el cliente tendrá el derecho de uso sobre el bien siempre que cumpla con el pago de las cuotas y otras condiciones especificadas en el contrato. Cumplido el plazo del contrato, y al haber pagado todas las cuotas pactadas, el cliente podrá ejercer la opción de compra previamente acordada. Los empresarios ven esta opción como alternativa de poder adquirir productos para sus empresas, debido a que la mayoría de estos arrendamientos incluyen también el mantenimiento de los bienes que están adquiriendo durante el tiempo del contrato, así como la compra del mismo y además de renovar durante un nuevo contrato los equipos con los que cuentan actualmente, para así tener una continuidad y mejora en el negocio. Con el uso de las nuevas tecnologías, es necesaria una herramienta para tener un registro de estas solicitudes, que permita almacenar este proceso mediante la utilización de una aplicación web, en la cual se implementa un servidor web y de base de datos, usados mayormente en el desarrollo de este tipo de proyectos. Una de sus principales ventajas es su acceso desde cualquier navegador sin tener que instalar algún software adicional, lo cual sería provechoso para los usuarios de las entidades que se encuentran lejos de las sedes centrales. El trabajo de investigación a presentar consta de los siguientes capítulos:

Capítulo I: Donde se describe la situación problemática, el problema general, los problemas específicos, objetivo general y objetivos específicos.

Capítulo II: Donde se describe el marco teórico, además de los antecedentes, las bases teóricas y el glosario de términos relacionados al trabajo de investigación.

Capítulo III: Donde se describe las variables, indicadores e hipótesis que se obtuvieron de acuerdo al desarrollo de nuestro trabajo.

Capítulo IV: Donde se describe la metodología usada y su realización en el desarrollo de este trabajo.

Capítulo V: Donde se describe la solución tecnológica y los artefactos usados en el uso de la metodología escogida.

Capítulo VI: Donde se describen los resultados obtenidos del sistema que se desarrolló en el trabajo.

Finalizando con las conclusiones y recomendaciones que se obtuvieron del presente trabajo de investigación.

CAPÍTULO I: PLANTEAMIENTO DEL PROBLEMA

1.1 Situación Problemática

Scotiabank Perú, de capitales canadienses, con sede principal en la Av. Dionisio Derteano 102 - San Isidro, Lima y agencias en todo el territorio nacional; como todas las entidades financieras, cuenta con diversas áreas operativas, donde se realizan las diferentes solicitudes de los clientes, donde encontramos las solicitudes de arrendamientos financieros que se brindan a los usuarios. Debido al recorte del presupuesto en diferentes áreas de la entidad bancaria, por un tema de inversión en construcción de nuevas sedes de Lima y provincias, el área de Tecnologías de la Información (TI) no fue ajena a dicha acción, por lo que tuvo que prescindir de personal de sus diferentes sub-áreas, dentro de las cuales se encuentra desarrollo de TI, ya que son los encargados de crear, implementar, modificar nuevas soluciones para la entidad, que ha originado que los proyectos y las actividades se hayan visto reducidos. La ausencia de una plataforma tecnológica para la aprobación de arrendamientos financieros, origina que se venga trabajando de manera semiautomatizada con aplicaciones ofimática y de manera física, que además la información de los clientes y solicitudes no se encuentren centralizado mediante un sistema, por ende, el tiempo de respuesta hacia el usuario final no es la adecuada para generar niveles de satisfacción y fidelizar a nuevos clientes, lo que está resultando que el proceso de solicitud de arrendamiento financiero genere demoras en el tiempo de respuesta al futuro cliente del banco. El proceso comienza cuando el área comercial inicia el registro de solicitud(es), ellos se encargan de la aprobación inicial; la solicitud pasa al área de riesgos, que hace una evaluación crediticia del potencial cliente, luego presentan la(s) solicitud(es) aprobadas al comité de inversiones; el comité a su vez aprueba la línea de crédito del cliente; luego éste pasa al área legal para la preparación del contrato (previa validación en una notaría) para la posterior firma del cliente, ya firmado dicho documento el área de operación hace la aprobación y desembolso del arrendamiento financiero. La demora se ve reflejada en el trámite interno post negociación con el cliente, para llegar a la aprobación final del arrendamiento financiero, éste tiene que pasar por varias áreas, las cuales no llevan un orden y tiempo adecuado para la atención de las solicitudes, cuya atención puede demorar entre 15 y 20 días útiles, por solicitud, sin considerar la poca coordinación entre las áreas involucradas, lo cual se evidencia ante la falta de un sistema que permita mejorar los tiempos de atención a los clientes.

1.2 Problema de la investigación

1.2.1 Problema General

¿Cómo influye el desarrollo de una aplicación web en la mejora de la aprobación de arrendamientos financieros en el banco Scotiabank Perú?

1.2.2 Problemas Específicos

- ¿Cómo influye el nivel de funcionalidad en la mejora de la aprobación de arrendamientos financieros en el banco Scotiabank Perú?

- ¿Cómo influye el nivel de usabilidad en la mejora de la aprobación de arrendamientos financieros en el banco Scotiabank Perú?
- ¿Cómo influye el nivel de mantenibilidad en la mejora de la aprobación de arrendamientos financieros en el banco Scotiabank Perú?
- ¿Cómo influye el nivel de eficiencia en la mejora de la aprobación de arrendamientos financieros en el banco Scotiabank Perú?

1.3 Objetivos

1.3.1 Objetivo General

Determinar la influencia del desarrollo de una aplicación web para la mejora de la aprobación de arrendamientos financieros en el banco Scotiabank Perú.

1.3.2 Objetivos Específicos

- Determinar la influencia del nivel de funcionalidad en la mejora de la aprobación de arrendamientos financieros en el banco Scotiabank Perú.
- Determinar la influencia del nivel de usabilidad en la mejora de la aprobación de arrendamientos financieros en el banco Scotiabank Perú.
- Determinar la influencia del nivel de mantenibilidad en la mejora de la aprobación de arrendamientos financieros en el banco Scotiabank Perú.
- Determinar la influencia del nivel de eficiencia en la aprobación de arrendamientos financieros en el banco Scotiabank Perú.

1.4 Justificación

El desarrollo del presente trabajo de investigación consiste en una solución informática para la mejora de la aprobación de arrendamientos financieros en el banco Scotiabank Perú. Entre los principales beneficios se encuentra:

- Agilizar el proceso de aprobación de líneas de crédito para arrendamiento financiero en el banco Scotiabank Perú.
- Facilitar la operatividad de los usuarios, permitiendo reducir los tiempos de respuesta para la atención de las solicitudes.
- Reducir el tiempo de procesamiento de nuevos contratos.
- Disponer de una herramienta tecnológica que brinde la disponibilidad permanente de aprobación de arrendamientos financieros.
- Disponer de un medio de consulta de: clientes, solicitudes, cotizaciones, cronograma de pagos, usuarios.
- Sistematizar las solicitudes de arrendamiento financiero, permitiendo a las áreas acceder a dicha información.

- Uso de software licenciado cumpliendo los estándares establecidos por el banco Scotiabank Perú para el diseño/desarrollo de aplicaciones/sistemas de información utilizando licencias originales.

1.5 Alcance

En el desarrollo del trabajo se considera lo siguiente:

- Análisis, diseño y desarrollo de la aplicación web para la aprobación de arrendamientos financieros en el banco Scotiabank Perú.
- Adaptación de metodología RUP para el desarrollo de la aplicación.
- La implementación se realiza en el banco Scotiabank Perú.
- Capacitación al personal del banco en el uso de la aplicación web.

Así mismo, la aplicación comprende los siguientes módulos:

- Inicio (donde se muestran todas las solicitudes agregadas con sus estados respectivos).
- Comercial (donde los empleados de esa área registran nuevas solicitudes, clientes, cotizaciones y la aprobación inicial de las mismas).
- Riesgos (donde los empleados de esa área registran la información de riesgos del cliente evaluado previamente).
- Comité (donde los miembros del comité de inversiones registran la aprobación de la línea de crédito del cliente).
- Aprobación (donde los empleados del área de operaciones registran la aprobación y desembolso del monto final para culminar con la solicitud).
- Seguridad (donde el administrador del sistema crea los usuarios y los perfiles para determinar los niveles de accesos).
- Mantenimiento (donde el administrador del sistema realiza el mantenimiento de la aplicación, módulos).

CAPÍTULO II: MARCO TEÓRICO

2.1 Antecedentes de la investigación

Comprende la revisión de trabajos de tesis relacionado al tema:

- **Enríquez Chávez, G. (2015). *Efectos del arrendamiento financiero en la gestión de las medianas empresas mineras del Perú* (Tesis de maestría). Universidad de San Martín de Porres, Lima, Perú.**

El arrendamiento financiero es un contrato en la que la empresa arrendadora, banco o sociedad de arrendamiento financiero, adquiere de un tercero determinados bienes que otra empresa ha elegido, habiéndose acordado el precio y forma de pago, la empresa arrendadora entrega el bien para su uso durante un tiempo establecido, llamado periodo irrevocable, que generalmente coincide con la vida útil probable del activo, siendo todos los gastos, seguros y riesgos por cuenta del arrendatario. Al finalizar la investigación se concluyó que el arrendamiento financiero es una fuente de financiamiento que se origina por la necesidad que tiene una empresa de adquirir un bien, ya sea por aumentar su capacidad productiva, mejorar el servicio o implementar nuevos proyectos. En conclusión, se ha determinado que los efectos del arrendamiento financiero, inciden directamente en la gestión financiera de las medianas empresas mineras del Perú.

- **Rodríguez Medina, C. (2015). *Experiencia profesional en el área de leasing en el BCP durante los años 2010 al 2013* (Tesis de pregrado). Universidad Católica Santa María, Arequipa, Perú.**

El trabajo tiene como objetivo principal mostrar las técnicas, las metodologías, los estándares que se emplean para la gestión de proyectos de software en una entidad bancaria, que son tipos de empresas que soportan mucho de su plan estratégico en las tecnologías de información. Ya que, el volumen de tecnologías y aplicaciones usadas en el banco es amplio, el trabajo pone como caso de estudio uno de los servicios que brinda en su negocio e identifica sus relaciones con los sistemas de información, las tecnologías usadas, arquitectura usada, interrelación con otros sistemas, módulos y roles del aplicativo, relación del negocio con el aplicativo, el ciclo de desarrollo de software para proyectos nuevos al aplicativo y su importancia. El trabajo, además, hace un énfasis en la metodología del desarrollo de software utilizada que logra atender cientos de proyectos en paralelo sin descuidar la calidad y el éxito de cada uno de ellos. La definición de procesos en el área TI es fundamental para encaminar la calidad de las soluciones tecnológicas en base a reglas, buenas prácticas, monitoreo y control. Más aún cuando se elaboran múltiples proyectos al mismo tiempo ya que se tiende a perder el control a mayor cantidad. Dentro de la programación, la implementación de manuales sobre uso de estándares y buenas prácticas funciona para reutilizar conocimiento y mejorar las soluciones, además los estándares ayudan a entender de forma más veloz la solución elaborada, como si todos hablaran el mismo idioma.

- **Sancho Chacón, K. (2013). *Cotizador leasing: aplicación web y móvil para que profesionales realicen cotizaciones leasing en el BAC San José*. (Tesis de pregrado). Tecnológico de Costa Rica: Escuela de Ingeniería en Computación, San Carlos, Costa Rica.**

Se presenta los puntos más relevantes desarrollados durante el periodo de práctica supervisada, en donde teniendo como punto de partida: un problema, especificado más adelante, este estudio a fondo con el fin de encontrar los puntos más críticos o de consideración, luego se realizó el levantamiento de requisitos, diseño y finalmente el desarrollo del sistema que solucionó el problema. El problema que se presentó fue en la gerencia de banca de empresas del BAC San José en el cual no existía un sistema para realizar las cotizaciones leasings ágiles, además dar seguimiento a dichas cotizaciones y llevar las estadísticas. Esta gerencia en la actualidad con la única herramienta que cuenta es una hoja de cálculo, la cual como se sabe es una herramienta muy estática y que dificulta muchísimo el proceso. A raíz de este problema fue que nació el proyecto sobre el cual se basó la práctica de especialidad en la que fue posible aplicar conocimientos adquiridos durante el estudio universitario parte de esto fue el levantamiento de requisitos y el diseño del software, que iba a llevar la aplicación que como lo veremos más adelante busca la fácil manipulación del sistema, y la integridad de los datos, el sistema se basa en el lenguaje JavaScript fomentando la capacidad de respuesta de parte del servidor. Para el presente proyecto, se establecieron objetivos con el afán de llevar una meta clara durante el desarrollo del mismo, estos durante el tiempo establecido para el desarrollo de la práctica supervisada se fueron cumpliendo uno a uno sin embargo uno de los objetivos específicos aún no ha podido ser concluido ya que otro se extendió mucho y abarco tiempo establecido para el objetivo específico inconcluso. Una de las etapas necesarias para el desarrollo del proyecto y por consiguiente una buena implementación de la solución, fue la etapa de diseño sin duda, esta se estará tratando más adelante, veremos algunos puntos importantes como el modelo de subsistemas, diagramas de las clases y algunas de las interfaces de usuario que muestran parte importante del diseño de las aplicaciones actuales. Finalmente, en este documento se estarán realizando las conclusiones respectivas y se estará hablando de la gran experiencia que representa esta etapa: la entrada al mundo laboral, la cual es demasiado diferente a lo que hasta aquí se ha vivido en la universidad.

- **Saldaña Alarcón, M. (2012). *Propuesta de mejora del proceso de archivo de la documentación de leasing de una entidad financiera* (Tesis de pregrado). Pontificia Universidad Católica del Perú, Lima, Perú.**

Este trabajo surge cuando la entidad financiera decide entregar en outsourcing la gestión de los documentos relacionados a las operaciones leasing a una empresa peruana especializada en la administración de archivos. Previamente, los documentos eran administrados íntegramente por personal del banco. Asimismo, se detalla el impacto que producirá la implementación de las propuestas, no solo en los procesos del archivo de los documentos leasing, sino también en la operatividad del banco. Las conclusiones se dividen para la empresa de gestión de archivos y para la empresa financiera, de la cual se detallan las más importantes: la propuesta de crear un sistema de gestión de documentos que permita registrar los files almacenados, pues se concentra la información en una base de datos que permite llevar un control ordenado de las existencias del archivo, reduciendo los tiempos de búsqueda, el aplicativo también permitirá acceder a la información en el momento que lo requiera. Las empresas financieras necesitan concentrar sus recursos en la creación de mayores oportunidades de negocio y no

en labores administrativas que pueden ser entregadas a empresas especializadas en las labores de administración de archivos.

- **Gualán Saavedra, M. & Montúfar Cevallos, D. (2012), *Aplicación web para la gestión de microcréditos* (Tesis de pregrado). Universidad de Cuenca, facultad de ingeniería, escuela de informática, Cuenca, Ecuador.**

Crear una herramienta que permita gestionar los procesos necesarios para la comercialización de microcréditos. Para esto se busca desarrollar una plataforma robusta que aprovecha los beneficios de utilizar un manejo transaccional de procesos, una arquitectura modular del sistema y el empleo de tecnologías recientes en el manejo de persistencias y desarrollo de interfaces web en Java como son JPA y GWT. Con los resultados obtenidos se pretende dar soporte a los asesores de microcrédito que ejercen su trabajo en zonas alejadas de la ciudad, llegando a ofrecer el producto crediticio a clientes que no disponen de facilidades para obtener estos créditos por medios tradicionales. Los asesores requieren de herramientas como computadores portátiles, teléfonos inteligentes o tablets para realizar su trabajo de campo. El presente sistema tiene como objetivo funcionar sobre dichos dispositivos, facilitando a los asesores la gestión de microcréditos. Durante el desarrollo del proyecto se obtuvo una gran cantidad de experiencia en todas las áreas aplicadas y entre otras cosas se pudo constatar la utilidad, la eficacia, las ventajas y desventajas de las herramientas de software que se emplearon.

2.2 Bases teóricas

El desarrollo de las bases teóricas comprende: aplicación web, niveles de medición para el desarrollo de software, metodologías para el desarrollo de software, arrendamiento financiero.

2.2.1 Aplicación web

En ingeniería de software se denomina aplicación web al software que los usuarios pueden utilizar accediendo a un servidor web, vía internet, intranet o extranet, utilizando para ello un navegador web. Se hacen populares porque para interactuar con cualquiera de este tipo de aplicaciones no requiere la utilización de un software añadido, todas las prestaciones son accesibles a través de una única aplicación, el navegador web. Otra forma de definirlo sería como un conjunto de páginas web que interactúan unas con otras y con diversos recursos en un servidor web consigo mismo o con otros, incluidas bases de datos. La interacción permite implementar aplicaciones como catálogos de productos virtuales o tienda virtual, administración de noticias y contenidos (Talledo, 2015, p.71)

Desde la perspectiva de un usuario, puede ser difícil percibir la diferencia entre un sitio web y una aplicación web. Según el diccionario Oxford en línea, una aplicación es "un programa o conjunto de programas para ayudar al usuario de un ordenador para procesar una tarea específica". Una aplicación web es básicamente una manera de facilitar el logro de una tarea específica... en la Web, a diferencia de un sitio web estático que es más bien una herramienta, no menos importante, para la comunicación. El término más decisivo de esta definición es "tarea específica". La aplicación web por lo tanto permite al usuario interactuar directamente contigo y tus datos, todo en forma personalizada, para llevar a cabo esa tarea específica (Menéndez-Barzanallana, 2012).

Según Lujan (2002) es un tipo especial de aplicación cliente/servidor, donde tanto el cliente (navegador, explorador o visualizador) como el servidor (el servidor web) y el protocolo mediante el que se comunican (HTTP) están estandarizados y no han de ser creados por el programador de aplicaciones.

Las aplicaciones web usan la infraestructura de la web (protocolos, lenguajes) para su funcionamiento. Hoy en día las aplicaciones web han crecido hasta convertirse en grandes sistemas distribuidos complejos y que pueden atender a millones de usuarios en forma simultánea. Utilizan tecnología basada en web como son los navegadores y los servidores web. Esto permite tener el acceso a un mundo de aplicaciones por medio de una sola interfaz que es el navegador (Tahuiton, 2011)

a. Tipos de aplicaciones web

Según Tahuiton (2011) tenemos las siguientes:

- **Estáticas:** Este tipo de aplicaciones representan a la primera generación y están compuestas por páginas web estáticas, imágenes y texto, pero no cuentan con una lógica de negocio. Dentro de este tipo de aplicaciones se encuentran las páginas personales.
- **Orientadas a servicios:** Este tipo de aplicaciones intentan ofrecer un servicio especializado, por lo cual implementan una lógica de negocio acorde al servicio ofrecido. Durante su mantenimiento los desarrolladores necesitan comprender claramente la lógica de negocio. Un ejemplo de este tipo de aplicaciones son los servidores de correo electrónico.
- **De datos:** Este tipo de aplicaciones está enfocado a proveer una interfaz para acceder a una gran cantidad de datos y no en la lógica de negocio, por lo tanto, los desarrolladores necesitan comprender el flujo de datos. Un ejemplo de este tipo de aplicaciones son los catálogos en línea de las bibliotecas.
- **Sistemas de información:** Combinan las aplicaciones orientadas a servicios y de datos. Los desarrolladores necesitan comprender claramente el flujo de datos y la lógica de negocio (especialmente en la manipulación de los datos). Ejemplos de este tipo de aplicaciones son la banca en línea y los portales de comercio electrónico.

b. Características de las aplicaciones web

De acuerdo a Talledo (2015) y Tahuiton (2011) se encuentran:

- Aunque se basa en el modelo cliente/servidor su forma de interactuar es distinta a aquel. Una diferencia es que las comunicaciones entre el cliente y el servidor no son persistentes. No necesita mantener abierta la comunicación el cliente y el servidor. Solo se comunicarán cuando el cliente lo solicite y se mantendrá hasta que el servidor finaliza la entrega del o de los documentos que el cliente necesita.
- Es multiplataforma porque no requiere de software adicional para funcionar. Al realizarse todas las tareas a través de la interfaz del navegador web vale cualquier navegador para acceder a la aplicación web.
- No requiere conocimientos sobre comunicaciones, va implícito en el navegador y servidor.
- Son más accesibles: gracias al uso de protocolos estándar como HTTP toda aplicación web puede ser usada por toda la web sin importar ubicación y plataforma.

- Son más fáciles de mantener: desde el momento en que una aplicación se despliega en el servidor, los clientes pueden acceder a ella, sin necesidad de una aplicación cliente. Esto permite que cualquier cambio sólo se haga en el servidor y no en el cliente, ahorrando tiempos de desarrollo.

c. Protocolos

Entre los protocolos que intervienen en el uso de una aplicación web, según Andreu (2011) tenemos:

- **TCP/IP:** Conjunto de protocolos, una pila de protocolos, de los cuales los más conocidos son TCP (en el nivel de transporte) e IP (en el nivel de internet). Además de estos dos, hay más protocolos propios de la arquitectura TCP/IP, como por ejemplo FTP, HTTP, SMTP, UDP y muchos más. Con la expansión de internet, la arquitectura TCP/IP se ha convertido en una de las arquitecturas de redes más difundidas.
- **TCP (transmisión control protocol, protocolo de control de transmisión):** Su función principal es el transporte confiable (entregados en su destino, sin errores y en el orden que se transmitieron) y bidireccional de datos. Trabaja en la capa de transporte. Proporciona un mecanismo, el puerto lógico, que permite distinguir distintas aplicaciones en una misma máquina. Sus comunicaciones se dan entre dos equipos mediante circuitos virtuales o enlaces punto a punto con el modelo cliente-servidor.
- **IP (internet protocol, protocolo de internet):** Su función principal es el envío de paquetes de datos tanto a nivel local como a través de redes. Trabaja en la capa de red. Este protocolo no necesita configuración inicial de un equipo que no se haya comunicado con anterioridad y se basa en paquetes o datagramas cuya cabecera lleva información de las direcciones de las máquinas origen y destino.

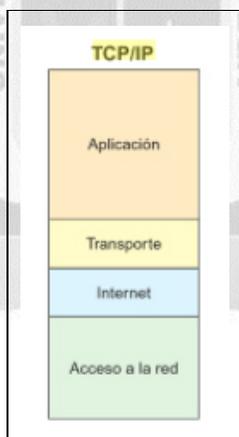


Figura 2.1: Arquitectura TCP/IP. Fuente: Barbancho, J. et al. (2014)

d. Servidores web

Es un programa diseñado para aceptar peticiones HTTP del navegador y servir las páginas web que tiene alojadas. Para navegar a un sitio web se necesita saber la dirección web exacta del sitio. Cada sitio tiene una dirección única conocida como URL. (Ramos & Ramos, 2014, p. 8)

Según Contreras, M. (2016.): “Podemos almacenar archivos en HTML y mostrar contenidos a los usuarios que dispongan de nuestra URL. Es aquí donde tenemos que tener la página web con la que el usuario posteriormente interactuará.” (p. 31)

Es un sistema distribuido que procesa una aplicación del lado del servidor. El código recibido por el cliente suele ser compilado y ejecutado por un navegador web. Para la comunicación entre el servidor y cliente se utiliza el protocolo de comunicación HTTP, perteneciente a la capa de aplicación del modelo OSI. (Yucra & Gonzales, 2012)

Es una aplicación informática del lado del servidor que realiza conexiones bidireccionales y/o unidireccionales y síncronas o asíncronas con el cliente generando una respuesta en cualquier lenguaje o aplicación del lado del cliente. (Talledo, 2015)

Según Yucra & Gonzáles (2012) un servidor web básico tiene un esquema de funcionamiento muy sencillo, ejecutando de forma infinita el bucle siguiente:

- Espera peticiones en el puerto TCP asignado.
- Recibe una petición de un cliente web.
- Busca el recurso en la cadena de petición.
- Envía el recurso por la misma conexión por donde ha recibido la petición.
- Vuelve a recibir otra petición.

Según Yucra & Gonzáles (2012) un servidor web que sigue este esquema cumple con los requisitos básicos de los servidores HTTP. Algunos servidores web importantes son: Apache, IIS, Cherokee, Tomcat, ighttpd, thttpd y muchos más. Es uno de los componentes más importantes del ecosistema web, también conocido como servidor HTTP, programa que procesa una aplicación del lado del servidor realizando conexiones bidireccionales y unidireccionales. El código recibido por el cliente suele ser compilado y ejecutado por un navegador web. Se componen básicamente de los siguientes componentes:
Núcleo: Es el servidor como tal, y es el centro del servidor web. Siempre está cargado y funcionando cuando el servidor está encendido.

Módulo: Los servidores web son bastante complejos. Estos módulos, conocidos como extensiones del servidor, se instalan cuando se configuran los servidores web.

Páginas: En una determinada carpeta están alojados las páginas webs donde está guardada la información, están estructurados en forma de árbol de directorios.

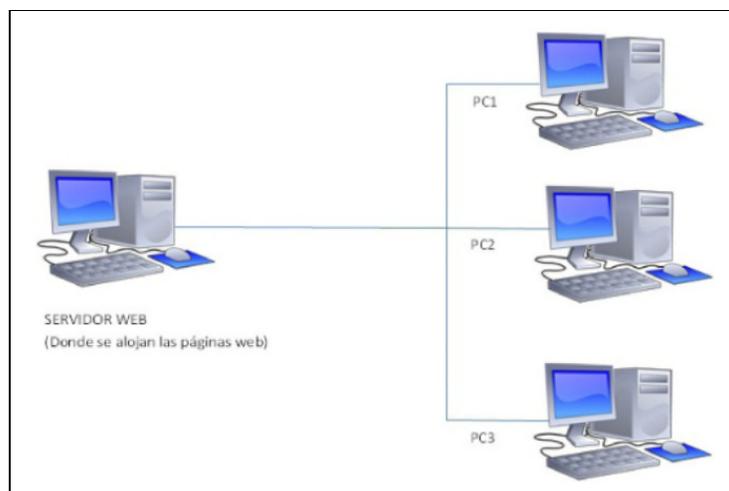


Figura 2.2: Servidor web. Fuente: Contreras, M. (2016)

2.2.2 Niveles de medición para el desarrollo de un software

Se dan definiciones para cada característica de calidad y las subcaracterísticas del software que influye en la característica de calidad. Para cada característica y subcaracterística, la capacidad del Software está determinado por un conjunto de atributos internos que pueden medirse. Ejemplos de las métricas se dan en ISO/IEC 9126-3. Las características y subcaracterísticas pueden medirse externamente por el grado en que la capacidad es proporcionada por el sistema que contiene el software. (ISO/IEC 9126-1:2001)

El ISO 9126 presenta seis características, las cuales tienen subcaracterísticas, estas son:

a. Funcionalidad

La capacidad del producto de software para proporcionar funciones que satisfacen las necesidades declaradas e implícitas cuando el software se utiliza bajo condiciones especificadas. (ISO/IEC 9126-1:2001)

Según Calero, Piattini & Moraga (2010) se subdivide en cinco características:

- **Adecuación.** La capacidad del producto software para proporcionar un conjunto apropiado de funciones para tareas específicas y objetivos de los usuarios.
- **Exactitud.** La capacidad del producto software para proporcionar los resultados o efectos correctos y con el grado de precisión acordado.
- **Interoperabilidad.** La capacidad del producto software para interactuar con uno más sistemas especificados.
- **Seguridad.** Referido a la capacidad del producto software para proteger la información y los datos.
- **Conformidad.** La capacidad del software para adaptarse a los estándares, convenciones o regulaciones en leyes y prescripciones relativas a la funcionalidad.

b. Fiabilidad

La capacidad del producto de software para mantener el nivel de performance especificado cuando es usado bajo las condiciones especificadas. (ISO/IEC 9126-1:2001)

Según Calero et al. (2010) se subdivide en cuatro características:

- **Madurez.** La capacidad del producto software para evitar fallos provocados por errores en el software.
- **Tolerancia a fallos.** La capacidad del producto software para mantener un nivel de rendimiento determinado en caso de defectos en el software o incumplimiento de su interfaz.
- **Recuperabilidad.** La capacidad del producto software para restablecer un determinado nivel de rendimiento y recuperar los datos afectados directamente en caso de ocurrir un fallo.
- **Conformidad.** La capacidad del producto software para adaptarse a estándares, convenciones y regulaciones referidas a la fiabilidad.

c. Usabilidad

La capacidad del producto de software para ser entendido, aprendido, utilizado y atractivo para el usuario, cuando se utiliza bajo condiciones especificadas. (ISO/IEC 9126-1:2001)

Según Calero et al. (2010) se subdivide en cinco características:

- **Comprensibilidad.** La capacidad del producto software para permitir al usuario que entienda si el software es adecuado, y como debe utilizarse para determinadas tareas y bajo ciertas condiciones de uso.
- **Facilidad de aprendizaje.** La capacidad del producto software para permitir al usuario aprender su aplicación.
- **Operabilidad.** La capacidad del producto software para permitir que el usuario lo opere y lo controle.
- **Atracción.** La capacidad del producto software para atraer al usuario.
- **Conformidad.** La capacidad del producto software para adaptarse a estándares, convenciones, guías de estilo y regulaciones relacionadas con la usabilidad.

d. Eficiencia

La capacidad del producto de software para proporcionar un rendimiento adecuado, en relación con la cantidad de los recursos utilizados, en las condiciones establecidas. (ISO/IEC 9126-1:2001)

Según Calero et al. (2010) se subdivide en tres características:

- **Comportamiento temporal.** La capacidad del producto software para proporcionar tiempos de respuesta y de procesamiento apropiados cuando realiza sus funciones bajo condiciones determinadas.
- **Utilización de recursos.** La capacidad del producto software para utilizar cantidades y tipos de recursos apropiados cuando el software realiza su función bajo determinadas condiciones.
- **Conformidad.** La capacidad del producto software para adaptarse a estándares o convenciones relacionados con la eficiencia.

e. Mantenibilidad

La capacidad del producto de software de ser modificado. Las modificaciones pueden incluir correcciones, mejoras o adaptación del software a los cambios en el ambiente, en los requisitos y especificaciones funcionales. (ISO/IEC 9126-1:2001)

Según Calero et al. (2010) se subdivide en cinco características:

- **Analizabilidad.** Capacidad del producto software de diagnosticar sus deficiencias o causas de fallos, o de identificar las partes que deben ser modificadas.
- **Cambiabilidad.** Capacidad del producto software de permitir implementar una modificación especificada. La implementación incluye los cambios en el diseño, el código y la documentación.
- **Estabilidad.** Capacidad del producto software de evitar los efectos inesperados de las modificaciones.
- **Facilidad de prueba.** Capacidad del producto software de permitir validar las partes modificadas.
- **Conformidad.** Capacidad del producto software de cumplir los estándares o convenciones relativas a la mantenibilidad.

f. Portabilidad

La capacidad del producto de software para ser transferido de un entorno a otro. (ISO/IEC 9126-1:2001)

Según Calero et al. (2010) se subdivide en cinco características:

- **Adaptabilidad.** Capacidad del producto software para ser adaptado para ambientes determinados sin realizar acciones o aplicar medios, más que los proporcionados para este propósito para el software considerado.

- **Facilidad de instalación.** Capacidad del producto software para ser instalado en un ambiente determinado.
- **Coexistencia.** Capacidad del producto software para coexistir con otro software independiente en un ambiente común compartiendo recursos.
- **Reemplazabilidad.** Capacidad del producto software para ser utilizado en lugar de otro producto de software para el mismo propósito en el mismo ambiente.
- **Conformidad.** Capacidad del producto software para adaptarse a estándares relacionados a la portabilidad.

2.2.3 Metodologías para el desarrollo de software

Al momento de crear un producto software, se necesita contar con una metodología para poder llevarlo a cabo, éstas se dividen en metodologías tradicionales y ágiles.

2.2.3.1. Metodologías tradicionales

Entre las más usadas tenemos **RUP (Rational Unified Process)**

Rational unified process, es más reconocido como un producto de software hecho por sus creadores y utilizado ya hace algunos años por IBM. Está muy bien orientado para proyectos que requieren infraestructuras grandes, ambientes myfriend muy grandes como el que proveen servidores de empresas multinacionales y exige un profundo conocimiento de sistemas de información, tecnologías IBM, avanzados conocimientos de integración. (Toro. 2013)

Según Wong & Torres (2010) RUP es un proceso de desarrollo de software y junto con el lenguaje unificado de modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. Asimismo, provee a través de un entorno web: Lineamientos, plantillas, workflows y herramientas, que guían una implementación efectiva de las mejores prácticas de la industria del software.

Según Tabares (2011) es una metodología de desarrollo de software que intenta integrar todos los aspectos a tener en cuenta durante todo el ciclo de vida del software, con el objetivo de hacer realizables tanto pequeños como grandes proyectos software. No es un sistema con pasos firmemente establecidos sino un conjunto de metodologías adaptables al contexto y necesidades de cada organización.

a. **Características**, según Molina & Quishpi (2015):

- Desarrollo de software iterativo
- Administración de requisitos
- Uso de arquitecturas basadas en componentes
- Software de modelado visual
- Verificación de la calidad del software
- Control de cambios en el software

Según Tabares (2011):

- **Guiado/manejado por casos de uso.** La razón de ser de un sistema software es servir a los usuarios, ya sean humanos y otros sistemas; un caso de uso es una facilidad que el software debe proveer a sus

usuarios. Los casos de uso constituyen la guía fundamental, establecida para las actividades a realizar durante todo el proceso de desarrollo, incluyendo el diseño, la implementación y las pruebas del sistema.

- **Centrado en la arquitectura.** La arquitectura involucra los elementos más significativos del sistema y está influenciada entre otros por plataformas software sistemas operativos, manejadores de bases de datos, protocolos, consideraciones de desarrollo como sistemas heredados y requisitos no funcionales. Es como una radiografía del sistema que estamos desarrollando, lo suficientemente completa como para que todos los implicados en el desarrollo tengan una idea clara de qué es lo que están construyendo, pero lo suficientemente simple. Se representa mediante varias vistas que se centran en aspectos concretos del sistema, abstrayéndose de lo demás. Todas las vistas juntas forman el llamado modelo 4+1 de la arquitectura, recibe este nombre porque lo forman las vistas lógica, de implementación, proceso y despliegue, más la de casos de uso que es la que da cohesión de datos.
- **Iterativo e incremental.** Para hacer más manejable un proyecto se recomienda dividirlo en ciclos. Para cada ciclo se establecen fases de referencia, cada una de las cuales debe ser considerada como un mini-proyecto, cuyo núcleo fundamental está constituido por una o más iteraciones de las actividades principales básicas de cualquier proceso de desarrollo.
- **Desarrollo basado en componentes.** La creación de sistemas de software necesita dividir el sistema en componentes con interfaces bien definidas, que posteriormente serán ensamblados para generar el sistema. Esta característica es un proceso de desarrollo permite que el sistema se vaya creando a medida que se obtienen, o que se desarrollan, y maduran sus componentes.
- **Utilización de un único lenguaje de modelado.** UML es adoptado como único lenguaje de modelado para el desarrollo de todos los modelos.
- **Proceso integrado.** Se establece una estructura que abarca los ciclos, fases, flujos de trabajo, mitigación de riesgos, control de calidad, gestión del proyecto y control de configuración. Además, esta estructura cubre a las herramientas para soportar la automatización del proceso, soportar flujos individuales de trabajo, para construir los diferentes modelos, e integrar el trabajo a través del ciclo de vida y a través de todos los modelos.

b. Elementos:

Según Kroll, P, & Kruchten, P. (2003), la estructura estática de RUP trata de cómo los elementos del proceso(actividades, disciplinas, artefactos y roles) se agrupan lógicamente en disciplinas de proceso centrales. Un proceso describe quién está haciendo qué, cómo y cuándo. Los cuatro elementos claves de modelado del RUP son:

- Roles, el quién
- Actividades, el cómo
- Artefactos, el qué
- Flujos de trabajo, cuando

- **Roles.** Un rol es como un “sombrero” que un individuo (o grupo) usa durante un proyecto. Un individuo puede usar muchos sombreros diferentes. Este es un punto importante porque es natural pensar en un rol como individuo en el equipo, o como un trabajo fijo, pero en la ruptura los roles simplemente definen como los individuos deben hacer el trabajo y especifican la competencia y responsabilidad que debe tener el individuo que desempeña ese papel. Una persona suele realizar una o más funciones, y varias personas pueden desempeñar el mismo rol.
- **Actividad.** Una actividad de un rol específico es una unidad de trabajo que un individuo en ese papel puede ser solicitado a realizar. La actividad tiene un propósito claro, usualmente expresado en términos de crear o actualizar algunos artefactos, como un modelo, un componente o un plan. Cada actividad se asigna a una función específica. Generalmente una actividad tarda unas pocas horas o unos pocos días en completarse, por lo general involucra a una persona, y afecta a uno o sólo a un pequeño número o artefactos. Una actividad debe ser utilizable como elemento de planificación y progreso, si es demasiado pequeña, será descuidada, y si es demasiado grande, el progreso tendría que ser expresado en términos de partes de una actividad. Las actividades pueden repetirse varias veces en el mismo artefacto -especialmente cuando se pasa de una iteración a otra, refinando una expansión del sistema- por el mismo papel, pero no necesariamente por el mismo individuo.

Pasos: Las actividades se dividen en pasos, que se dividen en tres categorías principales:

- Pasos de pensamiento, donde la persona que desempeña el papel entiende la naturaleza de la tarea, recoge y examina los artefactos de entrada y formula un resultado.
- Realizando pasos, donde el rol crea o actualiza algunos artefactos.
- Revisar los pasos, donde el rol inspecciona los resultados en función de algunos criterios.

No todas las etapas se realizan necesariamente cada vez que se invoca una actividad, por lo que los pasos pueden expresarse en forma de flujos alternativos.

- **Artefacto.** Un artefacto es una pieza de información que es producida, modificada o utilizada por un proceso. Los artefactos son los elementos tangibles del proyecto: las cosas que el proyecto produce o utiliza mientras trabaja hacia el producto final. Los artefactos se utilizan como entrada de roles para realizar una actividad y son el resultado o salida para otras actividades

Los artefactos pueden adoptar diversas formas:

- Un modelo, como el modelo de caso de uso o el modelo de diseño.
- Un elemento de modelo, es decir, un elemento dentro de un modelo, como una clase, un caso de uso o un subsistema.
- Un documento, como el caso de visión o negocio.
- Código fuente.
- Ejecutables, como un prototipo ejecutable.

Un artefacto puede ser documentado formalmente (usando una herramienta) o informalmente (capturado en un mensaje de correo electrónico o una pizarra blanca).

- **Flujo de trabajo (workflow).** Una simple enumeración de todos los roles, actividades y artefactos no constituye un proceso. Necesitan una manera de describir secuencias significativas de actividades que

producen algún resultado valioso y mostrar interacciones entre roles, esto es exactamente lo que hacen los flujos de trabajo.

Los flujos de trabajo vienen en diferentes formas, los dos flujos de trabajo más comunes son disciplinas, que son flujos de trabajo de alto nivel y detalles de flujo de trabajo, que son flujos de trabajo dentro de una disciplina.

En términos UML, un flujo de trabajo puede expresarse como un diagrama de secuencia, un diagrama de colaboración o un diagrama de actividad

- c. Según Toro (2013) divide en 7 **etapas o fases** el proyecto de desarrollo de aplicaciones de software:
- **Modelo del negocio (business modeling)**. El objetivo en esta etapa es hacer claro las reglas del negocio relativas al manejo de la información.
 - **Requisitos funcionales (requirements)**. En esta etapa el objetivo es diseñar los programas, módulos rutinas y demás componentes del sistema, buscando una arquitectura óptima del mismo.
 - **Implementación (implementation)**. Se instalan los bancos de datos y se montan igualmente las facilidades de comunicación de los programas e interfaces previstas.
 - **Pruebas (test)**. En esta fase se realizan todas las pruebas tanto a nivel de los módulos independientes como las resultantes de la integración de estos.
 - **Configuración y administración de cambios (configuration & change manage)**. El propósito aquí es llevar a cabo los cambios tanto en la configuración de equipos, servidores y programas, así como como en las diferentes interfaces.
 - **Instalación (deployment)**. El objetivo es poner en funcionamiento el producto del proyecto.
 - Cada una de estas etapas es desarrollada mediante un ciclo de iteraciones, lo cual consiste en reproducir el ciclo de vida en cascada a una menor y cada vez más menor escala. Los objetivos de una iteración se establecen en función de la evaluación que se haga de las cantidades y calidades de las iteraciones precedentes o precursoras.
- d. Según Toro (2013) el ciclo de vida que se desarrolla por cada iteración, es llevado a cabo bajo la guía combinada de dos **disciplinas** muy interrelacionadas, a saber:

Disciplina de desarrollo

- **Ingeniería de negocios**. Que consiste en entender las necesidades del negocio.
- **Análisis de requerimientos**. Trasladando las necesidades del negocio a un sistema automatizado de manejo de información.
- **Análisis y diseño**. Trasladando los requerimientos dentro de la arquitectura de software.
- **Implementación**. Creando software que se ajuste a la arquitectura y que tenga el comportamiento deseado.
- **Pruebas**. Asegurándose que el comportamiento requerido es el correcto y que todo lo solicitado está presente.

Disciplina de soporte

- **Configuración y administración del cambio.** Guardando todas las versiones del proyecto.
- **Administrando el proyecto.** Administrando horarios y recursos.
- **Ambiente.** Administrando el ambiente de desarrollo.
- **Distribución.** Hacer todo lo necesario para la salida del proyecto.

Según Tabares (2011), indica sobre el propósito y alcance de las disciplinas:

- **Modelado de procesos.** El propósito de la disciplina consiste en entender el negocio de la organización, y entender los procesos que se involucran para el desarrollo del producto. Su alcance es:
 - Evaluar el estado actual de la organización y su capacidad de adaptar nuevos sistemas, para apoyar procesos específicos de los cuales ha surgido una necesidad.
 - Explorar los procesos actuales del negocio, los roles y las responsabilidades.
 - Identificar y evaluar estrategias potenciales para modificar los procesos, si es necesario.
 - Desarrollar modelos de dominio que representen la unidad de negocio relacionada con el desarrollo de un nuevo proyecto.
- **Requisitos.** El propósito de la disciplina es establecer entre todos los interesados, un acuerdo frente a lo que el sistema pueda hacer y a su vez define el alcance total del sistema. Su alcance es:
 - Establecer y mantener un acuerdo sobre cuál es la funcionalidad esperada.
 - Entregar a los desarrolladores una descripción clara de los requisitos del sistema.
 - Definir los límites del sistema.
 - Definir el “look and feel” de la aplicación.
 - Proveer la base para la creación del plan de iteraciones, la estimación de costos y esfuerzos.
- **Análisis y diseño.** El propósito de la disciplina es transformar los requisitos de negocio en especificaciones de software, detallar y probar la arquitectura de la solución. Su alcance es:
 - Transformar los requisitos en el diseño del sistema a implementar.
 - Definir una arquitectura robusta para el sistema.
 - Entregar a los desarrolladores una especificación clara de los requisitos del sistema.
 - Adaptar el diseño al ambiente de implementación.
- **Implementación.** El propósito de la disciplina es codificar o configurar los componentes de software que conforman la solución para ensamblarlos y desplegarlos en unidades totalmente funcionales. Su alcance es:
 - Implementar la solución en términos de creación, codificación y personalización de componentes.
 - Definir la organización de componentes como subsistemas o módulos.
 - Probar los componentes como unidades aisladas.
 - Liberar módulos totalmente funcionales.
- **Pruebas.** El propósito de la disciplina es evaluar y determinar la calidad del producto. Su alcance es:
 - Buscar y documentar defectos encontrados en el software.
 - Determinar la calidad del software.
 - Validar la especificación de los requisitos frente al sistema liberado.

- Validar el diseño y arquitectura del producto de software.
- Validar la implementación de los requisitos.
- Asegurar que todos los defectos fueron corregidos antes de desplegar el producto.
- Asegurar la correcta ejecución del proceso de desarrollo.
- **Despliegue.** El propósito de la disciplina es asegurar que el producto quede disponible para su utilización, por parte de los usuarios finales del proyecto. Su alcance es:
 - Especificar la instalación del producto.
 - Probar la instalación del producto.
 - Crear el material para consulta y capacitación del usuario.
 - Validar el diseño del producto de software.
 - Asegurar la implementación de los requisitos suplementarios (requisitos no funcionales).
 - Entregar el producto funcional al usuario.
- **Gestión de la configuración y el cambio.** El propósito de la disciplina es la definición y conducción de los artefactos que deben ser controlados, dado su alto nivel de interacción dentro del proyecto. Su alcance es:
 - Controlar los cambios a los requisitos.
 - Controlar los cambios a los artefactos.
 - Mantener la integridad de los artefactos.
 - Versionar los artefactos.
 - Permitir conocer el estado de un artefacto.
- **Gestión del proyecto.** El propósito de la disciplina es proveer un marco de planeación, ejecución coordinación y control de todas las actividades que hacen parte del proyecto, además de unificar los canales de comunicación dentro del proyecto. Su alcance es:
 - Proveer un marco de trabajo para proyectos de implementación y desarrollo de software.
 - Proveer prácticas para la planeación, ejecución y monitoreo de proyectos.
 - Proveer un marco de trabajo para el manejo de riesgos.
- **Gestión del ambiente.** El propósito de la disciplina es proveer el soporte necesario para la implementación, describiendo los ambientes, las herramientas y la logística para facilitar la ejecución del proyecto. Su alcance es:
 - Proveer la infraestructura (hardware, software y comunicaciones) para la implementación.
 - Proveer las herramientas para la implementación.
 - Proveer las actividades, los artefactos y los lineamientos necesarios para el soporte del proceso.

e. Fases

De acuerdo a Quinatoa (2016) y Tabares (2011), el proceso de software diseñado por RUP se divide principalmente en cuatro fases, al completarlas se cumple un ciclo, logrando un producto para el cliente, estas son:

- **Inicio.** Es establecer un caso de negocio e identificar todas las entidades externas (personas y sistemas) que interactuarán con el sistema y definir estas interacciones, ya que esta información permite evaluar la aportación que el sistema hace al negocio.

Durante la fase de inicio se desarrolla una descripción del producto final y se presenta el análisis del negocio.

Objetivos:

- Especificar la visión final del producto y sus casos de negocios.
 - Afinar el alcance del proyecto.
 - Establecer los servicios de negocio a desarrollar.
- **Elaboración.** Es desarrollar una comprensión del dominio de problema, establecer un marco de trabajo arquitectónico para el sistema, desarrollar el plan del proyecto e identificar los riesgos del proyecto, al final de la fase se tendrá un modelo de los requerimientos del sistema (se detallan casos de uso UML), una descripción arquitectónica y un plan de desarrollo del software.

Durante la fase de elaboración se especifican en detalle la mayoría de los casos de uso del producto y se diseña la arquitectura.

Objetivos:

- Definir la arquitectura del sistema.
 - Definir y especificar los servicios.
 - Definir la interfaz gráfica del usuario.
- **Construcción.** Concierno el diseño del sistema, programación y pruebas, en el lapso de la fase se desarrollan e integran las partes del sistema, al final de la fase se debe tener un sistema software operativo y la documentación para entregar a los usuarios.

Durante la fase de construcción se crea el producto. La línea base de la arquitectura crece hasta convertirse en el sistema completo.

Objetivos

- Construir el producto, evolucionando la visión del negocio, la arquitectura y los planes; hasta que el producto esté listo para transferencia a la comunidad de usuarios.
 - Desarrollar los servicios, casos de uso (funcionalidades), módulos, subsistemas del proyecto.
 - Realizar desarrollo por iteraciones, priorizando los servicios, casos de uso, módulos y/o subsistemas más críticos para el negocio.
 - Realizar pruebas funcionales de cada uno de los servicios, casos de usos, módulos y/o subsistemas a medida que se vayan desarrollando.
- **Transición.** Se ocupa de mover el sistema desde la comunidad de desarrollo hacia la del usuario y hacerlo trabajar en un entorno real, pero se deja de lado en la mayor parte de los modelos de procesos del software ya que es una actividad de alto costo y a veces problemática y al final, se debe tener un sistema software documentado que funciona bien en su entorno operativo.

Durante la fase de transición se busca finalizar el producto para la entrega al usuario. Finalizada esta fase se obtiene una versión del producto.

Objetivo:

- Lograr la transición desde el producto a la comunidad de usuarios, la cual incluye: Manufactura, entrega, entrenamiento, soporte y mantenimiento del producto.

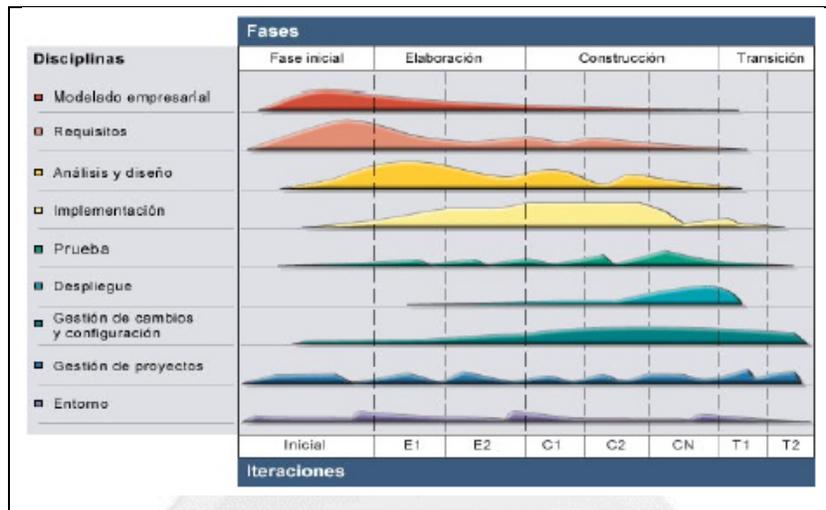


Figura 2.3: Fases, iteraciones, disciplinas y organizaciones del RUP. Fuente: Kruchten, P. (2004)

2.2.3.2. Metodologías ágiles

a. XP

Es una metodología complementaria a Scrum, ya que ésta se centra en la gestión y organización del proyecto, XP pone más énfasis en los procesos seguidos a la hora de producir código, dándole gran importancia a la adaptabilidad antes que la previsibilidad. (Molina & Quishpi, 2015)

- Según Quinatoa (2016) el ciclo de vida de XP se subdivide en las siguientes fases:
 - **Fase de la exploración.** Los clientes plantean rasgos de las historias de usuario para utilizar en la primera entrega del producto, y el equipo de desarrollo se familiariza con herramientas tecnológicas.
 - **Fase del Planeamiento.** Se priorizan las historias del usuario y se define el alcance del reléase y los programadores estiman el esfuerzo que requieren y de allí se define el primer reléase que no excede normalmente de dos meses, se deben incluir varias iteraciones para lograr un reléase y el cliente decide las historias para cada iteración y las pruebas funcionales las cuales se ejecutan al final de cada iteración, y en la última iteración el sistema está listo para producción.
 - **Fase de Producción.** Requiere pruebas y comprobación extra del funcionamiento del sistema antes de que éste se pueda liberar al cliente y en esta fase, los nuevos cambios son todavía encontrados y debe tomarse la decisión de si se incluyen o no en el reléase actual.
 - **Fase de Mantenimiento.** Requiere de un mayor esfuerzo para satisfacer también las tareas del cliente y la velocidad del desarrollo puede desacelerar después de que el sistema esté en la producción.
 - **Fase de Muerte.** Es cuando el cliente no tiene más historias para ser incluidas en el sistema y se han satisfecho sus necesidades, se ha cubierto aspectos como rendimiento y confiabilidad del sistema y se genera la documentación final, y no se realizan más cambios en la arquitectura o a su vez ocurre cuando el sistema no genera los beneficios esperados por el cliente o no hay dinero para mantenerlo.

- Según Toro (2013, p.24) Xtreme programming cuenta con los siguientes roles:
 - **Programador.** Escribe las pruebas unitarias y produce el código del sistema.
 - **Ciente.** Escribe las historias de usuario y las pruebas funcionales para validar su implementación. Además, asigna la prioridad a las historias de usuario y decide cuáles se implementan en cada iteración centrándose en aportar mayor valor al negocio.
 - **Encargado de pruebas (tester).** Ayuda al cliente a escribir las pruebas funcionales. Ejecuta las pruebas regularmente, difunde los resultados en equipo y es responsable de las herramientas de soporte para pruebas.
 - **Encargado del seguimiento (tracker).** Proporciona realimentación al equipo. Verifica el grado de acierto entre las estimaciones realizadas y el tiempo real dedicado, para mejorar futuras estimaciones. Realiza el seguimiento del progreso de cada iteración.
 - **Entrenador (coach).** Es responsable del proceso global. Debe proveer guías al equipo de forma que se apliquen las prácticas XP y se siga el proceso correctamente.
 - **Consultor.** Es un miembro extremo del equipo con un conocimiento específico en algún tema necesario para el proyecto, en el que puedan surgir problemas.
 - **Gestor (big boss).** Es el elemento humano que vincula los clientes y los usuarios con los programadores, colaborando a fin de que el equipo trabaje efectivamente y creando las condiciones adecuadas. Su labor esencial es de coordinación.
- Según Molina & Quishpi (2015) esta metodología sigue cinco máximas que considera necesarias para crear un buen software:
 - **Simplicidad.** Es la base de la programación extrema, que busca simplificar y clarificar el código al máximo para así agilizar el desarrollo y facilitar el mantenimiento. De esta manera se intenta evitar la complejidad generada al aumentar el tamaño del código y disminuir la cantidad de bugs que aparecen por ello.
 - **Comunicación.** Se refiere a cómo se comunica el código con el programador. Si este es complejo es mucho más difícil de entender mientras que sí es claro puede considerarse autodocumentado ya que es fácil conocer su funcionalidad. Las pruebas unitarias son otra forma efectiva de comunicación ya que describen el diseño de las clases y métodos al mostrar ejemplos de su funcionalidad.
 - **Retroalimentación.** Es muy importante que haya retroalimentación continua por parte del cliente ya que de esta forma se evita tener que volver a programar sesiones grandes del proyecto. El tener ciclos de desarrollo cortos ayuda de gran manera a este principio.
 - **Coraje.** Se refiere a la capacidad de programar pensando únicamente en las necesidades actuales, de manera que no se pierda tiempo planificando a futuro con funcionalidades que pueden no ser necesarias. El coraje permite que los programadores se sientan cómodos al modificar su código para adaptarlo a nuevas necesidades.
 - **Respeto.** El respeto se manifiesta de varias formas. Los miembros del equipo se respetan los unos a los otros, porque los programadores no pueden realizar cambios que hacen que las pruebas existentes fallen o que demore el trabajo de sus compañeros. Los miembros respetan su trabajo porque siempre están

luchando por la alta calidad en el producto y buscando el diseño óptimo o más eficiente para la solución a través de la refactorización del código. Los miembros del equipo respetan el trabajo del resto no haciendo menos a otros, una mejor autoestima en el equipo eleva su ritmo de producción.

- Según Láinez (2015, p.119-122) tiene una docena de prácticas derivadas de sus valores y son las siguientes:
 - **Ciente presente.** Uno de los paradigmas del desarrollo de software tradicional es que el cliente no necesita, o incluso no debe estar presente durante el proceso de desarrollo. La XP busca acabar con ese paradigma, haciendo que la presencia del cliente sea de vital importancia para el éxito del proyecto. El feedback que el cliente suministra es parte esencial de una iteración. Para que el cambio de feedback pueda ocurrir y el cliente pueda obtener el máximo de valor del proyecto, es esencial que este participe activamente del proceso de desarrollo.
 - **Juego de planificación.** En el inicio de cada iteración el cliente es invitado a escribir, a su manera, las funcionalidades que desea en el sistema, en pequeñas tarjetas llamadas historias de usuario, esa es la cantidad más pequeña de información que este puede especificar, Conocedor del tiempo y del coste, el cliente debe decidir el orden en que cada historia de usuario será desarrollada, es decir, este prioriza el desarrollo de las funcionalidades. Esta práctica es conocida como el juego de la planificación y asegura que el equipo trabaje en el que considera más importante para el cliente.
 - **Programación en par.** En la programación en par dos desarrolladores escogen una historia de usuario y se sientan en un único ordenador para codificar una determinada funcionalidad. El desarrollador con menos experiencia tiene como responsabilidad asumir el control del teclado y conducir activamente la programación del código fuente, mientras el otro con mayor experiencia inspecciona el código en busca de errores y defectos, cuestionando las decisiones y buscando estratégicamente las soluciones más simples para el código. Uno de los beneficios de esa práctica es la revisión constante del código, y también la diseminación del conocimiento entre pares, lo que ayuda a la nivelación técnica de todo el equipo.
 - **Releases cortos.** Esta práctica busca, por medio de releases cortos, entregar versiones actualizadas del software al cliente a lo largo del proceso de desarrollo.
 - **Desarrollo guiado por las pruebas.** Los desarrolladores escriben pruebas para cada funcionalidad antes de codificarlas. De esta forma las interfaces externas de los métodos y de las clases son planeadas antes de codificación. Esta práctica genera una masa de pruebas que puede ser usada en cualquier momento para validar todo el sistema.
 - **Refactoring.** El refactoring es el proceso de reorganizar el código fuente de un software para mejorar su calidad interna, facilitar la lectura y disminuir el tiempo desperdiciado con el mantenimiento, sin perjudicar el rendimiento y modificar su comportamiento externo. Esa técnica es fundamental para hacer el código más legible y detectar errores en ciertos algoritmos.
 - **Código colectivo.** En la XP, el sistema no está segmentado en partes. Los desarrolladores tienen acceso a todas las partes del código y pueden modificar aquello que juzguen importante: El código es colectivo. Eso suministra mayor agilidad al proceso y crea más un mecanismo de revisión y de verificación del

código, ya que aquello que es escrito por un par, acaba siendo manipulado por otro. Si alguna cosa está confusa en el código, este pasará por el refactoring.

- **Código estandarizado.** Para que todos los desarrolladores puedan manipular cualquier parte del software, de forma más rápida, el equipo establece estándares de codificación, que sirven también para hacer el sistema más homogéneo, permitiendo que cualquier mantenimiento futuro sea efectuado más rápidamente.
- **Integración continua.** La práctica de la integración continua es la actividad de unir el trabajo realizado por un par de programadores, al código como uno todo. Después de terminar determinada actividad, el par debe probar y juntar su código a la versión más reciente del código colectivo, Eso debe ser hecho varias veces al día para sincronizar las actividades individuales.
- **Ritmo sustentable.** Esa práctica consiste en trabajar respetando los límites físicos y demostrando respeto por la individualidad. Para eso, la XP recomienda que la carga horaria de trabajo no pase de las 8 horas diarias y 40 horas semanales.
- **Metáfora.** Esta práctica utiliza comparaciones que permiten al equipo transmitir ideas de modo que todos la entiendan. Muchas veces un miembro del equipo intenta explicar determinada actividad, pero no consigue ser entendido, por lo que el trabajo queda comprometido. El uso de metáforas posibilita transmitir la idea de modo de aclaración para los oyentes. Esta práctica facilita la comunicación entre desarrollador y el cliente, estableciendo un vocabulario común entre ambos.
- **Stand up meeting.** El equipo de desarrollo se reúne cada mañana para evaluar el trabajo que fue ejecutado el día anterior y priorizar aquello que será implementado el día que se inicia. Se trata de una reunión rápida que recibe el nombre de stand up meeting, que significa reunión en pie.

b. Scrum

Según Toro (2013, p.26-27) es especial para proyectos que están expuestos a sufrir rápidos cambios de los requisitos funcionales. Sus principales características son: El desarrollo de software se realiza mediante iteraciones, denominadas sprints, con una duración aproximada de unos 30 días. Otra característica es sostener reuniones tipo usuario-programador a lo largo de todo el proyecto, entre ellas destaca la reunión diaria de 15 minutos del equipo de desarrollo para una necesaria coordinación e integración. Para tener éxito con los métodos agile y scrum y según la complejidad de los proyectos, se requiere de un proceso de gestión coherente que incluya información para comprender los diversos roles involucrados en estos proyectos y los pasos para completar las tareas esenciales.

Es un framework o conjunto de buenas prácticas para la gestión de proyectos. Se basa en un estudio realizado sobre los procesos de desarrollo utilizados con éxito en Canon, Xerox, Honda o HP. En este estudio se comprobó que estos equipos partían de unos requisitos muy generales para sus productos y que además debían salir al mercado en muy poco tiempo. Eran equipos altamente productivos que seguían unos patrones muy similares. Se basa en entregas parciales y regulares del producto final, comenzando por aquellas funcionalidades más importantes para el cliente, por lo que Scrum está especialmente indicado para proyectos donde se necesita obtener resultados pronto o donde los requisitos son cambiantes y la

competitividad, la flexibilidad y la productividad son cruciales (¿dónde no lo son?). Esto permite no alargar demasiado las entregas y reaccionar antes de que los costes se disparen. (Martel, 2014)

- Según Martel (2014) cuenta con los siguientes roles:
 - **Dueño del producto.** Establece y prioriza la lista de funcionalidades a desarrollar en el proyecto (la pila del producto) de acuerdo con las necesidades de quien paga por el producto. Aporta la perspectiva del cliente y lo que éste necesita. Se encarga además de describir las tareas o funcionalidades que se recogen en la pila del producto y en la pila del sprint. Este papel lo juega normalmente alguien elegido por nuestro cliente para representar a todas las personas interesadas en obtener un producto final que les sea útil (usuarios, gestores o directivos del cliente). Un dueño del producto que define con claridad las funcionalidades que requiere del producto, que conoce bien su negocio y que mantiene unos objetivos claros, sencillos y lejos de complejidades innecesarias.
 - **Scrum master.** No es exactamente el jefe de proyecto tradicional. Ese rol se reparte en el propio scrum master y el dueño del producto. Su misión más importante es la de proteger al equipo de interrupciones mientras trabajan para completar el sprint y resolverles cualquier incidencia u obstáculo que les impida cumplir la meta del sprint. Preparará las reuniones y se asegura de que sean productivas. Asignará también las tareas al equipo de trabajo y hará un seguimiento de las ya asignadas.
 - **Equipo de trabajo.** Son los miembros del equipo que tienen la responsabilidad de entregar el producto. Al final de cada sprint deben entregar las historias de usuario revisadas y comprobadas. En caso fallarán en la reunión de demo.
- Según Molina & Quishpi (2015) el sprint es considerado como el corazón del scrum. Es un bloque de tiempo de un mes, o menos, de duración durante del cual se crea un incremento del producto terminado. Un sprint consiste en:
 - **Reunión de planificación de sprint (sprint planning meeting).** Es donde se planifica el trabajo a realizar durante el sprint. La reunión de planificación de sprint tiene una duración máxima de ocho horas para un sprint de un mes. Para sprints más cortos, el evento es usualmente más corto.
 - **Scrum diario (daily scrum).** Es una reunión diaria, de 15 minutos de duración, que tiene el objetivo de facilitar la transferencia de la información, la colaboración entre los miembros del equipo y se ponen de manifiesto puntos en la que se pueden ayudar unos a otros. Los objetivos del scrum diario son aumentar la productividad en el proyecto, potenciar el compromiso de equipo, fomentar el aprendizaje de los miembros del equipo y conocer el estado del sprint.
 - **Revisión de sprint (sprint review).** Se trata de una reunión informal donde el equipo presenta al cliente los requisitos completados en el sprint. En la revisión del sprint, el cliente puede ver de manera objetiva cómo han sido desarrollados los requisitos que proporcionó y ver si se cumplen sus expectativas.
 - **Retrospectiva de sprint (sprint retrospective).** El objetivo de la retrospectiva es el de mejorar de manera continua la productividad y la calidad de producto que está desarrollando, se analiza cómo ha sido la forma de trabajar durante el sprint y porqué se están consiguiendo o no los objetivos.
 - Cada sprint tiene una definición sobre qué se va construir, un diseño y un plan flexible que guiará la construcción, el trabajo y el producto resultante.

- **Lista de producto (product backlog).** Se trata de un listado que contiene los requisitos del proyecto, expresados en forma de historias del usuario. El cliente es el responsable de crear y gestionar la lista, asignando una importancia a cada historia de usuario para priorizarlas sobre las demás.
- **Lista de pendientes del sprint (sprint backlog).** Es una lista de tareas que el equipo elabora en la reunión de planificación como un plan para completar los requisitos seleccionados. El sprint backlog pertenece únicamente al equipo de desarrollo. Es actualizado a medida que el trabajo se completa.
- **El tablero de tareas (scrum taskboard).** Para poder gestionar la línea de historias del producto backlog y hacerla visible para el equipo de desarrollo, se puede realizar un tablero de tareas que puede ser dibujado en una pizarra y que actúa como un espacio donde el equipo puede interactuar e indicar su estado de progreso en el sprint.
- **Puntos de historia.** Son unidades de medida empleadas para medir y estimar el tiempo de desarrollo de las historias del usuario, su objetivo es catalogar la dificultad de las tareas y no se pueden comparar a horas de esfuerzo. Un punto de historia equivale a una jornada de trabajo de un miembro del equipo en dedicación exclusiva.
- Según Herranz (2016) se basa en cinco valores:
 - **Enfoque.** Permite lograr una sinergia entre los miembros del equipo que permite realizar un trabajo excelente y entregar valor más temprano.
 - **Coraje.** Cada miembro del equipo debe sentirse apoyado por el resto y, de este modo, lograrán tener el coraje para aceptar grandes retos.
 - **Transparencia.** Los equipos deben expresar con sinceridad cómo van, que se están encontrando en su camino y cuáles son sus preocupaciones de modo que estas pueden ser atendidas.
 - **Compromiso.** Al tener más control sobre su propio destino, los miembros del equipo están más comprometidos con el éxito del proyecto.
 - **Respeto.** A medida que trabajan juntos, compartiendo éxitos y fracasos, el respeto y la ayuda brindada al resto de compañeros hace a los miembros del equipo dignos de respeto.

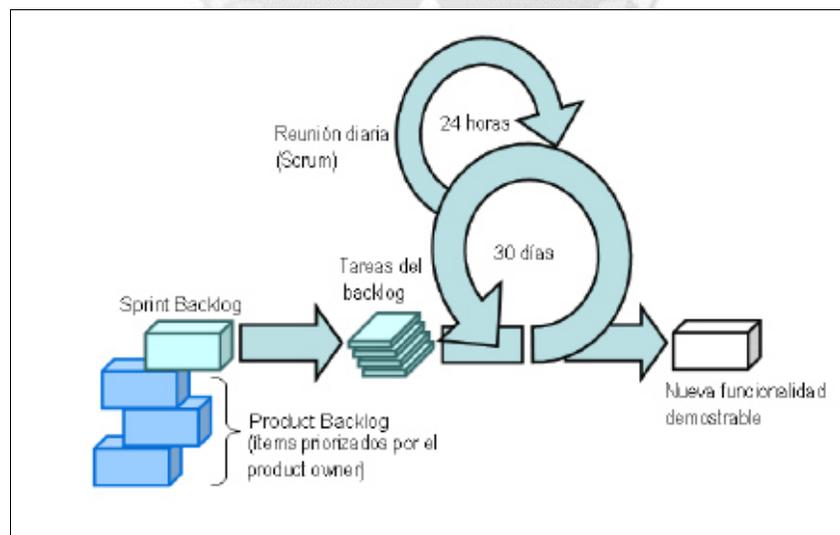


Figura 2.4: Proceso Scrum. Fuente: Tabares (2011)

- Según Quinatoa (2016) los artefactos más importantes en Scrum son:
 - **Lista de Objetivos/ Requerimientos Priorizada.** Este documento representa lo que el cliente espera del proyecto ya sea a objetivos y requerimientos, así como entregas o sprints y detalle de riesgos. El cliente se encarga de crear y gestionar esta lista, con ayuda de un facilitador.
 - **Lista de Tareas de la Iteración.** Refleja la totalidad de las tareas de iteración o Sprint en curso, con el fin de cumplir los objetivos o requerimientos, y entregar algo funcional al cliente acorde con lo esperado.
 - **Gráficos de Trabajo Pendiente.** Reflejan la velocidad con la que avanza el proyecto, permitiendo una vista general de la rapidez con la que el proyecto en general o el Sprint en curso en particular están avanzando.

2.2.4 Arrendamiento (leasing) financiero

En el desarrollo del trabajo también se usará el término traducido al inglés: “leasing”, debido a que el banco y sus trabajadores usan más este término para referirse a este tipo de operación.

“De Grosso (2001) interpreta al leasing, en primer lugar, como una operación financiera, y que al ser el tomador quien elige por sí mismo el bien del proveedor, la propiedad del dador queda despojada prácticamente de todos sus atributos normales, por lo que el interés del dador en la operación pasa ser puramente financiero y se limita a recuperar el monto de su inversión en capital; y en segundo lugar, que el contrato de locación entre el dador y el tomador se concluye por un período que toma en cuenta el tiempo que insume la amortización económica de la cosa, de donde resulta la naturaleza puramente financiera de la operación para el dador.” (Pérez Paputsachis, 2016, p.143)

Es un contrato mercantil que celebra una empresa locadora (arrendadora) para el uso de un bien mueble o inmueble por la arrendataria, a cambio de pagos periódicos que pueden ser por adelantado o al vencimiento y con opción a favor de la arrendataria para comprar dichos bienes por un valor previamente pactado. Mediante este contrato de leasing financiero, la empresa arrendadora, banco o sociedad de arrendamiento financiero, adquiere de un tercero determinados bienes que otra empresa ha elegido con anterioridad, habiéndose acordado previamente el precio y la forma de pago. La empresa arrendadora entrega el bien para su uso durante un tiempo establecido, llamado periodo irrevocable, que generalmente coincide con la vida útil probable del activo, siendo todos los gastos, seguros y riesgos por cuenta del arrendatario. (Mávila, 2003)

Se define como un tipo de arrendamiento en el que se transfieren sustancialmente todos los riesgos y ventajas inherentes a la propiedad del activo. También se transfiere el derecho de uso de activos, incluso en el caso de que el arrendador quedará obligado a suministrar servicios de cierta importancia en relación con la operación o el mantenimiento de los citados bienes. Entre los riesgos y ventajas inherentes a la propiedad del activo se encuentra la expectativa de operación; ganancia por revalorización; pérdidas por capacidad ociosa; obsolescencia tecnológica; variaciones en el desempeño (Díaz & Vásquez, 2013, p.235)

El arrendamiento financiero ha sido, tradicionalmente, una opción que puede sustituir a los créditos especializados, como el hipotecario, refaccionario simple o con garantía hipotecaria; su versatilidad lo ha llevado a financiar con plazos medianos y largos, equipos, inmuebles, unidades industriales completas, embarcaciones, aeronaves y cualquier otro equipo que, al amparo de la viabilidad del proyecto, sea susceptible de financiamiento. (Haime, 2003)

Las normas contables del arrendamiento financiero están regidas por la NIC (Norma Internacional de Contabilidad) N°17 "Tratamiento Contable de los Contratos de Arrendamiento", el FASB (Financial Accounting Standard Board) N° 13 "Contabilización de los Arriendos" y por la ley del Impuesto a la Renta. (Mávila, 2003)

a. Etimología y denominación

Según (Piña & Del Águila, 2014) la palabra leasing, de origen anglosajón, deriva del verbo inglés "to lease", que significa arrendar o dar en arriendo, y del sustantivo "lease" que se traduce como arriendo, escritura de arriendo, locación, etc. Así tenemos algunas derivaciones:

- En Estados Unidos "leasing"
- Francia "credit-bail", "équipement-bail", "location financiere avec promesse de vente"
- Bélgica "location-financement"
- Italia "locazione finanziaria"
- España "arrendamiento financiero"
- Brasil "arrendamiento mercantil"
- Argentina "locación de bienes de capital o locación financiera"
- Uruguay "arrendamiento financiero" y "crédito de uso"
- Perú "arrendamiento financiero"

b. Características del arrendamiento (leasing) financiero

Según Haime (2003):

- Es una herramienta de negociación ante proveedores de equipo, al tener la posibilidad de pagar de contado y negociar mejores precios, lo que puede reducir el costo total de la operación.
- El tratamiento fiscal está específicamente regulado en las leyes de la materia.
- Se conservan los escasos recursos monetarios de la empresa, al no requerir pagos anticipados importantes al inicio de la operación.
- A diferencia del arrendamiento puro, en el financiero se tiene la posesión y propiedad del bien, lo cual se formaliza al ejercer la opción de compra al final del contrato.
- Se dejan libres y sin afectación otras fuentes de financiamiento, pues, por lo general, los bienes financiados con el arrendamiento financiero sirven de garantía para el crédito obtenido, sin necesidad de comprometer otros bienes.
- Se puede recuperar la liquidez de la empresa por medio del arrendamiento ficticio (sale and leaseback).

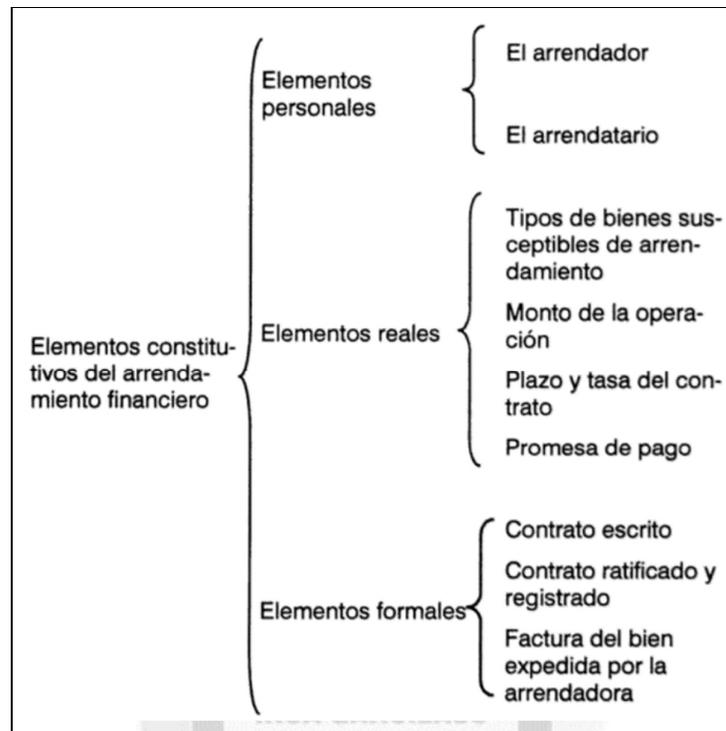


Figura 2.5: Elementos del contrato de arrendamiento financiero. Fuente: Haime (2003)

2.3 Glosario de términos

- **Arrendador:** Es aquel que se obliga a conceder el uso o goce temporal de los bienes solicitados por el arrendatario, y que son propiedad legal del primero. (Haime, 2003)
- **Arrendamiento:** Acción de arrendar, contrato por el cual se arrienda, precio en que se arrienda. (Diccionario de la lengua española, Real Academia Española - RAE)
- **Arrendatario:** Es la persona que adquiere el derecho a usar un activo a cambio del pago de un canon. Es la persona que toma en arriendo cualquier cosa. Persona física o jurídica que alquila un bien. (Diccionario de la lengua española, Real Academia Española - RAE)
- **Contrato de arrendamiento:** Contrato que se establece entre dos partes y que supone que la primera (el arrendador) le entrega algún elemento suyo (mueble o inmueble) a la segunda parte (el arrendatario) para que la utilice en su beneficio propio. Este contrato supone que la segunda parte debe pagar por ese préstamo a manera periódica a lo largo del tiempo que se establezca de común acuerdo en el contrato. En algunos casos, el pago puede ser en dinero y en otros casos puede ser otorgándosele al arrendador parte del producto que el arrendatario logra a partir del uso de ese espacio o elemento. (Piña & Del Águila, 2014)
- **Correspondencia externa:** correspondencia remitida a los clientes o proveedores. (Saldaña, 2012)
- **Correspondencia interna:** Correos electrónicos enviados entre personal del banco relacionados a la operación leasing. (Saldaña, 2012)

- **Covenants:** Conjunto de restricciones y de obligaciones que se imponen al deudor para que la posición de los acreedores, que han aceptado prestar en consideración de una situación financiera del deudor y de una posición otorgada a los acreedores, no se deteriore indebidamente, más allá de los riesgos inevitables que han asumido razonablemente. (Gestión, 2015)
- **Cronograma de pagos:** Documento que indica las fechas de pago de las cuotas del arrendamiento, así como los respectivos montos a cancelar. (Saldaña, 2012)
- **Dador:** Que da; portador de una carta de una persona a otra; librador de una letra de cambio (Diccionario de la lengua española, Real Academia Española - RAE)
- **Desembolsos:** Documentos que corresponden a los cargos y abonos realizados a nombre del cliente. (Saldaña, 2012)
- **Documentos de garantías y seguros:** Documentación relacionada a los seguros adquiridos para proteger los bienes adquiridos mediante leasing. Estos documentos son administrados por el proveedor encargado de realizar estos trámites. (Saldaña, 2012)
- **Financiación:** Es la acción y efecto de financiar. Consiste en aportar dinero y recursos para la adquisición de bienes o servicios. Es habitual que la financiación se canalice mediante créditos o préstamos. (Piña & Del Águila, 2014)
- **Gestión financiera:** Una de las áreas funcionales de la gestión, hallada en cualquier organización compitiéndole los análisis, decisiones y acciones relacionadas con los medios financieros necesarios a la actividad de dicha organización. (Enríquez, 2015)
- **Leasing financiero:** También conocido como arrendamiento financiero o arrendamiento con opción a compra, es un contrato mercantil que celebra una empresa locadora (arrendadora) para el uso de un bien mueble o inmueble por la arrendataria, a cambio de pagos periódicos que pueden ser por adelantado o a vencimiento y con opción a favor de la arrendataria para comprar dichos bienes por un valor previamente pactado. (Piña & Del Águila, 2014)
- **Leasing operativo:** Es un alquiler, que permite al cliente el uso de un bien por un plazo determinado. El leasing o arrendamiento operativo es una alternativa eficiente para las empresas que requieren inversión en activos o bienes de capital. (Piña & Del Águila, 2014)
- **Leaseback:** Conocido también como retroarrendamiento financiero, en virtud del cual una empresa transfiere un bien mueble o inmueble de su propiedad a una empresa especializada, que a la vez de manera inmediata le concede en leasing el uso del mismo bien, a cambio del pago de un monto periódico por un plazo determinado y una opción de compra al final del contrato por un valor residual. (Mamani, 2013)
- **Proveedor:** Dicho de una persona o de una empresa; que provee o abastece de todo lo necesario para un fin a grandes grupos, asociaciones. Comunidades. (Diccionario de la lengua española, Real Academia Española - RAE)
- **Usabilidad:** La usabilidad se refiere a la capacidad de un software de ser comprendido, aprendido, usado y ser atractivo para el usuario, en condiciones específicas de uso. (ISO/IEC 9126-1:2001)

CAPÍTULO III: VARIABLES E HIPÓTESIS

3.1 Variables e Indicadores

3.1.1 Identificación de Variables

- Variable Independiente: Desarrollo de una aplicación web
- Variable Dependiente: Mejora de la aprobación de arrendamientos financieros en el banco Scotiabank Perú

3.1.2 Operacionalización de Variables

- Indicadores Variable Independiente:
 - Nivel de funcionalidad
 - Nivel de usabilidad
 - Nivel de eficiencia
 - Nivel de mantenibilidad
- Indicadores Variable Dependiente:
 - Tiempo de realización y aprobación de consultas
 - Cuotas periódicas
 - Número de activos arrendados
 - Pagos realizados
 - Número de arrendamientos financieros

3.2 Hipótesis

3.2.1 Hipótesis General

El desarrollo de una aplicación web influye significativamente en la mejora de la aprobación de arrendamientos financieros en el banco Scotiabank Perú

3.2.2 Hipótesis Específicas

- El nivel de funcionalidad influye significativamente en la mejora de las aprobaciones de arrendamientos financieros en el banco Scotiabank Perú.
- El nivel de usabilidad influye significativamente en la mejora de las aprobaciones de arrendamientos financieros en el banco Scotiabank Perú.
- El nivel de eficiencia influye significativamente en la mejora de las aprobaciones de arrendamientos financieros en el banco Scotiabank Perú.
- El nivel de mantenibilidad influye significativamente en la mejora de las aprobaciones de arrendamientos financieros en el banco Scotiabank Perú.

3.3 Matriz de Consistencia

(Ver Anexo I: Matriz de Consistencia)

CAPÍTULO IV: METODOLOGÍA PARA EL DESARROLLO DE LA INVESTIGACIÓN

La metodología sugerida es una guía que permitirá definir un conjunto de actividades indispensables para convertir los requerimientos funcionales (qué debe hacer el sistema) del usuario en un conjunto de elementos necesarios en la construcción de productos orientados a la web. (Gil, 2008)

4.1 Metodología RUP - Rational Unified Process

a. Objetivo de la metodología: construir software de alta calidad en tiempo y presupuesto estimados

Según Gil (2008), RUP describe cómo utilizar de forma efectiva procedimientos comerciales probados en el desarrollo de software para equipos de desarrollo de software. Conocidos como las 6 mejores prácticas:

Administración de requerimientos			
Desarrollo iterativo	Modelamiento visual	Verificación de la calidad	Arquitectura con componentes
Control de cambios			

Tabla 4.1: Las seis mejores prácticas RUP. Fuente: Gil (2004)

Dada la complejidad de los sistemas actuales, no es posible hacer de manera secuencial la definición completa del problema, diseñar la solución completa, construir el software y por último probarlo.

La metodología está basada en componentes de software enlazados a través de interfaces. Un componente es una parte física y reemplazable del software que puede ser ajustado y proporciona la elaboración de un conjunto de interfaces. Una interfaz es la colección de operaciones que son utilizadas para especificar un servicio de una clase o de un componente.

La metodología utiliza el lenguaje de modelado unificado (UML) para preparar todos los diagramas de un sistema de software y está fundamentado en tres principios básicos:

b. Dirigido por casos de uso

Un sistema de software debe brindar servicios a sus usuarios, por lo que se debe conocer qué se necesita y desea a futuro

Un caso de uso es una fracción de la funcionalidad del sistema. Estos representan los requisitos funcionales (¿qué debe hacer el sistema?) y especifican una secuencia de acciones que el sistema puede llevar a cabo. Tomando el modelo de casos de uso como base, los desarrolladores crean una serie de modelos de diseño e implementación que llevan a cabo los casos de uso.

c. Centrado en la arquitectura

La arquitectura en un sistema de software es descrita por medio de diferentes vistas del sistema en proceso de desarrollo

La arquitectura surge de las necesidades de la organización, de cómo la perciben los usuarios y de cómo se reflejan en los casos de uso. Pero hay otros factores que también influyen tales como la plataforma en la que funcionará el software, los bloques de construcción de que se dispondrá, consideraciones de implementación, sistemas heredados y requisitos no funcionales.

- **Arquitectura multicapa.** Esta arquitectura presenta tres niveles a saber:
 - Capa de presentación

- Capa de lógica de aplicación
- Capa de persistencia de los datos

Ventajas de la arquitectura multicapa. Algunas de las ventajas son:

- Múltiples plataformas y escalabilidad para suplir variaciones de demanda del mercado
- Integración con sistemas de información existentes
- Brinda componentes portables a través de múltiples plataformas
- Promueve el aprovechamiento de los servicios provistos por los servidores de aplicaciones

d. Iterativo e incremental

Se aconseja dividir el trabajo en partes más pequeñas o subsistemas; donde cada subsistema es una iteración que resulta en un incremento. Las iteraciones hacen referencias a pasos en el flujo del trabajo, y los incrementos, al crecimiento del producto.

En cada iteración, los desarrolladores identifican y especifican los casos de uso relevantes, crean un diseño utilizando la arquitectura seleccionada como guía, implementan el diseño mediante componentes y verifican que los componentes satisfagan los casos de uso.

4.2 RUP como un proceso personalizable

Según Kroll, P, & Kruchten, P. (2003), cada proyecto y cada organización tienen necesidades únicas que requieren un proceso que se adapta a su situación específica. Para acomodar este requisito, el producto RUP constituye un marco de proceso completo compuesto de varias partes integradas:

- Mejores prácticas.** El RUP viene con una biblioteca de mejores prácticas, producida por IBM software y sus socios. Estas mejores prácticas están evolucionando continuamente y cubren un alcance amplio. Las mejores prácticas se expresan en forma de fases, funciones, actividades, artefactos y flujos de trabajo.
- Herramientas de entrega de procesos.** El RUP está literalmente en la punta de los dedos de los desarrolladores porque se entrega en línea usando la tecnología web, en lugar de usar libros o carpetas. Esta entrega permite que el proceso se integre con las muchas herramientas de desarrollo de software en la suite rational y con otras herramientas para que los desarrolladores puedan acceder a la guía de procesos dentro de la herramienta que están utilizando.
- Herramientas de configuración.** La forma modular y electrónica del RUP permite que sea adaptada y configurada para satisfacer las necesidades específicas de una organización de desarrollo.
- Herramientas de creación de procesos.** Rational process workbench (RPW) es una herramienta de creación de procesos, que le permite capturar sus propias mejores prácticas en el formato RUP.
- Comunidad / mercado.** La comunidad en línea de la red de desarrolladores racionales (RDN) permite a los usuarios intercambiar experiencias con colegas y expertos y proveedores de acceso a la información más reciente en términos de artefactos, artículos o contenido adicional RUP.

El producto RUP es diseñado, desarrollado, entregado y mantenido como cualquier herramienta de software. Se actualiza aproximadamente dos veces al año, por lo que el proceso nunca es obsoleto y sus usuarios se benefician de los últimos desarrollos.

4.3 Fases de los ciclos de vida, objetivos e hitos

Según Kroll, P, & Kruchten, P. (2003):

a. Fase de inicio. Establecer una buena comprensión de todos los requisitos y establecer el alcance del sistema. Mitigar muchos de los riesgos de negocio, producir el caso de negocios para la construcción del sistema y obtener la compra de todas las partes interesadas sobre si proceder con el proyecto

Objetivos:

- Entender el alcance del proyecto
- Construir el caso de negocio
- Invitar a los interesados a avanzar

Hito: Hito de objetivo del ciclo de vida

b. Fase de elaboración. Cuidar de muchas de las tareas más difíciles desde el punto de vista técnico: diseñar, implementar, probar y basar una arquitectura ejecutable, incluyendo subsistemas, sus interfaces, componentes clave y mecanismos arquitectónicos, como cómo tratar la comunicación interprocesos o la persistencia. Abordar los principales riesgos técnicos, tales como los riesgos de contención de los recursos, los riesgos de rendimiento y los riesgos de seguridad de los datos, implementando y validando el código real

Objetivos:

- Mitigar los principales riesgos técnicos
- Crear una arquitectura de línea base
- Entender lo que se necesita para construir el sistema

Hito: Hito de la arquitectura del ciclo de vida

c. Fase de construcción. Realizar la mayor parte de la implementación a medida que pasa de una arquitectura ejecutable a la primera versión operativa de su sistema. Despliegue varias versiones internas y alfa para asegurarse de que el sistema es utilizable y responde a las necesidades del usuario. Terminar la fase mediante la implementación de una versión beta totalmente funcional del sistema, incluida la instalación y la documentación de apoyo y el material de capacitación (aunque el sistema probablemente todavía necesitará ajuste de funcionalidad, rendimiento y calidad general)

Objetivo:

- Construir la primera versión operativa del producto

Hito: Hito de la capacidad operacional inicial

d. Fase de transición. Garantizar que el software responde a las necesidades de sus usuarios. Esto incluye probar el producto en preparación para la liberación y hacer ajustes menores basados en la retroalimentación del usuario se centra principalmente en el ajuste fino del producto, configuración, instalación y problemas de usabilidad, todos los principales problemas estructurales deberían haber sido elaborados mucho antes en el ciclo de vida del proyecto

Objetivo:

- Construir la versión final del producto y entregarla al cliente

Hito: Hito de la liberación del producto

4.4 Buenas prácticas

Según Tabares (2011), RUP identifica seis buenas prácticas, con las que define una forma efectiva de trabajar para los equipos de desarrollo de software.

a. Gestión de requisitos

RUP brinda una guía para encontrar, organizar, documentar y seguir los cambios de los requisitos funcionales y restricciones. Utiliza una notación de caso de uso y escenarios para representar los requisitos.

b. Desarrollo de software iterativo

Desarrollo del producto mediante iteraciones con hitos bien definidos, en las cuales se repiten las actividades, pero con distinto énfasis, según la fase del proyecto.

c. Desarrollo basado en componentes

La creación de sistemas intensivos en software requiere dividir el sistema en componentes con interfaces bien definidas, que posteriormente serán ensamblados para generar el sistema. Esta característica en un proceso de desarrollo permite que el sistema se vaya creando a medida que se obtienen o se desarrollan sus componentes.

d. Modelado visual (utilizando UML)

UML es un lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema software. Utilizar herramientas de modelado visual facilita la gestión de dichos modelos, permitiendo ocultar o exponer detalles cuando sea necesario. El modelado visual también ayuda a mantener la consistencia entre los artefactos del sistema: requisitos, diseños e implementaciones. En resumen, el modelado visual ayuda a mejorar la capacidad del equipo para gestionar la complejidad del software.

e. Verificación continua de la calidad

Es importante que la calidad de todos los artefactos se evalúe en varios puntos durante el proceso de desarrollo, especialmente al final de cada iteración. En esta verificación las pruebas juegan un papel fundamental y se integran a lo largo de todo el proceso. Para todos los artefactos no ejecutables, las revisiones e inspecciones también deben ser continuas.

f. Gestión de los cambios

El cambio es un factor de riesgo crítico en los proyectos de software. Los artefactos software cambian no sólo debido a acciones de mantenimiento posteriores a la entrega del producto, sino que durante el proceso de desarrollo, especialmente importantes por su posible impacto son los cambios en los requisitos. Por otra parte, otro gran desafío que debe abordarse es la construcción de software con la participación de múltiples desarrolladores, posiblemente distribuidos geográficamente, y quizás en distintas plataformas. La ausencia de disciplina rápidamente conduciría al caos. La gestión de cambios y de configuración es la disciplina de RUP encargada de este aspecto.

4.5 Artefactos RUP

En el desarrollo del proyecto, se usaron los siguientes artefactos por fases y flujos:

FASES	INICIO	ELABORACIÓN	CONSTRUCCIÓN
FLUJOS			
Modelo de negocio	-Modelo de casos de uso de negocio -Diagrama de actividades		
Requerimientos	-Matriz de actividades y requerimientos -Modelo de casos de uso -Plantillas de especificación de casos de uso		
Análisis y diseño		-Modelo de datos	
Implementación			-Diagrama de componentes -Diagrama de despliegue -Codificación

Tabla 4.2: Artefactos RUP utilizados. (Fuente: Elaboración propia)

Según Kruchten (2004), define los siguientes artefactos:

a. Modelo de casos de uso del negocio:

Representa la relación de los actores del negocio con los casos de uso del negocio (CUN). Describe cómo el negocio interactúa con el entorno.

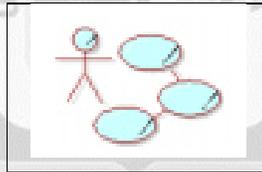


Figura 4.1: Artefacto modelo de casos de uso del negocio. Fuente: Kruchten (2004)

b. Actor del negocio:

Representa un rol de algún participante externo que interactúa con el negocio. Es quien está interesado en requerir o recibir algo.

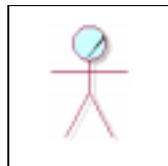


Figura 4.2: Artefacto actor del negocio. Fuente: Kruchten (2004)

c. Caso de uso del negocio (CUN):

Secuencia de acciones que realiza un proceso de negocio y que da un resultado observable y de valor a un actor de negocio.

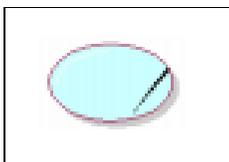


Figura 4.3: Artefacto caso de uso del negocio. Fuente: Kruchten (2004)

d. Metas del negocio:

Valor deseado de una medida en particular en el futuro, utilizados para planificar y gestionar las actividades del proceso de negocio.

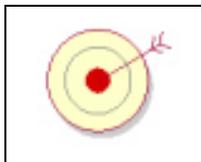


Figura 4.4: Artefacto metas del negocio. Fuente: Kruchten (2004)

e. Trabajadores del negocio:

Representa un rol que desempeñan los trabajadores del negocio, llamados también unidades funcionales, realizan actividades internas del negocio.

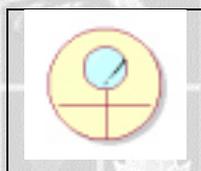


Figura 4.5: Artefacto trabajadores del negocio. Fuente: Kruchten (2004)

f. Realización de casos de uso del negocio: Representa las actividades que se realizan en un caso de uso, incluye quienes lo realizan y que entidades se utilizan.

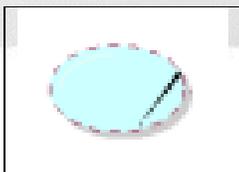


Figura 4.6: Artefacto realización de casos de uso del negocio. Fuente: Kruchten (2004)

g. Entidades del negocio:

Paquetes de información o documentos que se utilizan en una actividad.

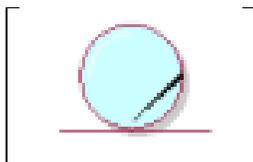


Figura 4.7: Artefacto entidades del negocio. Fuente: Kruchten (2004)

h. Matriz de proceso, servicio y funcionalidades:

Matriz que muestra la trazabilidad de los procesos, actividades, requerimientos funcionales y casos de uso.

PROCESO DE NEGOCIO (MBA)	ACTIVIDADES DEL PROCESO	RESPONSABLE DEL NEGOCIO	REQUERIMIENTOS FUNCIONALES	CASOS DE USO	ACTORES
Proceso de venta	Procesar solicitud de pedido	Cliente	RF01: El sistema debe permitir procesar pedidos en el lugar o línea		
Ident. Realiza en su sitio de venta de carne y reses.	Modificar pedido	Cliente	RF02: El sistema debe permitir modificar pedidos en el sitio de compra en línea o línea	CU01: Gestionar pedido de compra	Usuario Cliente
	Registrar pedido al pedido	Proveedor	RF03: El sistema debe permitir registrar pedidos de compra de compra en el lugar o línea		
Proceso de adquisición de carne	Registrar datos de compra	Proveedor	RF04: El sistema debe permitir registrar pedidos de compra en línea o línea	CU02: Gestionar pedidos de compra	Cliente proveedor
Ident. Realiza en su sitio de venta de carne y reses.	Registrar pedido pendiente de entrega carne de pedido	Proveedor	RF05: El sistema debe permitir registrar pedidos de compra en línea o línea	CU03: Actualizar pedido	Proveedor
	Pagar el pedido	Cliente	RF06: El sistema debe permitir realizar compras desde el lugar o línea	CU04: Realizar compra	Usuario Cliente

Figura 4.8: Artefacto matriz de proceso, servicio y funcionalidad. Fuente: Kruchten (2004)

i. Modelo de casos de uso:

Conjunto de secuencias de acciones, cada secuencia representa un comportamiento del sistema, está conformado por los actores del sistema y los casos de uso. Describe los requerimientos funcionales del sistema en forma de casos de uso.

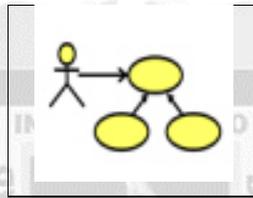


Figura 4.9: Artefacto modelo de casos de uso. Fuente: Kruchten (2004)

j. Actores:

Representa un rol que realiza un grupo de usuarios que interactúan directamente con el sistema.



Figura 4.10: Artefacto actores. Fuente: Kruchten (2004)

k. Casos de uso:

Conjunto de secuencias e interacciones entre el actor y el sistema durante un proceso, representan los objetivos y requisitos funcionales de forma comprensible para los actores involucrados.

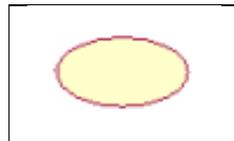


Figura 4.11: Artefacto casos de uso. Fuente: Kruchten (2004)

l. Especificación de casos de uso (ECU):

Documento en que se describe, de forma textual, que hace el actor y el sistema, en conclusiones, describir los requerimientos funcionales que conforma un caso de uso.



Figura 4.12: Artefacto especificación de casos de uso. Fuente: Kruchten (2004)

m. Modelo de datos:

Describe la lógica y las representaciones físicas de la persistencia de los datos utilizados por la aplicación.

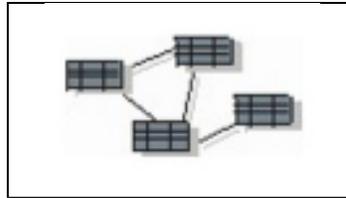


Figura 4.13: Artefacto modelo de datos. Fuente: Kruchten (2004)

n. Diagrama de componentes: Muestra las interfaces y dependencias entre los componentes del software, los componentes de código de fuente, código binario y ejecutable, un módulo de software puede representarse como un tipo de componente. (Kruchten, 2004)

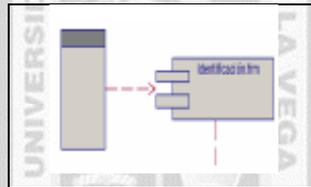


Figura 4.14: Artefacto diagrama de componentes. Fuente: Kruchten (2004)

o. Diagrama de despliegue:

Muestra la distribución física de los artefactos software en nodos unidos por conexiones de comunicación.

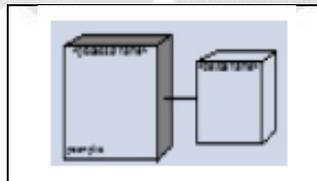


Figura 4.15: Artefacto diagrama de despliegue. Fuente: Kruchten (2004)

(Ver anexo VII: Código fuente)

CAPÍTULO V: SOLUCIÓN TECNOLÓGICA

5.1 Flujo del negocio

5.1.1. Modelo de casos de uso del negocio

En el desarrollo de la aplicación se identificaron los siguientes casos de uso de negocio:

- Proceso de solicitud de arrendamiento financiero
- Proceso de evaluación de solicitud
- Proceso de cierre de solicitud

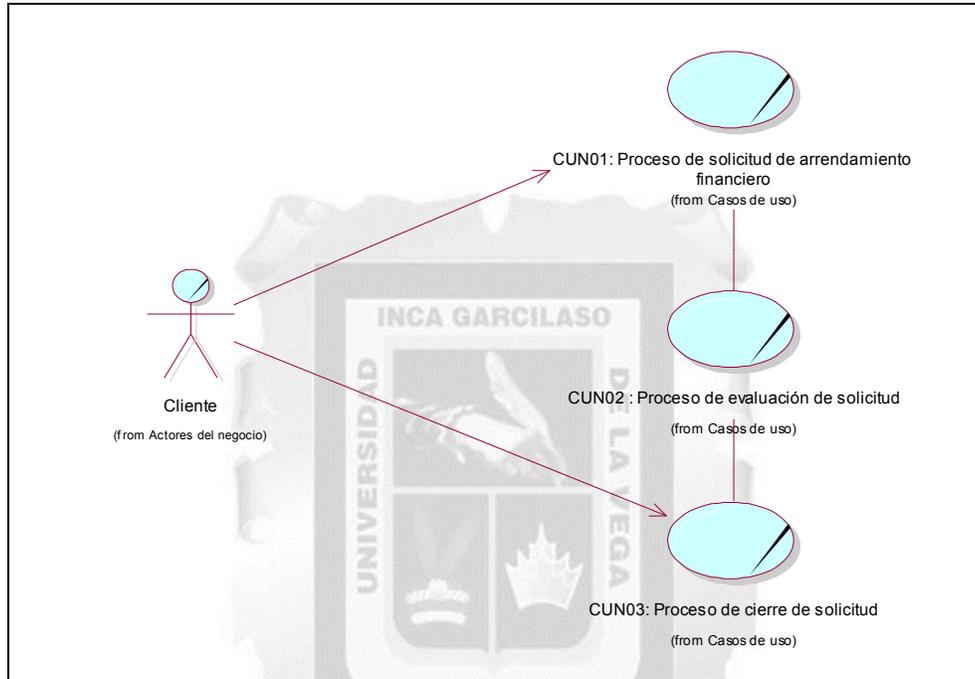


Figura 5.1: Diagrama de casos de uso del negocio. (Fuente: Elaboración propia)

5.1.2. Actor del negocio

- **Cliente.** Actor externo del negocio que se acerca al banco para solicitar el arrendamiento financiero



Figura 5.2: Actor del negocio. (Fuente: Elaboración propia)

5.1.3. Trabajadores del negocio

Encontramos a los siguientes trabajadores de las áreas involucradas:

- **Asesor área comercial.** Actor interno del negocio que recibe la solicitud de arrendamiento financiero, asimismo, envía a su gerencia de área para la aprobación inicial de la solicitud, para luego cotizar y aprobar las cotizaciones.

- **Gerente área comercial.** Actor interno del negocio que se encarga de la aprobación inicial de las solicitudes de arrendamientos financieros.
- **Analista área de riesgos.** Actor interno del negocio que se encarga de la evaluación financiera del potencial cliente.
- **Miembro del comité de inversiones.** Actor interno del negocio que aprueba o no el contrato de arrendamiento.
- **Asesor área de operaciones.** Actor interno del negocio que activa el contrato de arrendamiento y gestiona con el proveedor el desembolso del monto para la adquisición del bien solicitado.

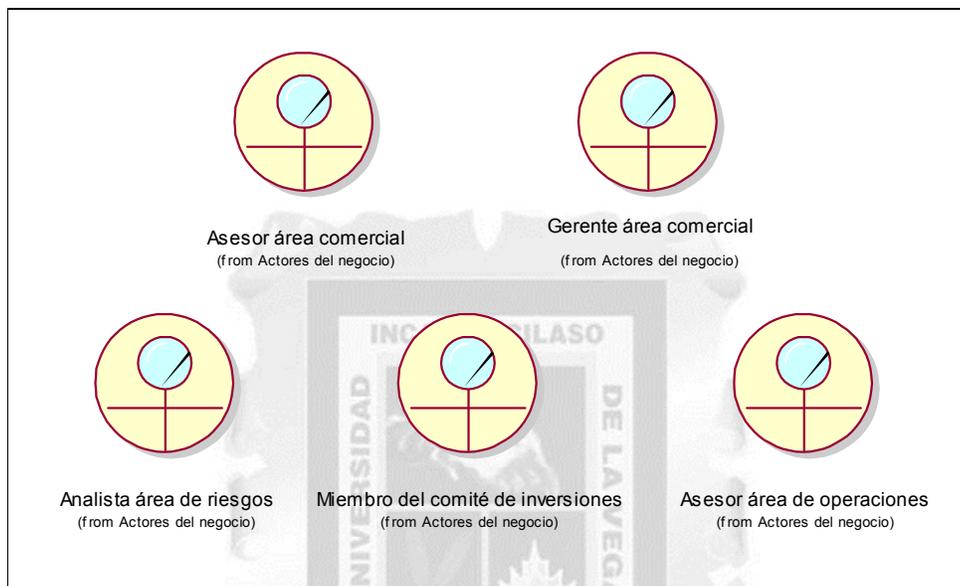


Figura 5.3: Trabajadores del negocio. (Fuente: Elaboración propia)

5.1.4. Casos de uso del negocio

Tenemos estos casos:

- **Proceso de solicitud de arrendamiento financiero.** Este proceso se inicia cuando el cliente se acerca al banco para solicitar el arrendamiento financiero, el asesor comercial registra sus datos, en caso no los tenga se lo pide, registra la solicitud, luego esta solicitud la revisa el gerente comercial, el cual puede aprobar o no la solicitud, para devolverla de nuevo al asesor, que cotiza y aprueba la solicitud, para finalizar el proceso.
- **Proceso de evaluación de solicitud.** Este proceso se inicia cuando la solicitud llega al empleado del área de riesgos, que son los encargados de evaluar la situación financiera del potencial cliente, las garantías, mediante consultas externas, una vez pasada esta prueba, se define los compromisos del cliente; luego la solicitud pasa para su evaluación del área de comité de inversiones, los cuales se reúnen para aprobar o no las solicitudes y generar un contrato para finalizar el proceso.
- **Proceso de cierre de solicitud.** El proceso final de nuestro proyecto sucede cuando, una vez aprobado el contrato, pasa al área legal para gestionar las firmas del cliente y asimismo legalizar el contrato en una notaría, una vez hecho esto, el área de operaciones activa la fecha de inicio de

contrato y gestiona el desembolso del monto solicitado al proveedor de bien para finalizar el proceso.

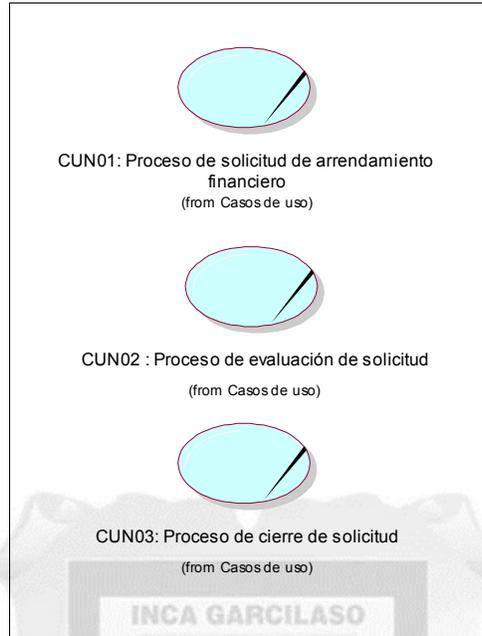


Figura 5.4: Casos de uso del negocio. (Fuente: Elaboración propia)

5.1.5. Metas del negocio

Se identificaron las siguientes metas por cada caso de uso de negocio:

- Reducir en un 50% el tiempo de la solicitud de arrendamiento financiero.
- Reducir en un 40% el tiempo de la evaluación de solicitud.
- Reducir en un 50% el tiempo del cierre de solicitud.

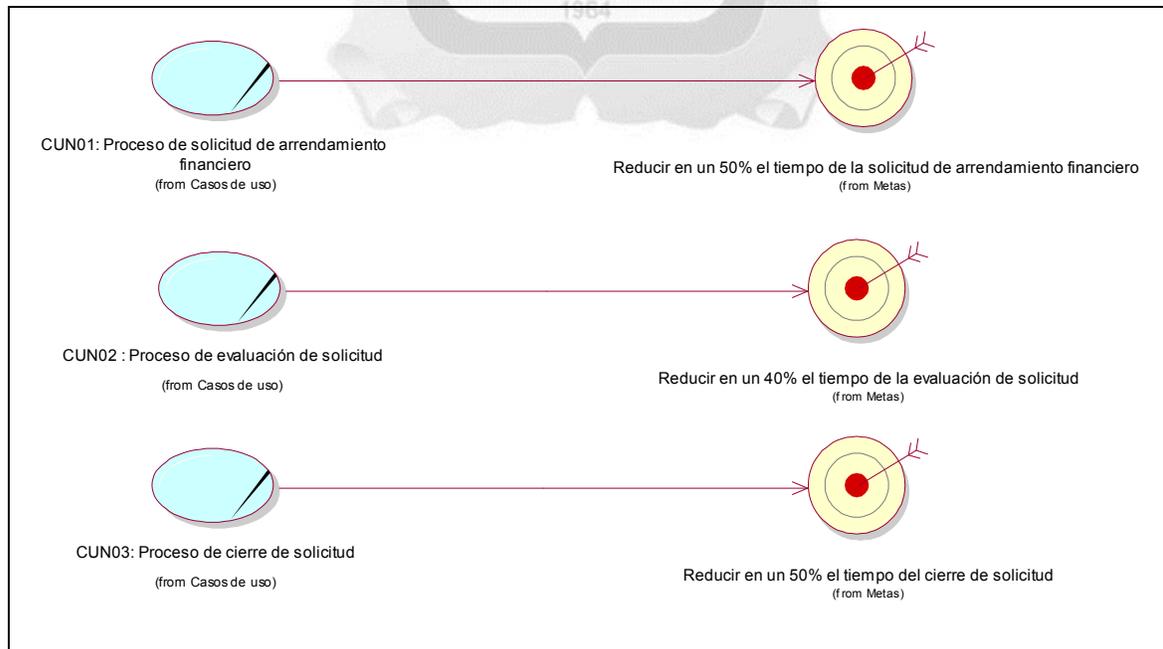


Figura 5.5: Metas del negocio. (Fuente: Elaboración propia)

5.1.6. Entidades del negocio

Encontramos las siguientes:

- **Archivo de clientes.** Archivar donde se encuentran las hojas con los datos de los clientes.
- **Ficha registro cliente.** Documento donde se registra a un nuevo cliente.
- **Ficha solicitud.** Documento donde se registra una nueva solicitud.
- **Solicitud.** Documento que contiene la información de la solicitud.
- **Monto.** Saldo que solicita el cliente para el arrendamiento.
- **Simulación pago de cuotas.** Hoja de cálculo donde se realiza la simulación de pagos que el cliente realizará mensualmente.
- **Lista de créditos.** Documento donde se guardan los créditos de los clientes.
- **Estados financieros.** Información financiera acerca del cliente.
- **Formato de contratos.** Documento donde se registrará el contrato.
- **Contrato.** Documento de acuerdo entre el banco y el cliente.

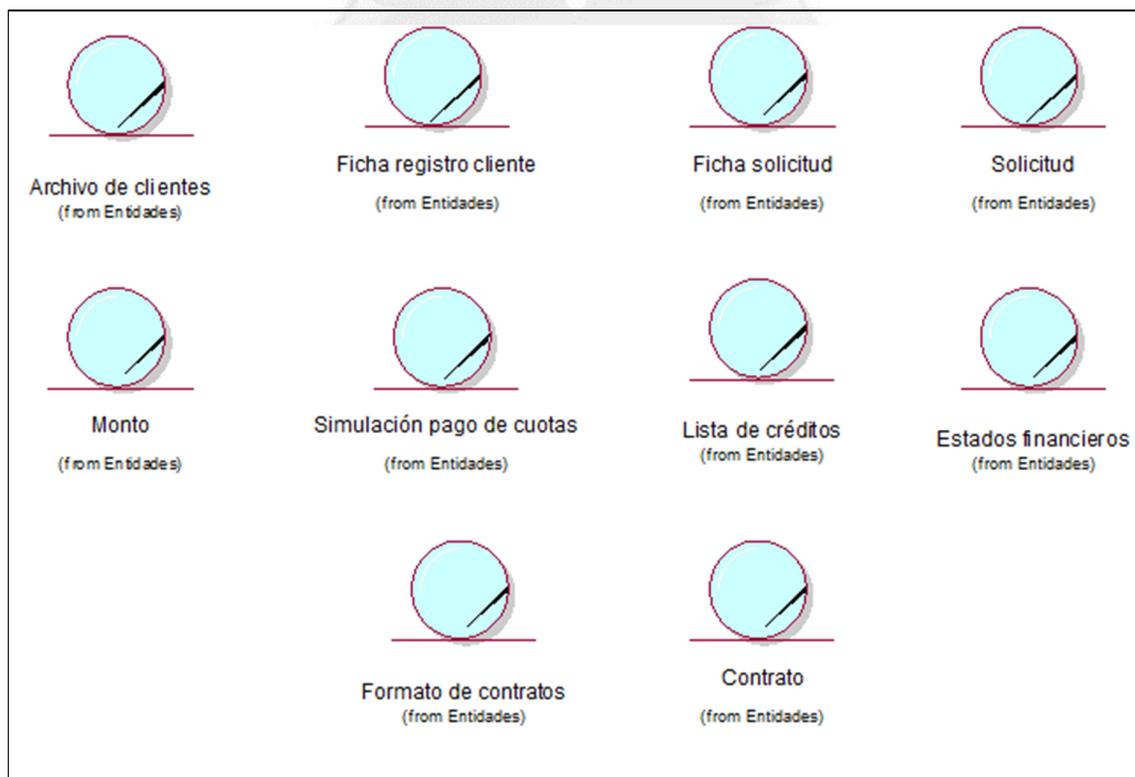


Figura 5.6: Entidades del negocio. (Fuente: Elaboración propia)

5.1.7. Diagrama de actividades

Se muestra los siguientes diagramas:

- Proceso de solicitud de arrendamiento financiero
- Proceso de evaluación de solicitud
- Proceso de cierre de solicitud

• **Proceso de solicitud de arrendamiento financiero**

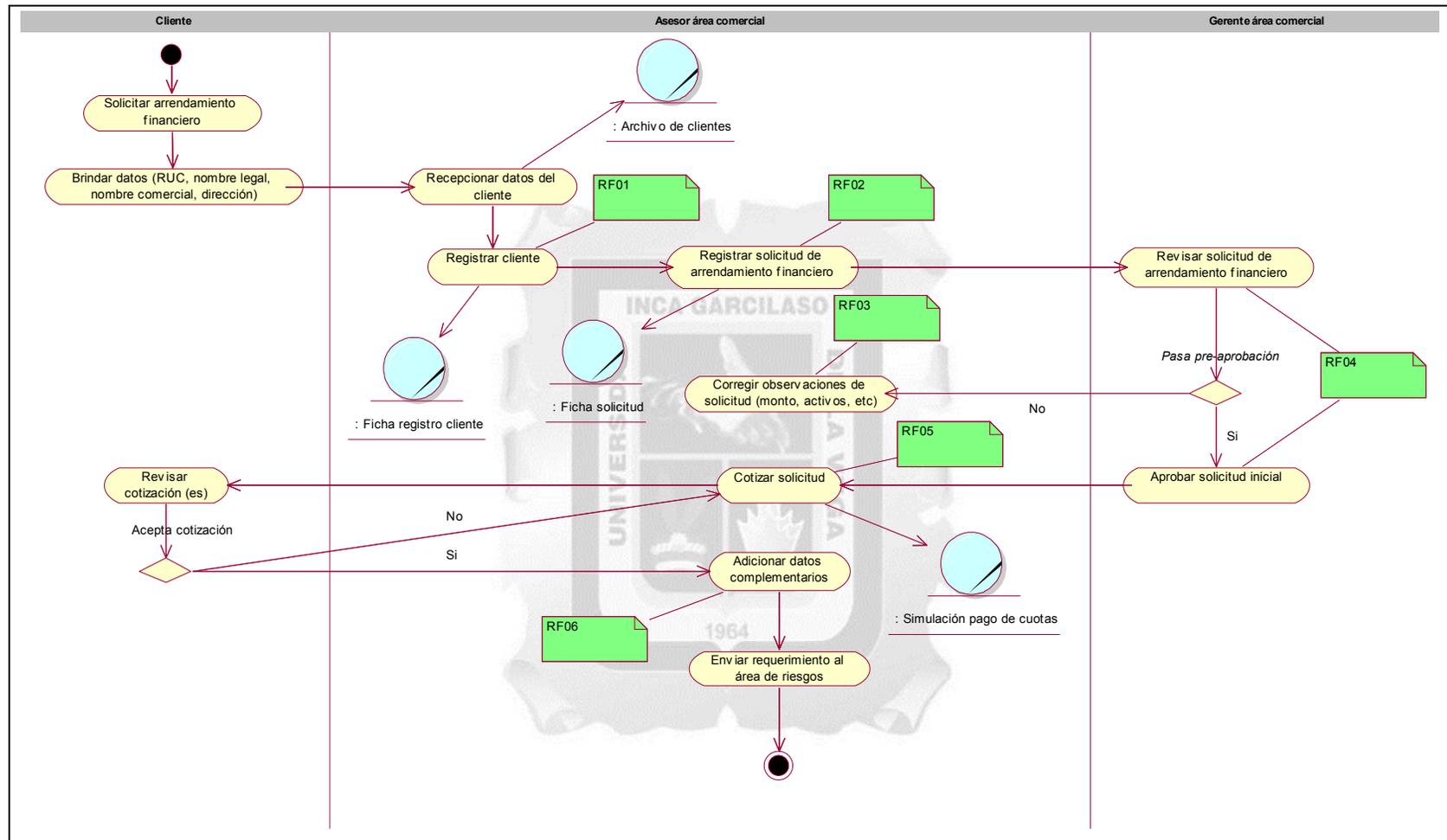


Figura 2.7: Diagrama de actividad - Proceso de solicitud de arrendamiento financiero. (Fuente: Elaboración propia)

- **Proceso de evaluación de solicitud**

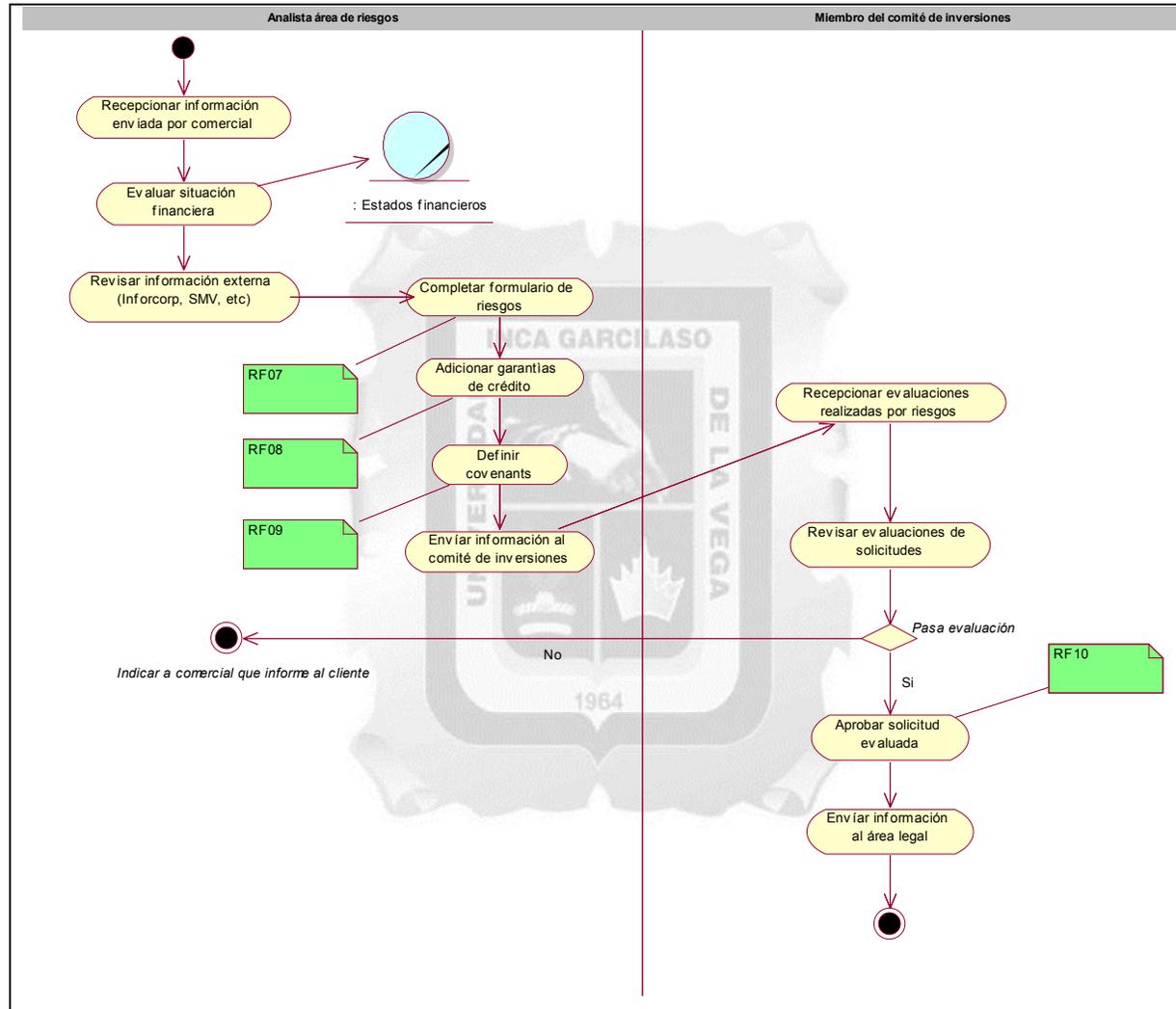


Figura 5.8: Diagrama de actividad - Proceso de evaluación de solicitud. (Fuente: Elaboración propia)

- **Proceso de cierre de solicitud**

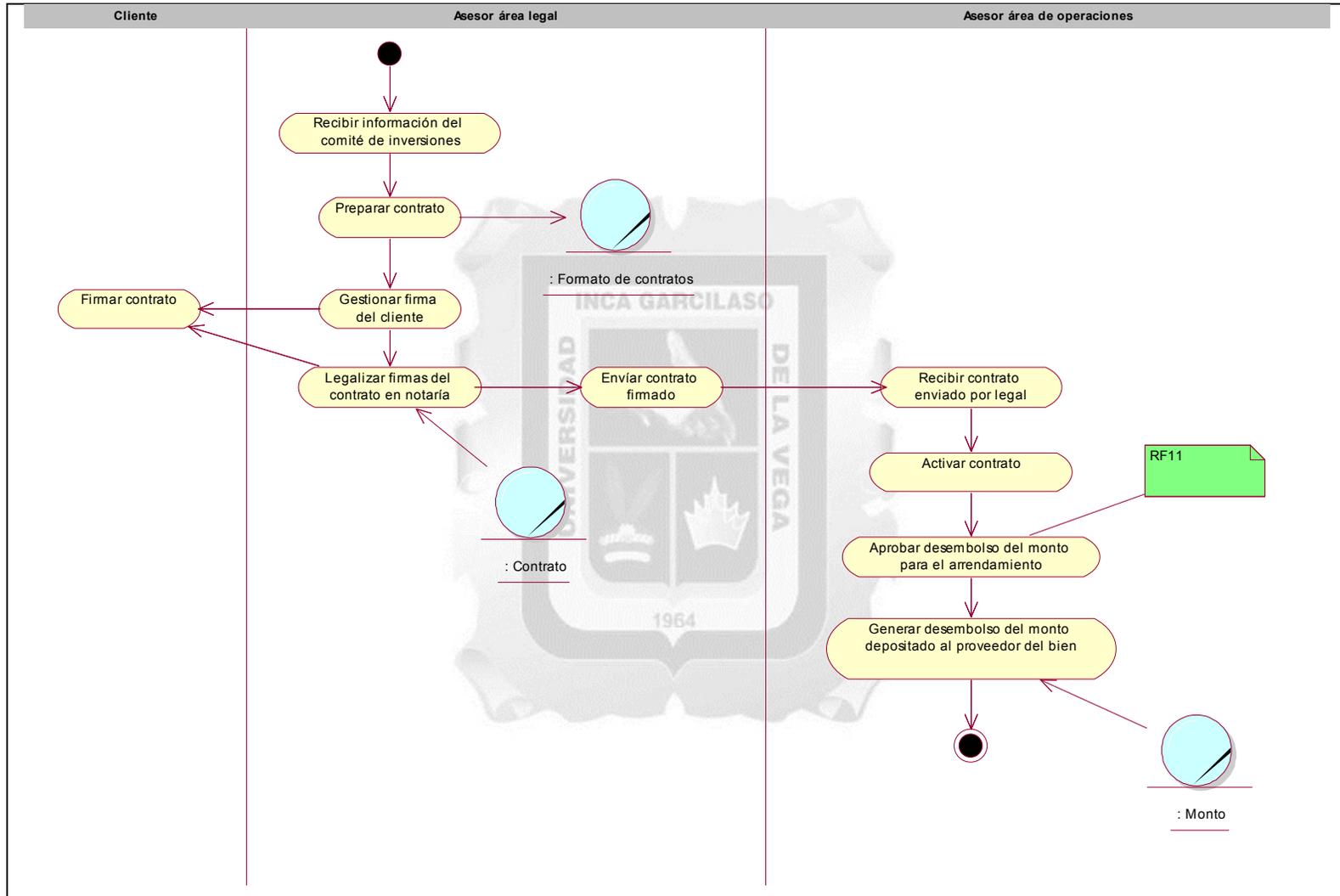


Figura 5.9: Diagrama de actividad - Proceso de cierre de solicitud. (Fuente: Elaboración propia)

5.1.8. Matriz de actividades y requerimientos

Matriz de Actividades y Requerimientos - Aplicación web para la mejora en la aprobación de arrendamientos financieros del banco Scotiabank Perú							
Proceso de Negocio "Meta"	Actividad del Negocio	Responsable del Negocio	Requerimiento funcional		Caso de Uso		Actores
Proceso de Solicitud de arrendamiento financiero "Reducir en un 50% el tiempo de la solicitud de arrendamiento financiero"	Registrar cliente	Comercial	RF01	El sistema debe permitir agregar nuevos clientes	CU01	Gestionar cliente	Asesor área comercial
	Registrar solicitud de arrendamiento financiero	Comercial	RF02	El sistema debe permitir crear nuevas solicitudes	CU02	Gestionar solicitud	
	Corregir observaciones de solicitud	Comercial	RF03	El sistema debe permitir modificar y guardar los cambios de las solicitudes			
	Revisar solicitud de arrendamiento financiero	Comercial	RF04	El sistema debe permitir modificar y aprobar solicitud inicial	CU03	Gestionar aprobación inicial	Gerente área comercial
	Aprobar solicitud inicial	Comercial					
	Cotizar solicitud	Comercial	RF05	El sistema debe permitir agregar, modificar y aprobar cotizaciones	CU04	Gestionar cotización	Asesor área comercial
	Adicionar datos complementarios	Comercial	RF06	El sistema debe permitir registrar y modificar datos complementarios de las cotizaciones			
Proceso de evaluación de solicitud "Reducir en un 40% el tiempo de la evaluación de solicitud"	Completar formulario de riesgos	Riesgos	RF07	El sistema debe permitir registrar un formulario de riesgos	CU05	Gestionar información de riesgos	Analista área de riesgos
	Adicionar garantías de crédito	Riesgos	RF08	El sistema debe permitir agregar diversas modalidades de garantía			
	Definir covenants	Riesgos	RF09	El sistema debe permitir agregar las restricciones y obligaciones a imponer al cliente			
	Aprobar solicitud evaluada	Comité de inversiones	RF10	El sistema debe permitir la aprobación de la solicitud evaluada	CU06	Gestionar aprobación de evaluación	Miembro del comité de inversiones
Proceso de cierre de solicitud "Reducir en un 50% el tiempo de cierre de solicitud"	Aprobar desembolso del monto para el arrendamiento	Operaciones	RF11	El sistema debe permitir la aprobación final del desembolso de monto para el bien solicitado	CU07	Gestionar aprobación final	Asesor área de operaciones

Tabla 5.1: Matriz de proceso, servicio y funcionalidades. (Fuente: Elaboración propia)

5.1.9. Matriz de requerimientos adicionales

Matriz de Requerimientos Adicionales - Aplicación web para la mejora de la aprobación de arrendamientos financieros en el banco Scotiabank Perú					
Paquete	Requisito Funcional		Caso de Uso		Actores
Consultas	RF16	El sistema debe permitir buscar cliente	CU08	Buscar cliente	Administrador, asesor área comercial
	RF19	El sistema debe permitir consultar número de solicitud	CU09	Buscar solicitud	Todos los usuarios
	RF20	El sistema debe permitir consultar estado de la solicitud			
	RF21	El sistema debe permitir consultar historial de solicitudes			
Solicitud	RF14	El sistema debe permitir exportar solicitud a pdf	CU10	Exportar solicitud	Todos los usuarios
Cotización	RF15	El sistema debe permitir exportar cronograma de pago a pdf	CU11	Exportar cotización	Todos los usuarios
Seguridad	RF22	El sistema debe permitir agregar, modificar y eliminar	CU12	Gestionar usuarios	Administrador
	RF23	El sistema debe permitir asignar perfiles a usuarios	CU13	Gestionar perfiles de usuario	
	RF24	El sistema debe permitir agregar, modificar y eliminar	CU14	Gestionar accesos por perfiles	
	RF25	El sistema debe permitir ingresar al mismo	CU15	Ingresar al sistema	Todos los usuarios
	RF26	El sistema debe permitir cambiar la contraseña	CU16	Restablecer contraseña	

Tabla 5.2: Matriz de requerimientos adicionales. (Fuente: Elaboración propia)

5.2. Flujo de requerimientos

5.2.1. Diagrama de casos de uso

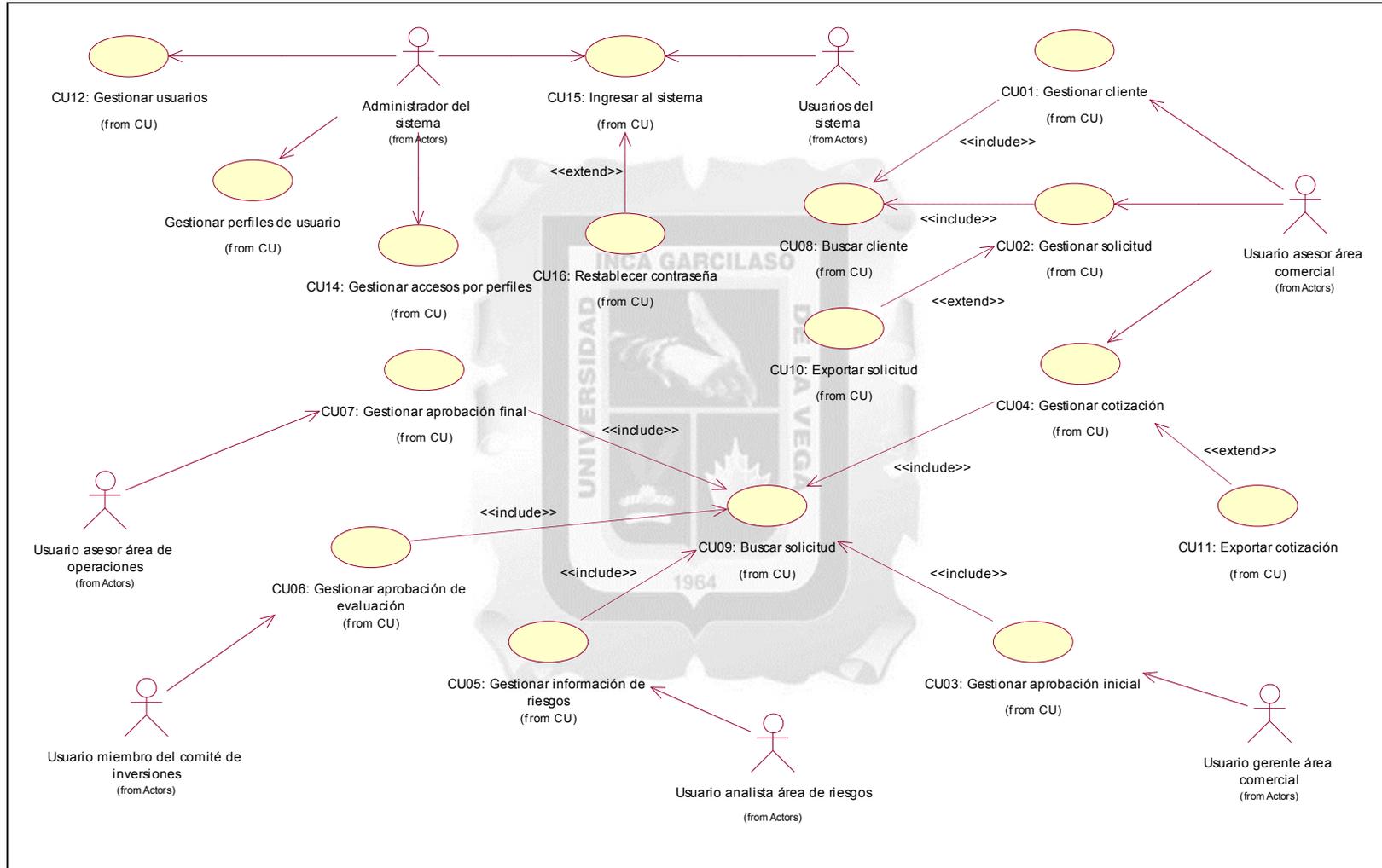


Figura 5.10: Diagrama de casos de uso. (Fuente: Elaboración propia)

5.2.2. Especificaciones de caso de uso

a. Especificación de caso de uso 01 - Gestionar cliente

1. Breve Descripción

Este caso de uso permite al **usuario asesor área comercial** buscar y agregar clientes al momento de crear una nueva solicitud.

2. Flujo de Eventos

Evento Disparador: El caso de uso comienza cuando el **usuario asesor área comercial** selecciona la opción **nuevo** en la interfaz principal del sistema.

2.1 Flujo Básico <<Gestionar cliente>>

2.1.1 El sistema muestra la interfaz **búsqueda de clientes**. La interfaz muestra la lista de clientes del banco.

La interfaz contiene los datos: **Razón social, RUC**.

Incluye las opciones: **Nuevo** y **buscar**.

2.1.2 El **usuario asesor área comercial** digita los datos (RUC o razón social) para realizar la búsqueda y luego hace clic en la opción **buscar**.

2.1.3 El sistema incluye el caso de uso **buscar cliente**.

2.1.4 El sistema muestra la empresa de acuerdo a los datos filtrados previamente.

2.1.5 Fin del caso de uso.

2.2 Sub Flujo <<Nuevo cliente>>

2.2.1 El **usuario asesor área comercial** hace clic en la opción **nuevo**.

2.2.2 El sistema muestra la interfaz **búsqueda de clientes**, donde activan los campos para ingresar los datos del cliente: **RUC, razón social, teléfono**; así como también los datos del contacto de la empresa: **Nombre, teléfono y e-mail**.

Incluye las opciones: **Guardar** y **cerrar**.

2.2.3 El **usuario asesor área comercial** digita los datos del cliente sobre los campos activos.

2.2.4 Si el **usuario asesor área comercial** selecciona **guardar**

2.2.4.1 El sistema actualiza la lista de clientes en la interfaz.

2.2.5 Si el **usuario asesor área comercial** selecciona **cerrar**

2.2.5.1 El sistema redirige a la interfaz **búsqueda de clientes**.

3. Flujos Alternativos

3.1 << No se encontraron registros >>

En el punto 2.1.2 del flujo básico **gestionar cliente**, en caso no existe el cliente, el sistema muestra el mensaje **no se encontraron registros**, el flujo continúa en el punto 2.1.2.

3.2 << Ingresar correo electrónico válido >>

En el punto 2.2.3 del sub flujo **nuevo cliente**, si el **usuario asesor área comercial** no digita un correo electrónico de la persona de contacto, al momento de seleccionar **guardar**, el sistema muestra el mensaje **debe ingresar el correo electrónico válido**, el flujo continúa en el punto 2.2.3.

4. Requerimientos Especiales

4.1 El caso de uso debe estar disponible a través del **navegador internet explorer**, previo logueo del **usuario asesor área comercial**.

5. Pre Condiciones

5.1 El **usuario asesor área comercial** se haya logueado en el sistema.

5.2 Los clientes estén disponibles en el sistema.

6. Post Condiciones

6.1 El cliente queda registrado en la base de datos.

7. Puntos de Extensión

Ninguno.

8. Prototipo (GUI)

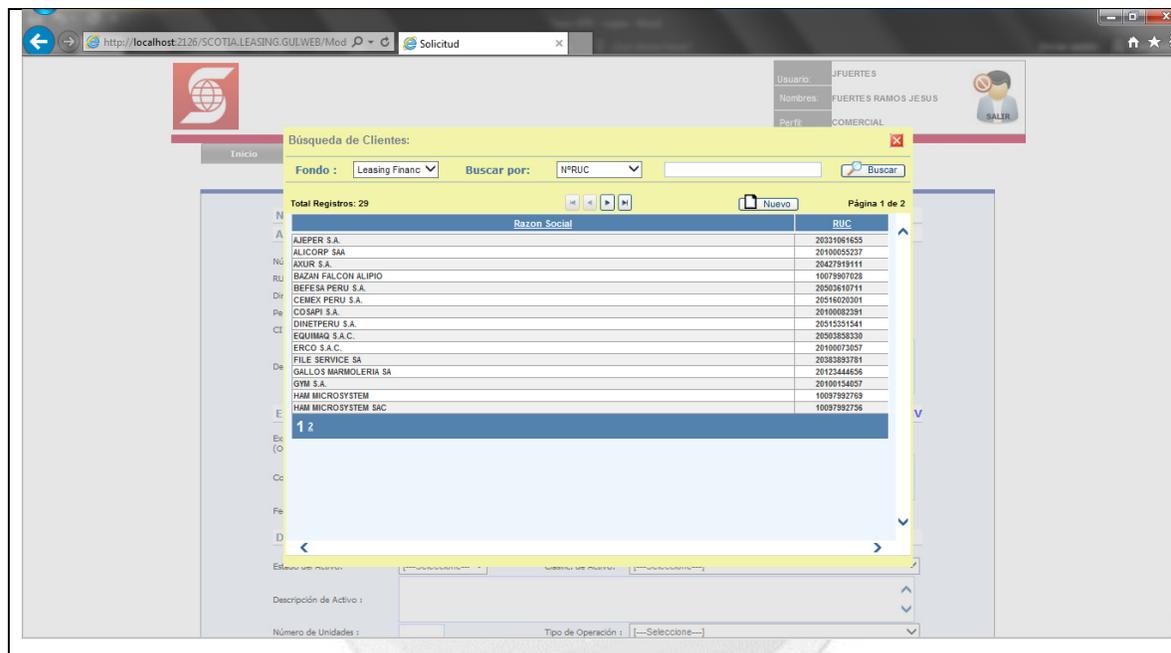


Figura 5.11: Flujo básico - Gestionar cliente. (Fuente: Elaboración propia)

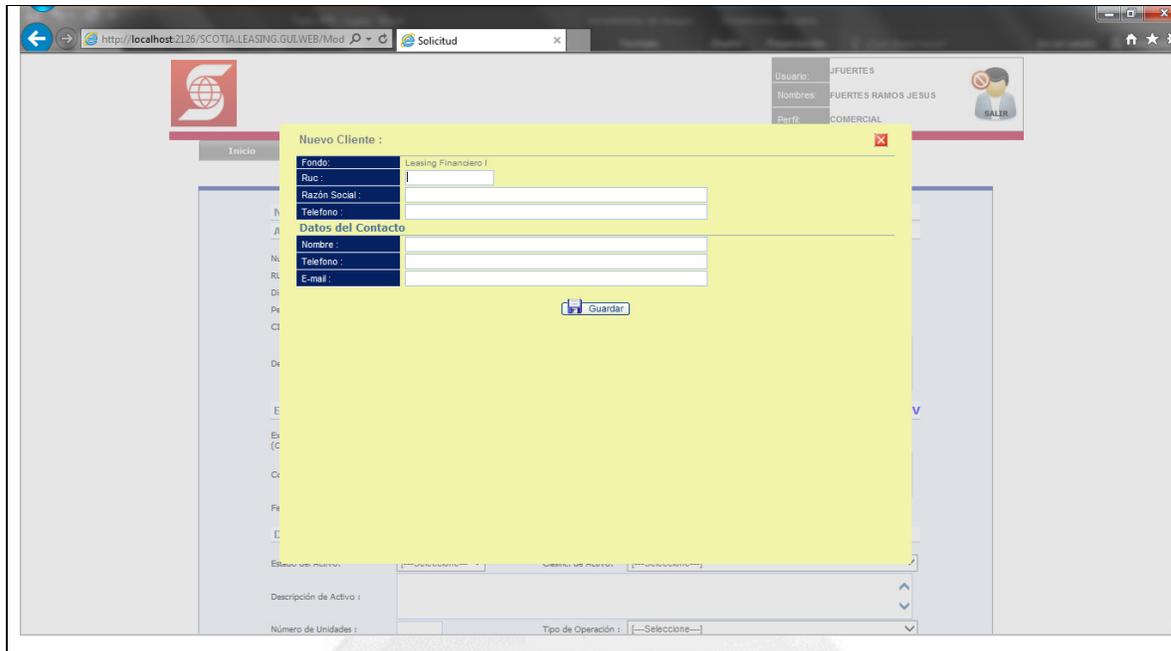


Figura 5.12: Sub flujo - Nuevo cliente. (Fuente: Elaboración propia)

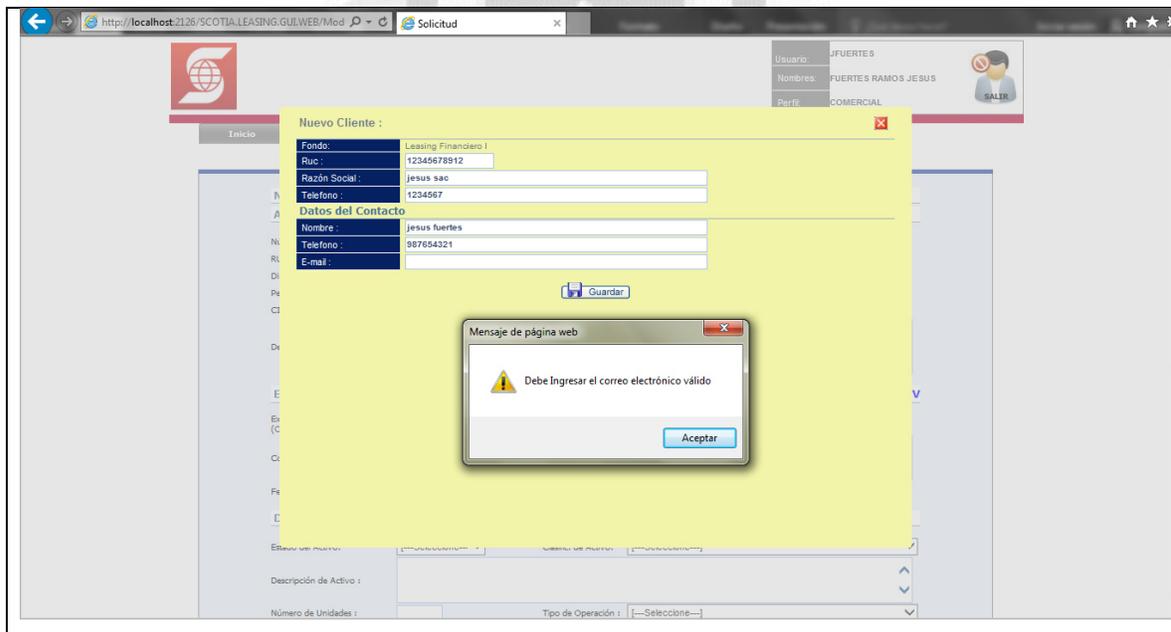


Figura 5.13: Flujo alternativo - Ingresar correo electrónico válido. (Fuente: Elaboración propia)

b. Especificación de caso de uso 02 - Gestionar solicitud

1. Breve Descripción

Este caso de uso permite al **usuario asesor área comercial** crear, modificar solicitudes.

2. Flujo de Eventos

Evento Disparador: El caso de uso comienza cuando el **usuario asesor área comercial** selecciona la opción **nuevo** en la interfaz principal del sistema.

2.1 Flujo Básico <<Gestionar solicitud>>

2.1.1 El sistema muestra la interfaz **búsqueda de clientes**. La interfaz muestra la lista de clientes del banco.

La interfaz contiene los datos: **Razón social, RUC**.

Incluye las opciones: **Nuevo** y **buscar**.

2.1.2 El **usuario asesor área comercial** digita los datos (RUC o razón social) para realizar la búsqueda y luego hace clic en la opción **buscar**.

2.1.3 El sistema incluye el caso de uso **buscar cliente**.

2.1.4 El sistema muestra la empresa de acuerdo a los datos filtrados previamente.

2.1.5 Fin del caso de uso.

2.2 Sub Flujo <<Nueva solicitud>>

2.2.1 El **usuario asesor área comercial** hace clic en la empresa encontrada en el punto 2.1.4.

2.2.2 El sistema muestra la interfaz **registro de solicitud**, donde activan los campos para ingresar los datos de la solicitud: **Descripción operación, línea aprobada, comentario exposición vigente, fecha estim. desem., estado del activo, clasific. de activo, descripción de activo, número de unidades, tipo de operación, inversión en activos, plazo(meses), TIR esperada (%), comisión estructuración (%), cuota adelantada (%), cuota adelantada aprox., valor residual (%), valor residual aprox., uso mensual promedio, opción de compra e información adicional**.

Incluye las opciones: **Guardar** y **cancelar**.

2.2.3 El **usuario asesor área comercial** digita la información de la solicitud sobre los campos activos.

2.2.4 Si el **usuario asesor área comercial** selecciona **guardar**

2.2.4.1 El sistema actualiza la lista de solicitudes en la interfaz.

2.2.4.2 Muestra el mensaje **el registro se ha creado correctamente** en la interfaz **lista de solicitudes**.

2.2.5 Si el **usuario asesor área comercial** selecciona **cancelar**

2.2.5.1 El sistema redirige a la interfaz **lista de solicitudes**.

2.3 Sub Flujo <<Modificar solicitud>>

2.3.1 El **usuario asesor área comercial** hace clic en la opción **editar registro**.

2.3.2 El sistema muestra la interfaz **registro de solicitud**, donde activan los campos para ingresar los datos de la solicitud a ser modificados: **Descripción operación, línea aprobada, comentario exposición vigente, fecha estim. desem., estado del activo, clasific. de activo, descripción de activo, número de unidades, tipo de operación, inversión en activos, plazo(meses), TIR esperada (%), comisión estructuración (%), cuota adelantada (%), cuota adelantada aprox., valor residual (%), valor**

residual aprox., uso mensual promedio, opción de compra, monto opción compra e información adicional.

Incluye las opciones: **Modificar, cancelar y exportar solicitud.**

2.3.3 Si el **usuario asesor área comercial** selecciona **modificar**

2.3.3.1 El sistema guarda los datos de la solicitud.

2.3.3.2 Cierra la interfaz **registro de solicitud.**

2.3.3.3 Muestra el mensaje **el registro se ha modificado correctamente** en la interfaz **lista de solicitudes.**

2.3.4 Si el **usuario asesor área comercial** selecciona **exportar solicitud**

2.3.4.1 El sistema extiende el caso de uso **exportar solicitud.**

2.3.5 Si el **usuario asesor área comercial** selecciona **cancelar**

2.3.5.1 El sistema redirige a la interfaz **lista de solicitudes.**

3. Flujos Alternativos

3.1 << Ingresar descripción de empresa >>

En el punto 2.2.3 del sub flujo **nueva solicitud**, si el **usuario asesor área comercial** no ingresa datos en el campo activo **descripción operación**, al momento de seleccionar **guardar**, el sistema muestra el mensaje **debe ingresar la descripción de empresa**, el flujo continúa en el punto 2.2.3.

3.2 << Ingresar descripción de activos>>

En el punto 2.2.3 del sub flujo **nueva solicitud**, si el **usuario asesor área comercial** no ingresa datos en el campo activo **descripción de activo**, al momento de seleccionar **guardar**, el sistema muestra el mensaje **debe ingresar la descripción de activos**, el flujo continúa en el punto 2.2.3.

4. Requerimientos Especiales

4.1 El caso de uso debe estar disponible a través del **navegador internet explorer**, previo logueo del **usuario asesor área comercial.**

5. Pre Condiciones

5.1 El **usuario asesor área comercial** se haya logueado en el sistema.

5.2 Las solicitudes estén disponibles en el sistema.

6. Post Condiciones

6.1 La solicitud queda registrada en la base de datos.

6.1 La solicitud queda cancelada en el sistema.

7. Puntos de Extensión

7.1 Caso de uso **exportar solicitud** en el punto 2.3.4.1 del sub flujo **modificar solicitud.**

8. Prototipo (GUI)

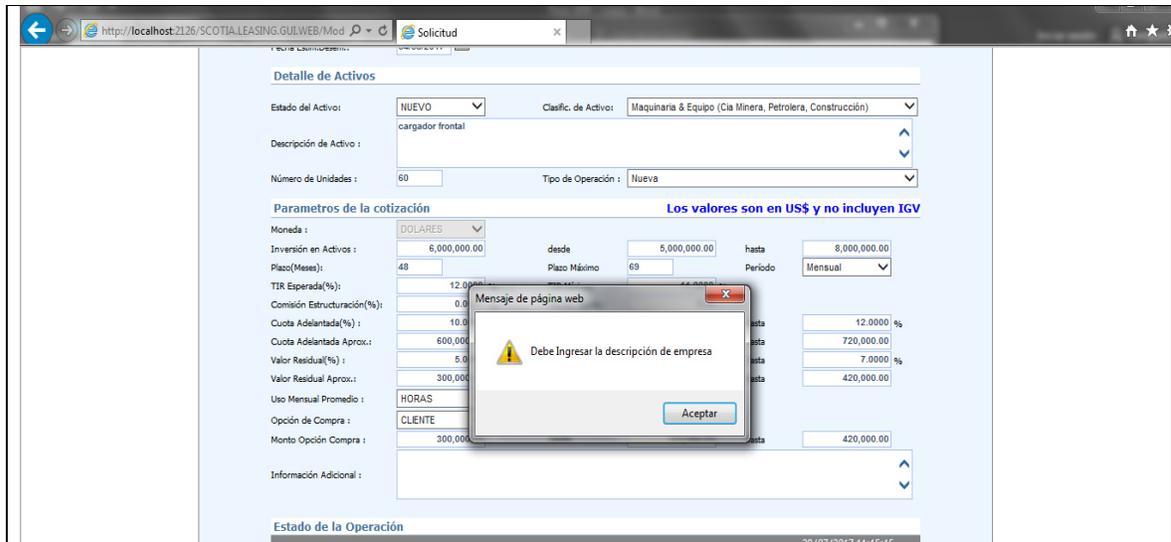


Figura 5.16: Flujo alternativo - Ingresar descripción de empresa. (Fuente: Elaboración propia)

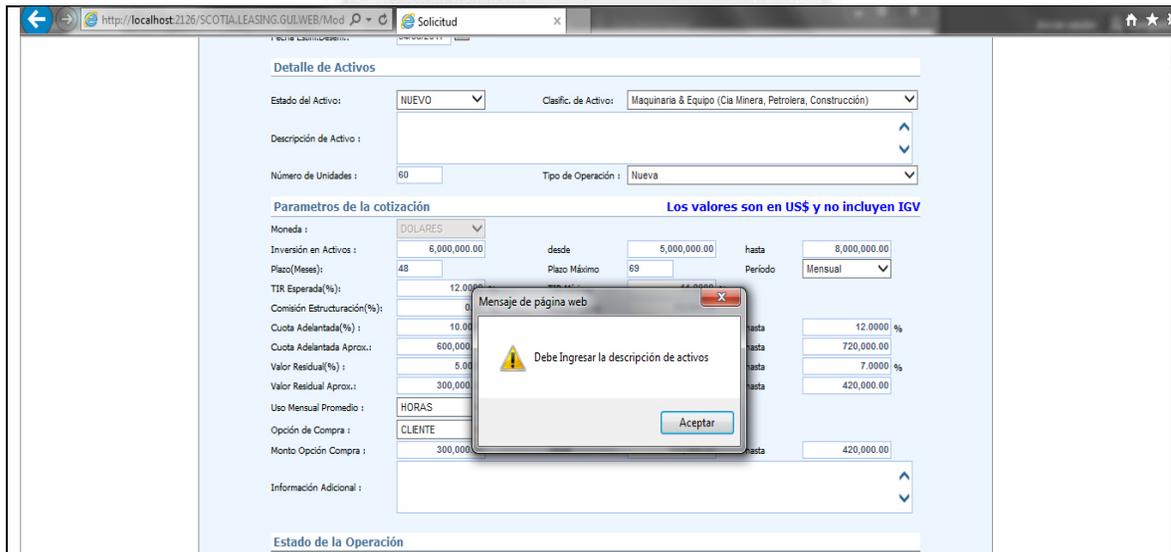


Figura 5.17: Flujo alternativo - Ingresar descripción de activos. (Fuente: Elaboración propia)

c. Especificación de caso de uso 03 - Gestionar aprobación inicial

1. Breve Descripción

Este caso de uso permite al **usuario gerente área comercial** modificar y realizar la aprobación inicial de las solicitudes.

2. Flujo de Eventos

Evento Disparador: El caso de uso comienza cuando el **usuario gerente área comercial** inicia sesión en el sistema.

2.1 Flujo Básico <<Gestionar aprobación inicial>>

2.1.1 El sistema muestra la interfaz **lista de solicitudes**. La interfaz muestra la lista de solicitudes registradas.

La interfaz contiene los datos: **Ver., apro. sol., solicitud, contrato, cliente, fondo, tipo operación, fecha solicitud y estado**.

Incluye las opciones: **Buscar, ver registro y aprobar/observar registro**.

2.1.2 El **usuario gerente área comercial** hace clic en la lista desplegable del campo **estado** y escoge la opción **registrado**, luego clic en la opción **buscar**.

2.1.3 El sistema incluye el caso de uso **buscar solicitud**.

2.1.4 El sistema muestra la solicitud de acuerdo a los datos filtrados previamente.

2.1.5 Fin del caso de uso.

2.2 Sub Flujo <<Realizar aprobación inicial>>

2.2.1 El **usuario gerente área comercial** hace clic en la opción **aprobar/observar registro**.

2.2.2 El sistema muestra la interfaz **aprobar/desaprobar/ observar solicitud (número de la solicitud)**, donde activan el campo para ingresar los datos de la solicitud: **Observación**

Incluye las opciones: **Aprobar, observar, desaprobar y cerrar**.

2.2.3 El **usuario gerente área comercial** digita la observación respectiva en el campo activo.

2.2.4 Si el **usuario gerente área comercial** selecciona **aprobar**

2.2.4.1 El sistema redirige a la interfaz **lista de solicitudes** con el mensaje **la solicitud fue aprobada correctamente**.

2.2.5 Si el **usuario gerente área comercial** selecciona **observar**

2.2.5.1 El sistema redirige a la interfaz **lista de solicitudes** con el mensaje **la solicitud fue observada correctamente**.

2.2.6 Si el **usuario gerente área comercial** selecciona **desaprobar**

2.2.6.1 El sistema redirige a la interfaz **lista de solicitudes** con el mensaje **la solicitud fue desaprobadada correctamente**.

2.2.7 Si el **usuario gerente área comercial** presiona el botón **cerrar**

2.2.7.1 El sistema redirige a la interfaz **lista de solicitudes**.

3. Flujos Alternativos

3.1 <<No se encontraron registros>>

En el punto 2.1.2 del flujo básico **gestionar aprobación inicial**, en caso no existe la solicitud, el sistema muestra el mensaje **no se encontraron registros**, el flujo continúa en el punto 2.1.2.

4. Requerimientos Especiales

4.1 El caso de uso debe estar disponible a través del **navegador internet explorer**, previo logueo del **usuario gerente área comercial**.

5. Pre Condiciones

5.1 El **usuario gerente área comercial** se haya logueado en el sistema.

5.2 Las solicitudes estén disponibles en el sistema.

6. Post Condiciones

6.1 La aprobación inicial de la solicitud queda registrada en la base de datos.

7. Puntos de Extensión

Ninguno.

8. Prototipo (GUI)

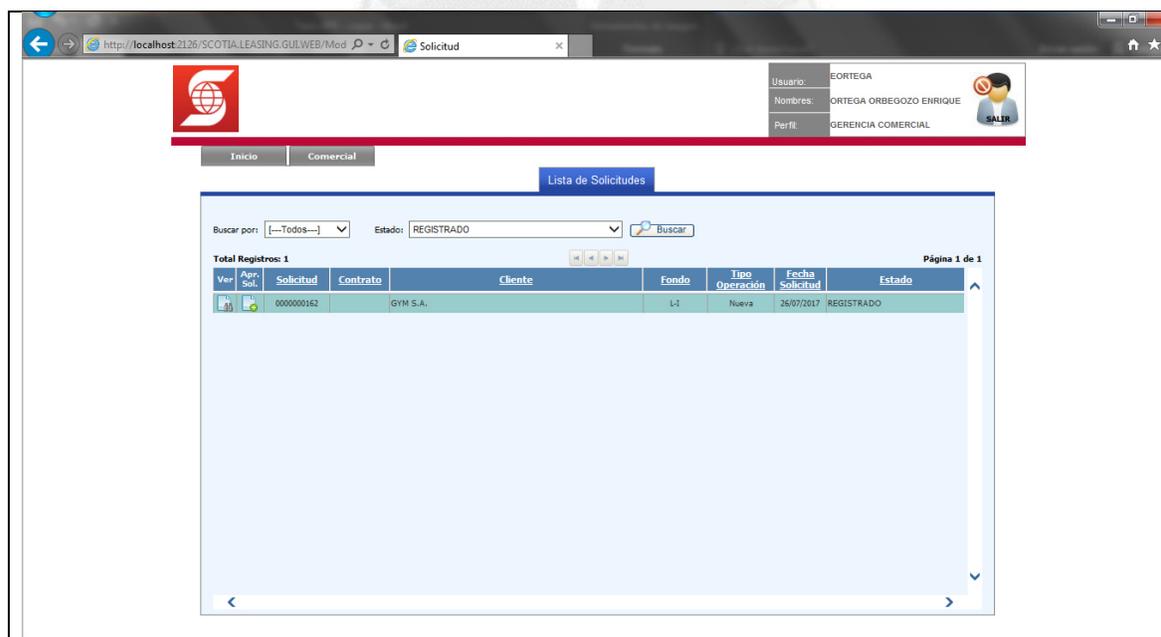


Figura 5.18: Flujo básico - Gestionar aprobación inicial. (Fuente: Elaboración propia)

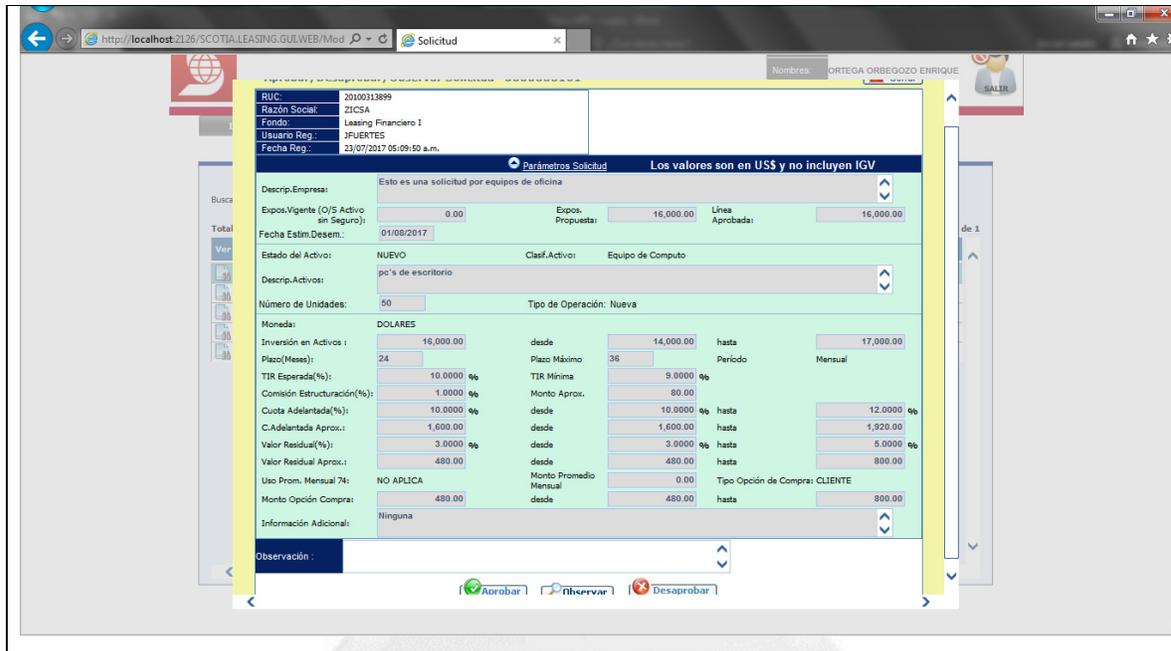


Figura 5.19: Sub Flujo - Realizar aprobación inicial. (Fuente: Elaboración propia)

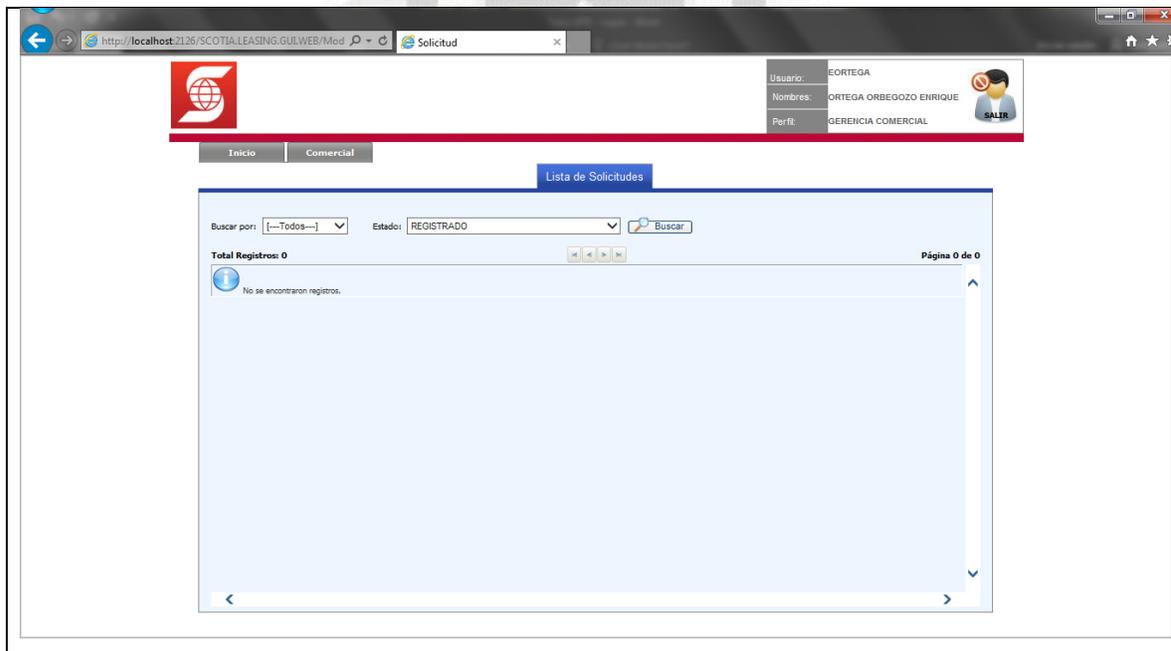


Figura 5.20: Flujo alternativo - No se encontraron registros. (Fuente: Elaboración propia)

d. Especificación de caso de uso 04 - Gestionar cotización

1. Breve Descripción

Este caso de uso permite al **usuario asesor área comercial** agregar, modificar y aprobar cotizaciones, así como registrar datos complementarios de la misma.

2. Flujo de Eventos

Evento Disparador: El caso de uso comienza cuando el **usuario asesor área comercial** hace clic en el menú **comercial**, opción **cotizador** en la interfaz principal del sistema.

2.1 Flujo básico<<Cotizar solicitud>>

2.1.1 El sistema muestra la interfaz **lista de cotizaciones**. La interfaz muestra la lista de solicitudes registradas.

La interfaz contiene los datos: **Ver, solicitud, contrato, cliente, fondo, tipo operación, fecha solicitud y estado**.

Incluye las opciones: **Ver registro** y **buscar**.

2.1.2 El **usuario gerente área comercial** hace clic en la lista desplegable del campo **estado** y escoge la opción **aprobado comercial**, luego clic en la opción **buscar**.

2.1.3 El sistema incluye el caso de uso **buscar solicitud**.

2.1.4 El sistema muestra la solicitud de acuerdo a los datos filtrados previamente.

2.1.5 Fin del caso de uso.

2.1.6 El **usuario asesor área comercial** hace clic en la opción **ver registro**.

2.1.7 El sistema muestra la interfaz **registro de cotizaciones**. La interfaz muestra información de la solicitud, así como el listado de cotizaciones relacionadas a dicha solicitud.

Incluye las opciones: **Nuevo** y **cancelar**.

2.1.8 El **usuario asesor área comercial** hace clic en la opción **nuevo**

2.1.9 El sistema muestra la interfaz **nueva cotización**, donde activan los campos para ingresar los datos de la cotización: **Seguros, activos netos, comentarios y observación**.

Incluye las opciones: **Agregar** en el ítem **conceptos**, **agregar** en el ítem **cuotas extraordinarias** y **guardar**.

2.1.6 El **usuario asesor área comercial** digita los datos de la cotización sobre los campos activos.

2.1.7 Si el **usuario asesor área comercial** selecciona **guardar**

2.1.7.1 El sistema muestra el mensaje **el registro se ha creado correctamente** y actualiza la lista de cotizaciones en la interfaz.

2.1.8 Si el **usuario asesor área comercial** selecciona **cancelar**

2.1.8.1 El sistema redirige a la interfaz **lista de cotizaciones**.

2.2 Sub Flujo <<Modificar cotización>>

2.2.1 El **usuario asesor área comercial** hace clic en la opción **ver registro**.

2.2.1 El sistema muestra la interfaz **registro de cotizaciones**. La interfaz muestra información de la solicitud, así como el **listado de cotizaciones** relacionadas a dicha solicitud, esta parte de la interfaz contiene los datos: **ver, edit., envío correo, n°, inversión en activos, TIR, plazo, cuota, cuota**

adelantada, valor residual, comisión estructuración, comisión seguro, TIR all in, TIR inc ce, fecha reg. y estado.

Incluye las opciones: **Nuevo, cancelar, ver registro, editar registro y registrada.**

2.2.3 El **usuario asesor área comercial** hace clic en la opción **editar**

2.2.4 El sistema muestra la interfaz **editar cotización**, donde activan los campos para ingresar los datos de la cotización: **Seguros, activos netos, comentarios y observación.**

Incluye las opciones: **Agregar** en el ítem **conceptos**, **agregar** en el ítem **cuotas extraordinarias**, **ver cronograma, guardar y cerrar.**

2.2.5 Si el **usuario asesor área comercial** selecciona **guardar**

2.2.5.1 El sistema guarda los datos de la cotización.

2.2.5.2 Cierra la interfaz **editar cotización.**

2.2.5.3 Muestra el mensaje **el registro se ha modificado correctamente** en la interfaz **registro de cotizaciones.**

2.2.6 Si el **usuario asesor área comercial** selecciona **ver cronograma**

2.2.6.1 El sistema extiende el caso de uso **exportar cotización.**

2.2.7 Si el **usuario asesor área comercial** selecciona **cerrar**

2.2.7.1 El sistema cierra la interfaz **editar cotización.**

2.3 Sub Flujo <<Aprobar cotización>>

2.3.1 El **usuario asesor área comercial** hace clic en la opción **registrada**

2.3.2 El sistema muestra la interfaz **aprobar/rechazar cotización**, donde activan el campo para ingresar los datos de la cotización: **Observación**

Incluye las opciones: **Aprobar, rechazar y cerrar.**

2.3.3 El **usuario asesor área comercial** digita la observación respectiva en el campo activo.

2.3.4 Si el **usuario asesor área comercial** selecciona **aprobar**

2.3.4.1 El sistema redirige a la interfaz **listado de cotizaciones** con el mensaje **La cotización se ha aprobado correctamente con el N°Contrato: X-XXXX.**

2.3.5 Si el **usuario asesor área comercial** selecciona **rechazar**

2.3.5.1 El sistema redirige a la interfaz **listado de cotizaciones** con el mensaje **la cotización se ha rechazado correctamente.**

2.3.6 Si el **usuario asesor área comercial** presiona el botón **cerrar**

2.3.6.1 El sistema cierra la interfaz **aprobar/rechazar cotización.**

2.4 Sub Flujo<<Registrar datos complementarios>>

2.4.1 El **usuario asesor área comercial** hace clic en el módulo **comercial**, opción **datos complementarios.**

La interfaz contiene los datos: **Ver, solicitud, contrato, cliente, fondo, tipo operación, fecha solicitud y estado.**

Incluye la opción: **Ver registro y buscar.**

2.4.2 El **usuario asesor área comercial** selecciona **ver registro.**

2.4.3 El sistema muestra la interfaz **detalle datos complementarios**.

La interfaz muestra información de la solicitud, datos de la cotización y los campos activos: **Consideraciones especiales, débito automático y banco asociado**.

Incluye las opciones: **Guardar y cancelar**.

2.4.4 El **usuario asesor área comercial** digita los datos complementarios de la cotización sobre los campos activos.

2.4.5 Si el **usuario asesor área comercial** selecciona **guardar**

2.4.5.1 El sistema muestra el mensaje **el registro se ha creado correctamente** y actualiza la lista de cotizaciones en la interfaz.

2.4.6 Si el **usuario asesor área comercial** escoge la opción **cancelar**

2.4.6.1 El sistema redirige a la interfaz **lista de cotizaciones**.

3. Flujos Alternativos

3.1 << Ingresar cantidad para activos netos >>

En el punto 2.1.6 del flujo básico **cotizar solicitud**, si el **usuario asesor área comercial** no digita una cantidad de activos a cotizar, al momento de seleccionar **guardar**, el sistema muestra el mensaje **ingrese una cantidad para el concepto activos netos**, el flujo continúa en el punto 2.1.6.

3.2 << Especificar las consideraciones especiales >>

En el punto 2.4.4 del sub flujo **registrar datos complementarios**, si el **usuario asesor área comercial** no digita las consideraciones especiales de la cotización, al momento de seleccionar **guardar**, el sistema muestra el mensaje **debe especificar las consideraciones especiales**, el flujo continúa en el punto 2.4.4.

4. Requerimientos Especiales

4.1 El caso de uso debe estar disponible a través del **navegador internet explorer**, previo logueo del **usuario asesor área comercial**.

5. Pre Condiciones

5.1 El **usuario asesor área comercial** se haya logueado en el sistema.

5.2 Las cotizaciones estén disponibles en el sistema.

6. Post Condiciones

6.1 La cotización queda registrada en la base de datos.

6.1 La cotización queda cancelada en el sistema.

7. Puntos de Extensión

7.1 Caso de uso **exportar cotización** en el punto 2.2.6.1 del sub flujo **modificar cotización**.

8. Prototipo (GUI)

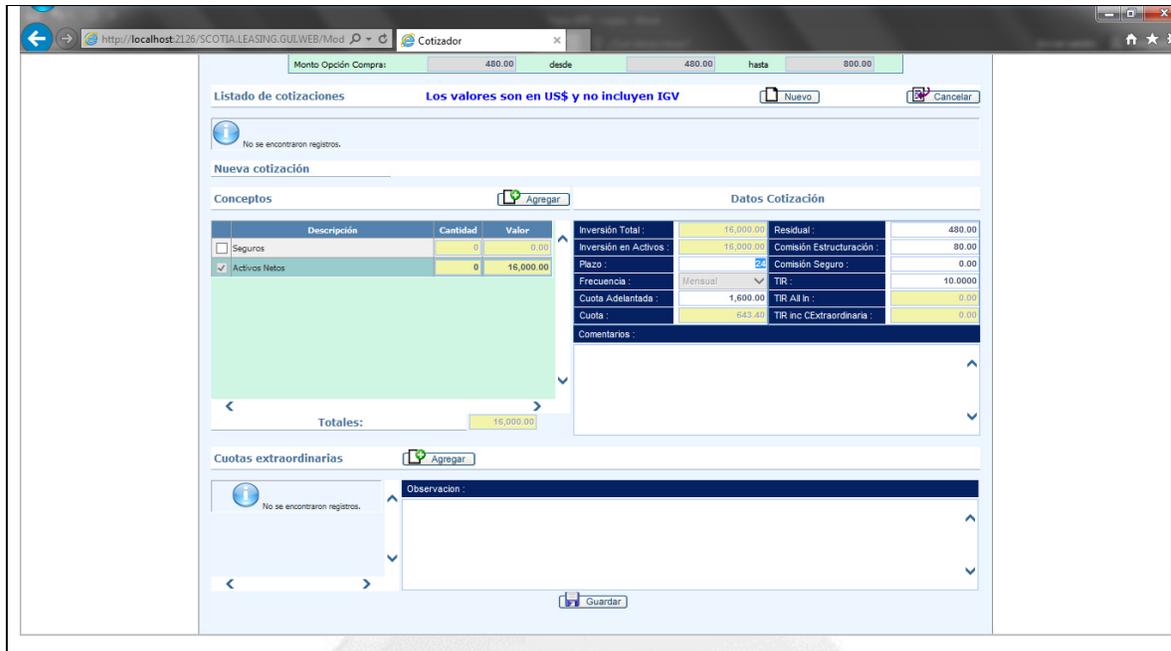


Figura 5.21: Flujo básico - Cotizar solicitud. (Fuente: Elaboración propia)

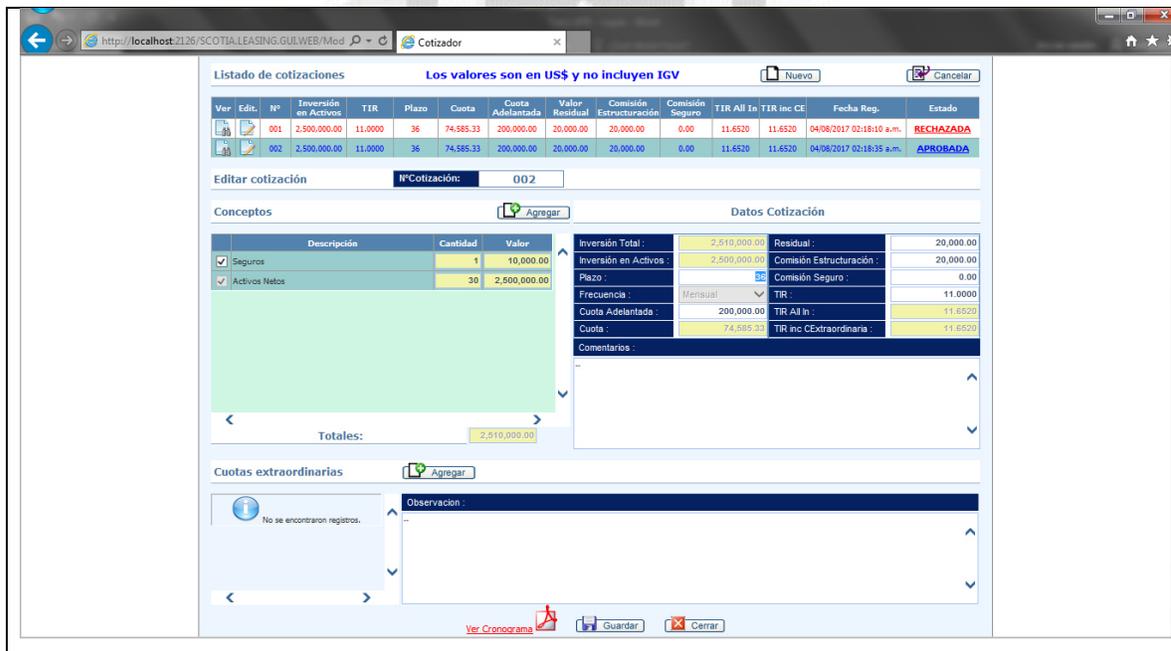


Figura 5.22: Sub flujo - Modificar cotización. (Fuente: Elaboración propia)

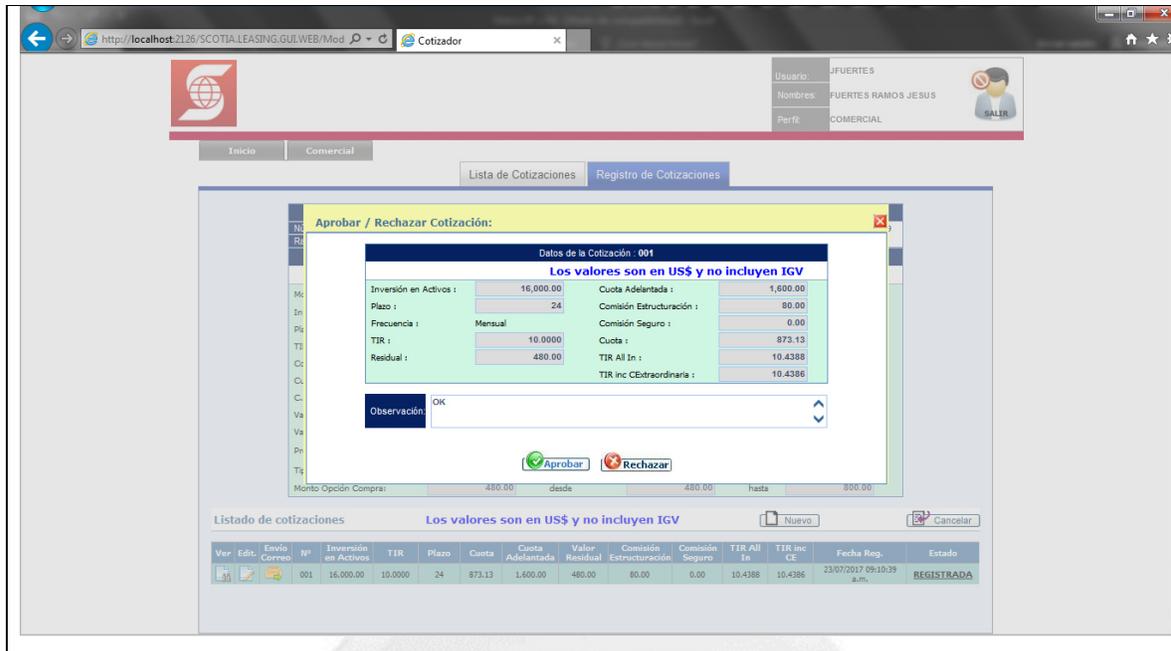


Figura 5.23: Sub flujo - Aprobar cotización. (Fuente: Elaboración propia)

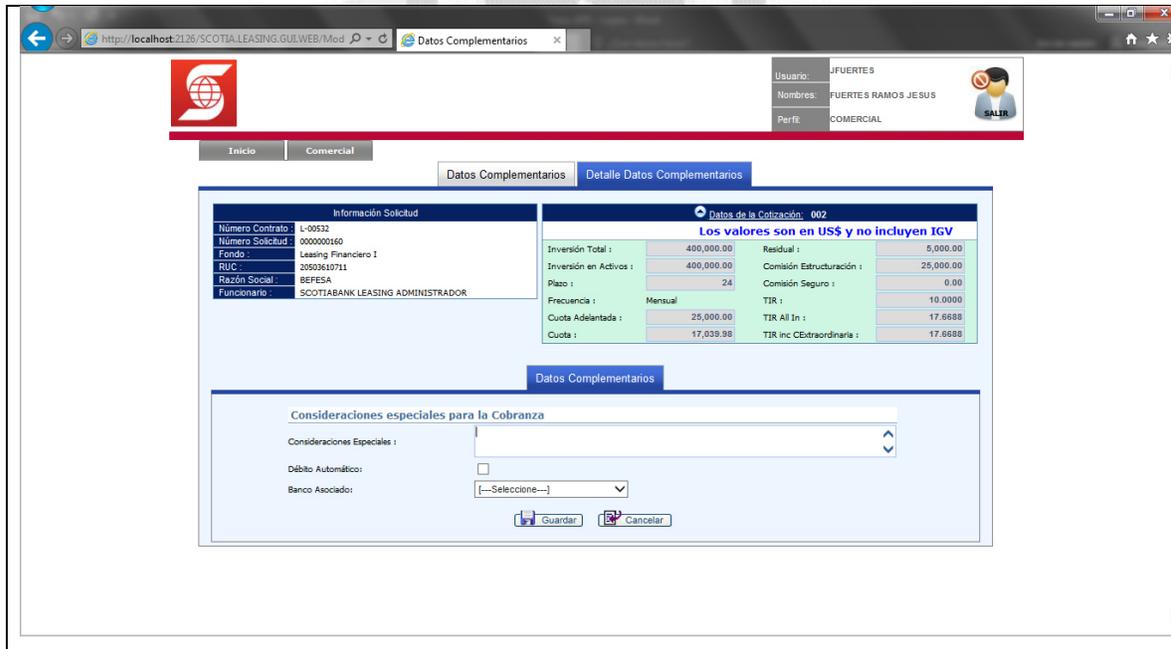


Figura 5.24: Sub flujo - Registrar datos complementarios. (Fuente: Elaboración propia)

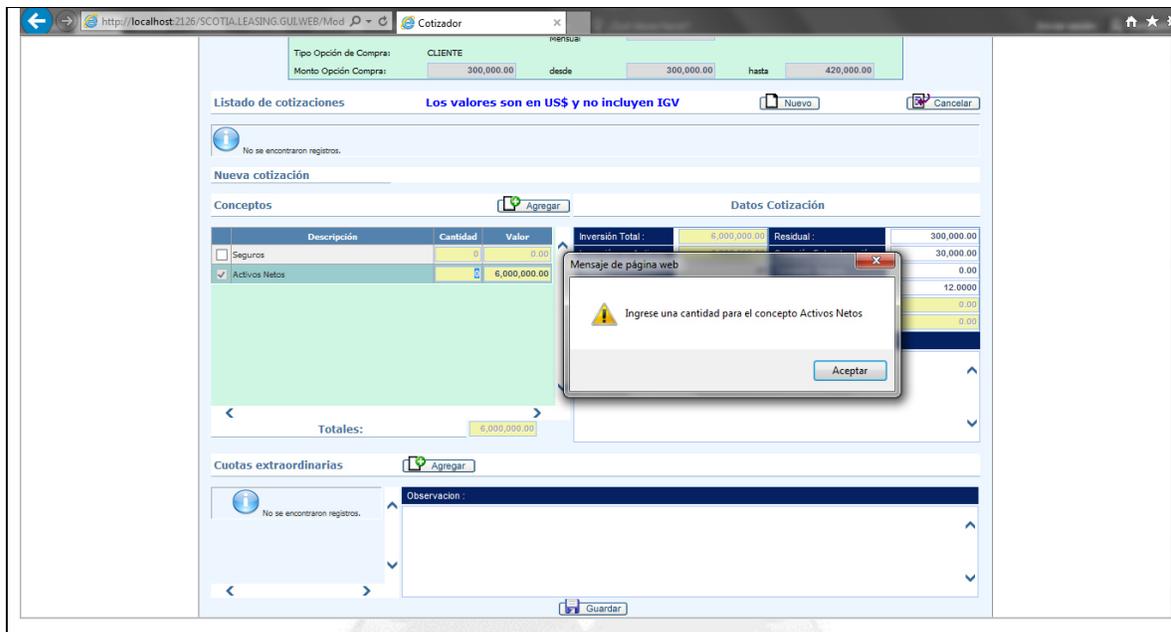


Figura 5.25: Flujo alternativo - Ingresar cantidad para activos netos. (Fuente: Elaboración propia)

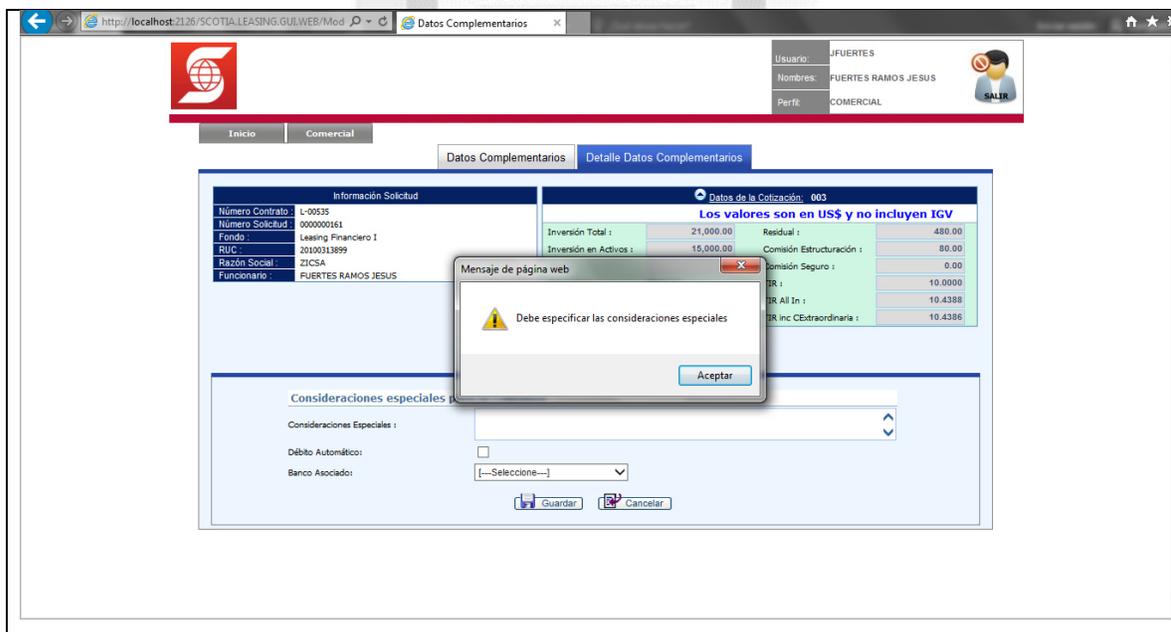


Figura 5.26: Flujo alternativo - Especificar las consideraciones especiales. (Fuente: Elaboración propia)

e. Especificación de caso de uso 05 - Gestionar información de riesgos

1. Breve Descripción

Este caso de uso permite al **usuario analista área de riesgos** registrar un formulario de riesgos, agregar modalidades de garantía y covenants de los clientes.

2. Flujo de Eventos

Evento Disparador: El caso de uso comienza cuando el **usuario analista área de riesgos** hace clic en el menú **riesgos**, opción **completar datos** en la interfaz principal del sistema.

2.1 Flujo Básico <<Gestionar información de riesgos>>

2.1.1 El sistema muestra la interfaz **riesgos**. La interfaz muestra la lista de solicitudes registradas.

La interfaz contiene los datos: **Ver, envío comité, solicitud, contrato, cliente, fondo, tipo operación, fecha solicitud, envío comité** y **estado**.

Incluye las opciones: **Buscar, ver registro** y **enviar solicitud al comité de inversión**.

2.1.2 El **usuario analista área de riesgos** escoge la opción **cotización aprobada** haciendo clic en la lista desplegable del campo **estado** y luego clic en la opción **buscar**.

2.1.3 El sistema incluye el caso de uso **buscar solicitud**.

2.1.4 El sistema muestra la solicitud de acuerdo a los datos filtrados previamente.

2.1.5 Fin del caso de uso.

2.2 Sub Flujo <<Registrar formulario de riesgos>>

2.2.1 El **usuario analista área de riesgos** hace clic en la opción **ver registro**.

2.2.2 El sistema muestra la interfaz **formulario de riesgos KYC**, donde activan los campos para ingresar la información: **Especificar (%)**, **comentario Infocorp**, **información de proveedores**, **información bancaria**, **info. reguladores gremio**, **especificar política** y **detallar acuerdos**.

Incluye las opciones: **Guardar** y **cancelar**.

2.2.3 El **usuario analista área de riesgos** digita la información en los campos activos.

2.2.4 Si el **usuario analista área de riesgos** selecciona **guardar**

2.2.4.1 El sistema guarda la información con el mensaje **el registro se ha modificado correctamente**.

2.2.5 Si el **usuario analista área de riesgos** selecciona **cancelar**

2.2.5.1 El sistema redirige a la interfaz **riesgos**.

2.3 Sub Flujo <<Registrar garantía>>

2.3.1 El **usuario analista área de riesgos** hace clic en la opción **ver registro** y luego en **garantías**.

2.3.2 El sistema muestra la interfaz **garantías**. La interfaz muestra la lista de garantías registradas.

La interfaz contiene los datos: **Ver, edit., elim., tipo garantía, monto garantía, detalle garantía, descripción garantía, usuario creación, fecha creación, usuario actualización, fecha actualización** y **estado**.

Incluye las opciones: **Nuevo** y **cancelar**.

2.3.3 El **usuario analista área de riesgos** hace clic en la opción **nuevo**.

2.3.4 El sistema muestra la interfaz **nueva garantía**, donde activan los campos para ingresar los datos de la garantía: **Tipo garantía, monto, detalle garantía, descripción garantía** y **estado**.

Incluye las opciones: **Guardar** y **cancelar**.

2.3.5 El **usuario analista área de riesgos** digita la información en los campos activos.

2.3.6 Si el **usuario analista área de riesgos** selecciona **guardar**

2.3.6.1 El sistema muestra el mensaje **el registro se ha creado correctamente** y actualiza la lista de garantías en la interfaz.

2.3.7 Si el **usuario analista área de riesgos** selecciona **cancelar**

2.3.7.1 El sistema cierra la interfaz **nueva garantía**.

2.4 Sub Flujo << Registrar covenant>>

2.4.1 El **usuario analista área de riesgos** hace clic en la opción **ver registro** y luego en **covenants**.

2.4.2 El sistema muestra la interfaz **covenants**. La interfaz muestra la lista de covenants registrados.

La interfaz contiene los datos: **Ver, edit., elim., tipo covenant, detalle covenant, periodicidad revisión, descripción covenant, fecha seguimiento, usuario creación, fecha creación, usuario actualización, fecha actualización** y estado.

Incluye las opciones: **Nuevo** y **cancelar**.

2.4.3 El **usuario analista área de riesgos** hace clic en la opción **nuevo**.

2.4.4 El sistema muestra la interfaz **nuevo covenant**, donde activan los campos para ingresar los datos de los covenants: **Tipo covenant, detalle covenant, periodicidad de revisión, descripción, fecha de seguimiento** y estado.

Incluye las opciones: **Guardar** y **cancelar**.

2.4.5 El **usuario analista área de riesgos** digita la información en los campos activos.

2.4.6 Si el **usuario analista área de riesgos** selecciona **guardar**

2.4.6.1 El sistema muestra el mensaje **el registro se ha creado correctamente** y actualiza la lista de covenants en la interfaz.

2.4.7 Si el **usuario analista área de riesgos** selecciona **cancelar**

2.4.7.1 El sistema cierra la interfaz **nuevo covenant**.

2.5 Sub Flujo<<Enviar solicitud>>

2.5.1 El **usuario analista área de riesgos** hace clic en la opción **enviar solicitud al comité de inversión**.

2.5.2 El sistema muestra el aviso **¿enviar a comité?**

Incluye las opciones: **Aceptar** y **cancelar**.

2.5.3 Si el **usuario analista área de riesgos** selecciona **aceptar**

2.5.3.1 El sistema muestra el mensaje **se notificó correctamente a los miembros del comité de inversión**.

2.5.4 Si el **usuario analista área de riesgos** escoge la opción **cancelar**

2.5.4.1 El sistema cierra el aviso.

3. Flujos Alternativos

3.1 << Especificar detalle de la garantía>>

En el punto 2.3.5 del sub flujo **registrar garantía**, si el **usuario analista área de riesgos** no digita el detalle de la garantía, al momento de seleccionar **guardar**, el sistema muestra el mensaje **debe especificar el detalle de la garantía**, el flujo continúa en el punto 2.3.5.

3.2 << Especificar detalle del covenant >>

En el punto 2.4.5 del sub flujo **registrar covenant**, si el **usuario analista área de riesgos** no digita el detalle del covenant, al momento de seleccionar **guardar**, el sistema muestra el mensaje **debe especificar el detalle de covenant**, el flujo continúa en el punto 2.4.5.

4. Requerimientos Especiales

4.1 El caso de uso debe estar disponible a través del **navegador internet explorer**, previo logeo del **usuario analista área de riesgos**.

5. Pre Condiciones

5.1 El **usuario analista área de riesgos** se haya logeado en el sistema.

5.2 Las solicitudes estén disponibles en el sistema.

6. Post Condiciones

6.1 El formulario queda registrado en la base de datos.

6.2 La garantía queda registrada en la base de datos.

6.3 El covenant queda registrado en la base de datos.

6.4 El formulario queda cancelado en el sistema.

6.5 La garantía queda cancelada en el sistema.

6.6 El covenant queda cancelado en el sistema.

7. Puntos de Extensión

Ninguno.

8. Prototipo (GUI)

Inicio | Riesgos

Riesgos

NOTA: Solo se mostrará Solicitudes con Contrato.

Buscar por: [---Todos---] Estado: COTIZACIÓN APROBADA [Buscar]

Total Registros: 2 | Página 1 de 1

Ver	Envío Comité	Solicitud	Contrato	Cliente	Fondo	Tipo Operación	Fecha Solicitud	Envío Comité	Estado
		0000000162	L-00537	GYM S.A.	Leasing Financiero 1	Nueva	26/07/2017	NO	COTIZACIÓN APROBADA
		0000000161	L-00536	ZICSA	Leasing Financiero 1	Nueva	23/07/2017	SI	COTIZACIÓN APROBADA

Figura 5.27: Flujo básico - Gestionar información de riesgos. (Fuente: Elaboración propia)

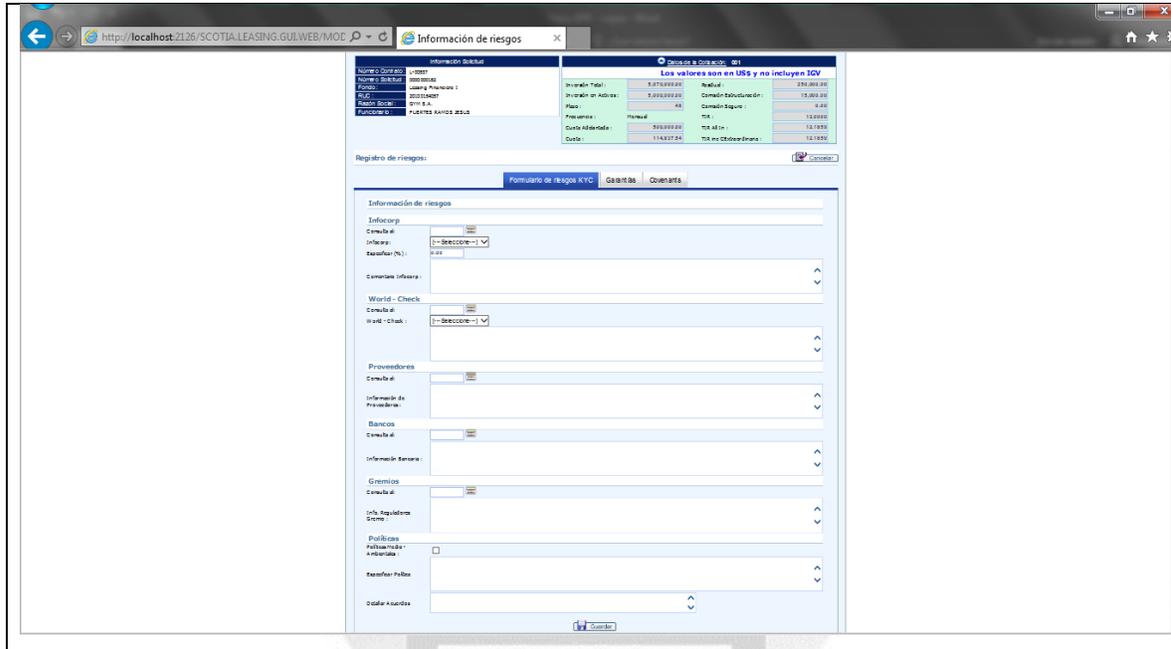


Figura 5.28: Sub flujo - Registrar formulario de riesgos. (Fuente: Elaboración propia)

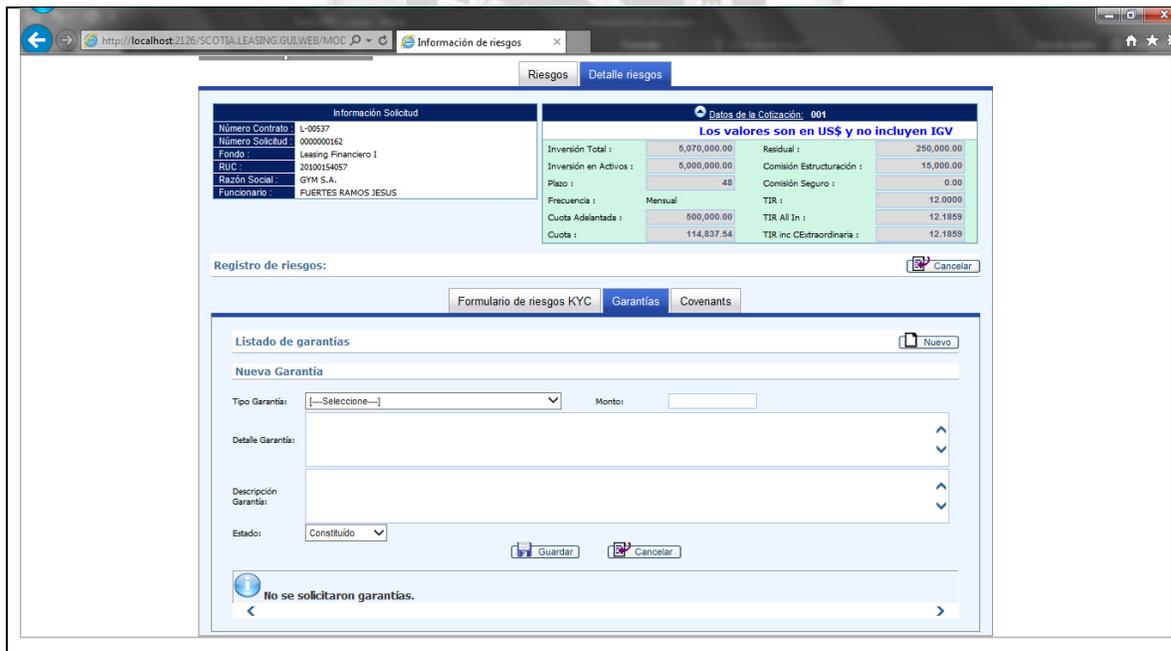


Figura 5.29: Sub flujo - Registrar garantía. (Fuente: Elaboración propia)

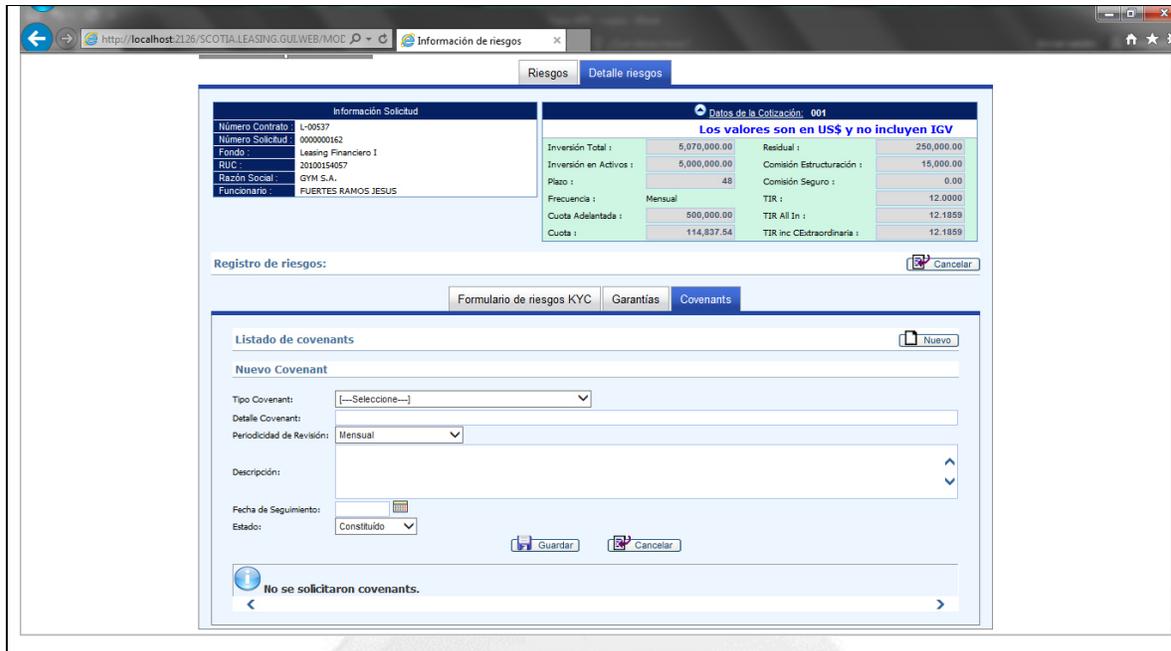


Figura 5.30: Sub flujo - Registrar covenant. (Fuente: Elaboración propia)

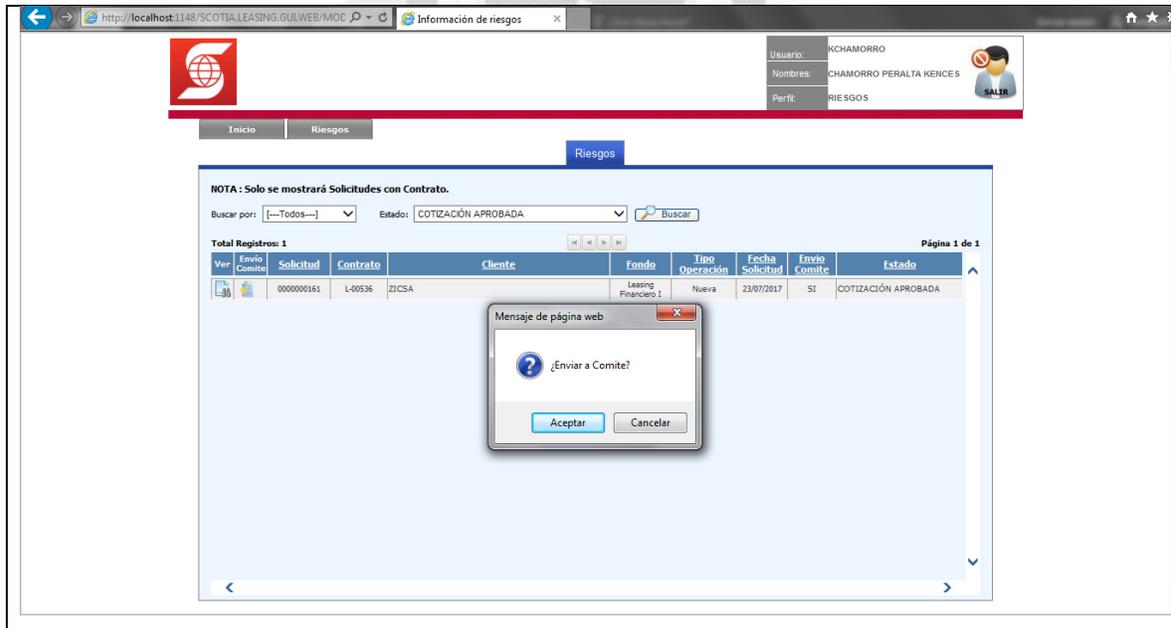


Figura 5.31: Sub flujo - Enviar solicitud. (Fuente: Elaboración propia)

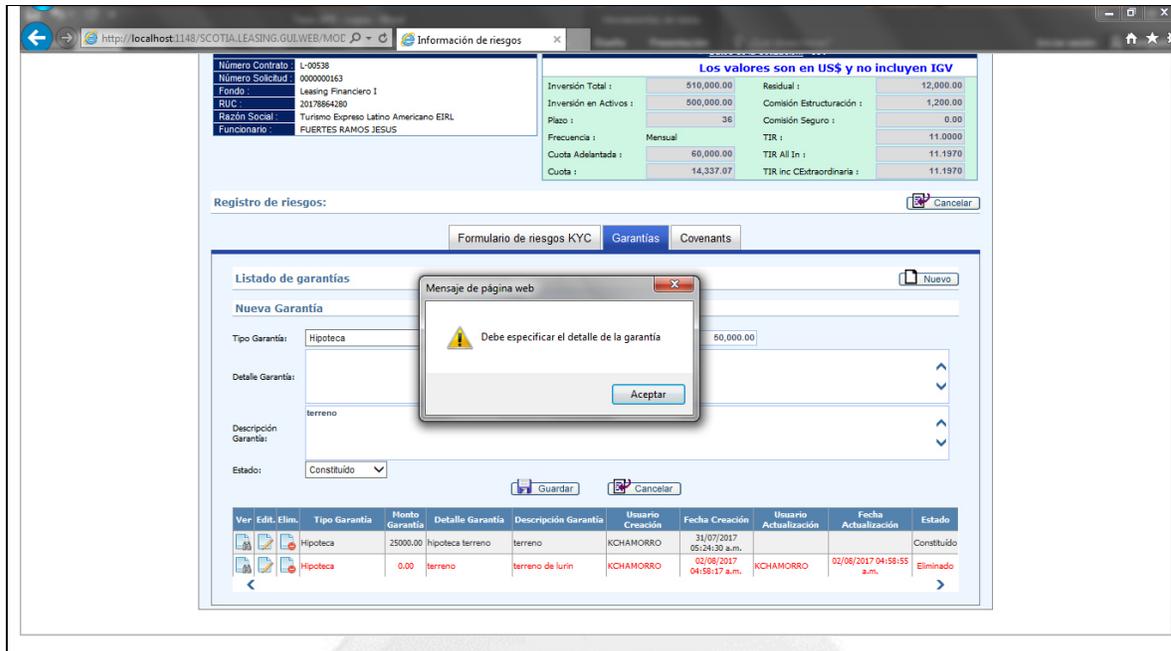


Figura 5.32: Flujo alternativo - Especificar detalle de la garantía. (Fuente: Elaboración propia)

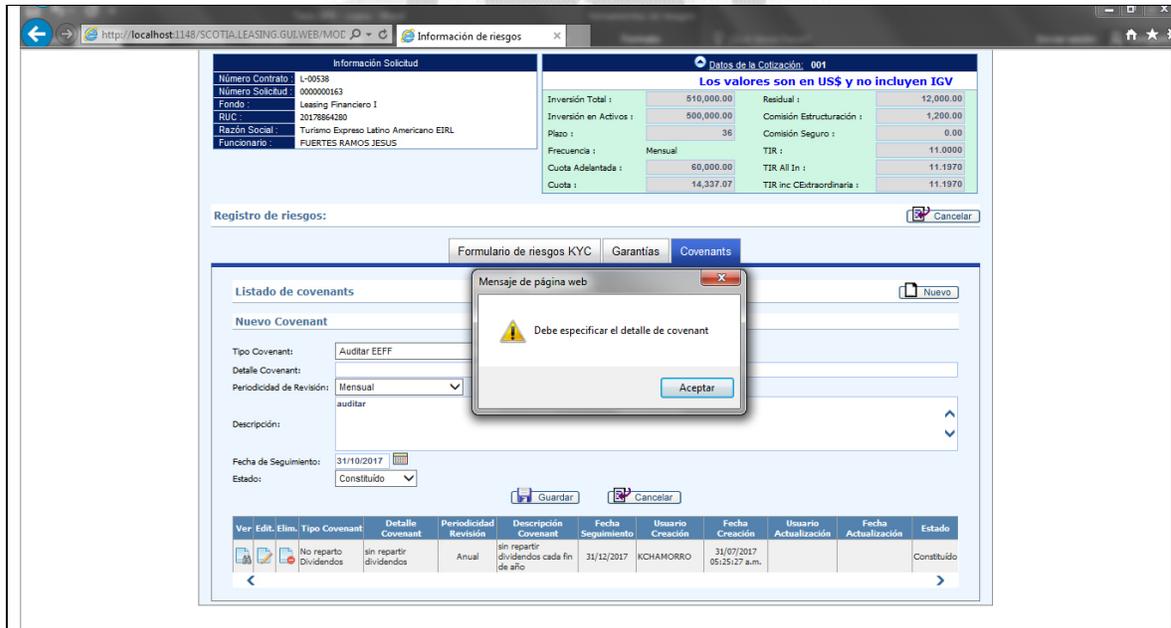


Figura 5.33: Flujo alternativo – Especificar detalle del covenant. (Fuente: Elaboración propia)

f. Especificación de caso de uso 06 - Gestionar aprobación de evaluación

1. Breve Descripción

Este caso de uso permite al **usuario miembro del comité de inversión** realizar la aprobación de la evaluación financiera de las solicitudes.

2. Flujo de Eventos

Evento Disparador: El caso de uso comienza cuando el **usuario miembro del comité de inversión** hace clic en el menú **comité**, opción **solicitudes** en la interfaz principal del sistema.

2.1 Flujo Básico <<Gestionar aprobación de evaluación>>

2.1.1 El sistema muestra la interfaz **lista de solicitudes**. La interfaz muestra la lista de solicitudes registradas.

La interfaz contiene los datos: **Apro. sol., ver hist., solicitud, contrato, cliente, fondo, tipo comité, tipo operación, fecha solicitud y estado**.

Incluye las opciones: **Aprobar/desaprobar, ver historial de aprobaciones y buscar**.

2.1.2 El **usuario miembro del comité de inversión** escoge la opción **pendiente aprobación comité de inversión** haciendo clic en la lista desplegable del campo **estado** y luego clic en la opción **buscar**.

2.1.3 El sistema incluye el caso de uso **buscar solicitud**.

2.1.4 El sistema muestra la solicitud de acuerdo a los datos filtrados previamente.

2.1.5 Fin del caso de uso.

2.2 Sub Flujo <<Aprobar evaluación>>

2.2.1 El **usuario miembro del comité de inversión** hace clic en la opción **aprobar/desaprobar**.

2.2.2 El sistema muestra la interfaz **comité aprobar/desaprobar - solicitud (número de la solicitud)**, donde activan el campo para ingresar los datos de la solicitud: **Observación**

Incluye las opciones: **Aprobar, observar, desaprobar y cerrar**.

2.2.3 El **usuario miembro del comité de inversión** digita la observación respectiva en el campo activo.

2.2.4 Si el **usuario miembro del comité de inversión** selecciona **aprobar**

2.2.4.1 El sistema redirige a la interfaz **lista de solicitudes** con el mensaje **el miembro del comité aprobó la solicitud nro xxxxxx**.

2.2.5 Si el **usuario miembro del comité de inversión** selecciona **desaprobar**

2.2.5.1 El sistema redirige a la interfaz **lista de solicitudes** con el mensaje **el miembro del comité desaprobo la solicitud nro xxxxxx**.

2.2.6 Si el **usuario miembro del comité de inversión** presiona el botón **cerrar**

2.2.6.1 El sistema redirige a la interfaz **lista de solicitudes**.

3. Flujos Alternativos

3.1 <<No se encontraron registros>>

En el punto 2.1.2 del flujo básico **gestionar aprobación de evaluación**, en caso no existe la solicitud, el sistema muestra el mensaje **no se encontraron registros**, el flujo continúa en el punto 2.1.2.

3.2 << Especificar observación de desaprobo >>

En el punto 2.2.3 del sub flujo **aprobar evaluación**, si el **usuario miembro del comité de inversión** no digita la observación de la solicitud, al momento de seleccionar **desaprobar**, el sistema muestra el mensaje **especifique una observación para la desaprobar de la solicitud**, el flujo continúa en el punto 2.2.5.

4. Requerimientos Especiales

4.1 El caso de uso debe estar disponible a través del **navegador internet explorer**, previo logeo del **usuario miembro del comité de inversión**.

5. Pre Condiciones

5.1 El **usuario miembro del comité de inversión** se haya logueado en el sistema.

5.2 Las solicitudes estén disponibles en el sistema.

6. Post Condiciones

6.1 La aprobación de la solicitud evaluada queda registrada en la base de datos.

7. Puntos de Extensión

Ninguno.

8. Prototipo (GUI)

The screenshot displays a web browser window with the URL `http://localhost:1148/SCOTIA.LEASING.GUL.WEB/MOD`. The page title is "COMITE - Solicitud". The user is logged in as "MACEVEDO" with the profile "ACEVEDO ACEVEDO MYTCHELL" and role "COMITE INVERSION 2". The main content area is titled "Lista de Solicitudes" and includes a search bar and a table of requests.

NOTA: El Listado solo estara disponible para los miembros de los Comites

Buscar por: [---Todos---] Estado: PENDIENTE APROBACIÓN COMITÉ INVERSIÓN [Buscar]

Total Registros: 2 Página 1 de 1

Aprob. Sol.	Ver Hist.	Solicitud	Contrato	Cliente	Fondo	Tipo Comite	Tipo Operación	Fecha Solicitud	Estado
		0000000163	L-00538	Turismo Expreso Latino Americano EIRL	Leasing Financiero 1	INVERSION	Nueva	31/07/2017	PENDIENTE APROBACIÓN COMITÉ INVERSIÓN
		0000000162	L-00537	GYM S.A.	Leasing Financiero 1	INVERSION	Nueva	26/07/2017	PENDIENTE APROBACIÓN COMITÉ INVERSIÓN

Figura 5.34: Flujo básico - Gestionar aprobación de evaluación. (Fuente: Elaboración propia)

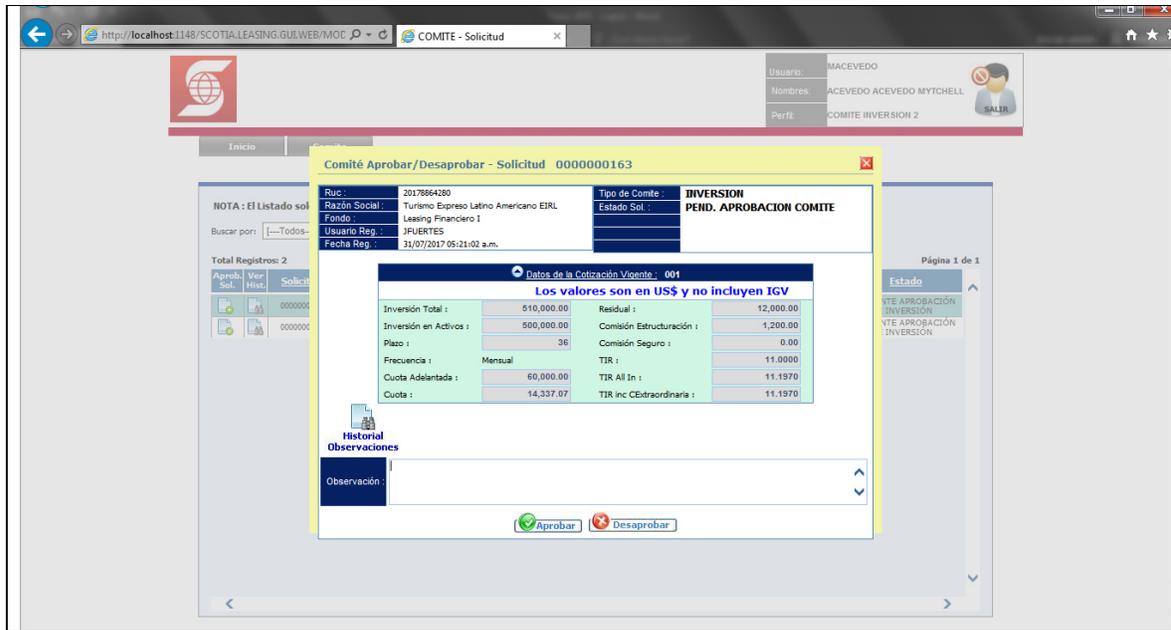


Figura 5.35: Sub flujo - Aprobar evaluación. (Fuente: Elaboración propia)

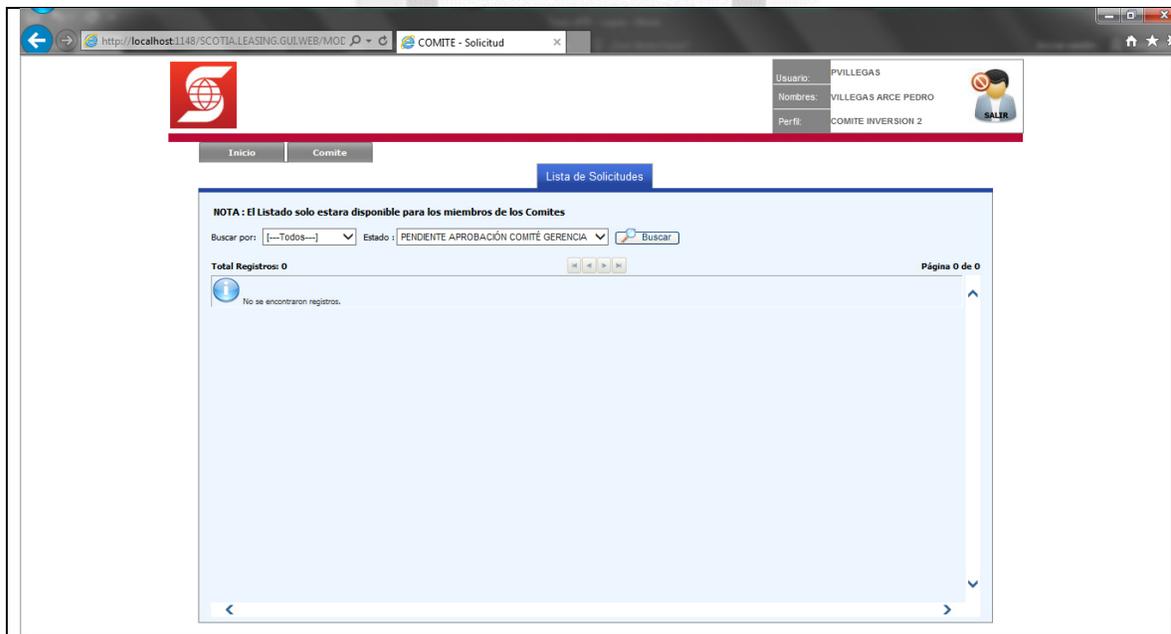


Figura 5.36: Flujo alternativo - No se encontraron registros. (Fuente: Elaboración propia)

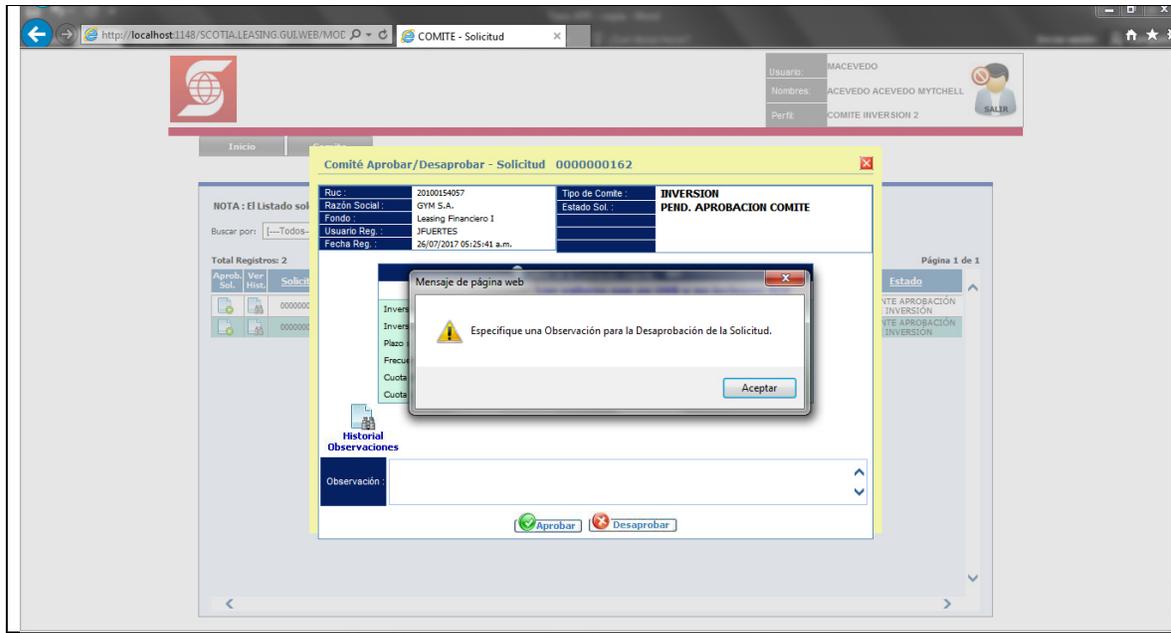
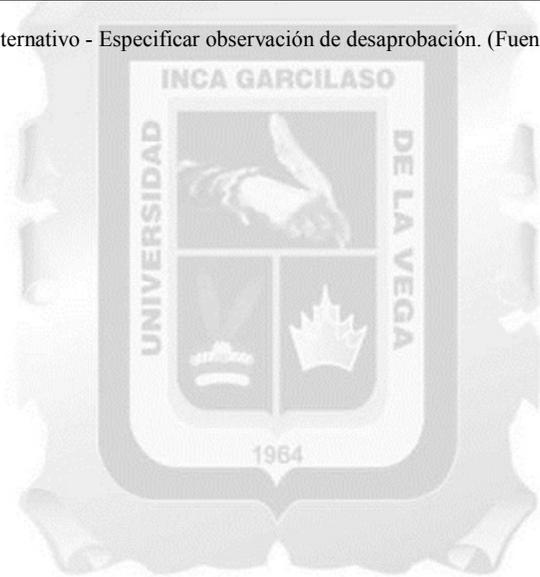


Figura 5.37: Flujo alternativo - Especificar observación de desaprobación. (Fuente: Elaboración propia)



g. Especificación de caso de uso 07 - Gestionar aprobación final

1. Breve Descripción

Este caso de uso permite al **usuario asesor área de operaciones** realizar la aprobación final de la solicitud.

2. Flujo de Eventos

Evento Disparador: El caso de uso comienza cuando el **usuario asesor área de operaciones** hace clic en el menú **aprobación**, opción **aprobación final solicitud** en la interfaz principal del sistema.

2.1 Flujo Básico <<Gestionar aprobación final>>

2.1.1 El sistema muestra la interfaz **lista de solicitudes-aprobación**. La interfaz muestra la lista de solicitudes registradas.

La interfaz contiene los datos: **Aprob., ver hist., solicitud, contrato, cliente, fondo, tipo comité, tipo operación, fecha solicitud** y estado.

Incluye las opciones: **Aprobar/desaprobar, ver historial** y **buscar**.

2.1.2 El **usuario asesor área de operaciones** escoge la opción **aprobado comité de inversión** haciendo clic en la lista desplegable del campo **estado** y luego clic en la opción **buscar**.

2.1.3 El sistema incluye el caso de uso **buscar solicitud**.

2.1.4 El sistema muestra la solicitud de acuerdo a los datos filtrados previamente.

2.1.5 Fin del caso de uso.

2.2 Sub Flujo <<Realizar aprobación final>>

2.2.1 El **usuario asesor área de operaciones** hace clic en la opción **aprobar/desaprobar**.

2.2.2 El sistema muestra la interfaz **aprobar/desaprobar solicitud: (número de la solicitud)**, donde activan el campo para ingresar los datos de la solicitud: **Observación**

Incluye las opciones: **Aprobar, observar, desaprobar** y **cerrar**.

2.2.3 El **usuario asesor área de operaciones** digita la observación respectiva en el campo activo.

2.2.4 Si el **usuario asesor área de operaciones** selecciona **aprobar**

2.2.4.1 El sistema redirige a la interfaz **lista de solicitudes** con el mensaje **se aprobó la solicitud nro xxxxxx por el área de operaciones**.

2.2.5 Si el **usuario asesor área de operaciones** selecciona **desaprobar**

2.2.5.1 El sistema redirige a la interfaz **lista de solicitudes** con el mensaje **se desaprobo la solicitud nro xxxxxx por el área de operaciones**.

2.2.6 Si el **usuario asesor área de operaciones** presiona el botón **cerrar**

2.2.6.1 El sistema redirige a la interfaz **lista de solicitudes-aprobación**.

3. Flujos Alternativos

3.1 <<No se encontraron registros>>

En el punto 2.1.2 del flujo básico **gestionar aprobación inicial**, en caso no existe la solicitud, el sistema muestra el mensaje **no se encontraron registros**, el flujo continúa en el punto 2.1.2.

3.2 << Especificar observación de desaprobarión>>

En el punto 2.2.3 sub flujo **realizar aprobación final**, si el **asesor área de operaciones** no digita la observación de la solicitud, al momento de seleccionar **desaprobar**, el sistema muestra el mensaje **especifique una observación para la desaprobarción de la solicitud**, el flujo continúa en el punto 2.2.5.

4. Requerimientos Especiales

4.1 El caso de uso debe estar disponible a través del **navegador internet explorer**, previo logeo del **usuario asesor área de operaciones**.

5. Pre Condiciones

5.1 El **usuario asesor área de operaciones** se haya logeado en el sistema.

5.2 Las solicitudes estén disponibles en el sistema.

6. Post Condiciones

6.1 La aprobación final de la solicitud queda registrada en la base de datos.

7. Puntos de Extensión

Ninguno.

8. Prototipo (GUI)

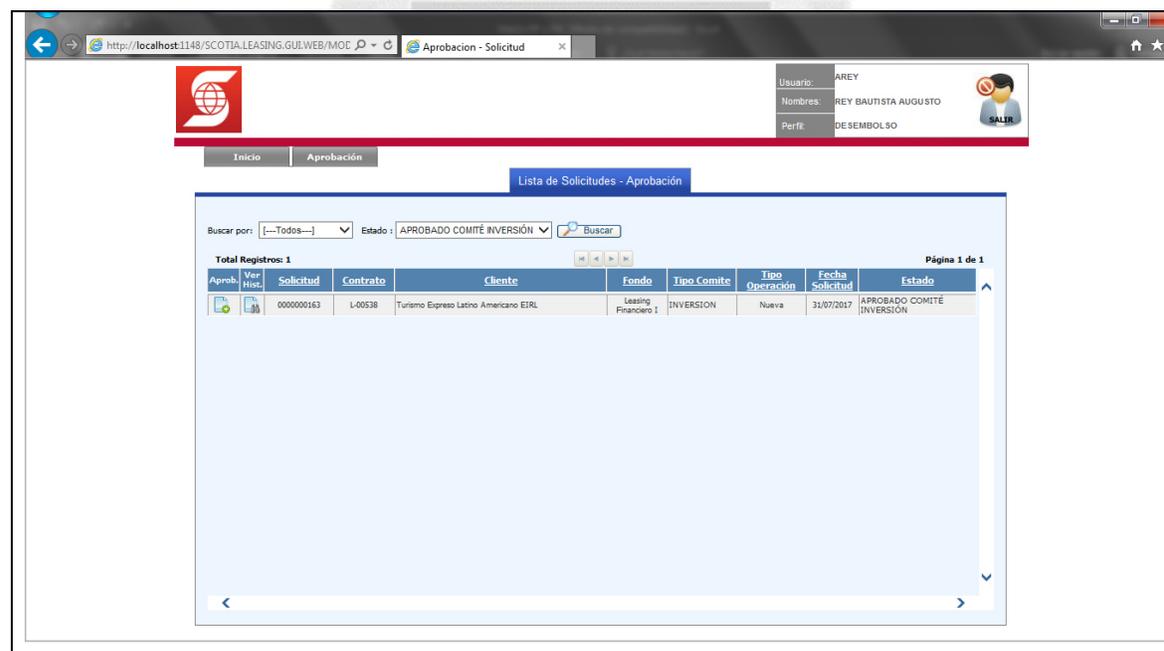


Figura 5.38: Flujo básico - Gestionar aprobación final. (Fuente: Elaboración propia)

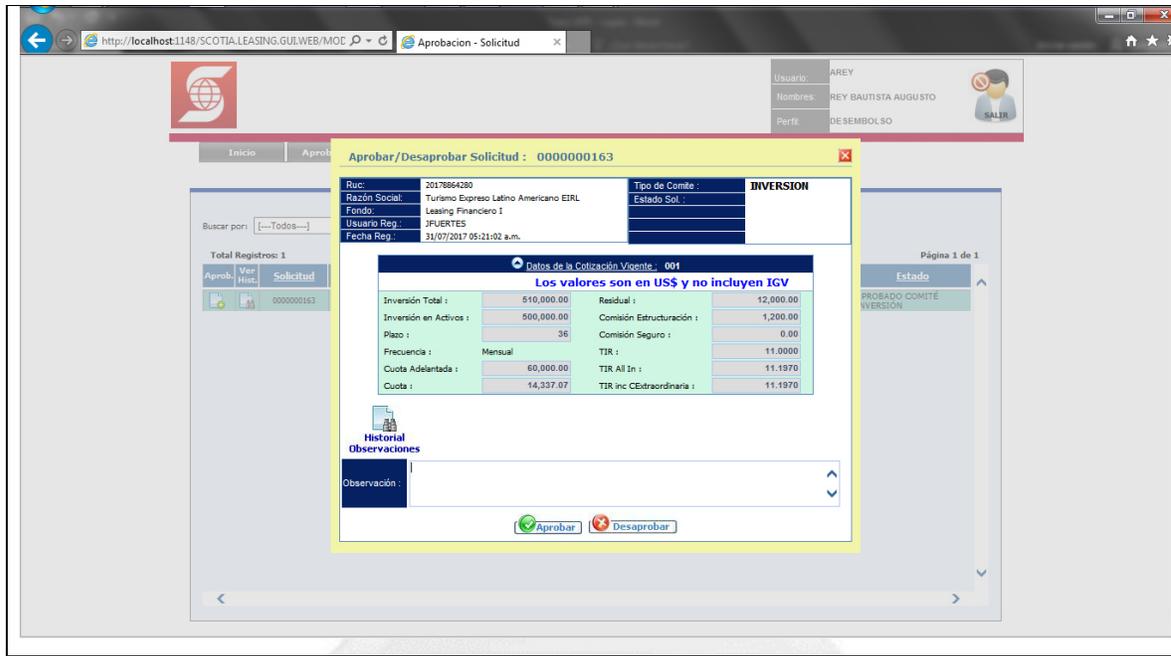


Figura 5.39: Sub flujo - Realizar aprobación final. (Fuente: Elaboración propia)

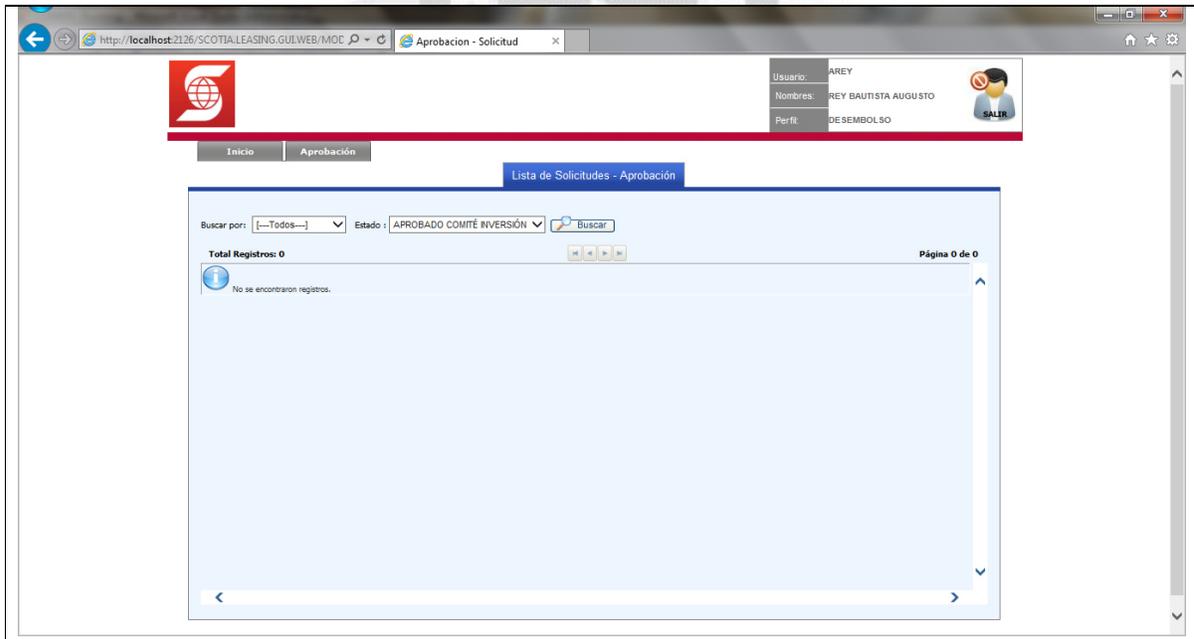


Figura 5.40: Flujo alternativo - No se encontraron registros. (Fuente: Elaboración propia)

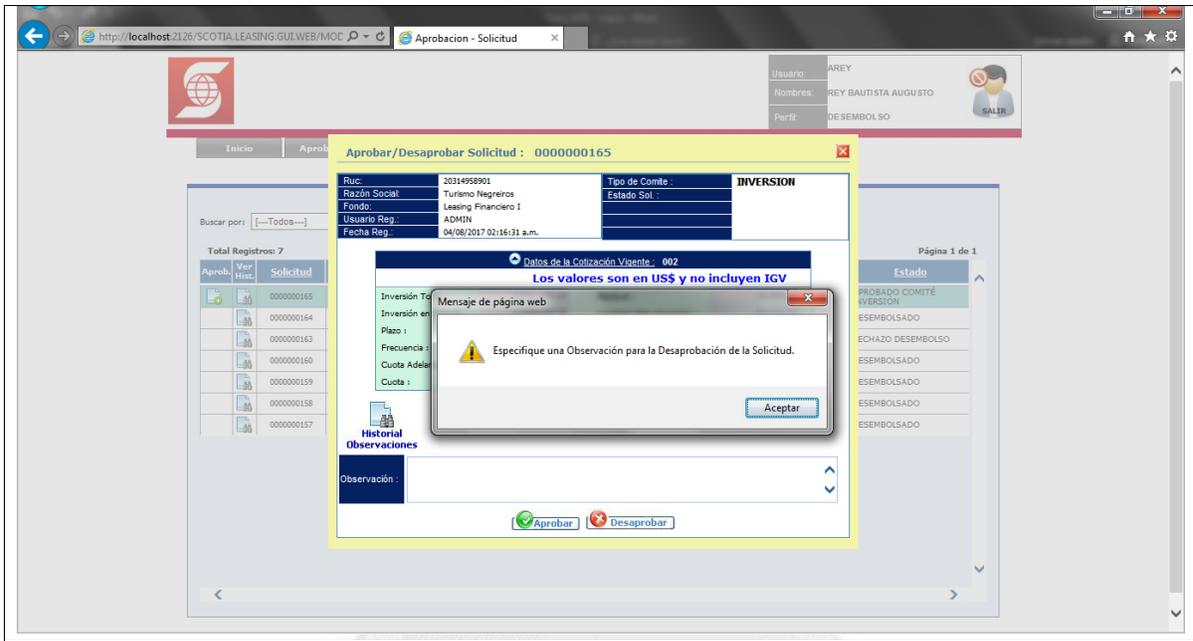


Figura 5.41: Flujo alternativo - Especificar observación de desaprobación. (Fuente: Elaboración propia)



h. Especificación de caso de uso 08 - Buscar Cliente

1. Breve Descripción

Este caso de uso permite a los **usuarios asesor área comercial** y **administrador** buscar clientes al momento de crear una nueva solicitud.

2. Flujo de Eventos

Evento Disparador: El caso de uso comienza cuando el **usuario** selecciona la opción **nuevo** en la interfaz principal del sistema.

2.1 Flujo Básico <<Buscar cliente>>

2.1.1 El sistema muestra la interfaz **búsqueda de clientes**. La interfaz muestra la lista de clientes del banco.

La interfaz contiene los datos: **Razón social, RUC**.

Incluye las opciones: **Nuevo** y **buscar**.

2.1.2 El **usuario** digita los datos (RUC o razón social) para realizar la búsqueda y luego hace clic en la opción **buscar**.

2.1.3 El sistema muestra la empresa de acuerdo a los datos filtrados previamente.

2.1.4 Fin del caso de uso.

3. Flujos Alternativos

3.1 << No se encontraron registros >>

En el punto 2.1.2 del flujo básico **buscar cliente**, en caso no existe el cliente, el sistema muestra el mensaje **no se encontraron registros**, el flujo continúa en el punto 2.1.2.

4. Requerimientos Especiales

4.1 El caso de uso debe estar disponible a través del **navegador internet explorer**, previo logueo del **usuario**.

5. Pre Condiciones

5.1 El **usuario** se haya logueado en el sistema.

5.2 Los clientes estén disponibles en el sistema.

6. Post Condiciones

6.1 El cliente ha sido buscado en el sistema.

7. Puntos de Extensión

Ninguno.

8. Prototipo (GUI)

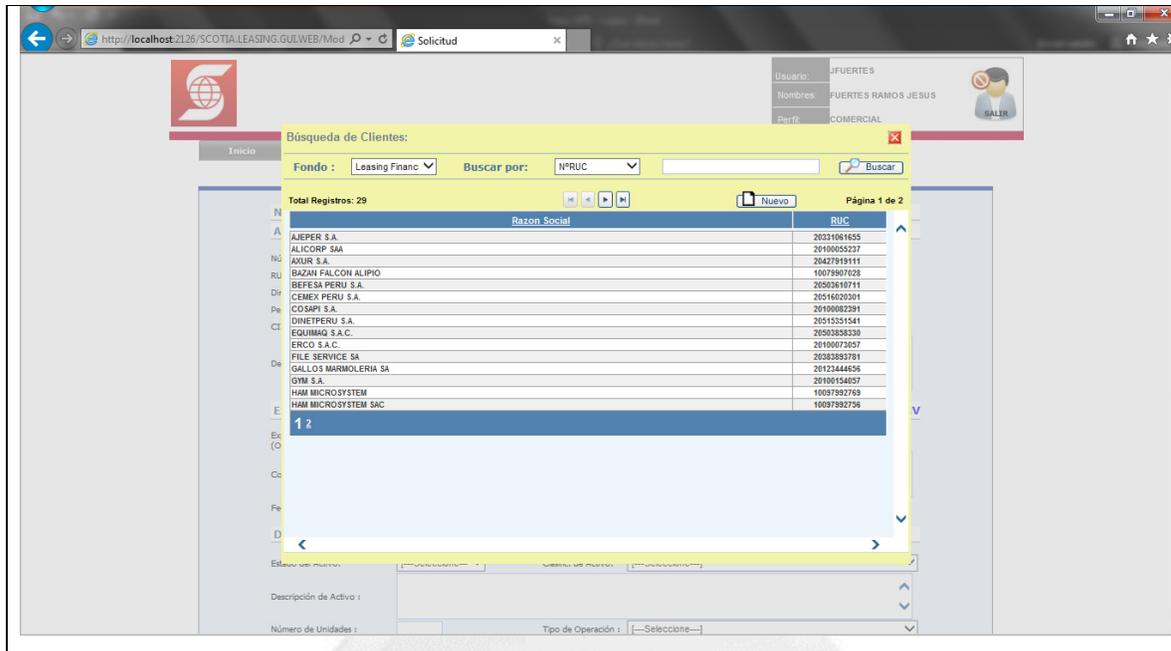


Figura 5.42: Flujo básico - Buscar cliente. (Fuente: Elaboración propia)

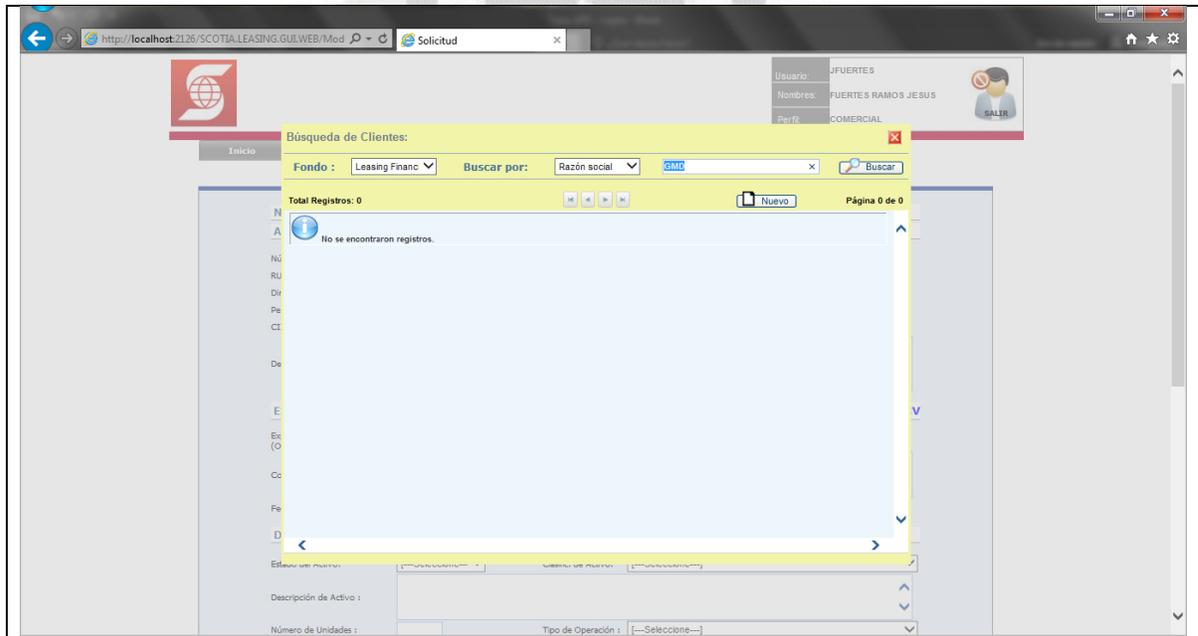


Figura 5.43: Flujo alternativo - No se encontraron registros. (Fuente: Elaboración propia)

i. Especificación de caso de uso 09 - Buscar solicitud

1. Breve Descripción

Este caso de uso permite a **todos los usuarios** del sistema buscar solicitudes con sus estados actuales respectivos.

2. Flujo de Eventos

Evento Disparador: El caso de uso comienza cuando el **usuario** hace clic en el menú **inicio**, opción **lista solicitud** en la interfaz principal del sistema.

2.1 Flujo Básico <<Buscar solicitud>>

2.1.1 El sistema muestra la interfaz **lista de solicitudes**. La interfaz muestra la lista de solicitudes registradas.

La interfaz contiene los datos: **Ver., solicitud, contrato, cliente, fondo, tipo operación, fecha solicitud y estado**.

Incluye las opciones: **Buscar y ver registro**.

2.1.2 El **usuario** hace clic en la lista desplegable del campo **estado**, selecciona el estado respectivo y luego hace clic en la opción **buscar**.

2.1.3 El sistema muestra la solicitud de acuerdo a los datos filtrados previamente.

2.1.4 Fin del caso de uso.

3. Flujos Alternativos

3.1 <<No se encontraron registros>>

En el punto 2.1.2 del flujo básico **buscar solicitud**, en caso no existe la solicitud con el estado escogido, el sistema muestra el mensaje **no se encontraron registros**, el flujo continúa en el punto 2.1.2.

4. Requerimientos Especiales

4.1 El caso de uso debe estar disponible a través del **navegador internet explorer**, previo logueo del **usuario**.

5. Pre Condiciones

5.1 El **usuario** se haya logueado en el sistema.

5.2 Las solicitudes estén disponibles en el sistema.

6. Post Condiciones

6.1 La solicitud ha sido buscada en el sistema.

7. Puntos de Extensión

Ninguno.

8. Prototipo (GUI)

Usuario: PILLEGAS
 Nombres: VILLEGAS ARCE PEDRO
 Perfil: COMITE INVERSION 2

Inicio | Comité

Lista de Solicitudes

Buscar por: [---Todos---] Estado: [---Todos---] Buscar

Total Registros: 9

Ver	Solicitud	Contrato		Fondo	Tipo Operación	Fecha Solicitud	Estado
	0000000165	L-00540	TURISMO	L-I	Nueva	04/08/2017	DESEMBOLSADO
	0000000164	L-00539	ALAS PERU	L-I	Nueva	03/08/2017	DESEMBOLSADO
	0000000163	L-00538	TURISMO	L-I	Nueva	31/07/2017	RECHAZO DESEMBOLSO
	0000000162	L-00537	GYM S.A.	L-I	Nueva	26/07/2017	DESAPROBADO COMITÉ INVERSIÓN
	0000000161	L-00536	ZICSA	L-I	Nueva	23/07/2017	COTIZACIÓN APROBADA
	0000000160	L-00532	BEFESA	L-I	Nueva	15/07/2017	DESEMBOLSADO
	0000000159	L-00530	COSAPI S.A.	L-I	Nueva	14/07/2017	DESEMBOLSADO
	0000000158	L-00529	COSAPI S.A.	L-I	Nueva	23/06/2017	DESEMBOLSADO
	0000000157	L-00528	ALICORP SAA	L-I	Nueva	21/06/2017	DESEMBOLSADO

Página 1 de 1

Figura 5.44: Flujo básico - Buscar solicitud. (Fuente: Elaboración propia)

Usuario: AREY
 Nombres: REY BAUTISTA AUGUSTO
 Perfil: DESEMBOLSO

Inicio | Aprobación

Lista de Solicitudes

Buscar por: [---Todos---] Estado: REGISTRADO Buscar

Total Registros: 0

No se encontraron registros.

Página 0 de 0

Figura 5.45: Flujo alternativo - No se encontraron registros. (Fuente: Elaboración propia)

j. Especificación de caso de uso 10 - Exportar solicitud

1. Breve Descripción

Este caso de uso permite a **todos los usuarios** del sistema exportar las solicitudes creadas.

2. Flujo de Eventos

Evento Disparador: El caso de uso **exportar solicitud** puede ser llamado por el caso de uso **gestionar solicitud**.

2.1 Flujo Básico <<Exportar solicitud>>

2.1.1 El sistema muestra la interfaz **lista de solicitudes**. La interfaz muestra la lista de solicitudes registradas.

La interfaz contiene los datos: **Ver., solicitud, contrato, cliente, fondo, tipo operación, fecha solicitud y estado**.

Incluye las opciones: **Buscar y ver registro**.

2.1.2 El **usuario** hace clic en la lista desplegable del campo **estado**, selecciona el estado respectivo y luego hace clic en la opción **buscar**.

2.1.3 El sistema muestra la solicitud de acuerdo a los datos filtrados previamente.

2.1.4 Fin del caso de uso.

2.1.5 El **usuario** hace clic en la opción **ver registro**.

2.1.6 El sistema muestra la interfaz **registro de solicitud**, donde se muestran los datos de la solicitud

Incluye las opciones: **Cancelar y exportar solicitud**.

2.1.7 Si el **usuario** selecciona **exportar solicitud**

2.1.7.1 El sistema exporta el archivo **SICA.pdf** con los datos de la solicitud.

2.1.8 Si el **usuario** selecciona **cancelar**

2.1.8.1 El sistema redirige a la interfaz **lista de solicitudes**.

3. Flujos Alternativos

3.1 <<No se encontraron registros>>

En el punto 2.1.2 del flujo básico **exportar solicitud**, en caso no existe la solicitud con el estado escogido, el sistema muestra el mensaje **no se encontraron registros**, el flujo continúa en el punto 2.1.2.

4. Requerimientos Especiales

4.1 El caso de uso debe estar disponible a través del **navegador internet explorer**, previo logueo del **usuario**.

5. Pre Condiciones

5.1 El **usuario** se haya logueado en el sistema.

5.2 Las solicitudes estén disponibles en el sistema.

6. Post Condiciones

6.1 La solicitud ha sido exportada del sistema.

7. Puntos de Extensión

Ninguno.

8. Prototipo (GUI)

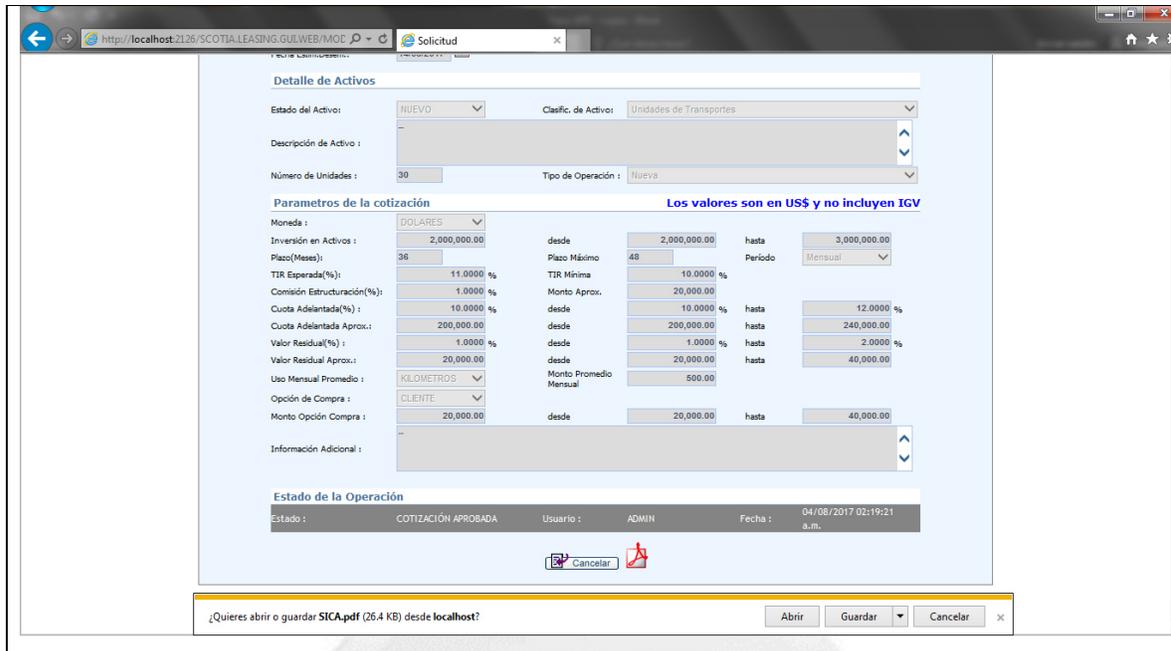


Figura 5.46: Flujo básico - Exportar solicitud. (Fuente: Elaboración propia)

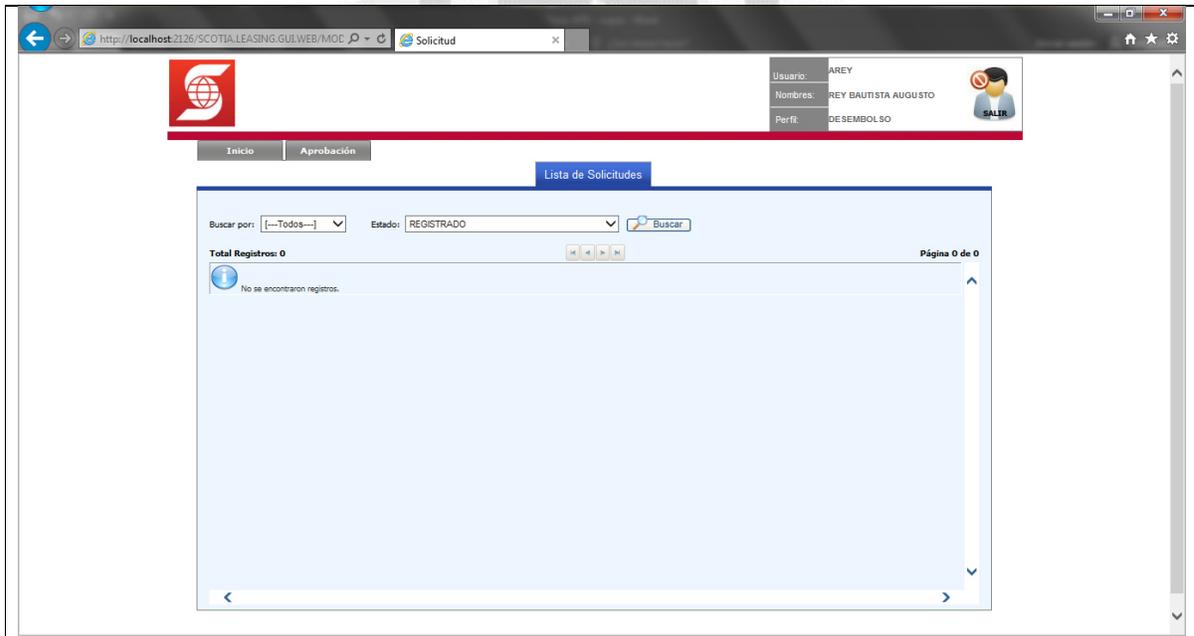


Figura 5.47: Flujo alternativo - No se encontraron registros. (Fuente: Elaboración propia)

k. Especificación de caso de uso 11 - Exportar cotización

1. Breve Descripción

Este caso de uso permite a **todos los usuarios** del sistema exportar las solicitudes creadas.

2. Flujo de Eventos

Evento Disparador: El caso de uso **exportar cotización** puede ser llamado por el caso de uso **gestionar cotización**.

2.1 Flujo Básico <<Exportar cotización>>

2.1.1 El sistema muestra la interfaz **lista de solicitudes**. La interfaz muestra la lista de solicitudes registradas.

La interfaz contiene los datos: **Ver., solicitud, contrato, cliente, fondo, tipo operación, fecha solicitud y estado**.

Incluye las opciones: **Buscar y ver registro**.

2.1.2 El **usuario** hace clic en la lista desplegable del campo **estado**, selecciona el estado respectivo y luego hace clic en la opción **buscar**.

2.1.3 El sistema muestra la solicitud de acuerdo a los datos filtrados previamente.

2.1.4 Fin del caso de uso.

2.1.5 El **usuario** hace clic en la opción **ver registro**.

2.1.6 El sistema te dirige a la pestaña **registro de solicitud**, hacemos clic en la pestaña **lista de cotizaciones**, donde se muestra la cotización relacionada a la solicitud

Incluye las opciones: **Ver registro y cancelar**.

2.1.7 El **usuario** hace clic en la opción **ver registro**

2.1.8 El sistema muestra la interfaz **ver cotización**, donde se muestran los datos de la misma.

Incluye las opciones: **Ver cronograma y cerrar**.

2.1.9 Si el **usuario** selecciona **ver cronograma**

2.1.9.1 El sistema exporta el archivo **Cronograma.pdf** con los datos del cronograma de pago de la cotización.

2.1.10 Si el **usuario** selecciona **cancelar**

2.1.10.1 El sistema redirige a la interfaz **lista de cotizaciones**.

3. Flujos Alternativos

3.1 <<No se encontraron registros>>

En el punto 2.1.2 del flujo básico **buscar solicitud**, en caso no existe la solicitud con el estado escogido, el sistema muestra el mensaje **no se encontraron registros**, el flujo continúa en el punto 2.1.2.

4. Requerimientos Especiales

4.1 El caso de uso debe estar disponible a través del **navegador internet explorer**, previo logueo del **usuario**.

5. Pre Condiciones

5.1 El **usuario** se haya logueado en el sistema.

5.2 Las cotizaciones estén disponibles en el sistema.

6. Post Condiciones

6.1 La cotización ha sido exportada del sistema.

7. Puntos de Extensión

Ninguno.

8. Prototipo (GUI)

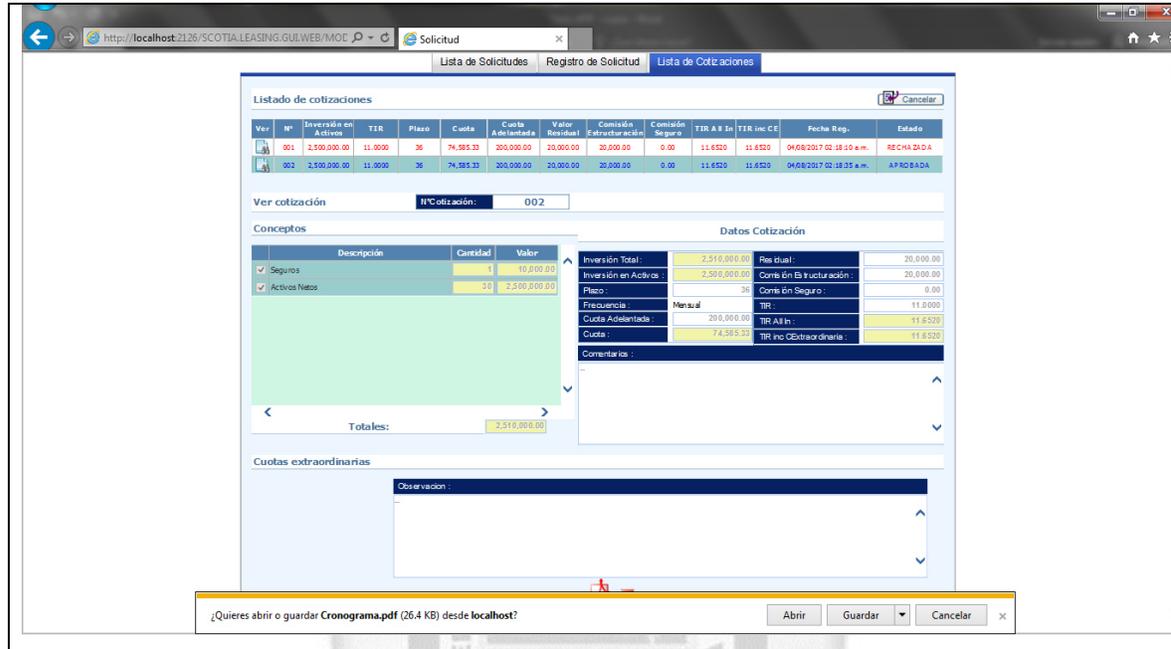


Figura 5.48: Flujo básico - Exportar cotización. (Fuente: Elaboración propia)

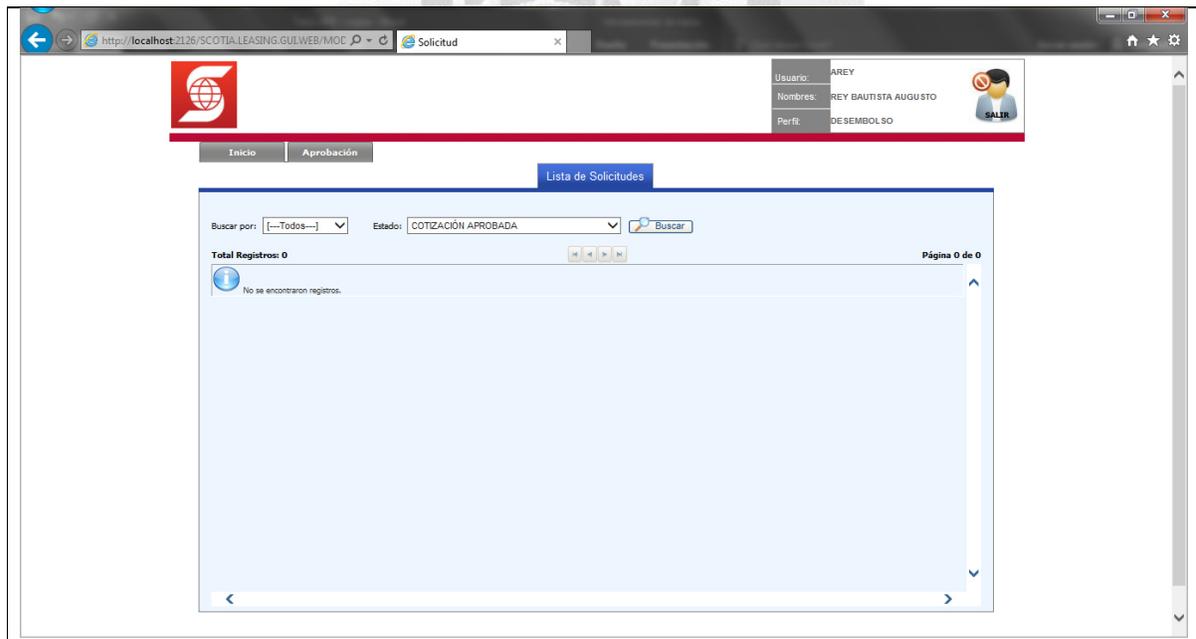


Figura 5.49: Flujo alternativo - No se encontraron registros. (Fuente: Elaboración propia)

1. Especificación de caso de uso 12 - Gestionar usuarios

1. Breve Descripción

Este caso de uso permite al **usuario administrador** agregar, modificar y eliminar usuarios en el sistema.

2. Flujo de Eventos

Evento Disparador: El caso de uso comienza cuando el **usuario administrador** hace clic en el menú **seguridad**, opción **usuarios** en la interfaz principal del sistema.

2.1 Flujo Básico <<Gestionar usuario>>

2.1.1 El sistema muestra la interfaz **lista de usuarios**. La interfaz muestra la lista de usuarios del sistema. La interfaz contiene los datos: **Ver, edit., elim., comité, código, usuario, apellidos y nombres, correo perfil, tipo de usuario, responsable documentos, usuario creación, fecha creación, usuario actualización, fecha actualización** y estado.

Incluye las opciones: **Nuevo, buscar, exportar, ver registro, editar registro, eliminar registro** y **dar permisos**.

2.1.2 El **usuario administrador** digita los datos para realizar la búsqueda y luego hace clic en la opción **buscar**.

2.1.3 El sistema muestra el usuario de acuerdo a los datos filtrados previamente.

2.1.4 Fin del caso de uso.

2.2 Sub Flujo <<Nuevo usuario>>

2.2.1 El **usuario administrador** hace clic en la opción **nuevo**.

2.2.2 El sistema muestra la interfaz **datos del usuario**, donde activan los campos para ingresar los datos del usuario: **Usuario, clave, apellido paterno, apellido materno, nombres, correo, perfil, tipo de usuario** y estado.

Incluye las opciones: **Guardar, cancelar** y **reset password**.

2.2.3 El **usuario administrador** digita los datos del usuario del sistema sobre los campos activos.

2.2.4 Si el **usuario administrador** selecciona **guardar**

2.2.4.1 El sistema actualiza la lista de usuarios del sistema en la interfaz.

2.2.4.2 Muestra el mensaje **el registro se ha creado correctamente** en la interfaz **lista de usuarios**.

2.2.5 Si el **usuario administrador** selecciona **cerrar**

2.2.5.1 El sistema redirige a la interfaz **lista de usuarios**.

2.3 Sub Flujo <<Modificar usuario>>

2.3.1 El **usuario administrador** hace clic en la opción **editar registro**.

2.3.2 El sistema muestra la interfaz **datos del usuario**, mostrando los datos del usuario a ser modificados: **Apellido paterno, apellido materno, nombres, correo, perfil, tipo de usuario** y estado.

Incluye las opciones: **Guardar** y **cancelar**.

2.3.3 Si el **usuario administrador** selecciona **guardar**

2.3.3.1 El sistema guarda los datos del usuario.

2.3.3.2 Cierra la interfaz **datos del usuario**.

2.3.3.3 Muestra el mensaje **el registro se ha modificado correctamente** en la interfaz **lista de usuarios**.

- 2.3.4 Si el **usuario administrador** selecciona **cancelar**
- 2.3.4.1 El sistema redirige a la interfaz **lista de usuarios**.

2.4 Sub Flujo <<Eliminar usuario>>

- 2.4.1 El **usuario administrador** hace clic en la opción **eliminar registro**.
- 2.4.2 El sistema muestra el mensaje **¿desea eliminar el registro?**
Incluye las opciones: **Aceptar** y **cancelar**.
- 2.4.3 Si el **usuario administrador** selecciona **aceptar**
- 2.4.3.1 El sistema actualiza la lista de usuarios.
- 2.4.3.2 Muestra el mensaje **el registro se ha eliminado correctamente** en la interfaz **lista de solicitudes**.
- 2.4.4 Si el **usuario administrador** selecciona **cancelar**
- 2.4.4.1 El sistema redirige a la interfaz **lista de solicitudes**.

3. Flujos Alternativos

3.1 << Ingresar el correo del usuario>>

En el punto 2.2.3 del sub flujo **nuevo usuario**, si el **usuario administrador** no digita el correo, al momento de seleccionar **guardar**, el sistema muestra el mensaje **debe ingresar el correo del usuario**, el flujo continúa en el punto 2.2.3.

4. Requerimientos Especiales

- 4.1 El caso de uso debe estar disponible a través del **navegador internet explorer**, previo logueo del **usuario administrador**.

5. Pre Condiciones

- 5.1 El **usuario administrador** se haya logueado en el sistema.
- 5.2 Los usuarios estén disponibles en el sistema.

6. Post Condiciones

- 6.1 El usuario queda registrado en la base de datos.

7. Puntos de Extensión

Ninguno.

8. Prototipo (GUI)

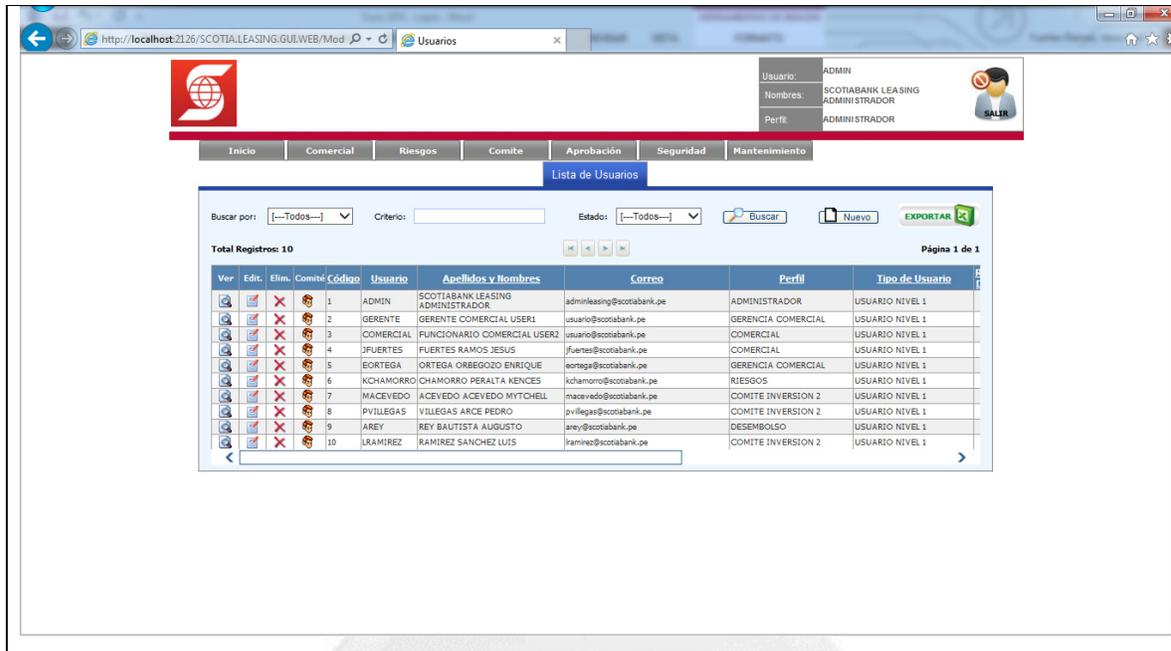


Figura 5.50: Flujo básico - Gestionar usuario. (Fuente: Elaboración propia)

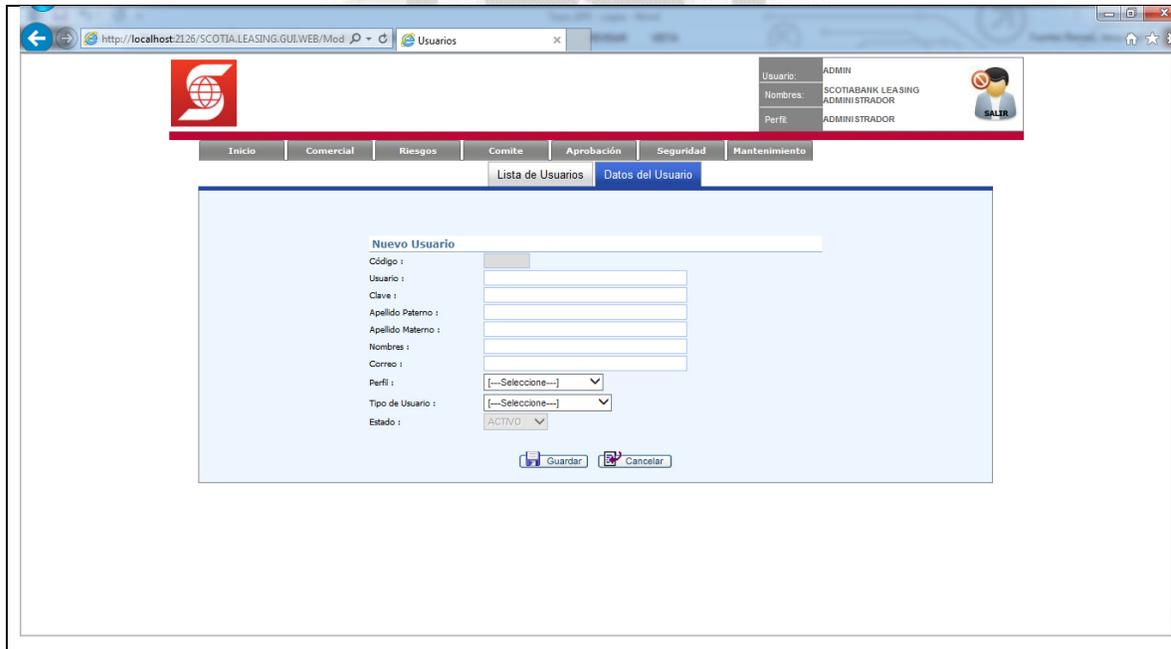


Figura 5.51: Sub flujo - Nuevo usuario. (Fuente: Elaboración propia)

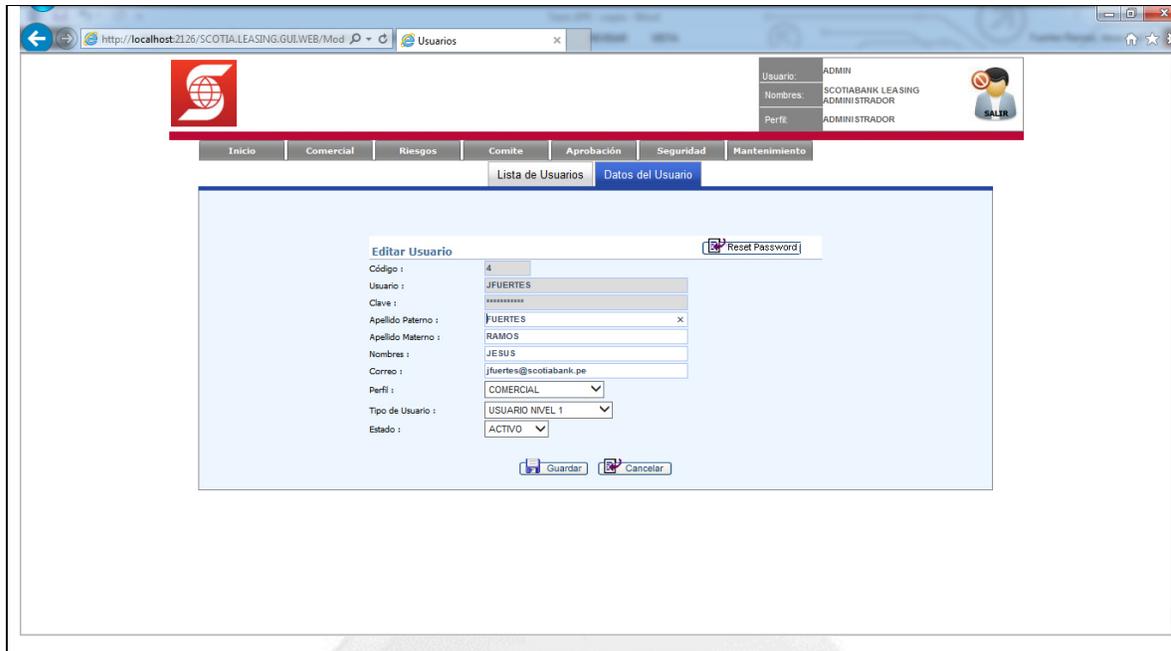


Figura 5.52: Sub flujo - Modificar usuario. (Fuente: Elaboración propia)

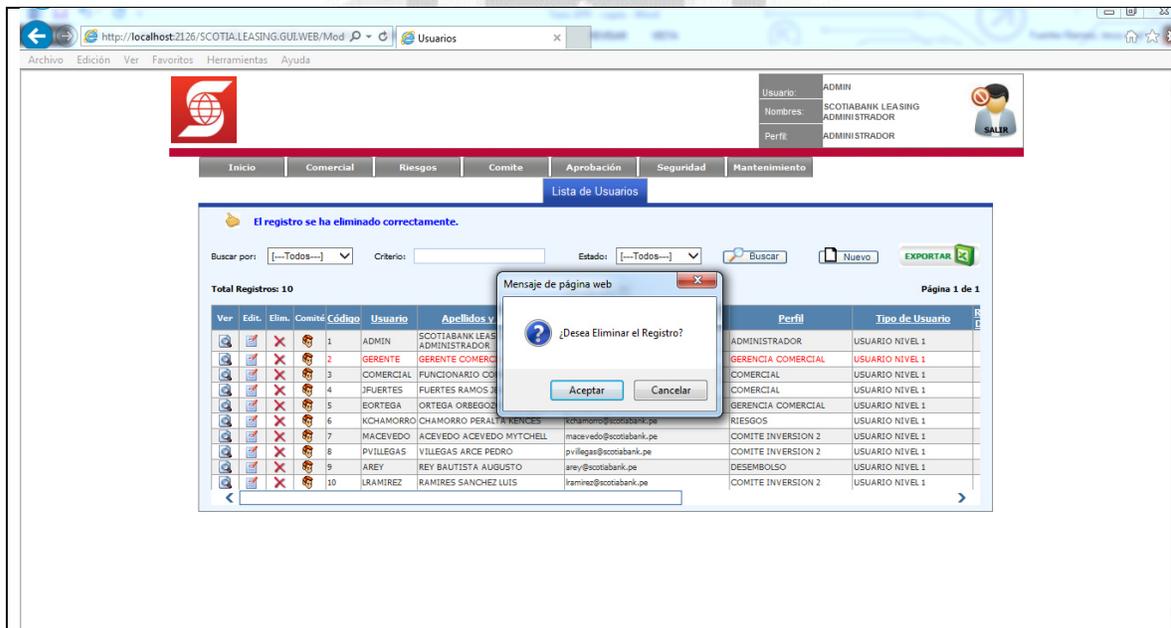


Figura 5.53: Sub flujo - Eliminar usuario. (Fuente: Elaboración propia)

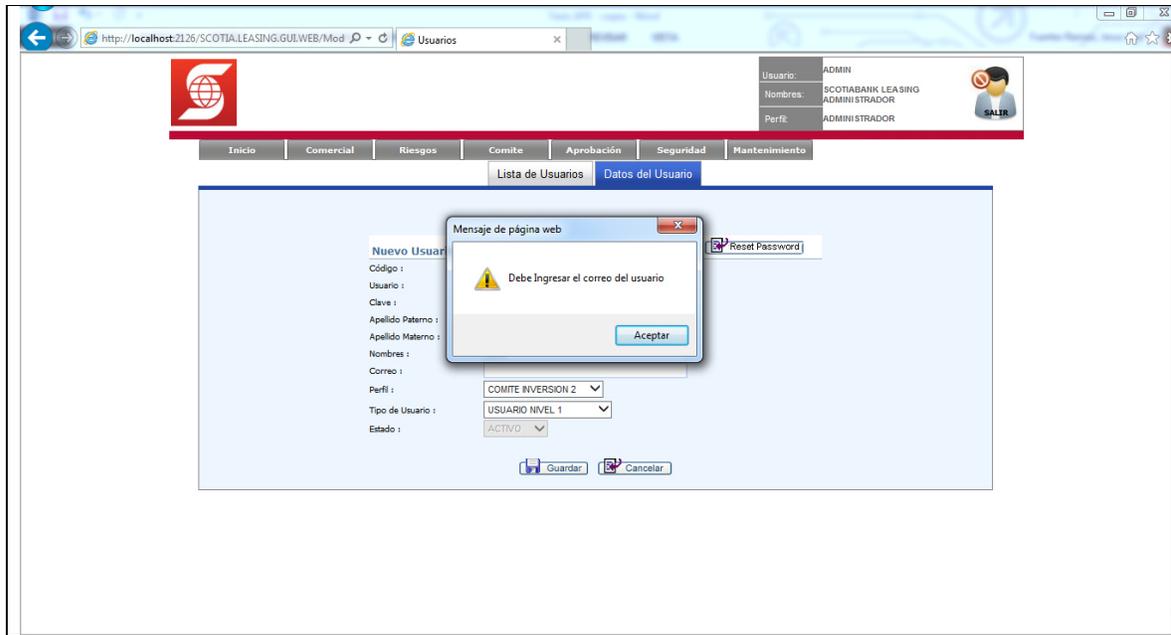


Figura 5.54: Flujo alternativo - Ingresar el correo del usuario. (Fuente: Elaboración propia)



m. Especificación de caso de uso 13 - Gestionar perfiles de usuario

1. Breve Descripción

Este caso de uso permite al **usuario administrador** agregar, modificar y eliminar perfiles en el sistema.

2. Flujo de Eventos

Evento Disparador: El caso de uso comienza cuando el **usuario administrador** hace clic en el menú **seguridad**, opción **perfil** en la interfaz principal del sistema.

2.1 Flujo Básico <<Nuevo perfil>>

2.1.1 El sistema muestra la interfaz **lista de usuarios**. La interfaz muestra la lista de usuarios del sistema. La interfaz contiene los datos: **Ver, edit., elim., código, descripción, tipo perfil, usuario creación, fecha creación, usuario actualización, fecha actualización y estado**.

Incluye las opciones: **Nuevo, buscar, exportar, ver registro, editar registro y eliminar registro**.

2.1.2 El **usuario administrador** hace clic en la opción **nuevo**.

2.1.3 El sistema muestra la interfaz **datos del perfil**, donde activan el campo para ingresar los datos del perfil: **Descripción**.

Incluye las opciones: **Guardar y cancelar**.

2.1.4 El **usuario administrador** digita los datos del perfil sobre el campo activo.

2.1.5 Si el **usuario administrador** selecciona **guardar**

2.1.5.1 El sistema actualiza la lista de perfiles en la interfaz.

2.1.5.2 Muestra el mensaje **el registro se ha creado correctamente** en la interfaz **lista de perfiles**.

2.1.6 Si el **usuario administrador** selecciona **cerrar**

2.1.6.1 El sistema redirige a la interfaz **lista de perfiles**.

2.2 Sub Flujo <<Modificar perfil>>

2.2.1 El **usuario administrador** hace clic en la opción **editar registro**.

2.2.2 El sistema muestra la interfaz **datos de perfil**, mostrando los datos del perfil a ser modificado: **Descripción**.

Incluye las opciones: **Guardar y cancelar**.

2.2.3 Si el **usuario administrador** selecciona **guardar**

2.2.3.1 El sistema guarda los datos del perfil.

2.2.3.2 Cierra la interfaz **datos de perfil**.

2.2.3.3 Muestra el mensaje **el registro se ha modificado correctamente** en la interfaz **lista de perfiles**.

2.2.4 Si el **usuario administrador** selecciona **cancelar**

2.2.4.1 El sistema redirige a la interfaz **lista de usuarios**.

2.3 Sub Flujo <<Eliminar perfil>>

2.3.1 El **usuario administrador** hace clic en la opción **eliminar registro**.

2.3.2 El sistema muestra el mensaje **¿desea eliminar el registro?**

Incluye las opciones: **Aceptar y cancelar**.

2.3.3 Si el **usuario administrador** selecciona **aceptar**

2.3.3.1 El sistema actualiza la lista de perfiles.

2.3.3.2 Muestra el mensaje **el registro se ha eliminado correctamente** en la interfaz **lista de perfiles**.

2.3.4 Si el **usuario administrador** selecciona **cancelar**

2.3.4.1 El sistema redirige a la interfaz **lista de perfiles**.

3. Flujos Alternativos

3.1 << Ingresar el nombre del perfil>>

En el punto 2.2.3 del sub flujo **nuevo perfil**, si el **usuario administrador** no digita el nombre del perfil, al momento de seleccionar **guardar**, el sistema muestra el mensaje **debe ingresar el nombre del perfil**, el flujo continúa en el punto 2.2.3.

4. Requerimientos Especiales

4.1 El caso de uso debe estar disponible a través del **navegador internet explorer**, previo logueo del **usuario administrador**.

5. Pre Condiciones

5.1 El **usuario administrador** se haya logueado en el sistema.

5.2 Los perfiles estén disponibles en el sistema.

6. Post Condiciones

6.1 El perfil queda registrado en la base de datos.

7. Puntos de Extensión

Ninguno.

8. Prototipo (GUI)

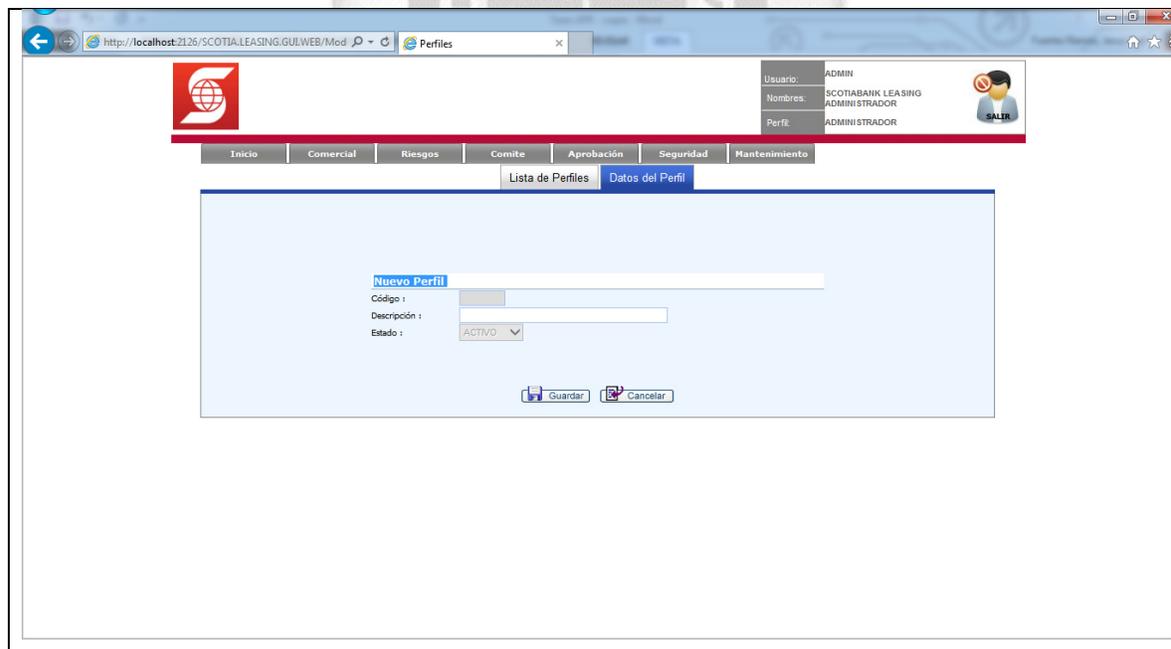


Figura 5.55: Flujo básico - Nuevo perfil. (Fuente: Elaboración propia)

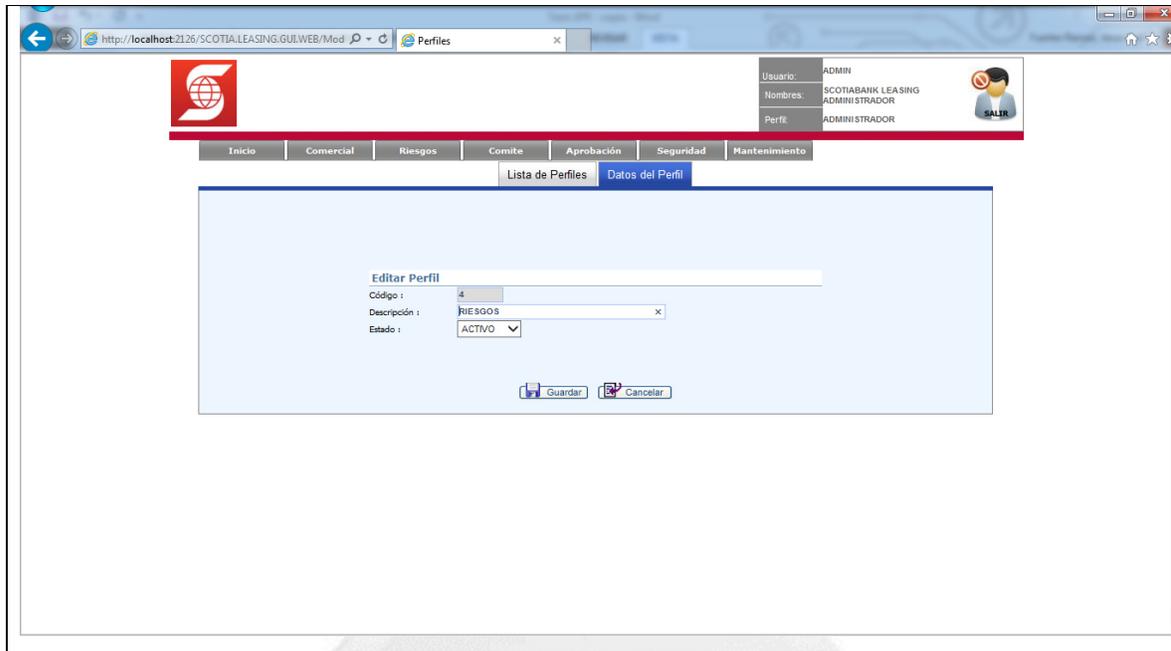


Figura 5.56: Sub flujo - Modificar perfil. (Fuente: Elaboración propia)

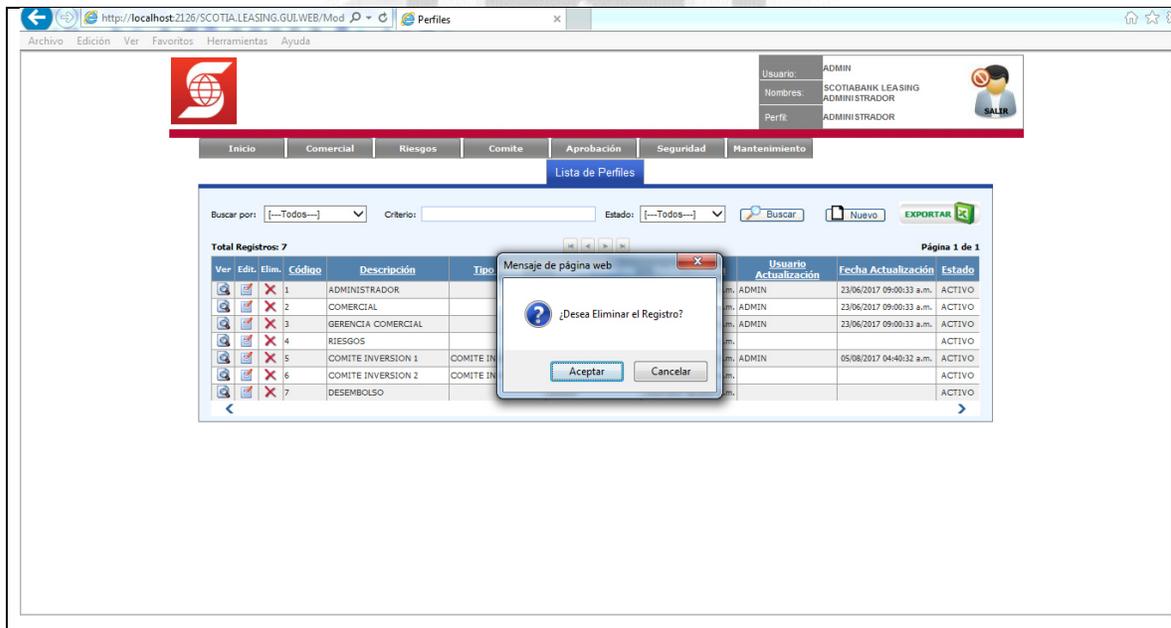


Figura 5.57: Sub flujo - Eliminar perfil. (Fuente: Elaboración propia)

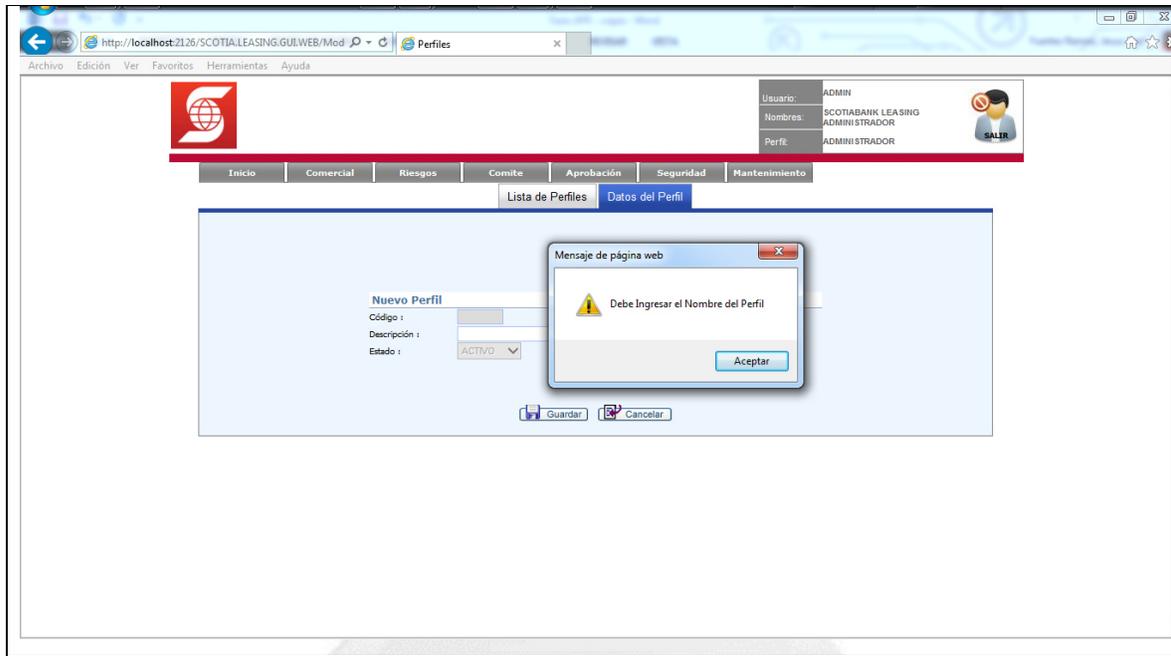


Figura 5.58: Flujo alternativo - Ingresar el nombre del perfil. (Fuente: Elaboración propia)



n. Especificación de caso de uso 14 - Gestionar accesos por perfiles

1. Breve Descripción

Este caso de uso permite al **usuario administrador** agregar, modificar y eliminar accesos de los perfiles de usuarios en los módulos del sistema.

2. Flujo de Eventos

Evento Disparador: El caso de uso comienza cuando el **usuario administrador** hace clic en el menú **seguridad**, opción **accesos** en la interfaz principal del sistema.

2.1 Flujo Básico <<Gestionar accesos>>

2.1.1 El sistema muestra la interfaz **opciones del menú**.

La interfaz contiene tres listas desplegables: **Módulo**, **venta** y **perfil**, donde figuran los datos de estos ítems del sistema.

Incluye las opciones: **Guardar cancelar** y **exportar**.

2.1.2 El **usuario administrador** selecciona cualquier ítem de las 3 listas desplegables.

2.1.3 El sistema muestra los datos **alias** y **descripción** que indican los accesos en el sistema a los cuales se les va brindar acceso, los cuales se activan/desactivan mediante una casilla de verificación

2.1.4 El **usuario administrador** activa o desactiva las casillas de verificación de acuerdo al perfil del usuario por área.

2.1.5 Si el **usuario administrador** selecciona **guardar**

2.1.5.1 El sistema actualiza la lista de perfiles en la interfaz.

2.1.5.2 Muestra el mensaje **el registro se ha modificado correctamente** en la interfaz **opciones del menú**.

2.1.6 Si el **usuario administrador** selecciona **cancelar**

2.1.6.1 El sistema limpia los datos de las listas desplegables.

3. Flujos Alternativos

3.1 <<No se encontraron registros>>

En el punto 2.1.2 del flujo básico **gestionar accesos**, si el usuario no elige ningún ítem de la lista desplegable **ventana**, el sistema muestra el mensaje **no se encontraron registros**, el flujo continúa en el punto 2.1.2.

4. Requerimientos Especiales

4.1 El caso de uso debe estar disponible a través del **navegador internet explorer**, previo logueo del **usuario administrador**.

5. Pre Condiciones

5.1 El **usuario administrador** se haya logueado en el sistema.

6. Post Condiciones

6.1 El acceso por perfiles queda registrado en la base de datos.

7. Puntos de Extensión

Ninguno.

8. Prototipo (GUI)

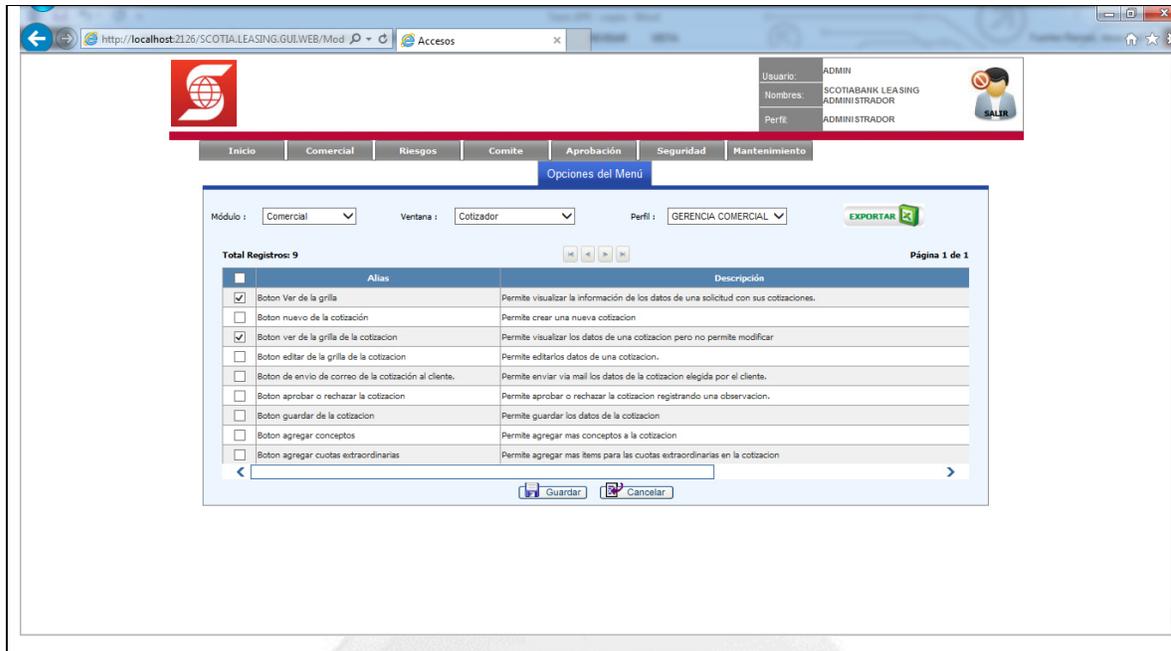


Figura 5.59: Flujo básico - Gestionar accesos. (Fuente: Elaboración propia)

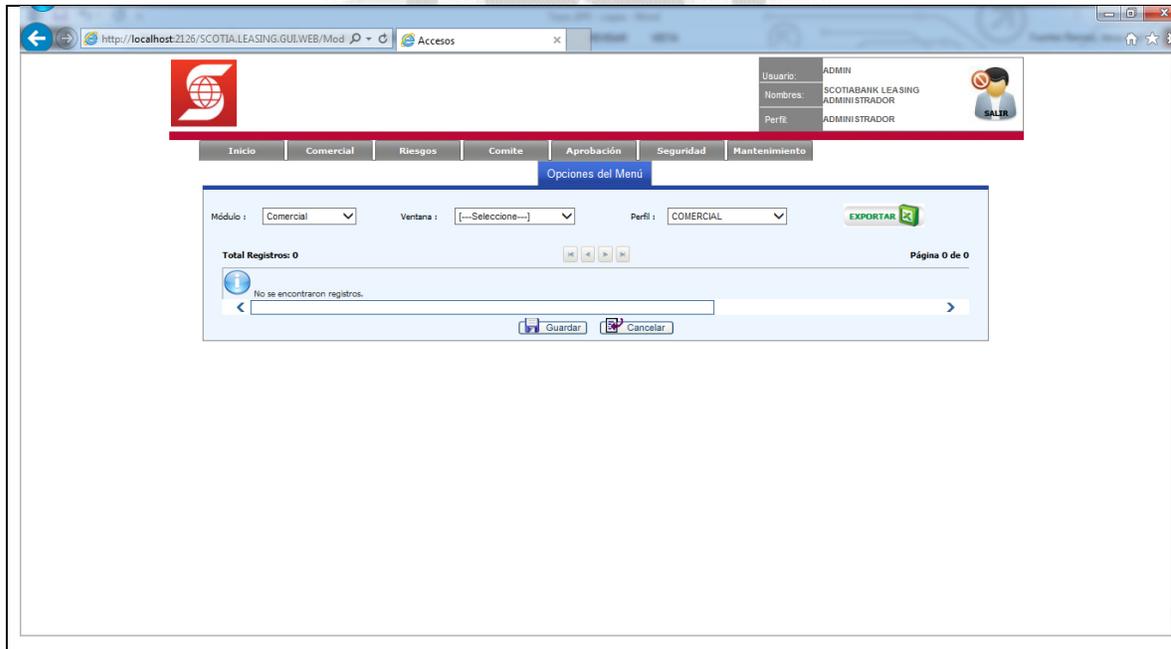


Figura 5.60: Flujo alternativo - No se encontraron registros. (Fuente: Elaboración propia)

o. Especificación de caso de uso 15 - Ingresar al sistema

1. Breve Descripción

Este caso de uso permite a **todos los usuarios** ingresar al sistema.

2. Flujo de Eventos

Evento Disparador: El caso de uso comienza cuando el **usuario** ingresa a la ruta web del sistema.

2.1 Flujo Básico <<Ingresar al sistema >>

2.1.1 El sistema muestra la interfaz de ingreso.

La interfaz contiene los campos activos para completar los datos: **Usuario** y **contraseña**.

Incluye las opciones: **Aceptar**, **salir** y **cambiar contraseña**.

2.1.2 El **usuario** digita la información en los campos activos.

2.1.3 Si el **usuario** selecciona **aceptar**

2.1.3.1 Ingresa a la interfaz principal del sistema.

2.1.4 Si el **usuario** selecciona **cambiar contraseña**

2.1.4.1 El sistema extiende el caso de uso **restablecer contraseña**.

2.1.5 Si el **usuario asesor área comercial** selecciona **salir**

2.1.5.1 Se cierra la ventana del navegador.

3. Flujos Alternativos

3.1 <<Verificar si usuario y contraseña son correctos>>

En el punto 2.1.2 del flujo básico **ingresar al sistema**, si el usuario digita un perfil o contraseña incorrecta, el sistema muestra el mensaje **favor, verificar si usuario y contraseña son correctos**, el flujo continúa en el punto 2.1.2.

3.1 <<Especificar su contraseña>>

En el punto 2.1.2 del flujo básico **ingresar al sistema**, si el usuario no digita una contraseña, el sistema muestra el mensaje **por favor debe especificar su contraseña**, el flujo continúa en el punto 2.1.2.

4. Requerimientos Especiales

4.1 El caso de uso debe estar disponible a través del **navegador internet explorer**.

5. Pre Condiciones

5.1 El **usuario** debe estar registrado en el sistema.

6. Post Condiciones

6.1 El **usuario** ha iniciado sesión en el sistema.

7. Puntos de Extensión

7.1 Caso de uso **restablecer contraseña** en el punto 2.1.4.1 del flujo básico **ingresar al sistema**.

8. Prototipo (GUI)

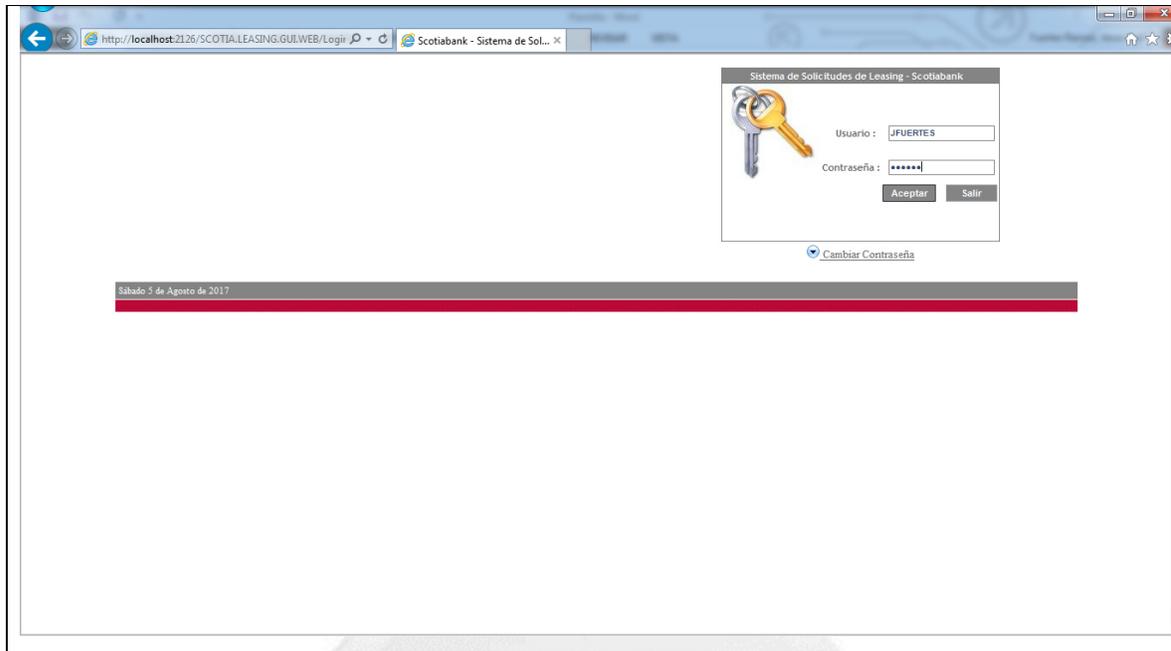


Figura 5.63: Flujo básico: Ingresar al sistema. (Fuente: Elaboración propia)

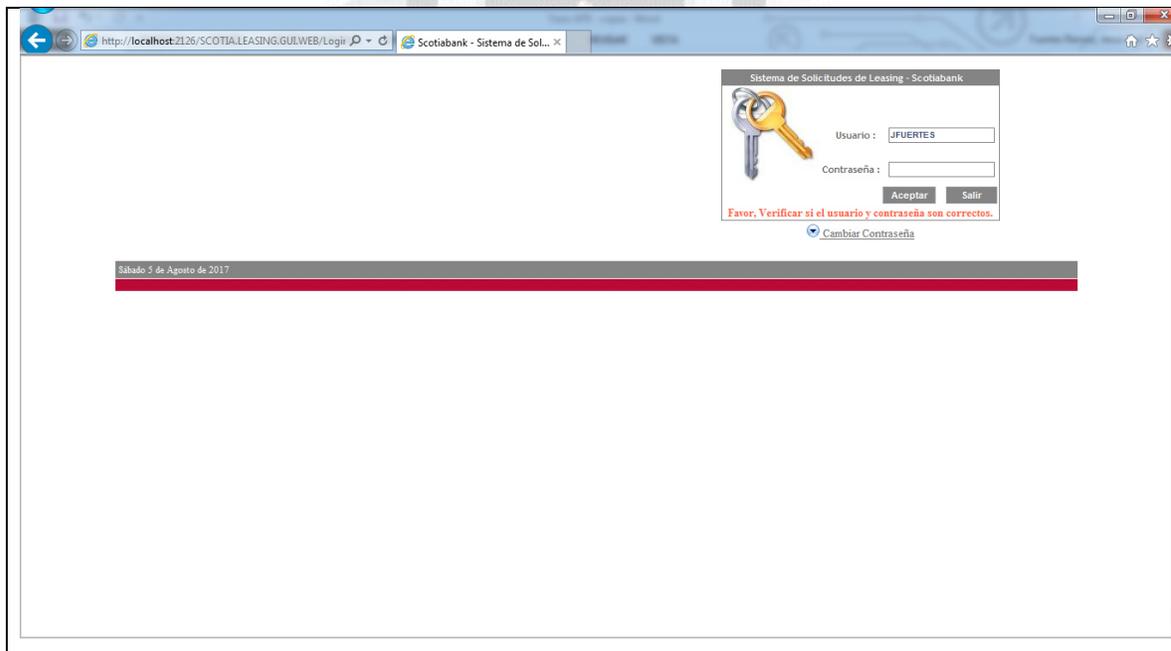


Figura 5.64: Flujo alternativo - Verificar si usuario y contraseña son correctos. (Fuente: Elaboración propia)

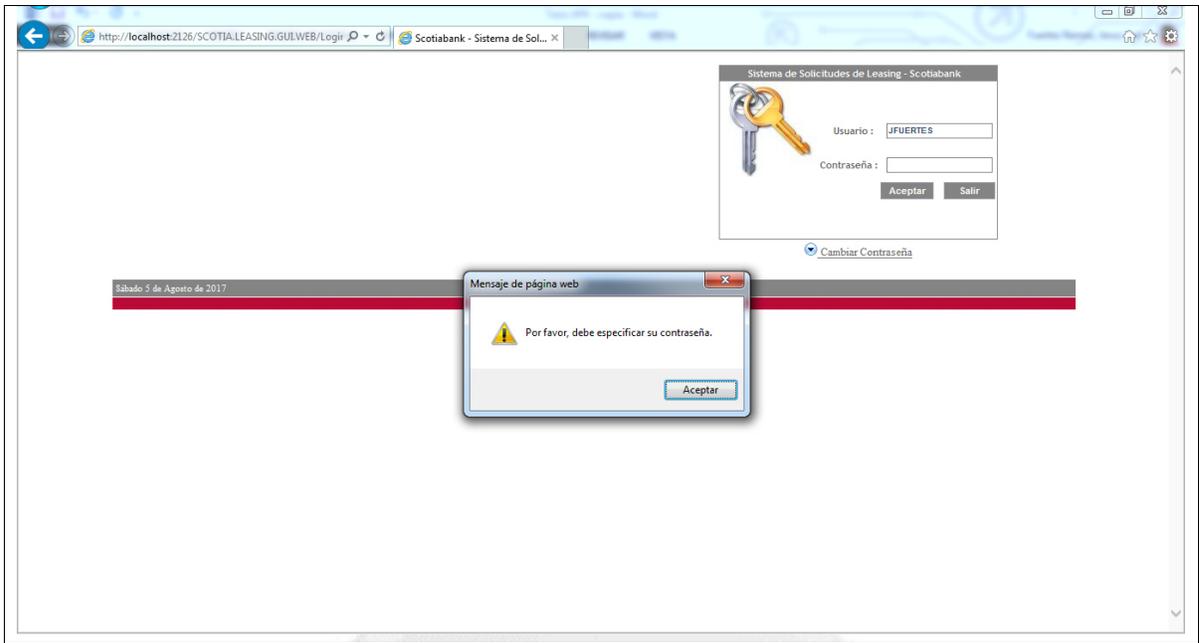


Figura 5.65: Flujo alternativo - Especificar su contraseña. (Fuente: Elaboración propia)



p. Especificación de caso de uso 16 - Restablecer contraseña

1. Breve Descripción

Este caso de uso permite a **todos los usuarios** del sistema cambiar la contraseña de sus cuentas.

2. Flujo de Eventos

Evento Disparador: El caso de uso **restablecer contraseña** puede ser llamado por el caso de uso **ingresar al sistema**.

2.1 Flujo Básico <<Restablecer contraseña>>

2.1.1 El sistema muestra la interfaz **cambiar contraseña**.

La interfaz contiene los campos activos para completar los datos: **Usuario, contraseña anterior y contraseña nueva**.

Incluye la opción: **Aceptar**.

2.1.2 El **usuario** digita la información en los campos activos.

2.1.3 El sistema genera una nueva contraseña con el mensaje **la clave del usuario xxxxx fue cambiada correctamente**.

2.1.4 Fin del caso de uso.

3. Flujos Alternativos

3.1 <<Clave anterior no coincide>>

En el punto 2.1.2 del flujo básico **restablecer contraseña**, en caso la contraseña anterior no esté correctamente digitada, el sistema muestra el mensaje **la clave anterior del usuario xxxxx no coincide**, el flujo continúa en el punto 2.1.2.

4. Requerimientos Especiales

4.1 El caso de uso debe estar disponible a través del **navegador internet explorer**.

5. Pre Condiciones

5.1 El **usuario** debe estar registrado en el sistema.

6. Post Condiciones

6.1 La contraseña del usuario ha sido restablecida.

7. Puntos de Extensión

Ninguno.

8. Prototipo (GUI)

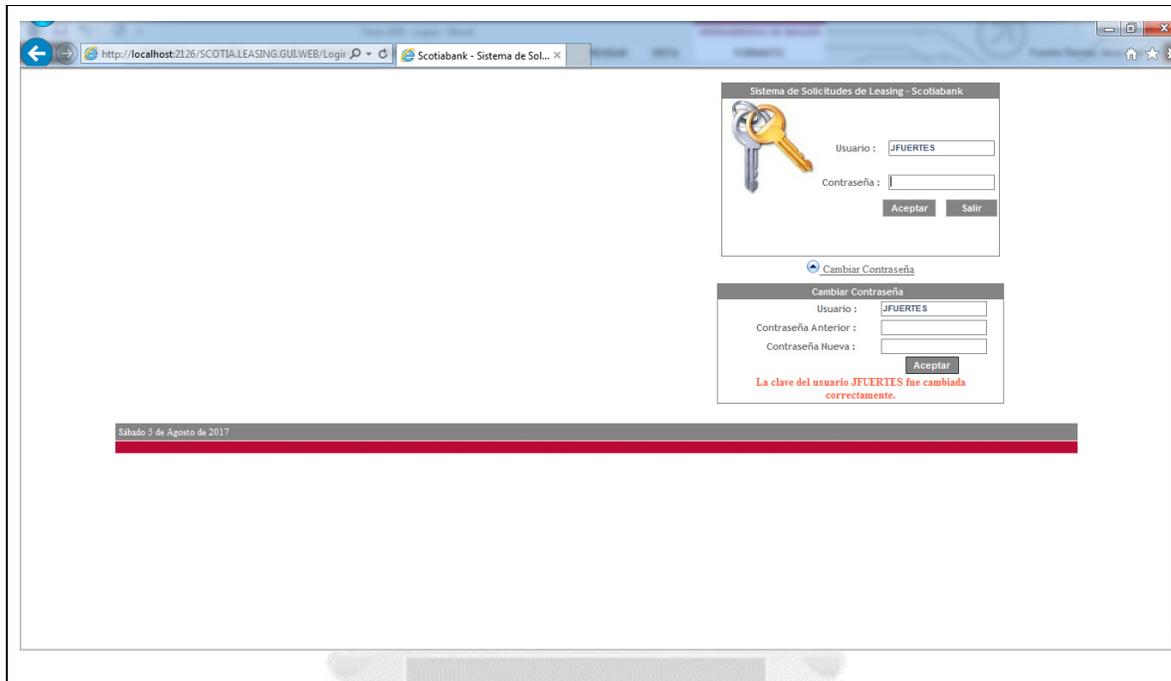


Figura 5.66: Flujo básico - Restablecer contraseña. (Fuente: Elaboración propia)

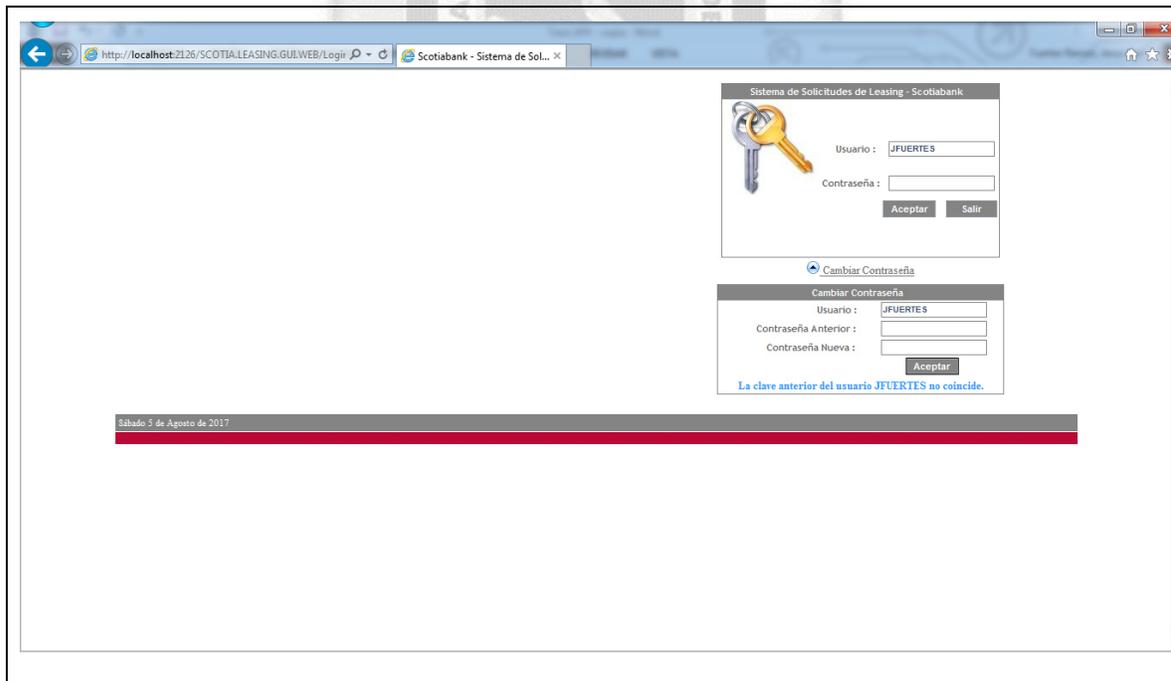


Figura 5.67: Flujo alternativo - Clave anterior no coincide. (Fuente: Elaboración propia)

5.3 Modelo de datos

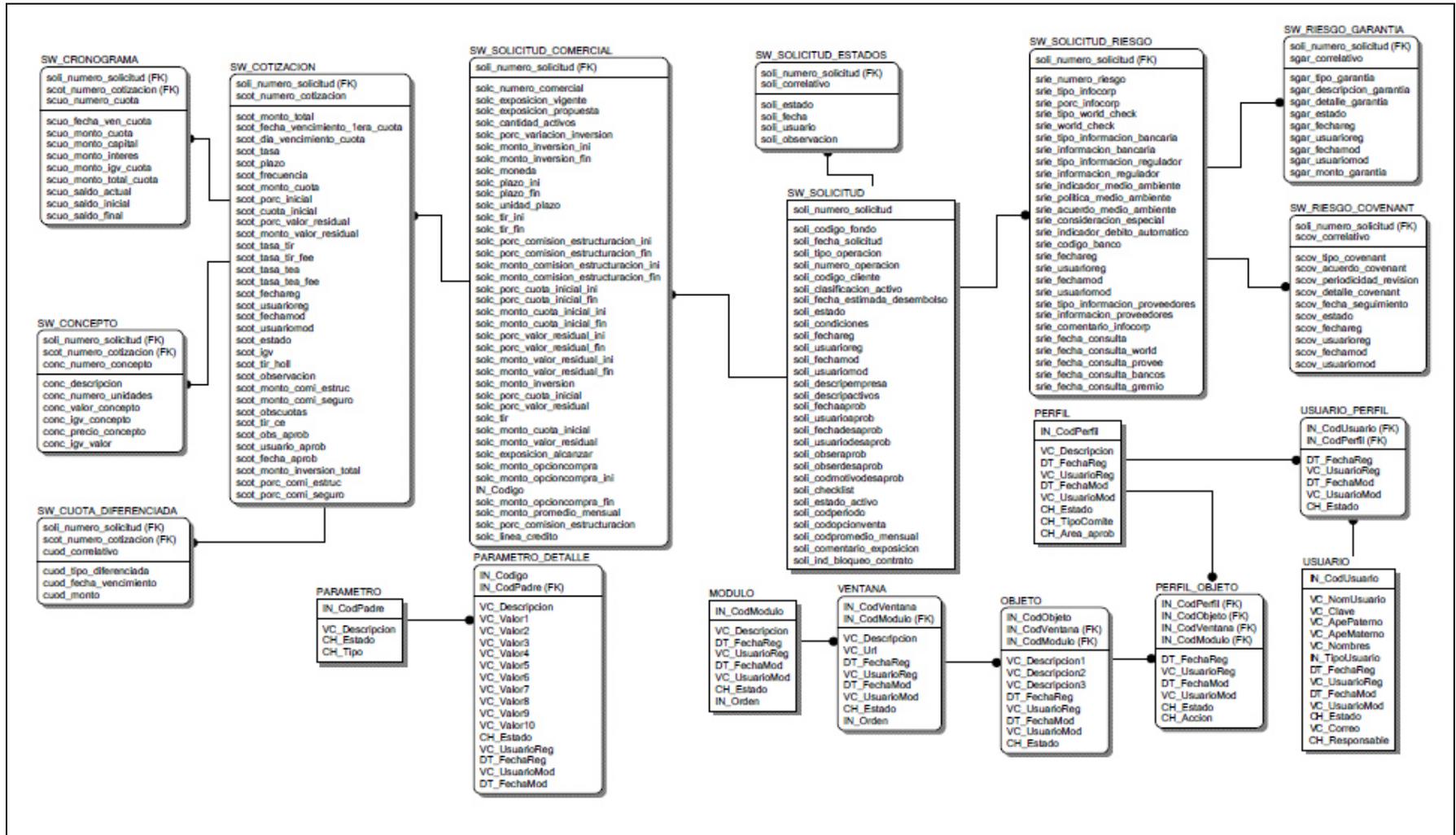


Figura 5.68: Base de datos solicitud. (Fuente: Elaboración propia)

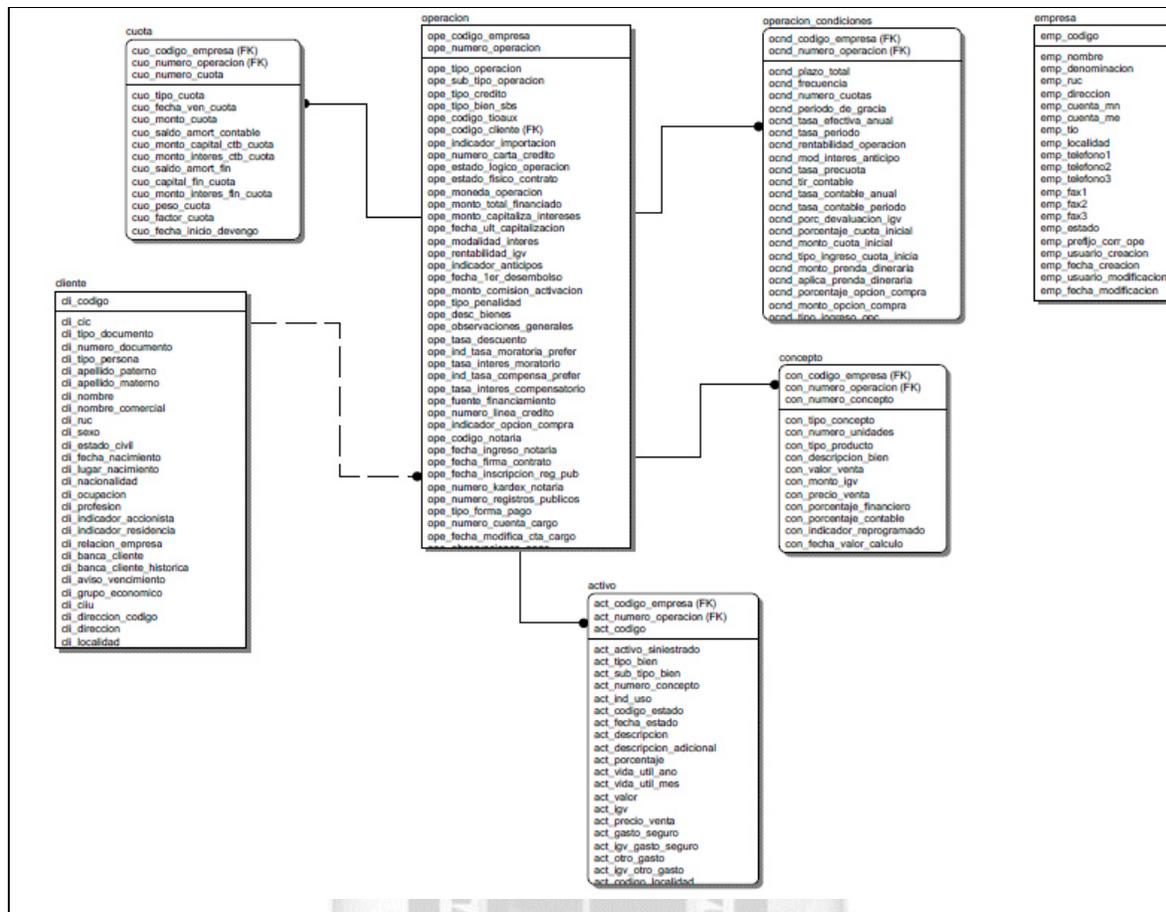


Figura 5.69: Base de datos fondo leasing. (Fuente: Elaboración propia)

En el desarrollo del presente trabajo se implementaron 2 bases de datos principales que se describen a continuación:

- La base de datos **solicitud** guarda los objetos relacionados a la aprobación de la solicitud, siendo las principales entidades: SW_solicitud, sw_solicitud_comercial, sw_solicitud_riesgos
- La base de datos **fondo leasing** guarda los objetos relacionados con las operaciones de arrendamiento financiero y que representa una base de datos de consulta para la base de datos **solicitud**, siendo sus principales entidades: cliente, operación y cuota

5.4 Diagrama de componentes

El sistema muestra sus componentes con sus dependencias y paquetes respectivos.

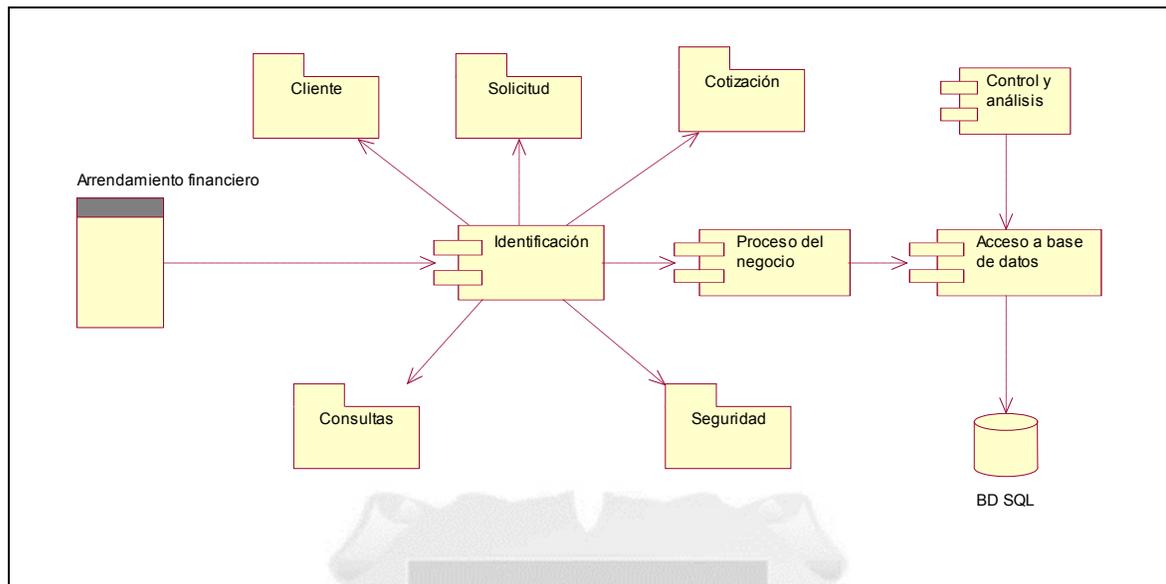


Figura 5.70: Diagrama de componentes. (Fuente: Elaboración propia)

5.5 Diagrama de despliegue

El sistema consta de 3 capas y 3 niveles, según la implementación realizada.

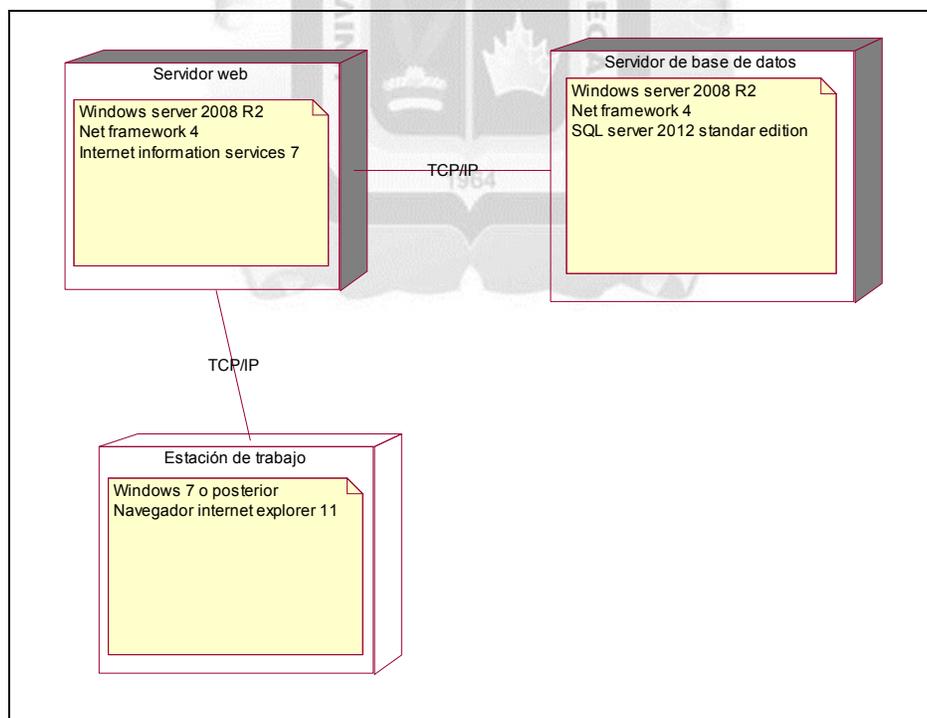


Figura 5.71: Diagrama de despliegue. (Fuente: Elaboración propia)

5.6 Diagrama de alto nivel

Según la arquitectura del sistema se presenta el siguiente diagrama:

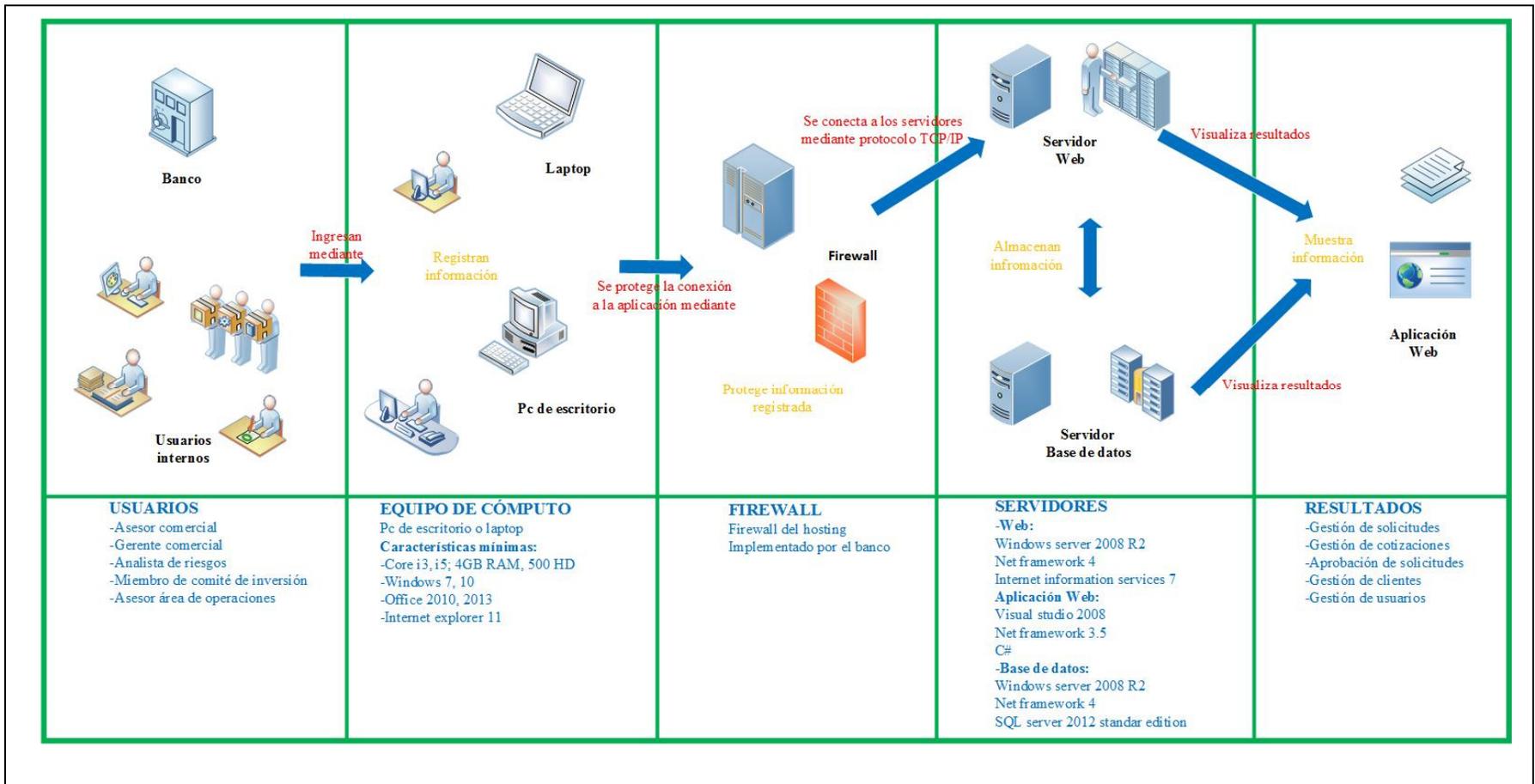


Figura 5.72: Diagrama de alto nivel. (Fuente: Elaboración propia)

CAPÍTULO VI: RESULTADOS

En este capítulo se muestra los resultados de la aplicación web, siguiendo el cumplimiento de los indicadores de calidad del ISO 9126 descritos en el marco teórico y en las hipótesis específicas. Para poder verificar dicho cumplimiento, se elaboró una encuesta (Ver anexo V: Encuesta de evaluación de calidad sobre la aplicación web) en la que participaron y respondieron las interrogantes 16 usuarios, luego del uso de la solución de forma local desde un equipo portátil, la descripción sobre los resultados se muestran a continuación:

6.1 Funcionalidad de la aplicación web

La capacidad del producto de software para proporcionar funciones que satisfacen las necesidades declaradas e implícitas cuando el software se utiliza bajo condiciones especificadas. (ISO/IEC 9126-1:2001)

Para verificar que se cumplía este indicador se evaluó el proceso de gestión de clientes, solicitudes, cotizaciones, aprobación de solicitudes e ingreso a la aplicación web.

Se concluye que la aplicación web cumplió con el indicador de funcionalidad por las siguientes razones:

- Un 87.5% afirma que fue muy satisfactorio el proceso de agregar y crear clientes en la aplicación web, mientras que un 12.5% afirma que fue satisfactorio.
- Un 94% afirma que fue muy satisfactorio el proceso de crear solicitudes en la aplicación web, mientras que un 6% afirma que fue satisfactorio.
- Un 81% afirma que fue muy satisfactorio el proceso de agregar cotizaciones en la aplicación web, mientras que un 19% afirma que fue satisfactorio.
- Un 100% afirma que fue muy satisfactorio el proceso de modificar datos de solicitudes, cotizaciones en la aplicación web.
- Un 100% afirma que fue muy satisfactorio la validación del ingreso de información en la aplicación web.
- Un 75% afirma que fue muy satisfactorio el proceso de aprobar cotizaciones, solicitudes en la aplicación web, mientras que un 25% afirma que fue satisfactorio.
- Un 100% afirma que fue muy satisfactorio el proceso del login del usuario en la aplicación web.

6.2 Usabilidad de la aplicación web

La capacidad del producto de software para ser entendido, aprendido, utilizado y atractivo para el usuario, cuando se utiliza bajo condiciones especificadas. (ISO/IEC 9126-1:2001)

Para verificar que se cumplía este indicador se evaluó la facilidad que ofrece la aplicación web en sus diferentes módulos.

Se concluye que la aplicación web cumplió con el indicador de usabilidad por las siguientes razones:

- Un 62.5% afirma que fue muy fácil realizar la creación de clientes en la aplicación web, mientras que un 37.5% afirma que fue fácil.

- Un 56% afirma que fue muy fácil realizar la creación de solicitudes en la aplicación web, mientras que un 44% afirma que fue fácil.
- Un 44% afirma que fue muy fácil realizar la creación de cotizaciones en la aplicación web, mientras que un 56% afirma que fue fácil.
- Un 69% afirma que el contenido de la aplicación web se muestra muy entendible para su uso, mientras que un 31% afirma que es entendible.
- Un 100% afirma que fue muy fácil ingresar a la aplicación web.
- Un 50% afirma que fue muy fácil la navegación dentro de la aplicación web, mientras que un 50% afirma que fue fácil.

6.3 Eficiencia de la aplicación web

La capacidad del producto de software para proporcionar un rendimiento adecuado, en relación con la cantidad de los recursos utilizados, en las condiciones establecidas. (ISO/IEC 9126-1:2001)

Para verificar que se cumplía este indicador se evaluó la rapidez de la navegación, módulos y exportación de archivos desde la aplicación web.

Se concluye que la aplicación web cumplió con el indicador de funcionalidad por las siguientes razones:

- Un 37.5% afirma que fue muy rápida la exportación de cotizaciones y solicitudes en la aplicación web, mientras que un 62.5% afirma que fue rápida.
- Un 31% afirma que fue muy rápido el tiempo de respuesta de los módulos en la aplicación web, mientras que un 69% afirma que fue rápido.
- Un 50% afirma que fue muy satisfactorio el tiempo de respuesta de la bandeja de solicitudes en la aplicación web, mientras que un 50% afirma que fue satisfactorio.
- Un 75% afirma que fue muy rápida la búsqueda de clientes y solicitudes en la aplicación web, mientras que un 25% afirma que fue rápida.

6.4 Mantenibilidad de la aplicación web

La capacidad del producto de software de ser modificado. Las modificaciones pueden incluir correcciones, mejoras o adaptación del software a los cambios en el ambiente, en los requisitos y especificaciones funcionales. (ISO/IEC 9126-1:2001)

Para verificar el cumplimiento de este indicador se desarrolló la aplicación web basándose en el grupo de políticas TI que utiliza el banco en este tipo de trabajos, las cuales se acataron y son las siguientes:

- **Políticas de uso de software**

Una de las principales políticas del banco en materia de TI es el uso de software licenciado en el desarrollo/implementación de sus diferentes aplicaciones/sistemas de información, en el presente trabajo se hizo uso de los siguientes programas:

- SQL 2012 licenciado
- Visual Studio 2008 licenciado

- Windows Server 2008 licenciado

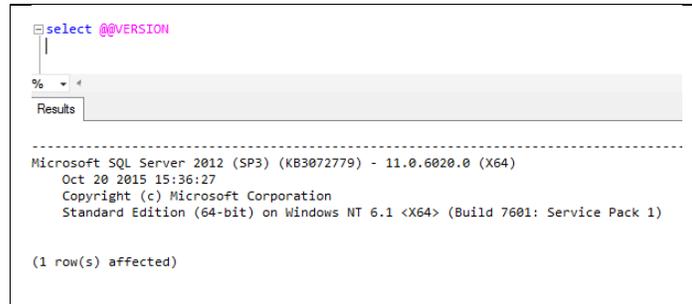


Figura 6.1: SQL 2012. (Fuente: Aplicación)

- **Políticas de programación**

- Lenguaje de programación basado en C#
- Versión Framework 3.5
- Motor de base de datos SQL 2012

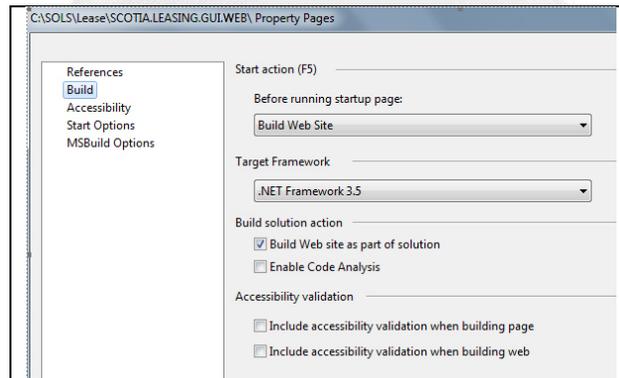


Figura 6.2: Framework 3.5. (Fuente: Aplicación)

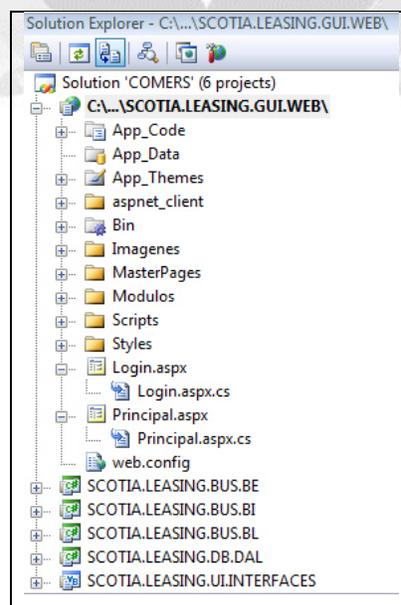


Figura 6.3: Lenguaje C#. (Fuente: Elaboración propia)

- **Políticas de aplicación web**
 - Uso del navegador Internet Explorer versión 11
 - Número de registros por página (clientes, solicitud): 15 registros
 - Plazo máximo de contratos: 36 meses



Figura 6.4: Navegador internet explorer 11. (Fuente: Elaboración propia)

The screenshot displays a web application interface with a table of records. The table has two columns: 'Razon Social' and 'RUC'. The table contains 15 rows of data. The interface also shows 'Total Registros: 30', navigation buttons, a 'Nuevo' button, and 'Página 1 de 2'.

Razon Social	RUC
AJEPER S.A.	20331061655
ALICORP SAA	20100055237
AXUR S.A.	20427919111
BAZAN FALCON ALIPIO	10079907028
BEFESA PERU S.A.	20503610711
CEMEX PERU S.A.	20516020301
COSAPI S.A.	20100082391
DINETPERU S.A.	20515351541
EQUIMAQ S.A.C.	20503858330
ERCO S.A.C.	20100073057
FILE SERVICE SA	20383893781
FUERTE IMPORTACIONES	10097992889
GALLOS MARMOLERIA SA	20123444656
GYM S.A.	20100154057
HAM MICROSYSTEM	10097992769

Figura 6.5: Lista de registros. (Fuente: Elaboración propia)

Estas políticas se fueron implementando en las diferentes fases del desarrollo, para así llegar al resultado final, cumpliendo los requerimientos solicitados, los cuales también se pueden modificar en caso sea necesario, ya que es una de las características de este indicador.

(Ver anexo VI: Resultados de la encuesta de evaluación de calidad sobre la aplicación web)

CONCLUSIONES

- La importancia del arrendamiento financiero en la actualidad para la obtención de bienes en corto a mediano plazo, lo que conlleva a que las entidades financieras deseen brindar de manera continua ese producto.
- Se investigó y analizó diversas soluciones o proyectos TI para arrendamientos financieros, en la mayoría de los casos se pudo comprobar un impacto positivo en los escenarios donde se han aplicado.
- El uso de metodología RUP para llevar a cabo el presente proyecto, basándose en las necesidades del negocio.
- La identificación de los procesos de negocio relacionados al arrendamiento financiero, para garantizar la viabilidad del proyecto y que la solución se ajuste a las necesidades del cliente.
- Como base de la aplicación web se tiene los diagramas desarrollados, ya que a partir de estos se obtuvo la funcionalidad para su diseño completo, mediante la metodología RUP, siendo el marco de referencia para la elaboración de la aplicación web.
- La aplicación desarrollada está orientada a servir como soporte en el proceso de mejora en la aprobación de arrendamientos financieros.
- La aplicación web facilita la tarea administrativa de los usuarios de las áreas involucradas, brindando una información actualizada con rapidez y eficacia sobre las diferentes solicitudes.
- En conclusión, con la implementación de la aplicación web se mejora el proceso de la aprobación de arrendamientos financieros, lo que impacta positivamente en el banco para captar y ganar nuevos clientes.

RECOMENDACIONES

- Se recomienda desarrollar un módulo para el uso del área legal con la opción de insertar archivos externos que son los contratos escaneados.
- Se recomienda desarrollar/habilitar la opción de envío de correos en los diferentes módulos, para que los usuarios reciban notificaciones de los estados actuales en sus buzones de correo.
- Se recomienda interconectar con sistemas externos para implementar la funcionalidad de realizar la evaluación de riesgos en la misma aplicación.
- Se recomienda ampliar el alcance de la aplicación para el uso en dispositivos móviles.
- Se recomienda respaldar las bases de datos periódicamente y alojar los respaldos en una ubicación externa, en caso llegara a pasar alguna calamidad.
- El desarrollo del presente trabajo de investigación puede ser tomado como modelo para nuevos proyectos en otras entidades financieras.



REFERENCIAS BIBLIOGRÁFICAS

- Barbancho, J. et al. (2014). *Redes locales*. Madrid, España: Ediciones Paraninfo.
- Belaunde, G. (2015, 31 de mayo). *Conociendo a los "Covenants"* [web log post]. Recuperado de <http://blogs.gestion.pe/riesgosfinancieros/2015/05/conociendo-a-los-covenants.html>
- Calero, C., Piattini, M. & Moraga, M. (2010). *Calidad del producto y proceso software*. Madrid, España: Editorial Ra-Ma.
- Contreras, M. (2016). *Desarrollo de aplicaciones web multiplataforma*. Madrid, España: Ministerio de Educación. Secretaría general técnica. Subdirección general de documentación y publicaciones.
- Díaz, M. & Vásquez, N. (2013). *Normas internacionales de información financiera (NIIF-IFRS): Principios y ejercicios básicos de contabilidad global*. Barcelona, España: Profit Editorial.
- Enríquez, G. (2015). *Efectos del arrendamiento financiero en la gestión de las medianas empresas mineras del Perú* (Tesis de maestría). Universidad de San Martín de Porres, Lima, Perú.
- Gil, C. (2008). RUP: Metodología en los sistemas y aplicaciones basada en la web. *Avances Investigación en Ingeniería*, 8(1), 83-87.
- Granados, R. (2015). *Desarrollo de aplicaciones web en el entorno servidor*. Málaga, España: IC Editorial.
- Gualán, M. & Montúfar, D. (2012). *Aplicación web para la gestión de microcréditos* (Tesis de pregrado). Universidad de Cuenca, facultad de ingeniería, escuela de informática, Cuenca, Ecuador.
- Haime, L. (2003). *El Arrendamiento Financiero*. México D.F., México: Ediciones Fiscales ISEF.
- Herranz, R. (2016). *Despegar con Scrum*. España: Utópica informática
- ISO/IEC 9126-1:2001, Information technology -- Software product quality -- Part 1: Quality model, -- International Standard
- Kroll, P. & Kruchten, P. (2003). *The Rational Unified Process Made Easy: A Practitioner's Guide to the RUP*. Boston, MA, USA: Addison-Wesley Professional.
- Kruchten, P. (2004). *The Rational Unified Process: An Introduction*. Boston, MA, USA: Addison-Wesley Professional.
- Laínez, J. (2015). *Desarrollo de Software ÁGIL: Extreme Programming y Scrum*. IT Campus Academy. Recuperado de books.google.es/books?isbn=150295222X
- Luján, S. (2002). *Programación de aplicaciones web: Historia, principios básicos y clientes web*. Alicante: Club Universitario
- Mamani, J. (2013). El Financiamiento a través del Leaseback (Retroarrendamiento). *Actualidad empresarial*, 4(290), 8.
- Martel, A. (2014). *Gestión Práctica de Proyectos Con Scrum: Desarrollo de Software Ágil Para El Scrum Master*. Edición illustrated. Recuperado de books.google.es/books?isbn=1517192366
- Mávila, D. (2003). Leasing financiero. *Notas científicas de aplicaciones*, 6(1), 86-88. Recuperado de http://sisbib.unmsm.edu.pe/bibvirtualdata/publicaciones/indata/Vol6_n1/pdf/leasing.pdf

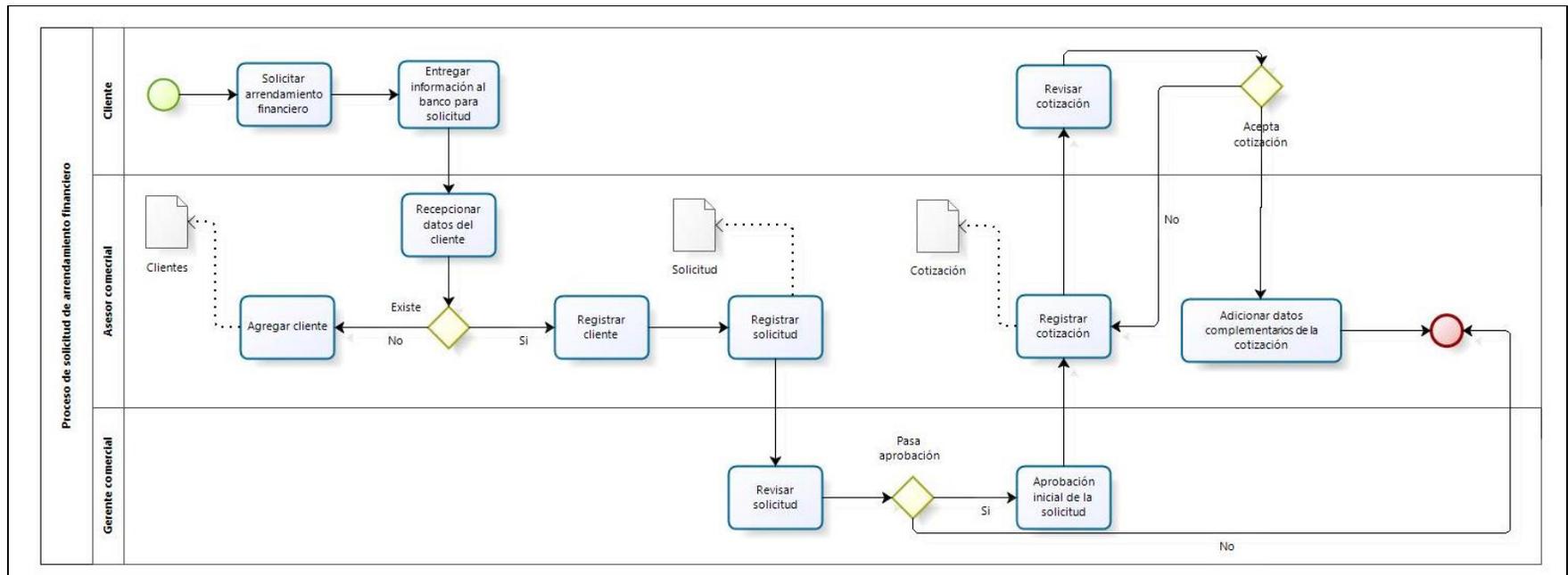
- Menéndez-Barzanallana, R. (2012). *Historia del desarrollo de aplicaciones web*. Murcia: Universidad de Murcia. Recuperado de <http://www.um.es/docencia/barzana/DIVULGACION/INFORMATICA/Historia-desarrollo-aplicaciones-web.html>
- Molina, J. & Quishpi, L. (2015), *Desarrollo de herramienta de gestión de proyectos RUP usando metodología Scrum + XP* (Tesis de maestría). Universidad Politécnica de Madrid, escuela técnica superior de ingeniería de sistemas informáticos, Madrid, España.
- Pérez, W. (2016). El leasing financiero un medio para mejorar los resultados de la industria. *Fides et Ratio-Revista de Difusión cultural y científica de la Universidad La Salle en Bolivia*, 11(11), 139-154. Recuperado de http://www.scielo.org.bo/scielo.php?script=sci_arttext&pid=S2071-081X2016000100010&lng=es&tlng=es
- Piña, J. & Del Águila, S. (2014). *Implicancia del leasing financiero en la rentabilidad de la empresa Global Rent en el periodo 2011-2013* (Tesis de pregrado). Universidad Nacional de San Martín, Tarapoto, Perú.
- Quinatoa, E. (2016). *Desarrollo de una metodología de trabajo basado en RUP para la creación de aplicaciones de software en la dirección de desarrollo institucional del IESS de la ciudad de Quito* (Tesis de maestría). ESPE Universidad de las fuerzas armadas, Latacunga, Quito, Ecuador.
- Ramos, A. & Ramos, M. (2014). *Sistemas informáticos y redes: Aplicaciones Web*. Madrid, España: Ediciones Paraninfo.
- Rodríguez, C. (2015). *Experiencia profesional en el área de leasing en el BCP durante los años 2010 al 2013* (Tesis de pregrado). Universidad Católica Santa María, Arequipa, Perú.
- Saldaña, M. (2012). *Propuesta de mejora del proceso de archivo de la documentación de leasing de una entidad financiera* (Tesis de pregrado). Pontificia Universidad Católica del Perú, Lima, Perú.
- Sancho, K. (2013). *Cotizador leasing: aplicación web y móvil para que profesionales realicen cotizaciones leasing en el BAC San José* (Tesis de pregrado). Tecnológico de Costa Rica: Escuela de Ingeniería en Computación, San Carlos, Costa Rica.
- Tabares, L. (2011). *Personalización de RUP para proyectos académicos de desarrollo de software* (Tesis de pregrado). Universidad EAFIT, escuela de ingeniería, Medellín, Colombia.
- Tahuiton, J. (2011). *Arquitectura de software para aplicaciones web* (Tesis de maestría). Instituto Politécnico Nacional, unidad Zacatenco, departamento de computación, México D.F., México.
- Talledo, J. (2015). *Implantación de aplicaciones web en entorno internet, intranet y extranet*. Madrid, España: Ediciones Paraninfo.
- Toro, F. (2013). *Administración de proyectos de informática*. Bogotá, Colombia: Ecoe Ediciones.
- Wong, L. & Torres, F. (2010). Mejorando las debilidades de RUP para la gestión de proyectos. *Revista de investigación de sistemas e informática*, 7(2), 49-56.
- Yucra, D. & Gonzáles, S. (2012). *Tecnologías web*. Lima, Perú: Fondo editorial de la UIGV.

ANEXOS

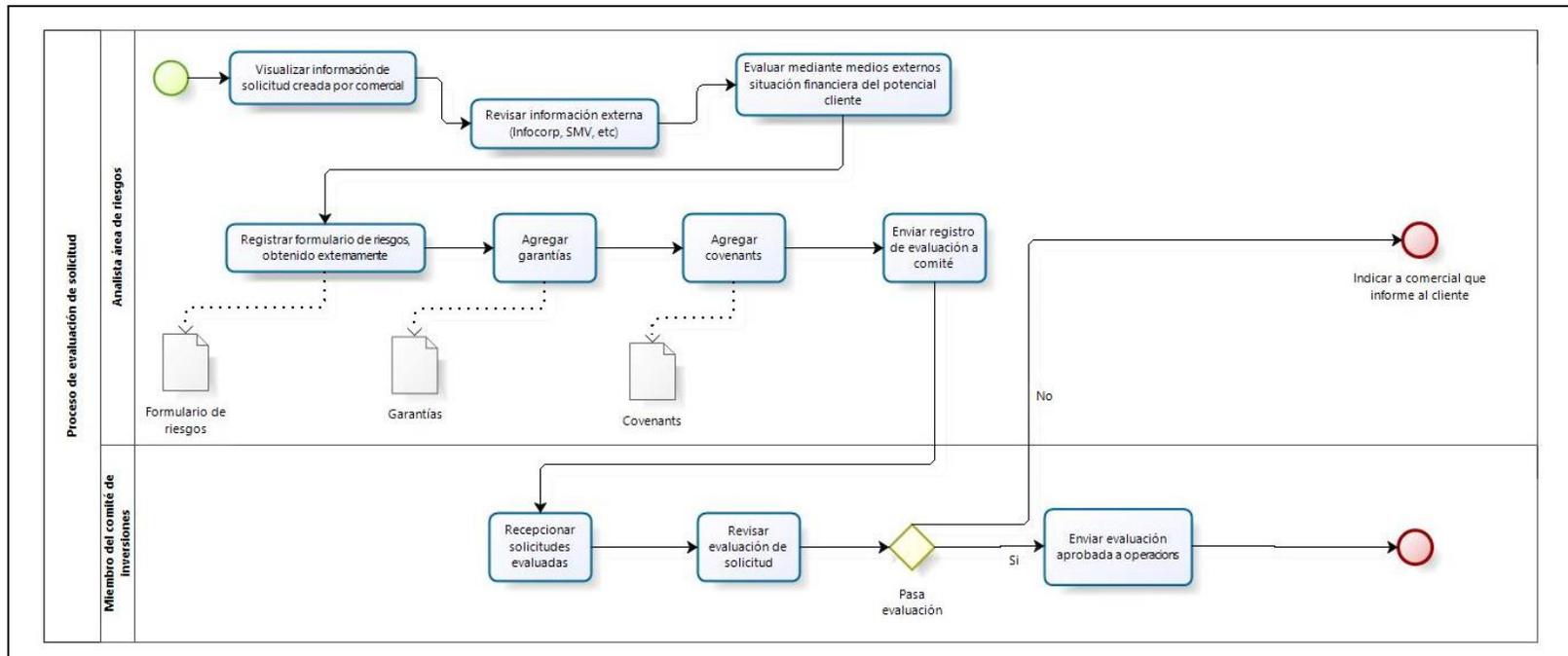
ANEXO I: MATRIZ DE CONSISTENCIA

	PROBLEMAS	OBJETIVOS	HIPÓTESIS	VARIABLES	INDICADORES
GENERAL	¿Cómo influye el desarrollo de una aplicación web en la mejora de la aprobación de arrendamientos financieros en el banco Scotiabank?	Determinar la influencia de una aplicación web para la mejora de la aprobación de arrendamientos financieros en el banco Scotiabank	El desarrollo de una aplicación web influye significativamente en la mejora de la aprobación de arrendamientos financieros en el banco Scotiabank	Independiente: Desarrollo de una aplicación web	<ul style="list-style-type: none"> • Nivel de funcionalidad • Nivel de usabilidad • Nivel de mantenibilidad • Nivel de eficiencia
ESPECÍFICO	¿Cómo influye el nivel de funcionalidad en la mejora de la aprobación de arrendamientos financieros en el banco Scotiabank?	Determinar la influencia del nivel de funcionalidad en la mejora de la aprobación de arrendamientos financieros en el banco Scotiabank	El nivel de funcionalidad influye significativamente en la mejora de la aprobación de arrendamientos financieros en el banco Scotiabank	Dependiente: Mejora de la aprobación de arrendamientos financieros en el banco Scotiabank	<ul style="list-style-type: none"> • Tiempo de realización y aprobación de consultas • Cuotas periódicas • Número de activos arrendados • Pagos realizados • Número de arrendamientos financieros
	¿Cómo influye el nivel de usabilidad en la mejora de la aprobación de arrendamientos financieros en el banco Scotiabank?	Determinar la influencia del nivel de usabilidad en la mejora de la aprobación de arrendamientos financieros en el banco Scotiabank	El nivel de usabilidad influye significativamente en la mejora de la aprobación de arrendamientos financieros en el banco Scotiabank		
	¿Cómo influye el nivel de eficiencia en la mejora de la aprobación de arrendamientos financieros en el banco Scotiabank?	Determinar la influencia del nivel de eficiencia en la mejora de la aprobación de arrendamientos financieros en el banco Scotiabank	El nivel de eficiencia influye significativamente en la mejora de la aprobación de arrendamientos financieros en el banco Scotiabank		
	¿Cómo influye el nivel de mantenibilidad en la mejora de la aprobación de arrendamientos financieros en el banco Scotiabank?	Determinar la influencia del nivel de mantenibilidad en la mejora de la aprobación de arrendamientos financieros en el banco Scotiabank	El nivel de mantenibilidad influye significativamente en la mejora de la aprobación de arrendamientos financieros en el banco Scotiabank		

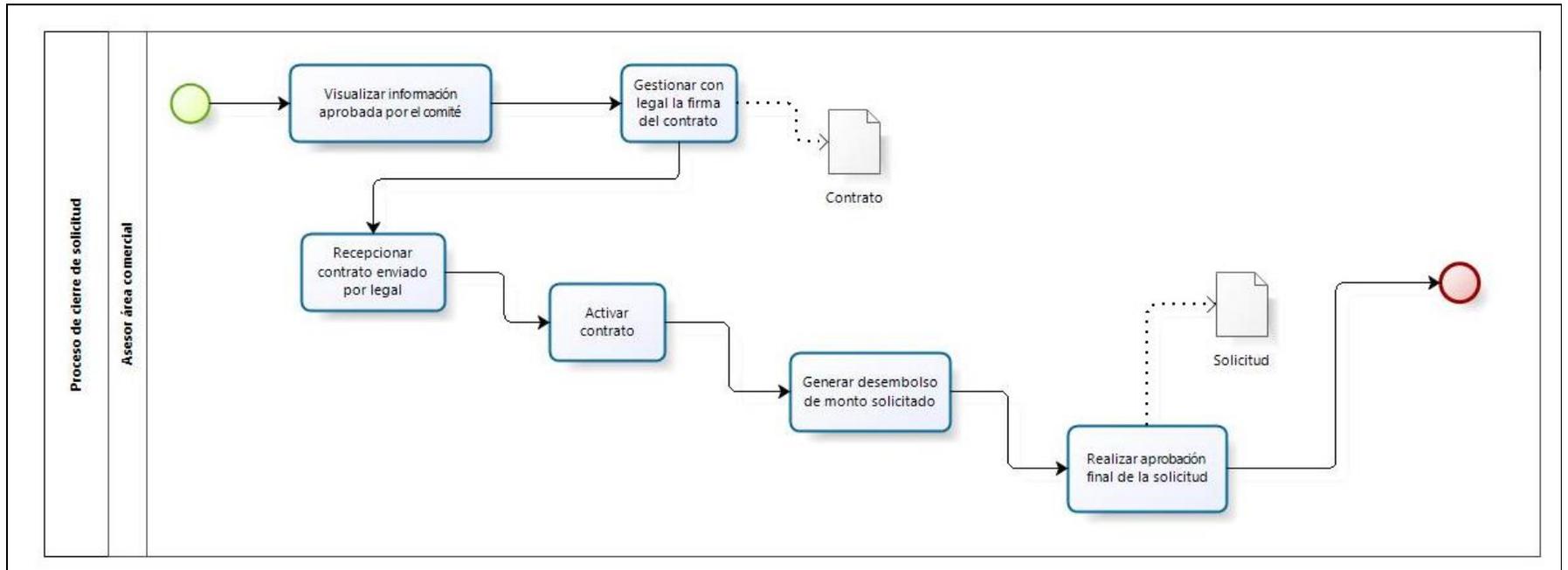
ANEXO II: PROCESO DE SOLICITUD DE ARRENDAMIENTO FINANCIERO TO BE



ANEXO III: PROCESO DE EVALUACION DE SOLICITUD TO BE



ANEXO IV: PROCESO DE CIERRE DE SOLICITUD TO BE



ANEXO V: ENCUESTA DE EVALUACIÓN DE CALIDAD SOBRE LA APLICACIÓN WEB



ENCUESTA INTERNA PARA EVALUACION DE CALIDAD DE APLICACIÓN WEB PARA LA MEJORA DE LA APROBACIÓN DE ARRENDAMIENTOS FINANCIEROS EN EL BANCO SCOTIABANK PERU

Apellidos y nombres:

Fecha: _____

Nota: Esta encuesta se realiza para evaluar la calidad de la aplicación web de parte de los usuarios que la utilizaron; basándose en preguntas sobre funcionalidad, usabilidad y eficiencia de la misma

Marcar con una **X** en el recuadro debajo de la opción indicada

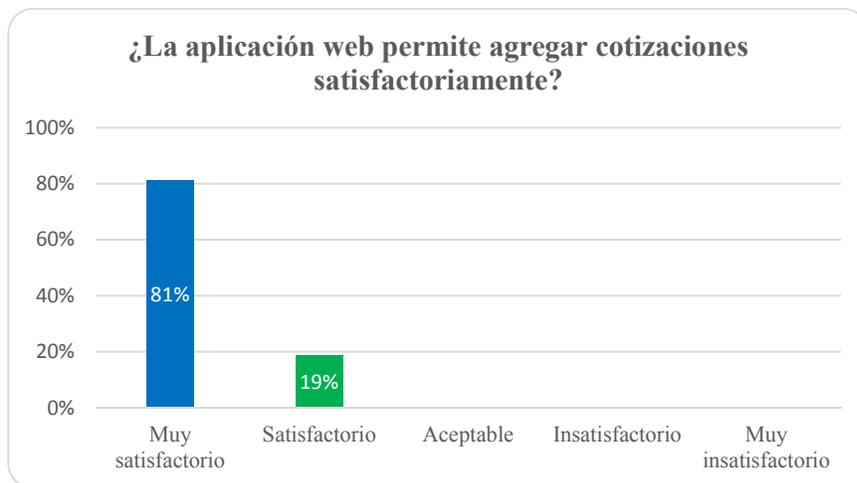
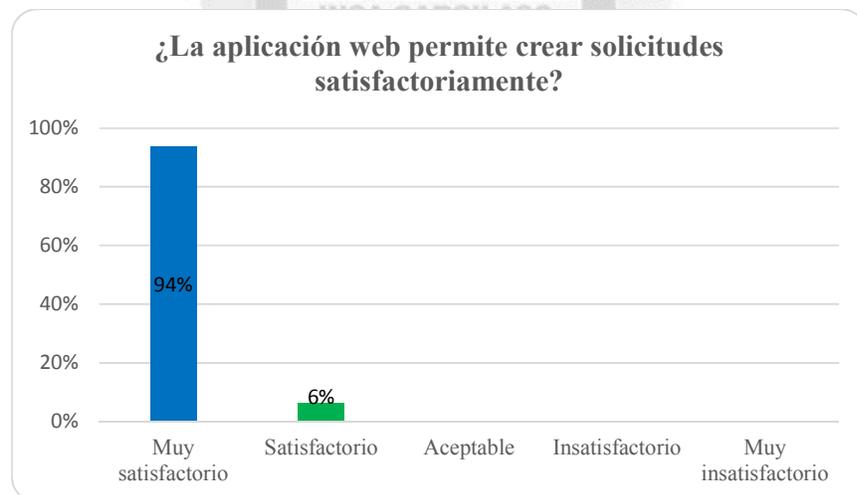
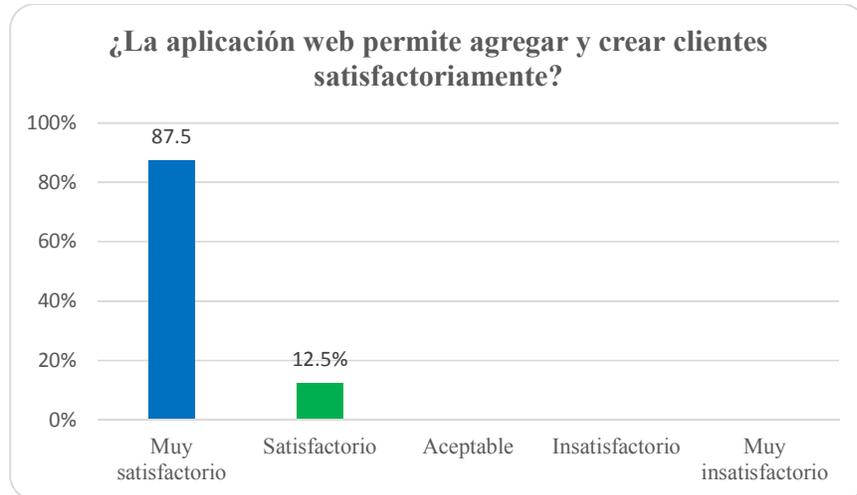
FUNCIONALIDAD DE LA APLICACIÓN WEB						
1	¿La aplicación web permite agregar y crear clientes satisfactoriamente?	Muy satisfactorio	Satisfactorio	Aceptable	Insatisfactorio	Muy insatisfactorio
		<input type="checkbox"/>				
2	¿La aplicación web permite crear solicitudes satisfactoriamente?	Muy satisfactorio	Satisfactorio	Aceptable	Insatisfactorio	Muy insatisfactorio
		<input type="checkbox"/>				
3	¿La aplicación web permite agregar cotizaciones satisfactoriamente?	Muy satisfactorio	Satisfactorio	Aceptable	Insatisfactorio	Muy insatisfactorio
		<input type="checkbox"/>				
4	¿La aplicación web permite modificar datos de solicitudes, cotizaciones satisfactoriamente?	Muy satisfactorio	Satisfactorio	Aceptable	Insatisfactorio	Muy insatisfactorio
		<input type="checkbox"/>				
5	¿La aplicación web valida el ingreso de información satisfactoriamente?	Muy satisfactorio	Satisfactorio	Aceptable	Insatisfactorio	Muy insatisfactorio
		<input type="checkbox"/>				
6	¿La aplicación web permite aprobar cotizaciones, solicitudes satisfactoriamente?	Muy satisfactorio	Satisfactorio	Aceptable	Insatisfactorio	Muy insatisfactorio
		<input type="checkbox"/>				
7	¿La aplicación web permite el login del usuario satisfactoriamente?	Muy satisfactorio	Satisfactorio	Aceptable	Insatisfactorio	Muy insatisfactorio
		<input type="checkbox"/>				
USABILIDAD DE LA APLICACIÓN WEB						
8	¿Es fácil realizar la creación de clientes en la aplicación web?	Muy fácil	Fácil	Normal	Difícil	Muy difícil
		<input type="checkbox"/>				

9	¿Es fácil realizar la creación de solicitudes en la aplicación web?	Muy fácil	Fácil	Normal	Difícil	Muy difícil
10	¿Es fácil realizar la creación de cotizaciones en la aplicación web?	Muy fácil	Fácil	Normal	Difícil	Muy difícil
11	¿El contenido de la aplicación web se muestra entendible para su uso?	Muy entendible	Entendible	Normal	No es entendible	Nada entendible
12	¿Es fácil ingresar a la aplicación web?	Muy fácil	Fácil	Normal	Difícil	Muy difícil
13	¿Es fácil la navegación dentro de la aplicación web?	Muy fácil	Fácil	Normal	Difícil	Muy difícil
EFICIENCIA DE LA APLICACIÓN WEB						
14	¿La exportación de cotizaciones y solicitudes en la aplicación web fue rápida?	Muy rápida	Rápida	Regular	Lenta	Muy lenta
15	¿El tiempo de respuesta de los módulos en la aplicación web fue rápida?	Muy rápida	Rápida	Regular	Lenta	Muy lenta
16	¿Es satisfactorio el tiempo de respuesta de la bandeja de solicitudes?	Muy satisfactorio	Satisfactorio	Aceptable	Insatisfactorio	Muy insatisfactorio
17	¿Considera que la búsqueda de clientes y solicitudes es rápida?	Muy rápida	Rápida	Regular	Lenta	Muy lenta

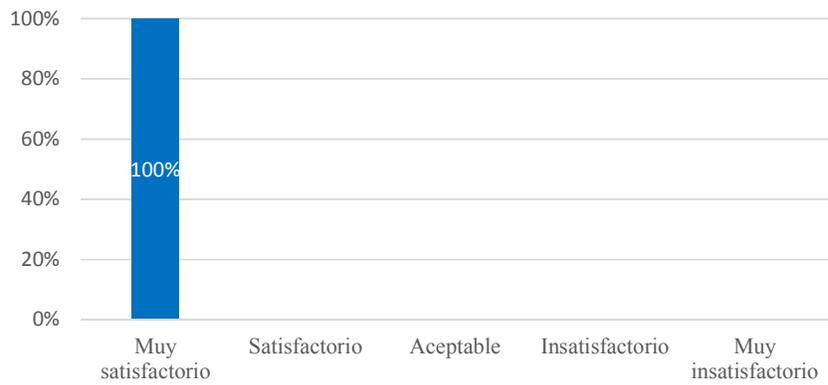


ANEXO VI: RESULTADOS DE LA ENCUESTA DE EVALUACIÓN DE CALIDAD SOBRE LA APLICACIÓN WEB

Resultados basados sobre encuesta a 16 personas:



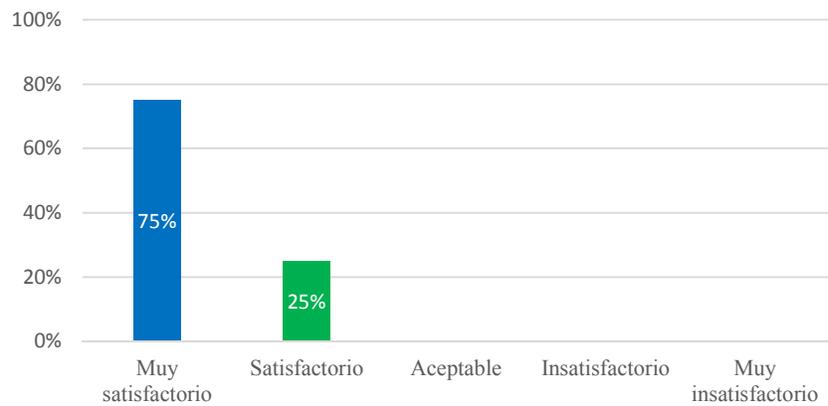
¿La aplicación web permite modificar datos de solicitudes, cotizaciones satisfactoriamente?



¿La aplicación web valida el ingreso de información satisfactoriamente?



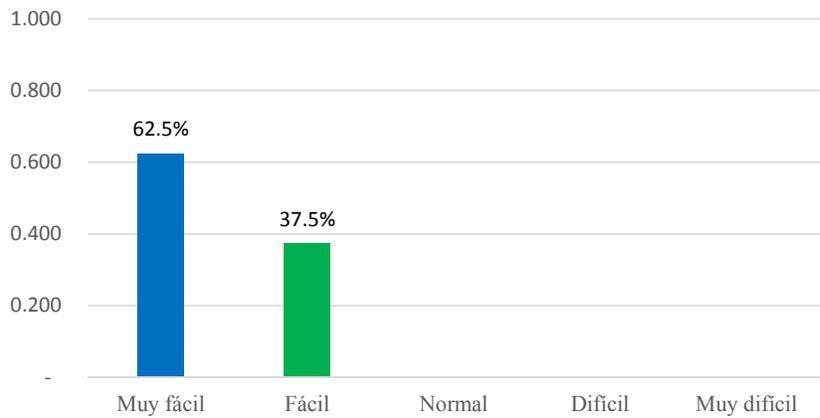
¿La aplicación web permite aprobar cotizaciones, solicitudes satisfactoriamente?



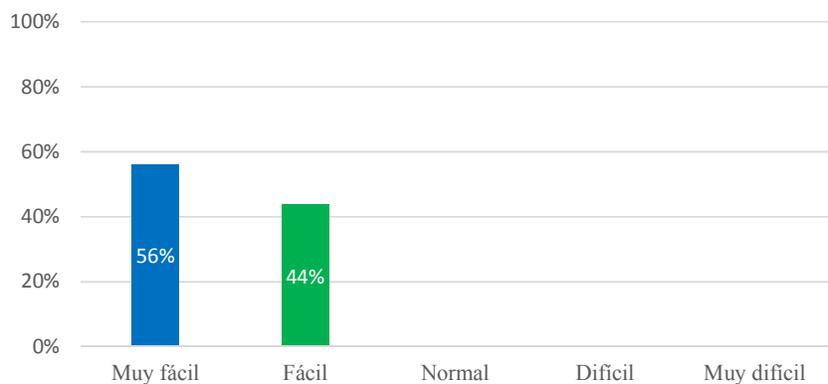
¿La aplicación web permite el login del usuario satisfactoriamente?



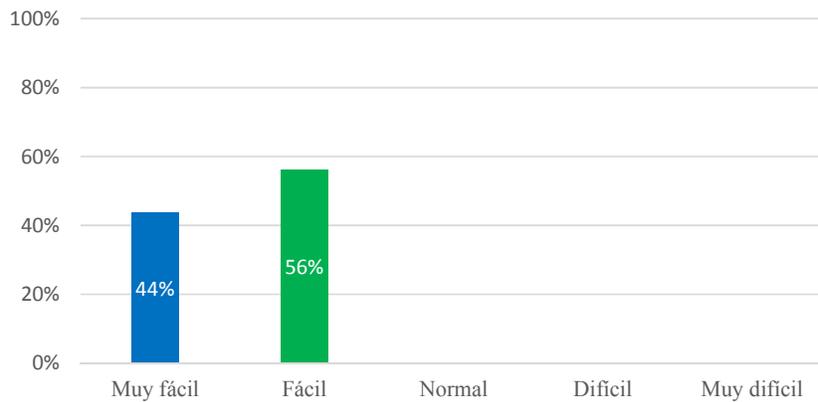
¿Es fácil realizar la creación de clientes en la aplicación web?



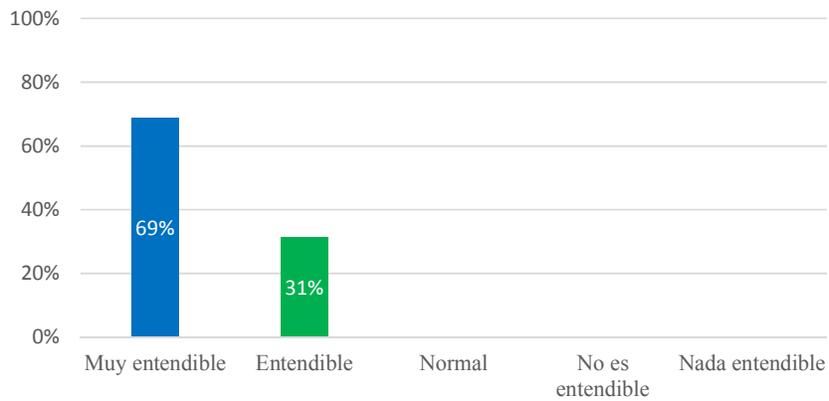
¿Es fácil realizar la creación de solicitudes en la aplicación web?



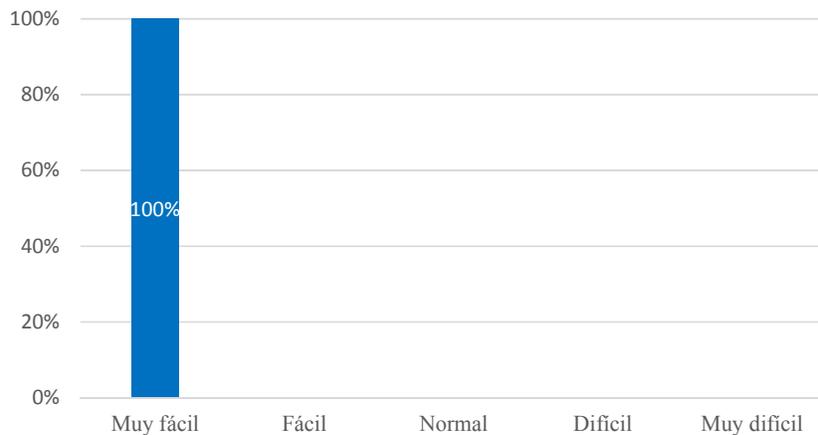
¿Es fácil realizar la creación de cotizaciones en la aplicación web?

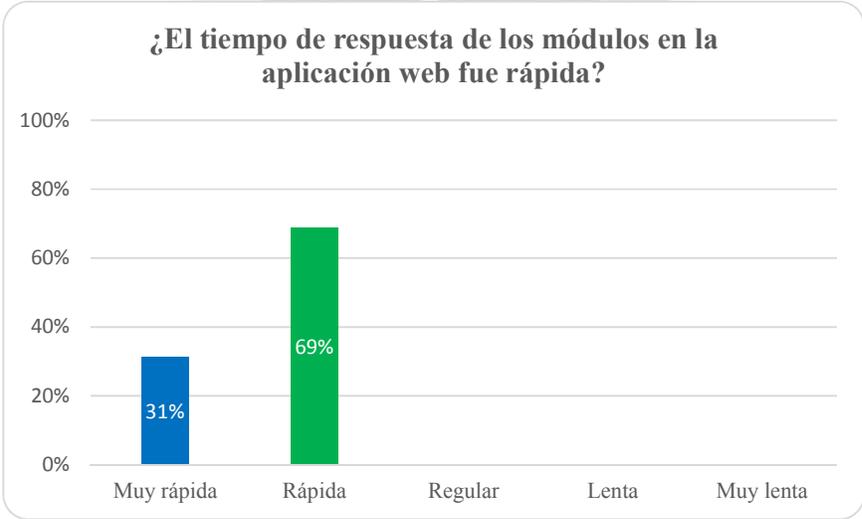
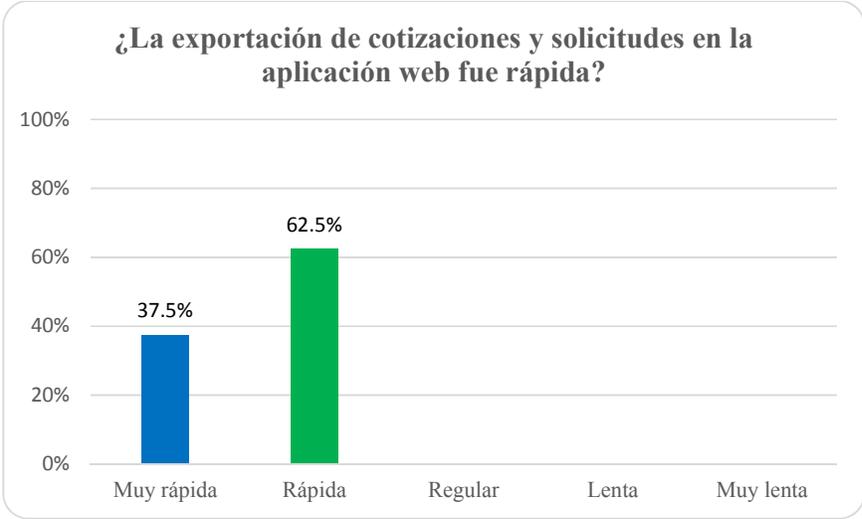
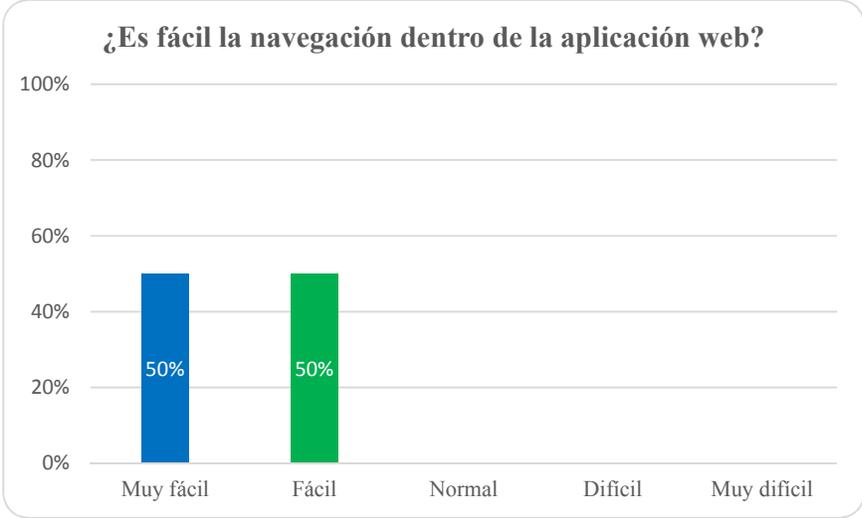


¿El contenido de la aplicación web se muestra entendible para su uso?

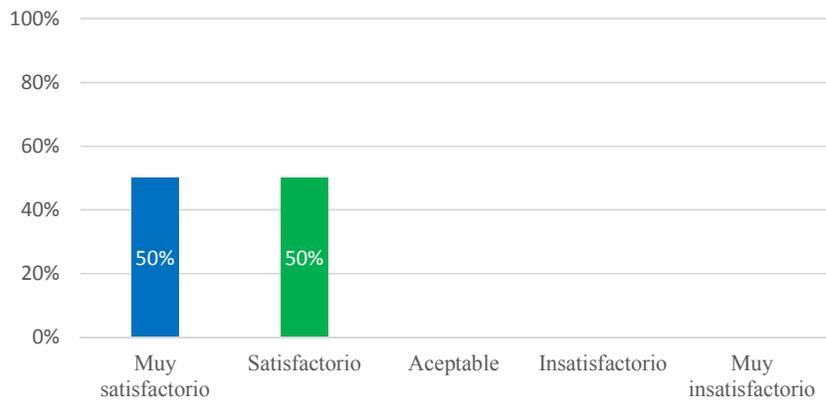


¿Es fácil ingresar a la aplicación web?

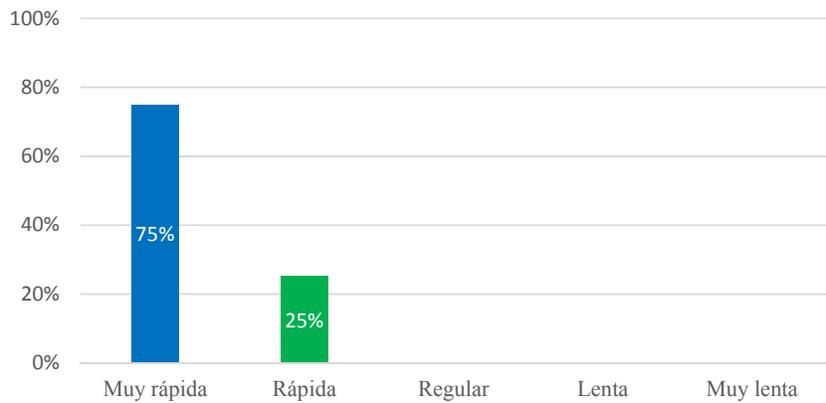




¿Es satisfactorio el tiempo de respuesta de la bandeja de solicitudes?



¿Considera que la búsqueda de clientes y solicitudes es rápida?



ANEXO VII: CÓDIGO FUENTE

Código Fuente GuardarSolicitud
Inicio
<pre> private void mGuardarSolicitud() { SolicitudBLL objBSolicitud = new SolicitudBLL(); SolicitudBE objESolicitud = new SolicitudBE(); Int32 Resultado; String Mensaje = String.Empty; String CodSol = String.Empty; Int32 Resultado2 = 0; String Mensaje2 = String.Empty; try { PintarMensajeError(false, "", 2); objESolicitud.CodFondo = hidFondo.Value; objESolicitud.NumeroComercial = hidNumeroComercial.Value; objESolicitud.CodCliente = txtCodCliente.Text.Trim(); objESolicitud.DescripEmpresa = txtEmpresa.Text.Trim(); objESolicitud.MontoExpoVigente = 0; if (txtExpoVigente.Text.Trim() != "") objESolicitud.MontoExpoVigente = Convert.ToDecimal(txtExpoVigente.Text.Trim()); objESolicitud.MontoExpoPropuesta = Convert.ToDecimal(txtExpoPropuesta.Text.Trim()); objESolicitud.MontoExpoAlcanzar = Convert.ToDecimal(txtExpoAlcanzar.Text.Trim()); objESolicitud.FechaDesem = txtFechaDesem.Text.Trim(); objESolicitud.CodEstadoActivo = cboEstadoActivo.SelectedValue.Trim(); objESolicitud.CodTipoOperacion = cboTipoOperacion.SelectedValue.Trim(); objESolicitud.CodActivo = cboActivo.SelectedValue.Trim(); objESolicitud.DescripActivos = txtDescripActivos.Text.Trim(); objESolicitud.CantActivos = Convert.ToInt32(txtCantActivos.Text.Trim()); objESolicitud.CodMoneda = cboMoneda.SelectedValue.Trim(); objESolicitud.MontoVariacionInver = Convert.ToDecimal(txtInversion.Text.Trim()); objESolicitud.MontoVariacionInverIni = Convert.ToDecimal(txtInversionIni.Text.Trim()); objESolicitud.MontoVariacionInverFin = Convert.ToDecimal(txtInversionFin.Text.Trim()); objESolicitud.PorcVariacionInver = Convert.ToDecimal(hidPorcentaje.Value); objESolicitud.PlazoIni = Convert.ToInt32(txtPlazoIni.Text.Trim()); objESolicitud.PlazoFin = Convert.ToInt32(txtPlazoFin.Text.Trim()); objESolicitud.CodPeriodo = cboPeriodo.SelectedValue.Trim(); objESolicitud.PorcTir = Convert.ToDecimal(txtTirEsperada.Text.Trim()); objESolicitud.PorcTirIni = Convert.ToDecimal(txtTirIni.Text.Trim()); objESolicitud.PorcComisionFin = Convert.ToDecimal(txtComiEstruc.Text.Trim()); objESolicitud.MontoComisionFin = Convert.ToDecimal(txtComiEstrucMonto.Text.Trim()); objESolicitud.PorcCuota = 0; } } </pre>

```

if (txtCuotaPor.Text.Trim()!="")objESolicitud.PorcCuota =
Convert.ToDecimal(txtCuotaPor.Text.Trim());
objESolicitud.PorcCuotaIni = Convert.ToDecimal(txtCuotaPorIni.Text.Trim());

objESolicitud.PorcCuotaFin = 0;

if (Convert.ToDecimal(txtCuotaPor.Text.Trim())>0)objESolicitud.PorcCuotaFin =
Convert.ToDecimal(txtCuotaPorFin.Text.Trim());

objESolicitud.PorcResidual = Convert.ToDecimal(txtValorResi.Text.Trim());
objESolicitud.PorcResidualIni = Convert.ToDecimal(txtValorResiIni.Text.Trim());
objESolicitud.PorcResidualFin = Convert.ToDecimal(txtValorResiFin.Text.Trim());
objESolicitud.Condiciones = txtCondiciones.Text.Trim();

objESolicitud.MontoCuota = Convert.ToDecimal(txtMontoCuotaAprox.Text.Trim());
objESolicitud.MontoCuotaIni = Convert.ToDecimal(txtMontoCuotaAproxIni.Text.Trim());
objESolicitud.MontoCuotaFin = Convert.ToDecimal(txtMontoCuotaAproxFin.Text.Trim());

objESolicitud.MontoResidual = Convert.ToDecimal(txtMontoValResiAprox.Text.Trim());
objESolicitud.MontoResidualIni = Convert.ToDecimal(txtMontoValResiAproxIni.Text.Trim());
objESolicitud.MontoResidualFin = Convert.ToDecimal(txtMontoValResiAproxFin.Text.Trim());

objESolicitud.CodOpcionVenta = cboTipoVenta.Selected.Value.Trim();
objESolicitud.MontoOpcionVenta = Convert.ToDecimal(txtMontoVenta.Text.Trim());
objESolicitud.MontoOpcionVentaIni = Convert.ToDecimal(txtMontoVentaIni.Text.Trim());
objESolicitud.MontoOpcionVentaFin = Convert.ToDecimal(txtMontoVentaFin.Text.Trim());

objESolicitud.CodPromedioMensual = cboPromMensual.Selected.Value.Trim();
objESolicitud.MontoPromedioMensual = Convert.ToDecimal(txtMontoPromMensual.Text.Trim());

objESolicitud.ComentarioExposicionVigente = txtComentarioExpo.Text.Trim();

if (hidAccion.Value == "1")
{
objESolicitud.UsuarioCreacion = NombreUsuario;
Mensaje = objBSolicitud.mInsertarSolicitud(objESolicitud);
}
else
{
if (hidAccion.Value == "3")
{
objESolicitud.NumeroSol = hidCodSolicitud.Value.Trim();
objESolicitud.UsuarioModificacion = NombreUsuario;

Mensaje = objBSolicitud.mModificarSolicitud(objESolicitud);
}
}

Resultado = Convert.ToInt32(Mensaje.Substring(0, 1));
Mensaje = Mensaje.Substring(1, Mensaje.Length - 1);

if (Resultado > 0)
{
if (hidAccion.Value == "1")
{
mListar(0, Convert.ToInt32(hidPaginacion.Value), true);
}
}

```

```

CodSol = objBSolicitud.mSolicitud_Maximo();

Mensaje2 = objBSolicitud.mSolicitud_Insertar_Estado(CodSol,
PARAMETRO_ESTADO_SOLICITUD.REGISTRADO, NombreUsuario, String.Empty);
Resultado2 = Convert.ToInt32(Mensaje2.Substring(0, 1));
Mensaje2 = Mensaje2.Substring(1, Mensaje2.Length - 1);

mPintarGrillaSolicitud(CodSol);
PintarMensajeTransaccion(true, Mensaje + " N°Solicitud: " + CodSol , 1, Resultado);
}
else {

if (hidAccion.Value == "3")
{
mListar(Convert.ToInt32(hidPageIndex.Value), Convert.ToInt32(hidPaginacion.Value), true);
mPintarGrillaSolicitud(txtNumeroSol.Text.Trim());
PintarMensajeTransaccion(true, Mensaje, 1, Resultado);
}

}

TabContenedor.ActiveTabIndex = 0;
this.TabContenedor.Tabs[0].Visible = true;
this.TabContenedor.Tabs[1].Visible = false;
this.TabContenedor.Tabs[2].Visible = false;

}
else {

PintarMensajeTransaccion(true, Mensaje, 2, Resultado);

}

}
catch (Exception lexcError)
{
PintarMensajeError(true, lexcError.Message.ToString(), 2);
}
finally
{
objBSolicitud = null;
objESolicitud = null;
}
}
}

```

Fin