



UNIVERSIDAD DE GUAYAQUIL
Facultad de Ciencias Matemáticas y Físicas
Carrera de Ingeniería en Sistemas
Computacionales

“Aplicación que Almacene la Información Crediticia de todas las Entidades que los otorgan, obteniendo una Base de Datos del Historial de las Personas, para poder brindar el Servicio con esta Información para que sirva para la Toma de Decisiones de Crédito Futuras en Línea”

TOMO I

SÉPTIMO CURSO DE GRADUACIÓN

Previo a la Obtención del Título de:

INGENIERO EN SISTEMAS COMPUTACIONALES

Autores:

Medina Murillo Denisse Vanessa

Román Rosales Jayron Stalin

Vera Quiróz Darwin Vicente

GUAYAQUIL-ECUADOR

Año: 2010

Lomo de la monografía

UG
CISC
7mo. C.G.

BURÓ DE CRÉDITO

**Medina Murillo
Denisse Vanessa**

**Román Rosales
Jayron Stalin**

**Vera Quiróz
Darwin Vicente**

**TOMO
I**

AGRADECIMIENTO

Damos gracias a Dios por habernos permitido alcanzar la meta profesional que nos propusimos.

A nuestros padres que con amor y sacrificio acompañaron cada paso de nuestras vidas estudiantiles y nos supieron conducir por el camino de los grandes ideales.

A nuestros amigos más cercanos que nos dieron todo su apoyo de manera incondicional.

A los profesores y compañeros que han iluminado y compartido cada uno de los rincones de nuestras etapas de estudios.

DEDICATORIA

Dedicamos este trabajo, que simboliza la culminación de una jornada más de nuestras vidas, primero a Dios que nos ha guiado por el sendero del bien y está con nosotros en este triunfo, y nos acompañara siempre.

A nuestros padres, hermanos y a todos quienes quieran compartir y han compartido con nosotros las experiencias de esta etapa profesional.

TRIBUNAL DE GRADUACIÓN

Presidente

1er. Vocal

2do. Vocal

Secretario

DECLARACIÓN EXPRESA

“La autoría de la tesis de grado corresponde exclusivamente a los suscritos, perteneciendo a la Universidad de Guayaquil los derechos que generen la aplicación de la misma”

(Reglamento de Graduación de la Carrera de Ingeniería en sistemas Computacionales, Art. 26)

Medina Murillo Dennisse Vanessa

Román Rosales Jayron Stalin

Vera Quiróz Darwin Vicente

RESUMEN

Infocredito es una aplicación desarrollada para ayudar a empresas emisoras de crédito en la toma de decisiones al momento de emitir un crédito. Está diseñada en dos ambientes, un ambiente intranet para la administración de su operación y un ambiente web para el acceso de afiliados; Se utiliza Oracle como Servidor de base de datos, OC4J como servidor web además del Suite de herramientas de desarrollo de Oracle (Forms Builder, Reports Builder y Jdeveloper) como herramientas de programación.

Para acceder al servicio es necesario un usuario y su respectiva clave de acceso a través del sitio web de Infocredito, además del respectivo certificado digital que se entregará al cliente al momento de iniciar la relación contractual con el Buró.

Los afiliados a Infocredito podrán encontrar en el sistema la información necesaria para su decisión en formatos html y pdf.

INDICE GENERAL

AGRADECIMIENTO	II
DEDICATORIA	III
TRIBUNAL DE GRADUACIÓN	IV
DECLARACIÓN EXPRESA	V
RESUMEN	VI
INDICE GENERAL	VIII

CAPÍTULO 1

1. INTRODUCCIÓN

1.1. Antecedentes	1
1.2. Generalidades	2
1.3. Problemática	5
1.4. Solución a la Problemática	6
1.5. Perspectivas	7
1.6. Visión	11
1.7. Misión	12
1.8. Beneficiarios	12
1.9. Objetivo General del Proyecto	13
1.10. Objetivos Específicos del Proyecto	13

1.11. Metas y Propósitos	15
1.12. Alcances	18
1.13. Arquitectura	21
1.14. Metodología	23
1.14.1. Base de Datos	26
1.14.2. Levantamiento de Información	26
1.14.3. Administración (Intranet)	28
1.14.4. Portal de Servicios (Internet)	30
1.15. Recursos para el desarrollo	30
1.16. Cronograma de Actividades	32

CAPÍTULO 2

2. ANÁLISIS

2.1. Levantamiento de Información	33
2.2. Investigación por Internet	39
2.3. Investigación de Mercado	40
2.3.1. Encuestas	40
2.3.2. Análisis de los Resultados	44
2.4. Análisis de Requerimientos	46
2.5. Análisis de Riesgos	48
2.6. Diagrama Entidad Relación	49
2.6.1. Análisis y descripción	50

2.6.2.	Relación entre Tablas	55
2.7.	Diagrama de Flujo de Procesos	61
2.7.1.	Proceso de Administración	61
2.7.2.	Portal de Servicios	62
2.8.	Diagrama de Clases	64
2.8.1.	Análisis y Descripción	65
2.9.	Casos de Uso	68

CAPÍTULO 3

4. CODIFICACIÓN

4.1.	Codificación de Procesos Principales (PL-SQL)	95
4.2.	Codificación de Procesos Principales (JDEVELOPER)	157

ANEXOS 170

ANEXO 1	(Ley de Buró de crédito)	171
ANEXO 2	(Formato de Archivo de carga)	178
ANEXO 3	(Cronograma de Actividades)	179
ANEXO 4	(Diagrama de Entidad-Relación)	180
ANEXO 5	(D.F Usuario Administrador)	181
ANEXO 6	(D.F Usuario Operador)	182
ANEXO 7	(DF Público en General)	183
ANEXO 8	(DF Usuario Afiliado)	184

ANEXO 9(Diagrama de Clases)	64
-----------------------------	----

FIGURAS

FIGURA 1.1 (Buró de créditos en el Ecuador)	11
FIGURA 1.2 (Arquitectura)	21
FIGURA 1.3 (Arquitectura 3 capas)	23
FIGURA 1.4 (Metodología)	25
FIGURA 1.5 (Fuentes de información)	27
FIGURA 1.6 (Archivos de carga)	27
FIGURA 2.1	43
FIGURA 2.2	43
FIGURA 2.3 (Caso de uso 1)	70
FIGURA 2.4 (Caso de uso 2)	71
FIGURA 2.5 (Caso de uso 3)	89

BIBLIOGRAFÍA	185
---------------------	------------

CAPÍTULO 1

INTRODUCCION

1.1. Antecedentes

El sistema financiero de todo país en el mundo se rige por la situación política de este, Ecuador no es una excepción y dado que su situación política es bastante volátil el sistema financiero ecuatoriano tiene las mismas características. Viéndose afectado en los últimos años principalmente por las implicaciones económicas que se generaron a raíz del surgimiento izquierdista en Sudamérica que originó la disminución en la relación comercial de Ecuador con Colombia; la desconfianza que se generó por el anuncio del no pago de la deuda externa, la baja drástica en el precio del petróleo y sobre todo la baja de remesas de emigrantes originada por la crisis mundial.

Sin embargo las empresas que forman parte del sector financiero ecuatoriano han buscado métodos para fortalecer su estructura económica. A pesar de la

presente crisis mundial en el Ecuador las instituciones financieras han desarrollado líneas de crédito de forma tal, que puedan acceder la mayor cantidad de beneficiarios posibles.

Es así que por ejemplo Unibanco a través de su producto cuota fácil ha llegado a tener 260.000 clientes, Banco Territorial fusionó su sistema crediticio con Casa Tosi para llegar a tener 1.000 clientes y así como ellos cientos de empresas en Ecuador han abierto sus líneas de crédito con la finalidad de incrementar sus ventas.

1.2. Generalidades

Hoy en día existe una gran necesidad en las grandes entidades comerciales para elegir con precisión y confiabilidad a qué persona otorgar una línea de crédito. Son cada vez más las empresas que deciden lanzar al mercado su propia tarjeta de crédito, por citar algunos ejemplos tenemos a La Ganga con su tarjeta GangaCard, De Prati con su tarjeta De Prati, Casa Tosi con CrediTosi, Pycca con Credipycca, Créditos Económicos con su Credicard, entre otras.

Esto se evidencia debido a que estas empresas han puesto sus ojos a un nuevo mercado al cual no estaban enfocadas antes: el mercado de la clase económica media baja y baja, que se les dificultaba acceder a créditos.

Se sabe que el mercado de la clase media, media alta y alta ya está casi saturado, así que la única alternativa de crecimiento son los sectores antes mencionados; esta tendencia se puede ver en los bancos como el Pichincha, Unibanco, Banco Solidario, Banco Territorial, etc. Que han lanzado tarjetas y líneas de crédito enfocadas al sector bajo, como ejemplo Banco solidario con los créditos de "La Olla de Oro" y "Crédito al Paso"; Unibanco con Tarjeta "Cuota Facil" e "Impulso Microcrédito al paso", Supermaxi con su tarjeta "Supermaxi", Banco Pichincha con la tarjeta "Visa Experta", Banco Territorial con la tarjeta Creditosi, Etafashion con su tarjeta "Planeta", etc. para citar algunos ejemplos de productos que son entregados sin muchos requerimientos.

Pero la lista de las empresas que no tienen su propia tarjeta de crédito y que sí ofrecen como alternativa de pago el crédito directo es superior, y no solo son instituciones bancarias y financieras, pues dentro de las empresas otorgantes de crédito suelen estar también casas comerciales, consorcios, concesionarios de automóviles, personas jurídicas y naturales. Algunas son empresas grandes y

otras pequeñas que lo que buscan es captar mas clientes e incrementar sus ventas ofreciendo esta alternativa; y como consecuencia de esta estrategia asumen también un gran riesgo.

Sin embargo todas estas empresas comerciales cuentan con métodos tradicionales de investigación o técnicas aisladas para poder inferir de alguna manera el perfil crediticio de un potencial sujeto de crédito, estas técnicas suelen ser: solicitar documentos personales, referencias escritas o solicitar referencias vía telefónica. Pero ninguno de estos métodos les dice con certeza cuál es el perfil crediticio al momento de tomar la decisión de otorgar o no un crédito.

Lo que las empresas y cadenas comerciales buscan es saber con precisión que tan buenos clientes y que tan cumplidores son en sus pagos de crédito obtenidos anteriormente, y así reducir significativamente los riesgos, los costos de cobranzas y las pérdidas por incobrables.

1.3. Problemática

Pero el éxito de la apertura de las líneas de crédito en las instituciones crediticias depende de la selección del destino de dichos créditos que se resumen en los siguientes puntos:

- Saber SI los solicitantes de créditos están en condición de responder a la responsabilidad que implica la obtención de dicha obligación.
- Conocer si en el pasado han cumplido con este tipo de obligaciones adquiridas.
- Si el solicitante ha tenido un buen comportamiento de pago, conocer su nivel de endeudamiento y analizar si es beneficioso otorgarle un mayor monto de crédito. U otorgarle un crédito adicional a los que tiene actualmente.
- En caso de los nuevos solicitantes, es necesario conocer si su situación económica actual lo habilita o no para ser receptor de un crédito.
- Analizar si a pesar de su mal comportamiento de pago sería oportuno otorgarle un monto de crédito menor al que sé le otorgo antes.

Para responder estas interrogantes será necesario:

- Tener información obtenida de diversos puntos de origen.

- Estandarizar la información que servirá para que se haga análisis.
- Combinar resultados de los diversos análisis realizados.
- Aplicar formulas de cálculos correctas

1.4. Solución a la problemática

La solución que se propone es la elaboración de un sistema de información crediticia (Buró de Crédito) que tiene como fin recopilar, manejar, procesar y entregar información relativa a operaciones de crédito que conforman el historial crediticio de personas o empresas.

Este servicio se brinda a las empresas afiliadas por medio de un portal web objetivo, confiable y de fácil manejo, que brinda información de los titulares (deudores), simulación de cálculos correspondiente a créditos, reportes, etc. Por medio de esta herramienta se puede consultar el historial crediticio de una persona o compañía ingresando su número de identificación, sea cédula o Ruc; el portal presenta la información más relevante en cuanto los créditos obtenidos, apegándose a la confidencialidad y respeto a la privacidad; así como también brindando información actualizada y de calidad. Esta información es de suma importancia al momento de tomar la decisión de otorgar un crédito o no;

complementándose con la opción de generar reportes resumidos o detallados. El Buró de Crédito únicamente recibe datos de los acreedores, los procesa con el fin de generar los reportes (sin modificar datos) y muestra la información con respecto del comportamiento crediticio de la persona, la cual es parte de la información que la empresa comercial e institución financiera incluye para su análisis de otorgamiento de crédito, es decir, el Buró actúa como un sistema de información para el apoyo a toma de decisiones.

El Buró se actualiza constantemente con información proveniente de clientes afiliados quienes tienen la obligación de reportar créditos nuevos y pagos realizados.

1.5. Perspectivas ¿Qué? ¿Quién? ¿Cómo? ¿Dónde?

¿Qué es el Buró de Crédito?

Es una Sociedad de Información Crediticia que tiene como fin recopilar, manejar, procesar y entregar información relativa a operaciones de crédito que conforman el historial crediticio de personas físicas y morales, incluye los créditos pagados y no pagados que una persona tenga tanto con empresas comerciales como con instituciones financieras.

¿Qué es el reporte de crédito?

Es la información que se proporciona sobre quienes realizan actividades de crédito. También incorpora con cuántas y en cuáles empresas o instituciones financieras ha realizado operaciones de crédito y cómo ha sido su comportamiento.

Los Reportes incluyen también un histórico de instituciones que han consultado su historial crediticio.

¿Quién alimenta la base de datos del Buró de Crédito?

Todas las instituciones del sistema financiero están obligadas a suministrar la información de sus créditos otorgados a la Superintendencia de Bancos y Seguros (SBS) en la forma y frecuencia que ella determine. La información indicará también el comportamiento del crédito de las personas y compañías. La SBS recibe la información y la valida pero no modifica ningún dato, luego la entrega a los buró a fin de que ellos presten el servicio de referencias crediticias. La información se va actualizando periódicamente conforme las instituciones financieras envíen los reportes mensuales a la SBS y está a los Burós

¿Quién decide si dar un crédito o no?

La empresa comercial e institución financiera que vaya a otorgarlo decide si da el crédito o no. El Buró de Crédito únicamente le entrega la información que se encuentra en su base de datos respecto del comportamiento crediticio de la persona, la cual es parte de la información que la empresa comercial e institución financiera incluye para su análisis de otorgamiento de crédito.

¿Dónde se puede consultar un reporte de crédito?

El interesado del reporte deberá acercarse a las oficinas de uno de los afiliados al servicio de Buró. También puede contactarse directamente con el Buró de Crédito mediante la opción de “Contáctenos”, o enviando sus datos al mail para ser atendidos posteriormente. El reporte crediticio de la misma persona que lo solicita no tendrá ningún costo, sin embargo los certificados si lo tienen. En el caso de este Buró el valor por consulta será de \$5,00.

¿Cómo modificar un reporte?

Presentando una reclamación ante el Buró de Crédito por correo electrónico o acercándose a las oficinas del Buró, para así brindar una rápida atención a las solicitudes. Si la reclamación fue procedente, se realizará el cambio en el

reporte, sin embargo, en caso de que no proceda o sólo proceda parcialmente, se hará la anotación correspondiente.

¿Qué es el Derecho al Olvido?

Es el derecho que tienen las personas físicas de que se les borre su historial crediticio bueno o malo después de tres años. Legalmente los Sistemas de Buró de Crédito en Ecuador sólo pueden almacenar información de los tres últimos años, según la ley de burós.

¿Qué Burós de crédito autorizados existen actualmente en ECUADOR?

Existen cuatro burós de crédito autorizados en el Ecuador:

- MULTIBURÓ
- CALTEC
- CREDIT REPORT
- ACREDITA



Figura 1.1: Burós de Crédito existentes en Ecuador

1.6. Visión

Ser un portal que aporte para el mejoramiento y crecimiento del sector crediticio ofreciendo a los clientes un servicio optimo de calidad en el análisis de riesgos crediticios y que la compañías afiliadas reconozcan y evidencien los beneficios de la información que brindamos reflejados en sus cobranzas.

1.7. Misión

Entregar un sistema informático que cumpla con los requerimientos del sector crediticio brindando información clara, veraz concisa y útil para los clientes traduciéndose esto en reducir significativamente los riesgos, los costos de cobranzas y las pérdidas por cartera no recuperada.

1.8. Beneficiarios

Al país: Democratiza el crédito, educa y promueve sanas costumbres del crédito, y disminuye el riesgo a través información depurada, exacta, veraz y oportuna.

Al Ciudadano: Facilita el acceso a crédito reduciendo los trámites para la colocación de créditos y haciendo público el comportamiento crediticio histórico de las personas.

A las Empresas que otorgan crédito: Suministra una herramienta objetiva para el análisis de riesgo haciendo más eficiente y ágil la asignación del crédito a un menor costo y riesgo, disminución de los niveles de morosidad al tener

mayor cantidad de información al momento de decidir sobre un crédito, acceso a una gran cantidad de información de referencias crediticias.

1.9. Objetivo General del Proyecto

Desarrollar una herramienta que sirva para brindar información sencilla, rápida, confiable y transparente de historiales de créditos personales o empresariales con el objetivo que dicha información sirva de apoyo a la toma de decisiones para otorgar créditos.

1.10. Objetivos Específicos

- Desarrollar un sistema de información que interactúe con el sistema financiero y comercial para proporcionar información sobre personas (naturales o jurídicas) y su capacidad de recibir créditos.
- Proporcionar información sobre créditos que se hubiesen otorgado y el comportamiento de pago del crédito usado por la persona o empresa.
- Proporcionar información fidedigna, alimentando la base de datos con información de instituciones financieras, comerciales y públicas otorgantes de créditos. La Superintendencia de Bancos y Seguros es

quien recopila y valida la información, luego la almacena en su Central de Riesgo y envía a los Burós la información para ser actualizada.

- Cumplir con el correspondiente marco legal, en este caso el establecido por la Superintendencia de Bancos y Seguros (SBS). (ANEXO 1).
- Ofrecer un servicio disponible las 24 horas del día, 365 días del año.
- Proveer reportes de créditos personales o empresariales, con la finalidad de reducir el riesgo de créditos, siendo estos reportes un apoyo a la toma de decisiones.
- Diseñar un sistema seguro para evitar el mal uso de la información y asegurar la integridad de los datos.

1.11. Metas y Propósitos

- **Objeto de interés público.-**

Presentar información transparente y objetiva que indique el cumplimiento o incumplimiento de las obligaciones crediticias; contribuyendo al desarrollo de las actividades comerciales dentro del país, además de fomentar la cultura de crédito y los principios que orientan el ejercicio responsable de su actividad.

- **Respeto por la Privacidad.-**

Preservar la intimidad de las personas. En ningún caso acceder a datos sensibles sobre la privacidad de las personas, tales como religión, raza, ideología o cualquier otro dato reservado, cuyo tratamiento pueda generar una discriminación fuera del ámbito que delimita el objeto de nuestro servicio.

- **Pertinencia.-**

Recabar la información disponible en tanto resulte pertinente, útil o necesaria para la realización de operaciones crediticias en general, y prescinde de toda

otra información que no esté directamente relacionada con el objeto de su actividad.

- **Objetividad**

Presentar información objetiva, a diferencia de la necesaria subjetividad de la decisión crediticia del otorgante, que depende de su interpretación frente a la información proporcionada según sus propios parámetros.

- **Identidad de la fuente**

Garantizar la transparencia de las fuentes de información. Recopilar información de diversas fuentes, identificando la misma en todos los casos, para luego someterla a un proceso de validación y ordenamiento, con el fin de generar servicios que se conviertan en soluciones.

- **Calidad de los datos**

La información recaudada será sometida a procesos de validación, donde se descartan aquellos datos en los que se adviertan incongruencias con la realidad, para así maximizar la calidad de datos incorporados.

- **Buen uso de la información**

Asegurar el buen uso de la información. Obligar a los afiliados a que cada reporte solicitado sea destinado para su uso reservado, exclusivo y confidencial para la evaluación de una posible transacción crediticia dentro de su organización; haciéndose responsables directos en caso de utilización inadecuada de dicha información.

- **Actualización**

Actualización mensual de la información.

- **Confidencialidad**

Mantener la confidencialidad de la información de la base de datos, con respecto a personas no autorizadas, e incluso las empresas afiliadas deben tener una carta de autorización antes de hacer una consulta.

- **Seguridad**

Proteger que la información presentada sea autentica y mantener la confidencialidad de los datos proporcionados; además evitar cualquier tipo de alteración y sustracción por parte de terceros

1.12. Alcances

- Crear una aplicación que almacene información crediticia de personas naturales o jurídicas y que permita que los sujetos autorizados accedan a ella con el fin de tomar decisiones de crédito futuras en línea.
- La herramienta será accesible mediante un sitio Web, presentado a modo de un portal, lo cual permitirá el fácil acceso desde cualquier lugar y será presentada de manera amigable.
- El manejo de la base de datos y servicio será administrado por medio de una aplicación disponible por intranet, donde existirán dos tipos de usuario: Administrador y Operador, cada tipo con su usuario, contraseña y privilegios.
- La información crediticia será recibida al final de cada mes en un CD o por vía electrónica con uno o varios archivos planos con formato predeterminado (ANEXO 2) que serán cargados a la base de datos.
- El portal tendrá un proceso de inicio de sesión. Este proceso de seguimiento permitirá o denegará el acceso a la aplicación. Sólo podrán

tener acceso los sujetos **afiliados** al servicio por medio de un usuario y contraseña que deben ser autorizados y autenticados por la administración del buró.

- La información será procesada, no modificada y se ofrecerán tres tipos de reportes (general, resumido y detallado), con gráficas, estadísticas, proyecciones, etc. Según el tipo de reporte.
- Los reportes se generarán en un archivo PDF. o HTML (Formato de documento portable) para ser impresos o guardados.

Los afiliados tendrán una opción de mensajería, exclusiva para comunicación con el Buró. Se enviará notificaciones, alertas, dudas, requerimientos, etc. Para esta función utilizaremos un aplicativo llamado sendemail.

- Consultas por unidad: digitando el documento de identificación del sujeto (cedula, RUC, pasaporte).
- El servicio de información crediticia se brinda previa firma de un contrato. Se presentará alertas a los afiliados, un mes antes de la terminación de su contrato (trimestral, semestral o anual), para renovación del mismo.

- Historial de consultas: el afiliado podrá revisar las 30 últimas consultas efectuadas, el administrador y operador podrán revisar las consultas realizadas en el último mes.
- Para lograr mayor seguridad debido a que se maneja información confidencial, se crearán puertos seguros http con certificados SSL, con algoritmo de encriptación RSA.

El Portal ofrece otros servicios para los cuales no se necesita ser afiliado como:

- Noticias sobre el sector financiero, tasas de interés actuales, boletines de la SBS, información bancaria, etc.
- Herramientas de cálculo como: cálculo de interés; valor de cuota a pagar según monto, interés, y plazo; cálculo de monto máximo a solicitar en créditos con datos como ingresos, gastos, etc. Estas herramientas serán solo para aclarar dudas de público en general.

- Opción de contáctenos: para personas que quieran solicitar su reporte crediticio, para reclamos en caso de información errónea, para afiliaciones, etc. Esta opción responderá las inquietudes vía mail.

1.13. Arquitectura

El diseño arquitectónico a utilizar en este proyecto será de tres capas. En esta arquitectura, según las decisiones de diseño que se tomen, se puede balancear la carga de trabajo entre el proceso cliente y el proceso correspondiente al servidor de aplicación.

El esquema utilizado es el siguiente:

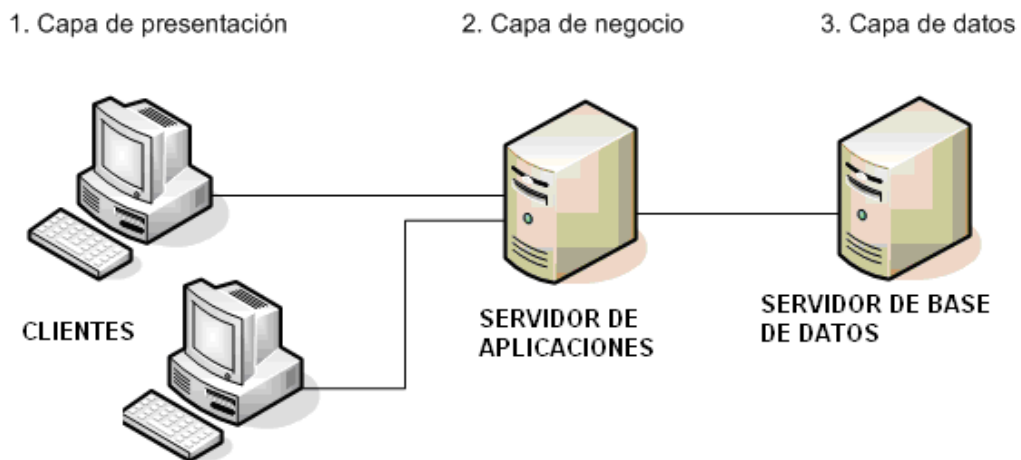


Figura 1.2: Arquitectura

Capa de presentación:

Es la interfaz de usuarios (usuario afiliado) que van a interactuar con el portal por medio de un navegador de Internet (Internet Explorer, Mozilla Firefox). La aplicación funciona con cualquier explorador existente en el mercado, para mayor comodidad de los clientes, en el desarrollo de la aplicación considera los navegadores para sistemas operativos Windows.

Capa de Negocio:

Es la capa intermedia que servirá para la comunicación entre la composición visual y los datos almacenados en la base de datos.

El servidor recibe requerimientos del cliente (http) y envía respuestas (html).

Es el servidor, dependiendo del tipo de solicitud, es quien accede y se conecta con la base de datos. Este se encarga de compilar los jsp, servlet y demás clases a utilizar. El Servidor utilizado es el OC4J, que es el acrónimo de **O**racle **A**pplication Server **C**ontainers for **(4)** **J**2EE Es decir, el servidor de aplicaciones J2EE de oracle.

Capa de datos:

Base de datos donde se encuentra almacenada la información, base de datos creada en Oracle 10g Enterprise Edition. Toda la información crediticia que se

ha recibido por parte de la SBS o de la central de riesgo, información de afiliados, usuarios, etc.

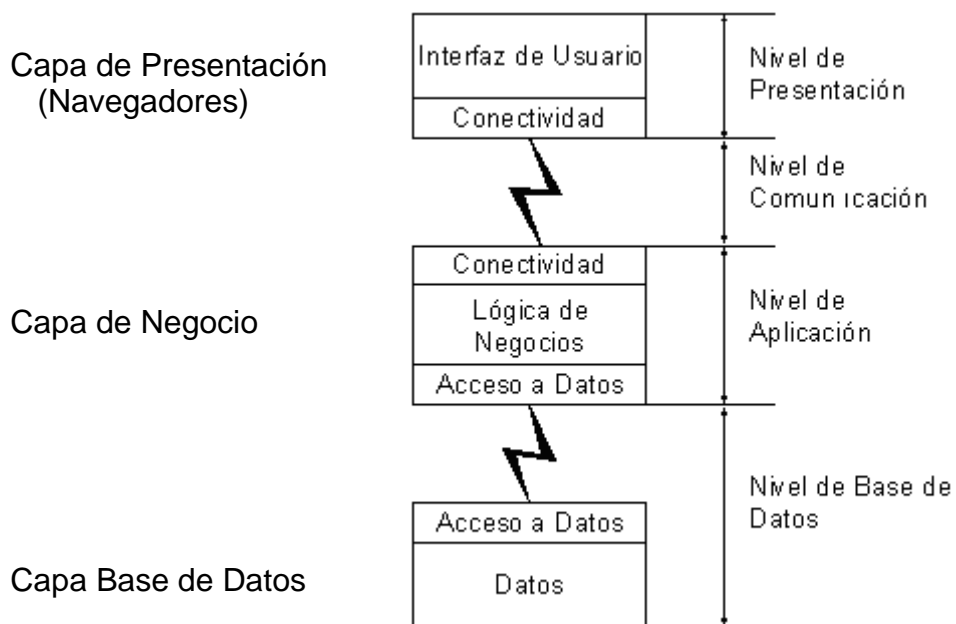


Figura 1.3: Arquitectura tres capas

1.14. Metodología

La metodología a utilizar será Orientado a Objetos con UML basado en ingeniería Web, y tomando en consideración la información que se tiene para desarrollar el proyecto, los requerimientos y alcances definidos se eligió **el modelo en espiral**, ya que este modelo esta orientado a riesgos que se pueden

presentar en el camino, además este ciclo permite que dividamos el proyecto en mini proyectos en donde se irá percibiendo el avance de este sistema crediticio, partiendo de una escala pequeña a una visión general de los procesos o servicios a brindar.

El modelo de ciclo de vida espiral para este proyecto, que se define como un proyecto de desarrollo rápido (5 meses) resulta excelente y ayuda a disminuir retrasos en caso que se hayan identificado mal los requerimientos y alcances: permite la gestión de riesgos, evaluando cada alternativa de solución posible para poder pasar al siguiente nivel o iteración. Además la elección del ciclo de vida se basa en que el modelo espiral es flexible a las modificaciones a medio camino que con seguridad se presentarán en el desarrollo de la herramienta ya que en su momento se querrá brindar reportes mas completos y de fácil entendimiento, otros datos o tipos de reportes, herramientas que ayuden a tomar la decisión de otorgar un crédito o no, y muchas otras modificaciones que se podrían presentar a futuro y que actualmente no se podrían mencionar; en fin ampliar el concepto inicial que hemos definido del software, ya sea para lograr ser un producto más atractivo a las sujetos otorgantes de crédito, y tener mejor posicionamiento y competencia en el mercado.

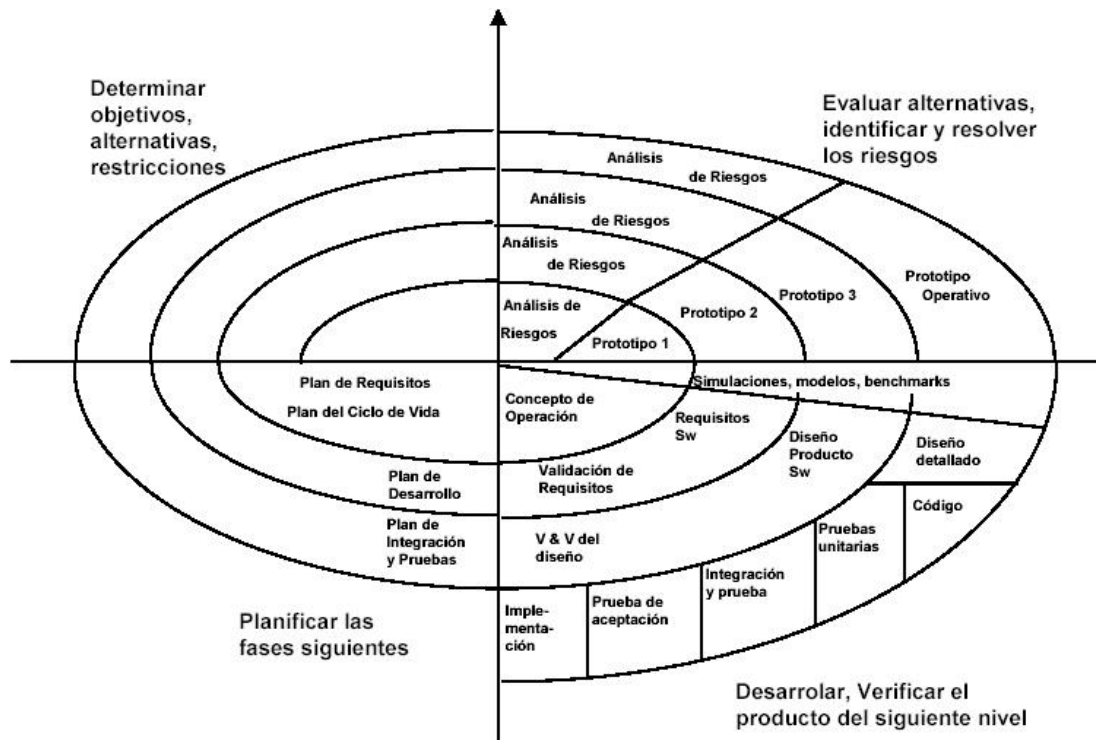


Figura 1.4: Metodología: Modelo en Espiral

El modelo de desarrollo a mas de ser de ciclo de vida en espiral, supone también un prototipado evolutivo, empezando a diseñar e implementar las partes mas importantes de un sistema de información crediticio, que son la recolección y actualización de información, y la generación de reporte que sirvan de ayuda a tomar una decisión, disminuyendo riesgos de pérdidas a nuestros afiliados; para luego ampliar y refinar el prototipo hasta que se termine satisfactoriamente; tomando en consideración que este proceso se llevará a cabo mientras el factor tiempo lo permita.

A continuación se detalla brevemente las iteraciones definidas inicialmente y algunos de sus requerimientos y alternativas de solución

1.14.1. Iteración 1: Base de Datos

1. Modelo lógico de base de datos
 - a. Establecer que información será requerida para el servicio de información crediticia
 - b. Definir tablas, campos y tipos de datos
 - c. Definir relación de datos entre tablas
2. Modelo físico de base de datos
 - a. Creación de tablas
 - b. Restricción de datos y relación entre tablas (constraints)

1.14.2 Iteración 2: Levantamiento de Información

La información contenida en la base de datos para ser ofrecida a los usuarios de Buró provendrá del organismo encargado de regular la banca pública, bancos, sociedades financieras, mutualistas, cooperativas, empresas de seguros privados y seguridad social, en este caso la *Superintendencia de Bancos y Seguros (SBS)*. Estas instituciones alimentan la base de datos de la SBS (Central de Riesgo), además también contiene la información de Riesgo comercial que almacena información de personas naturales o comerciantes que

por sus propios medios denuncian la morosidad de personas naturales o jurídicas con pruebas, como ejemplo la entrega de cheques sin fondos.

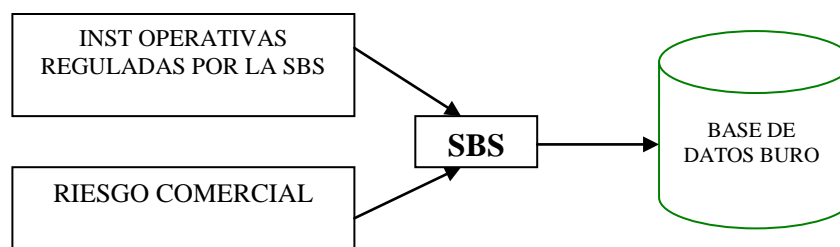


Figura 1.5: Fuentes de Información

Requerimientos

Para el levantamiento de información será necesario definir lo siguiente

1. Datos a recibir por parte de la SBS
2. Archivos aceptados para la carga de información y formato de la información contenida en cada archivo. **(ANEXO 2)**



Figura 1.6: Archivos de Carga

3. Carga de uno o varios archivos a la vez.

4. Carga de archivos y extracción automática de información hacia la base de datos del buró.

1.14.3. Iteración 3: Administración de Buró (Intranet)

La administración de los datos del buró se realizará por medio de una aplicación en intranet, y existirán dos tipos de usuario con acceso a esta: usuario administrador y usuario operador.

USUARIO ADMINISTRADOR

1. Creación de usuarios (operador o afiliado), generación de contraseñas.
2. Privilegios de administrador
 - a. Ingreso y monitoreo de afiliados.
 - b. Generación de claves para nuevos afiliados, o por pérdida.
 - c. Consulta de información crediticia contenida en la base de datos.
 - d. Bloqueo/desbloqueo de usuarios operador o afiliados según el caso. Recibirá alertas en caso de darse uno de estos casos.
 - e. Actualización de datos de usuarios afiliados, administradores y operadores.
 - f. Carga de información crediticia contenida en archivos enviados por la SBS.

3. Recibirá alertas trimestrales para la información que supere los 4 años de almacenamiento, será respaldada en una base de datos externa o desechada.

USUARIO OPERADOR

1. Creación de usuarios afiliados, generación de contraseña y asignación de privilegios
2. Privilegios de Operador
 - a. Ingreso y monitoreo de afiliados.
 - b. Generación de claves para nuevos afiliados o por pérdida.
 - c. Consulta de información crediticia contenida en la base de datos tendrá acceso a la información desde 4 años atrás.
 - d. Bloqueo/Desbloqueo de usuarios afiliados según el caso. Recibirá alertas.
 - e. Actualización de datos de usuarios afiliados.
 - f. Carga de información crediticia, contenida en archivos enviados por la SBS.
4. Recibirá alertas trimestrales para la información que supere los 4 años de almacenamiento, pero no podrá realizar ninguna acción al respecto, esa es tarea del usuario administrador.

1.14.4. Iteración 4: Portal de Servicios (Internet)

USUARIO AFILIADO

El usuario afiliado podrá realizar las siguientes acciones:

1. Cambio de contraseña
2. Consulta de reportes crediticios de los 3 últimos años (según reglamento para Burós de créditos establecidos por la SBS).
 - a. Consultas se realizarán para personas naturales con su número de cédula o pasaporte, y en caso de personas jurídicas con el número de RUC, en todos los casos deberá tener autorización escrita para dicha consulta.
 - b. Disponible tres tipos de reportes (general, resumido y completo)
 - c. Los reportes se presentarán en PDF o HTML para ser impresos o guardados.
3. Histórico de la últimas consultas realizadas (30 últimas)

1.15. Recursos para el desarrollo del Proyecto

1.15.1. Recursos de Hardware

Para el desarrollo de la aplicación se utilizarán tres computadores con las siguientes características:

- 1 Computadora con procesador Pentium IV Intel Inside 2.4 GHZ, 1Gb de memoria RAM y Disco Duro de 80GB.
- 1 Computadora Pentium IV Intel Incide 1.6 GHZ, 736 MB de memoria RAM y Disco Duro de 40 GB.
- 1 Computadora Pentium IV Intel 2.80 GHZ de memoria RAM y Disco Duro de 150GB.
- 1 MODEM ARRIS para el servicio de Internet banda ancha de 150KBP.
- 1 ROUTER D-LINK.

1.15.2. Recursos de Software

- Sistema Operativo: Microsoft Windows XP Professional SP3.
- Base de Datos: ORACLE ENTERPRISE Edition 10G. Versión 10.2.0.1.0.
- Servidor de Aplicaciones: OC4J. Versión 10.1.2.0.2
- Diseño de Páginas: DREAMWEAVER 8.
- Entorno de Desarrollo: JDEVELOPER. Versión 10.1.2.1.0

1.15.3. Recurso Humano

Se requerirán tres desarrolladores cuyos nombres se citan a continuación:

- Medina Murillo Dennisse Vanesa.
- Román Rosales Jayron Stalin
- Vera Quiroz Darwin Vicente

Para lograr el objetivo los 3 miembros ya mencionados cuentan con el conocimiento necesario en programación, redes, base de datos. Entre las cualidades que cumplen son: disciplina, responsabilidad, eficiencia, eficacia, colaboración, iniciativa, comunicación y respeto.

1.16. Cronograma de Actividades

Ver Anexo 3. (**ANEXO 3**)

CAPÍTULO 2

ANÁLISIS

2.1. Levantamiento de Información

Para el análisis de requerimientos y definición del proyecto se utilizaron como medio de investigación la entrevista personal/telefónica a dos de los cuatro burós autorizados para operar en el Ecuador y la investigación por Internet.

2.1.1 Entrevistas

2.1.1.1. Entrevista 1

Entrevistado: Sr. Ernesto Plaza

Empresa: CALTEC

Modo de entrevista: Telefónica

Descripción:

Se realizaron seis preguntas de tipo abiertas, y la entrevista se limitó a un tiempo de 15 minutos aproximadamente, a continuación se detallan las preguntas formuladas y las respuestas obtenidas:

1. ¿Qué servicios brinda CALTEC?

Reportes de Información sobre las deudas que tienen las personas en todo el Ecuador, la calificación global del deudor (según su forma de pago puede ser A-B-C-D-E) cada que se demora en un pago su calificación va bajando.

Conclusión: Objetivo principal es general los reportes.

2. ¿Cada cuanto tiempo es actualizada la información de los deudores?

La información se actualiza mensualmente

3. ¿CALTEC decide si una persona está apta o no para recibir un crédito?

No, el Buró se encarga de presentar la información del comportamiento de pago de la persona evaluada, ya es cuestión de la compañía que otorga crédito decidir y asumir el riesgo, puede que el historial demuestre que es mal pagador, y aun así le quiere dar crédito; la compañía está en su derecho pero CALTEC, no asume responsabilidad.

4. ¿CALTEC indica el monto máximo al que sujeto deudor puede solicitar?

No, existen herramientas para simular según los ingresos y gastos del individuo y se calcula un monto aproximado, en general son tablas que utilizan los bancos y tarjetas de crédito para hacer esos cálculos, pero no indica valor. Como le expliqué anteriormente solo muestra información y el afiliado toma su decisión

5. ¿De donde proviene la información almacenada en la base de datos de CALTEC y como asegura la veracidad de los datos?

Es verídica por que proviene de la Superintendencia de Bancos y Seguros que es quien rige y controla a los Burós de Crédito, y en caso de ser necesario los audita, la SBS es quien mensualmente nos envía la información.

2.1.1.2. Entrevista 2:

Entrevistado: Sr. Máximo Pérez

Empresa: MULTIBURÓ

Modo de entrevista: Presencial

Descripción:

Se realizaron once preguntas abiertas y cerradas, la entrevista tuvo una duración aproximada de 40 minutos, a continuación se detallan las preguntas y parte de las respuestas:

1. ¿Qué servicios ofrece un Buró de Crédito?

Información crediticia: sobre préstamos y créditos de personas y compañías.

2. ¿Cuáles son los costos por servicio?

Semestral: \$200.00

Anual: \$ 400.00

Reporte personales por unidad \$ 6.00

3. ¿Quiénes pueden obtener el servicio de MULTIBURO?

Cualquier persona sea natural o compañía, con tal que paguen la suscripción.

4. De donde obtiene información que brinda el Buró?

La Superintendencia de bancos y seguros.

La SBS tiene su central de riesgo que es alimentada por los datos de todas las instituciones que maneja: bancos, cooperativas, etc. Esta información la envían mensualmente a los burós de créditos (30 de c/mes).

Existe también la central de riesgo de información comercial que es en la que cualquier persona o compañía denuncian a la SBS si alguien les quedo mal en algún pago o crédito, por ejemplo un cheque protestado.

5. ¿Cómo reciben la información? Existe un formato general para todos los Burós?

Por medio de varios archivos planos que envía cada mes la SBS. LA Superintendencia tiene su formato.

6. ¿Qué proceso realizan con la información?

El Buró lo único que hace con la información es arreglarla, ponerle una presentación bonita, y que se entienda. No se modifica ningún dato.

7. ¿Cada que tiempo se actualiza los datos?

El proceso funciona de la siguiente manera:

- La SBS entrega información recopilada en fin de mes aproximadamente no hay un día fijo.
- Burós de crédito hacen sus actualizaciones
- Y la información esta lista para las consultas de nuestros afiliados (MULTIBURÓ) entre los días 20 y 25 de siguiente mes.

8. ¿Quién se encarga de controlar los Burós de crédito?

Superintendencia de Bancos y Seguros

9. ¿SBS impone estándares para alimentar el Buró de crédito, dar calificaciones y reportes?

Bueno las calificaciones ya las envía la Superintendencia, por ejemplo: si recién recibe un crédito tiene calificación A, se demora mas de 30 días

pasa a B, se demora mas de 60 días pasa a C y así continua hasta que ya lleguen a un proceso judicial y se dé por perdido ese valor en tal caso ya es un deudor con calificación E.

Los reportes, depende del Buró según los que quiera elaborar.

Y la información, pues el estándares básicamente que todos los Burós reciben la misma información, solo cambia la manera en que ellos la presenten.

2.2. Investigación por Internet

La investigación por Internet fue para analizar los servicios que brindan los sistemas de información crediticia, tanto en el País como en el Exterior. La finalidad de investigar a la competencia es para poder desarrollar en nuestro sistema los servicios básicos que prestan los Burós e incluso aumentarlos de manera que se ofrezca al mercado soluciones acordes a sus estándares y exigencias y a las regulaciones de las leyes ecuatorianas; y lograr convertirnos en líder del mercado

2.3. Investigación de Mercado

Se realizaron encuestas para definir cuales son los factores mas importantes que personas o empresas desean conocer al momento de otorgar un crédito, el muestreo inicial fue de 15 personas por el factor tiempo.

2.3.1. Resultado de las encuestas

Los resultados de la encuesta están dados en porcentajes para una mejor comprensión de las entrevistas realizadas:

1.- Políticas de su empresa para otorgar un crédito

-Ser de Nacionalidad ecuatoriana	100%
-Ser mayor de 18 años y menor de 50	100%
-1 año de estabilidad económica o laboral	60%
-2 años de estabilidad económica o laboral	40%
-Ingresos individuales superiores a US.\$ 250	60%
-Ingresos individuales superiores a US.\$ 350	20%
-Ingresos individuales superiores a US.\$ 500	20%
-Poseer Cuenta corriente o de ahorro (referencias)	80%
-Poseer mínimo una Referencia comercial	60%
-Poseer Tarjeta de Crédito	20%

2. ¿Qué técnicas emplea su empresa para evaluar el perfil crediticio de un potencial cliente?

Comprobar datos proporcionados por el solicitante en la solicitud de crédito.	100%
Convenios con empresas comerciales para intercambiar referencias de sus clientes.	40%
Aprobación inmediata si el solicitante maneja adecuadamente alguna tarjeta de crédito bancaria de prestigio como Diners Club, Visa Oro o Mastercard Oro o American Express.	80%

3. ¿Qué desventajas ha encontrado en estas técnicas?

Análisis de la solicitud de crédito:

Información no determina perfil crediticio del solicitante	60%
Debido a la demanda, no se logra eficiencia en tiempos de respuesta para el análisis de una solicitud	40%

Convenios con otras empresas similares:

<p>Es difícil tener estos convenios con todas las casas comerciales que otorgan créditos, debido al tiempo que les tomaría a estas empresas llamar a todas. Resultaría demasiado ineficiente en cuanto a tiempos de respuesta; y muchas de ellas no estarían dispuestas a compartir datos de sus buenos clientes por riesgo a perderlos con la competencia.</p>	<p>100%</p>
---	-------------

Otorgar crédito cuando cliente posee una tarjeta de crédito de prestigio:

<p>No garantiza que el solicitante mantenga un buen record de pagos, y aunque lo tuviese tampoco es garantía que tenga mismo comportamiento en el futuro. Aun así se toma el riesgo</p>	<p>60%</p>
---	------------

4. ¿Es necesario un sistema completo y específico en el que se pueda determinar el comportamiento crediticio de determinada persona o empresa?

En esta pregunta se consideró solo el 80% del total de muestreo, debido a que el otro 20% ya contaba con el servicio de sistema de información crediticia y se demostraron positivos a la presencia de este tipo de servicio.

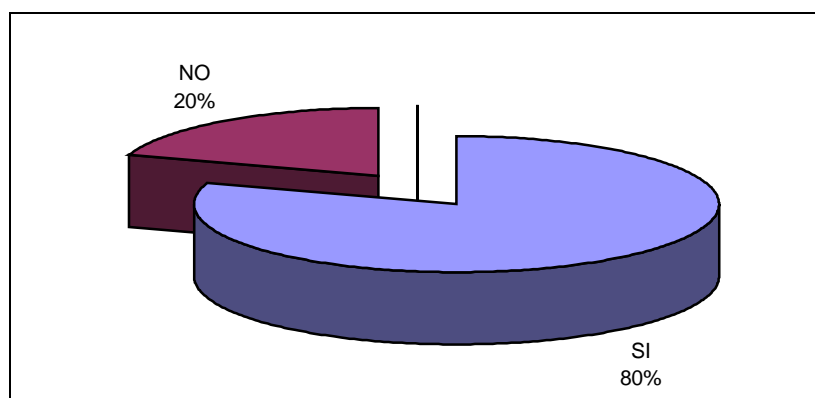


Figura 2.1

5. ¿Quiere su empresa protegerse de los malos créditos?

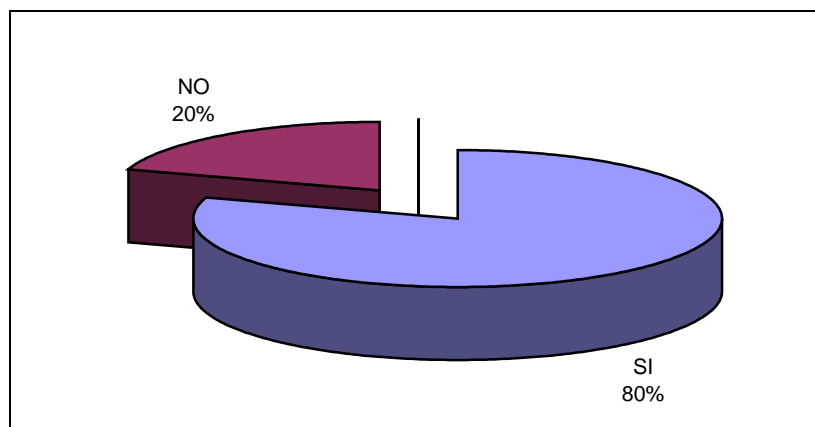


Figura 2.2:

7. ¿Cuáles serían sus requerimientos para adaptarlos al sistema?

Servicio de calidad	60%
Fácil manejo de consultas	40%
Acceso a la información de forma segura	20%
Agilidad en la entrega de informes o reportes	20%

2.3.2. Análisis de lo Resultados

Políticas de la empresa

Las políticas para otorgar créditos varían según la actividad de la misma, la mayoría coincide en los requisitos básicos, pero difieren en los ingresos individuales de los solicitantes y en las referencias comerciales y bancarias

El 60% de la muestra pide como mínimo a sus clientes una referencia comercial, la cual les da una idea de su posible comportamiento crediticio.

Recursos o Técnicas para evaluar el perfil crediticio de un cliente

Las técnicas que muchas empresas utilizan no han cambiado desde hace muchos años, cabe recalcar que los encuestados, un 30% poseían el servicio de sistema crediticio, por lo cual este enfoque es para aquellas que no lo poseen, aun así se da a notar el valor que todos dan a un Buró de Crédito que brinde toda la información crediticia de un sujeto de manera tan sencilla y rápida, evitando llamadas, confirmaciones, referencias comerciales, etc.; los cuales le toman tiempo y no siempre son confiables, ni aseguran un buen comportamiento de pago.

En aquellos encuestados que utilizan técnicas diferentes a los sistemas crediticios, expresaron que se arriesgan a malos clientes y a tener una cartera vencida que en muchos casos supera los 5, 6 o 7 meses en adelante.

Necesidad de un sistema completo y específico:

El 80% de los entrevistados que no contaban ya con un sistema de información crediticia (80% del total de entrevistados) coincidió que se necesita un sistema idóneo en que se pueda establecer el perfil crediticio de un solicitante para evitar “malos créditos”.

Y el 20% de los entrevistados, que ya poseían el servicio afirmó que es una gran ayuda y respaldo para evitar pérdidas y gastos de cobranzas.

¿Quiere su empresa protegerse de los malos créditos?

El 80% reconocen que su empresa necesita urgentemente protegerse de los “malos créditos”. El otro 20% aseguran que los métodos actuales a pesar de no siempre resultar efectivos, son una buena forma de protegerse.

Requerimientos en un nuevo sistema

Los requerimientos varían mucho en los encuestados pero se logró resaltar los más importantes para cada uno de ellos, que es contar con un servicio excelente, de calidad, de fácil acceso y uso, que entregue información de manera rápida y oportuna.

2.4. Análisis de Requerimientos

Debido a la gran cantidad de empresas, microempresas que otorgan créditos a sus clientes con inseguridad, por motivos de falta de información de los clientes, se considera importante saber a que tipo de cliente le darán un crédito.

Bajo esta circunstancia hemos decidido realizar un portal Web que permita de manera muy fácil a los afiliados y no afiliados dar a entender el manejo de dicho portal.

Este portal se basa en el Sistema de Información Crediticia, es decir, un portal que guarda información de deudores de créditos ya sea natural o jurídico.

El **afiliado** podrá ingresar de manera sencilla y muy fácil de entender, por medio de su usuario y contraseña podrá obtener las consultas y reportes que desee según el convenio obtenido y de manera rápida.

El **No afiliado** no podrá hacer lo mismo que un afiliado a nuestro sistema, pero visualizará y analizará el servicio que brindamos y además tiene la opción de contactarnos, también tienen la opción de suscribirse para obtener nuestro servicio informativo para la ayuda de toma de decisiones.

2.5. Análisis de Riesgos

N^a	DESCRIPCIÓN	ACCIÓN A SEGUIR
1	Al no tener la información apropiada esto ocasionaría a que los afiliados (clientes) desprestigien el sistema de nuestro producto, encaminándolo a cambiar de servicio	Darnos a conocer ante la SBS para que nos proporcione una información plena y de confianza para brindar con seguridad nuestro servicio.
2	Si no tenemos una B.D suficientemente amplia, se podría tener problemas por la gran cantidad de información a adquirir.	Contar con una base suficientemente amplia y concisa para la gran cantidad de información que nos proporcionaran.
3	Si la página no es lo suficientemente adecuada para el entendimiento del visitante o afiliado, podría generar una serie de quejas y controversias.	En el caso de la interfase de usuario, tendremos una página atractiva y sumamente cómoda para el entendimiento de todos los clientes, con la finalidad de no tener inconvenientes.
4	Que el usuario cuente con un navegador cuya versión sea inferior en la que nuestro proyecto funcione, el cual limitaría el uso de nuestra página.	Advertir al usuario los requerimientos para que el sistema funcione con normalidad, caso contrario no funcionará a su máxima capacidad al no contar con la

- versión especificada.
- 5 Si no se genera una información (reportes o consulta) rápida, segura y efectiva, esto podría provocar incomodidad o queja de parte del cliente. Para un caso de estos se trabajara con herramientas que trabajan de manera rápida y dejen satisfecho al cliente.
 - 6 Sobrepasar el tiempo de culminación propuesto para el proyecto. Planificar el tiempo de cada actividad en un cronograma, y no incumplir las fechas indicadas.

2.6. Diagrama Entidad Relación (DER)

El modelo entidad-relación aporta una herramienta de modelado para representar las entidades, relaciones y atributos: **los diagramas Entidad-Relación**. Mediante éstos, el esquema conceptual abstracto puede ser mostrado gráficamente y mantener una independencia conceptual con respecto a la implementación propiamente dicha. Ver Anexo 3. (ANEXO 3)

Para el desarrollo del sistema se ha generado el siguiente Diagrama Entidad Relación con sus respectivas Entidades y campos:

2.6.1. Descripción de Tablas

- **BC_USUARIO:**

La tabla usuario tiene la finalidad de almacenar los datos de quienes manejan el sistema de información crediticia. Esta tabla se diferenciara por los tipos de usuario que existirán en el sistema para un mejor control y orden de quienes interactúan en dicho sistema. Los usuarios según su rango y privilegios pueden ser de dos tipos: ADMINISTRADOR y el OPERADOR.

USUARIO ADMINISTRADOR es aquel que posee mas privilegios sobre el manejo del sistema de información crediticia, tales como creación de usuarios operador, afiliados u otros administradores; ingreso y monitoreo de afiliados y operadores; acceso a información histórica contenida en las bases de datos; bloqueo, desbloqueo o eliminación de usuarios, etc.

USUARIO OPERADOR sus privilegios son mas limitados, se encargará de los ingresos, actualización y monitoreo de afiliados; carga de archivos que alimenten la base de datos; consulta de información crediticia menor a 4 años; bloqueo, desbloques. Claro esta que podrá hacer estos movimientos a nivel de operador y no como el que posee el administrador

- **BC_MOVIMIENTO_USUARIO:**

Esta tabla será utilizada para control y monitoreo de los movimientos que realice cada usuario en el sistema (administrador u operador), en otras palabras lleva un control del ingreso y salida diaria y a la vez de los movimientos que a realizado en el sistema en dicha jornada.

- **BC_CONSULTA_USUARIO:**

Ayudara en las consultas que se le haga a los usuario sobre a que afiliado o titular ellos hicieron movimientos (consultar), indicando la IP, fecha y hora.

- **BC_SUCURSAL:**

Esta tabla contendrá los datos de las distintas sucursales que manejaran el servicio de información crediticia y serán diferenciadas por un código para detectar en que ciudad se encuentra localizada y que usuarios laboran en dicha sucursal.

- **BC_EMPRESA:**

Es la matriz que contiene todo el entorno de las sucursales.

- **BC_CIUADAD:**

Esta tabla contendrá las ciudades más importantes de cada provincia del país en donde funciona el sistema de información crediticia, en este caso ECUADOR.

- **BC_PROVINCIA:**

Esta tabla guardara la información de las provincias más importantes del país que ejercerán el servicio del sistema de información crediticio.

- **BC_PAIS:**

Tabla importante para el origen de la provincia y ciudades.

- **BC_AFILIADO:**

Esta tabla contendrá los datos principales para así tener una mejor visualización y conocimiento del afiliado que ha contratado el servicio de información crediticia, cada afiliado tendrá su propia identificación para mayor seguridad. Los tipos de afiliados pueden ser dos: Natural o Jurídico, y según el caso se guarda la información necesaria para cada uno de ellos.

- **BC_CONSULTA_AFILIADO:**

Se identifica al afiliado por medio de la identificación en donde demostrara la

fecha de consulta, hora y la IP donde se hizo la consulta.

- **BC_TITULAR (DEUDORES):**

Contiene la información más importante de los deudores. Estos también pueden ser personas naturales o jurídicas. Estos deudores tendrán su propia identificación para una mejor búsqueda de parte del afiliado.

- **BC_DETALLE_TITULAR**

Hace la descripción mas especifica del titular (DEUDOR), tales como dirección, teléfono, etc.

- **BC_ACREEDOR:**

Esta tabla contendrá la identidad de quien otorgo un crédito a determinado deudor.

- **TIPO_ACREEDOR:**

Indica que el acreedor puede ser Natural o Jurídico con su respectivo id de tipo de acreedor, para así poder diferenciar de manera mas fácil y rápido el tipo.

- **BC_CREDITO:**

La tabla crédito almacena los datos generalizados de un crédito para una rápida visualización y fácil entendimiento por medio de los campos: monto de crédito, plazo, fecha de actualización y responsabilidad con respecto al crédito.

- **BC_TIPO_CREDITO:**

El tipo de crédito representa será: Comercial, Consumo, Diferido, Hipotecario, Rotativo, Vivienda, Pyme. Con su respectivo id.

- **BC_DETALLE_CREDITO:**

La tabla detalle_crédito almacena datos más específicos del crédito como fechas de aperturas, monto a vencer, monto vencido, tipo de moneda y morosidad, en fin datos para el entendimiento de nuestros afiliados para cuando deseen un reporte o información de este tipo.

- **BC_PAGOS:**

Esta es la tabla que nos indicara el número de pago que el deudor va efectuando cada mes, si lo hace con responsabilidad o es moroso y a la vez se le asigna una calificación mensual.

- **BC_CALIFICACION:**

1	A	- 30 MORA	500
2	B	31 A 90 MORA	400
3	C	91 A 120 MORA	300
4	D	91 A 180 MORA	200
5	E	+180 DIAS MORA	100

El id son las letras del A al E.

La descripción son los días de MORA.

El valor referencial es del 500 al 100.

- **BC_MAIL_CONTENT:**

La tabla contiene numero de mail, emisor, remitente, copias, asuntos, mensaje, fecha y hora, en donde cada vez que se envíe un mail se insertaran en las celdas correspondientes, vale indicar que al enviarse el mail se activa un trigger que se encarga de enviar el mail.

- **BC_MAIL_CONTENT_AF:**

Tabla que registra comunicaciones con afiliados de sus operaciones del sistema.

2.6.2. Relación entre tablas

- **BC_USUARIO – BC_MOVIMIENTO_USUARIO: (1 a muchos)**

Indica que un usuario puede tener muchos movimientos y que un movimiento específico fue realizado por un usuario. Por medio de su identificación puede llevar el control de la hora de entrada, salida y también los movimientos que ha realizado, movimientos como por ejemplo el ingreso de un afiliado u

operador si es administrador o de afiliados si es un operador, actualizaciones de datos, cargas de archivos etc.

- **BC_SUCURSAL-BC_USUARIO: (1 a muchos)**

Indica a que sucursal pertenece un usuario, no es otra cosa que la identificación del usuario y sucursal donde se encuentra operando el sistema de información crediticia.

- **BC_EMPRESA-BC_SUCURSAL: (1 a muchos)**

Indica la pertenencia de una sucursal a una empresa y de una empresa con sus sucursales. En este caso como es una sola empresa por ende entenderemos que toda sucursal que maneje nuestro producto de información crediticia pertenecerá a esta única empresa.

- **BC_PROVINCIA_ESTADO-BC_CIUDAD: (1 a muchos)**

Indica la pertenencia de la ciudad a una provincia y de la existencia de muchas ciudades en una provincia.

- **BC_PAIS- BC_PROVINCIA_ESTADO: (1 a muchos)**

Indica la pertenencia de una provincia a un país y de la existencia de muchas provincias en un país.

- **BC_CIUADAD-BC_AFILIADO: (1 a muchos)**

Indica que en una ciudad existen 0,1 o muchos afiliados y que un afiliado pertenecerá a una sola ciudad.

- **BC_AFILIADO-BC_CONSULTA_AFILIADO: (1 a muchos)**

Indica que un afiliado puede tener muchas consultas y que una consulta específica fu realizada por un solo usuario.

- **BC_TITULAR-BC_CREDITO: (1 a muchos)**

Indica que un deudor puede tener muchos créditos y que un crédito específico pertenece a un solo deudor.

- **BC_CREDITO-BC_PAGO: (1 a muchos)**

Indica que un crédito puede tener 1 o muchos pagos y que un pago solamente se le hace a un crédito, es decir, la descripción de cómo el deudor va cancelando sus números de pagos o cuotas mensuales a dicho crédito, en la cual se detallara como la fecha de vencimiento mensual, estado y la calificación por el acreedor.

- **BC_CREDITO – BC_DETALLE_CREDITO: (1 a 1)**

Indica que 1 crédito podrá tener 1 solo detalle y 1 detalle corresponde a un solo crédito que indicara los campos mas específicos de un crédito, campos tales como: fecha de apertura, fecha de reporte, tipo moneda, monto por vencer y vencido, morosidad, etc.

- **BC_ACREEDOR- BC_CREDITO: (1 a muchos)**

Un acreedor puede otorgar 1 o varios créditos pero 1 crédito solo puede ser otorgado por 1 acreedor, relación que tiene como objetivo de dar a conocer el acreedor que otorga el crédito al deudor, en donde el acreedor se identificara por medio de RUC, pasaporte o numero de cedula.

- **BC_TITULAR-BC_PAGOS: (1 a muchos)**

Un titular puede hacer 1 o varios pagos de un crédito, pero un pago solo puede hacer exclusivamente un titular, esta relación da a saber el numero de pagos que va haciendo un titular en sus créditos.

- **BC_TITULAR-BC_DETALLE_TITULAR: (1 a 1)**

Indica que 1 titular podrá tener 1 solo detalle y 1 detalle corresponde a un solo titular que indicara los campos más específicos de un titular, para así dar un claro entendimiento del titular.

- **BC_TITULAR-BC_CALIFICACION: (1 a muchos)**

Por medio de la identificación del titular se sabrá la calificación del mismo, se asegura que un titular puede tener varias calificaciones pero una calificación solo le corresponde a un solo titular.

- **BC_CREDITO - BC_TIPO_CREDITO: (1a muchos)**

La relación entre estas tablas indicara que un crédito puede ser de varios tipos ya sea financiero, comercial, etc. Vale aclarar que un crédito puede ser de varios tipos pero un tipo de crédito solo pertenece a un crédito.

- **BC_USUARIO - BC_CONSULTA_USUARIO: (1a muchos)**

Para la descripción de esta relación es que demuestra la fecha, hora y la ip de donde se realiza dicha consulta. Esta relación indica que un usuario puede tener varias consultas pero una consulta solo le pertenece a un usuario.

- **BC_USUARIO – BC_MOVIMIENTO_USUARIO: (1a muchos)**

Los usuarios pueden tener varios movimientos pero un movimiento le pertenece a un solo usuario, aquí se detalla el id_usuario, el detalle del movimiento, hora inicio, fin y la IP.

- **BC_TITULAR – BC_DETALLE_CREDITO: (1a muchos)**

Un titular puede tener varios detalles de créditos pero un detalle de crédito le pertenece exclusivamente a un solo titular.

- **BC_PROVINCIA_ESTADO – BC_EMPRESA: (1a 1)**

Una empresa solamente pertenece a una sola provincia y una provincia solo contiene una sola empresa (matriz).

- **BC_PROVINCIA_ESTADO – BC_SUCURSAL: (1a muchos) EN DUDA**

Una provincia puede contener varias sucursales de una empresa pero una sucursal solamente pertenece a una provincia. Esto indicara el número de sucursales de una provincia con su respectivo nombre.

- **BC_ACREEDOR – BC_DETALLE_CREDITO: (1a muchos)**

El acreedor al otorgar crédito a la vez detalla dicho crédito con todos sus datos correspondientes, es decir un acreedor puede detallar varios créditos pero un detalle solo pertenece a un acreedor.

- **BC_ACREEDOR – BC_TIPO_ACREEDOR: (1a muchos)**

Da a conocer que un acreedor puede ser de varios tipos de acreedor en este caso acreedor público y jurídico, pero un tipo de acreedor solamente pertenece a un acreedor.

2.7. Diagrama de Flujo de Procesos

2.7.1. Flujo de Procesos:

El Diagrama de Flujo de Procesos es la representación gráfica de la secuencia de todas las operaciones, secuencias, esperas, almacenamientos, generación de reportes, y en general todo lo que ocurre dentro de la aplicación, tanto en intranet como en Internet (Portal Web).

2.7.2. Procesos de Administración del Sistema (Intranet)

2.7.2.1. Proceso Usuario Administrador

- Ingreso al sistema de administración de Buró
- Ingreso de usuario y contraseña de administrador
- Acceso al menú principal
- Interacción con los controles
- Ingreso de usuarios Afiliados u Operadores.
- Monitoreo de Usuarios Afiliados y Operadores

(Ver ANEXO 5)

2.7.2.2. Proceso Usuario Operador

- Ingreso al sistema de administración de Buró
- Ingreso de usuario y contraseña de operador
- Acceso al menú principal
- Interacción con los controles
- Ingreso de usuarios Afiliados.
- Monitoreo de Usuarios Afiliados.

(Ver ANEXO 6)

2.7.3. Procesos del Portal Web (Internet)

2.7.3.1. Proceso Ingreso a la página web, público en general

- Ingreso a la página web
- Visualización del sitio
- Acceso al menú principal
- Interacción con controles
- Opción: ¿Quiénes somos?
- Opción: Afiliados
- Acceso a suscripciones, descarga de formulario de afiliación
- Opción: preguntas frecuentes

- Opción: Contáctenos
- Otros Servicios: herramientas de cálculo y estadísticas

(Ver ANEXO 7)

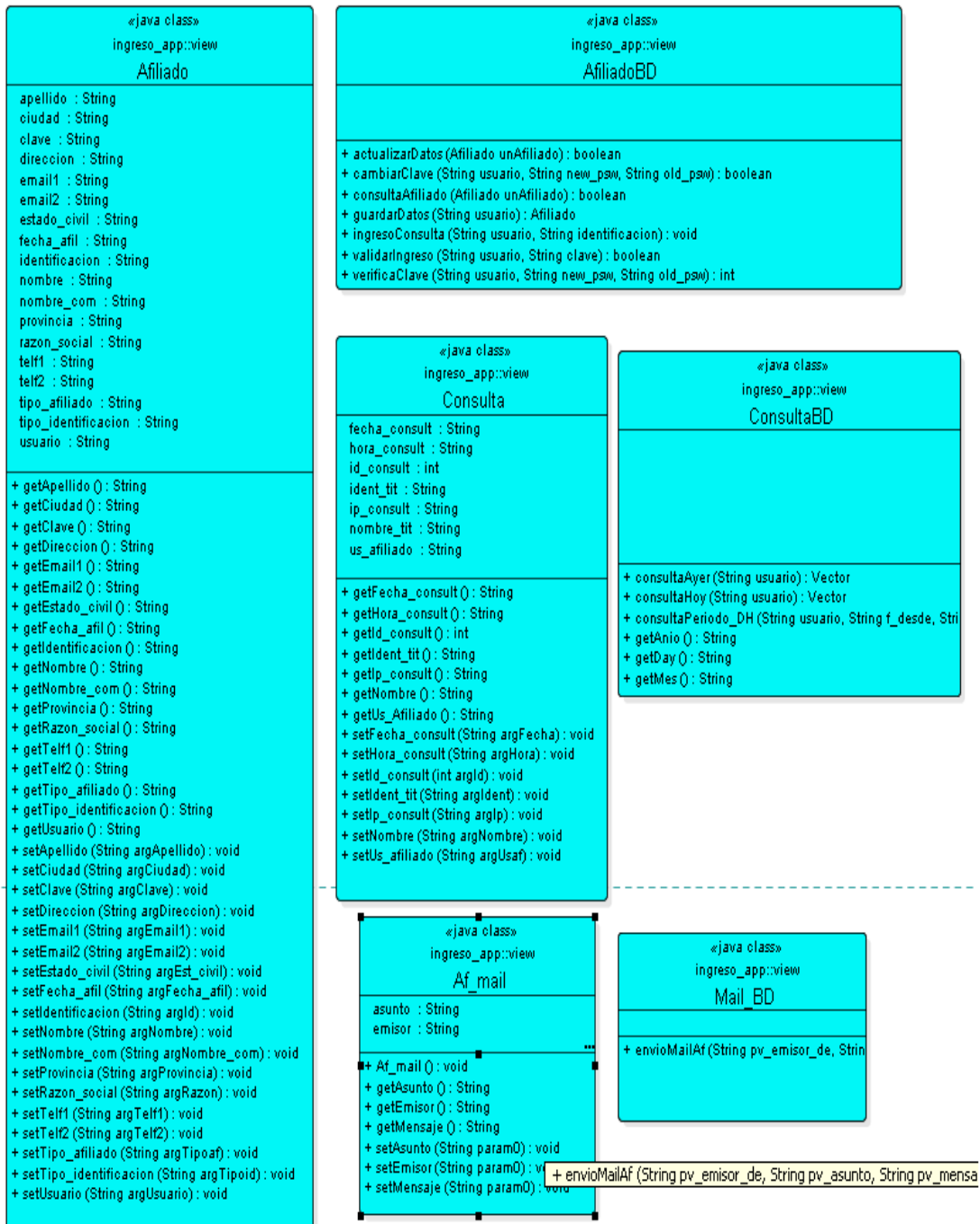
2.7.3.2 Procesos de Usuario Afiliado

- Ingreso al Portal Web
- Visualización del sitio
- Acceso al menú principal, público en general
- Acceso al menú afiliado
- Interacción con controles
- Actualizar datos de afiliado
- Consultas en línea
- Generación de reportes
- Historial de consultas
- Cambio de clave

(Ver ANEXO 8)

2.8. Diagrama de Clases

(ANEXO 9)



Este diagrama de clases utilizado en el ambiente WEB, muestra las principales clases, atributos y métodos utilizados para el manejo y ejecución del portal. La herramienta utilizada: JDEVELOPER 10.

A continuación se detalla la función de los principales métodos utilizados, para ampliar la comprensión del diagrama. En necesario recalcar que dentro de los métodos se facilita también la conexión a la base de datos Oracle.

CLASE AFILIADOBD

- **VALIDARINGRESO**

Se encarga de autenticar el ingreso al Sistema de un Afiliado, recibe como parámetros usuario y clave y compara los datos ingresados con los que se encuentran almacenados en la tabla *BC_AFILIADO*. Debido a que las claves se encuentran encriptadas en la base de datos, se llama a un procedimiento (*BC_DESENC_CLAVE_AF*) que se encarga de desencriptar la misma.

- **GUARDARDATOS**

Dentro de la interfaz web del afiliado se aplica el concepto de Sesiones. Este método guarda los datos del Afiliado actual (aquel que ha sido

autenticado) en la sesión activa con la finalidad de tener acceso rápido a los datos del mismo.

- **ACTUALIZARDATOS**

Utilizado dentro de la opción de actualizar perfil de Afiliado en el menú de Afiliación. El afiliado conectado podrá actualizar sus datos personales y de contacto.

- **CAMBIARCLAVE**

Cambia la clave del Afiliado – Sesión actual, primero verifica la clave actual y solicita la clave nueva. Este método se encarga también de actualizar la clave autenticada en la tabla correspondiente.

- **VERIFICACLAVE**

Verifica que la nueva clave ingresada cumpla con los estándares de claves de afiliados entre ellos que sea mayor de 5 dígitos, valida que no ingrese espacios en blanco en toda la clave o claves predecibles como “123456”, “welcome” o “computer”.

- **INGRESOCONSULTA**

Almacena en la tabla *BC_CONSULTA_AFILIADO* todas las consultas (generación de reportes crediticios) que el Afiliado ha generado. Esta tabla servirá para posteriores consultas de Historiales.

CLASE AFILIADOBD

- **CONSULTAHOY**

Recupera una lista de las consultas realizadas Hoy por el usuario de sesión actual. Los datos los lee de la tabla *BC_CONSULTA_AFILIADO AF*. Opción del menú de afiliado: “Historial de Consultas - Hoy”

- **CONSULTAAYER**

Recupera una lista de las consultas realizadas Ayer por el usuario de sesión actual. Los datos los lee de la tabla *BC_CONSULTA_AFILIADO AF*. Opción del menú de afiliado: “Historial de Consultas - Ayer”.

- **CONSULTAPERIODO_DH**

Recupera una lista de las consultas realizadas por período por el usuario de sesión actual, la consulta puede realizarse en un rango de fechas determinado

por el afiliado. Los datos los lee de la tabla *BC_CONSULTA_AFILIADO AF*.

Opción del menú de afiliado: "Historial de Consultas – Por período"

- **GETDAY / GETMES / GETANIO**

Recupera el día actual en número. Utilizada en la validación de filtros para consultar Historial de consultas por Período. Si los campos son nulos asigna el día, mes y año actual.

CLASE AFILIADOBD

- **BC_MAIL_CONTENT_AF**

Este método se utiliza en la opción "ENVÍO MAIL" de los afiliados; hace una inserción en la tabla *BC_MAIL_CONTENT_AF*, luego al insertar se activa un trigger *BC_TR_SEND_MAIL_CONT_AF*, que se encarga de realizar el envío del correspondiente correo.

2.9. Casos de Uso

Los casos de uso son la representación de manera textual de una secuencia de interacciones entre el sistema y uno o más actores en la que se considera al

sistema como algo desconocido y en la que los actores obtienen resultados observables, además pueden servir para elaborar casos de pruebas del sistema. En los casos de uso se incluyen a los usuarios, conocidos como actores pues son los que interactuarán directamente con el sistema.

Los diagramas de caso de uso son la representación gráfica de cada caso de uso, y proporcionan una visión global del conjunto de casos de uso de los actores interactuando con el sistema.

Los actores que intervienen en los casos de uso serán:

- Usuario Administrador
- Usuario Operador
- Usuario Afiliado

Caso de Uso 1: Usuario Administrador

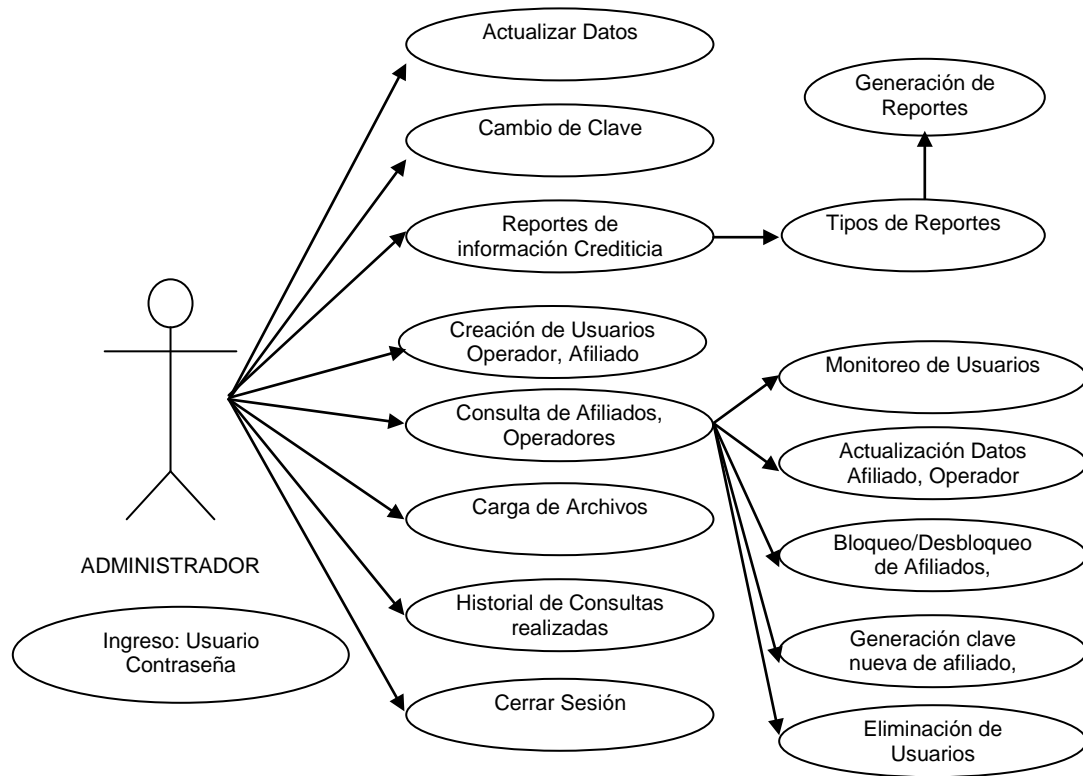


Figura 2.3: Caso de uso 1 – Usuario Administrador

Caso de Uso 2: Usuario Operador

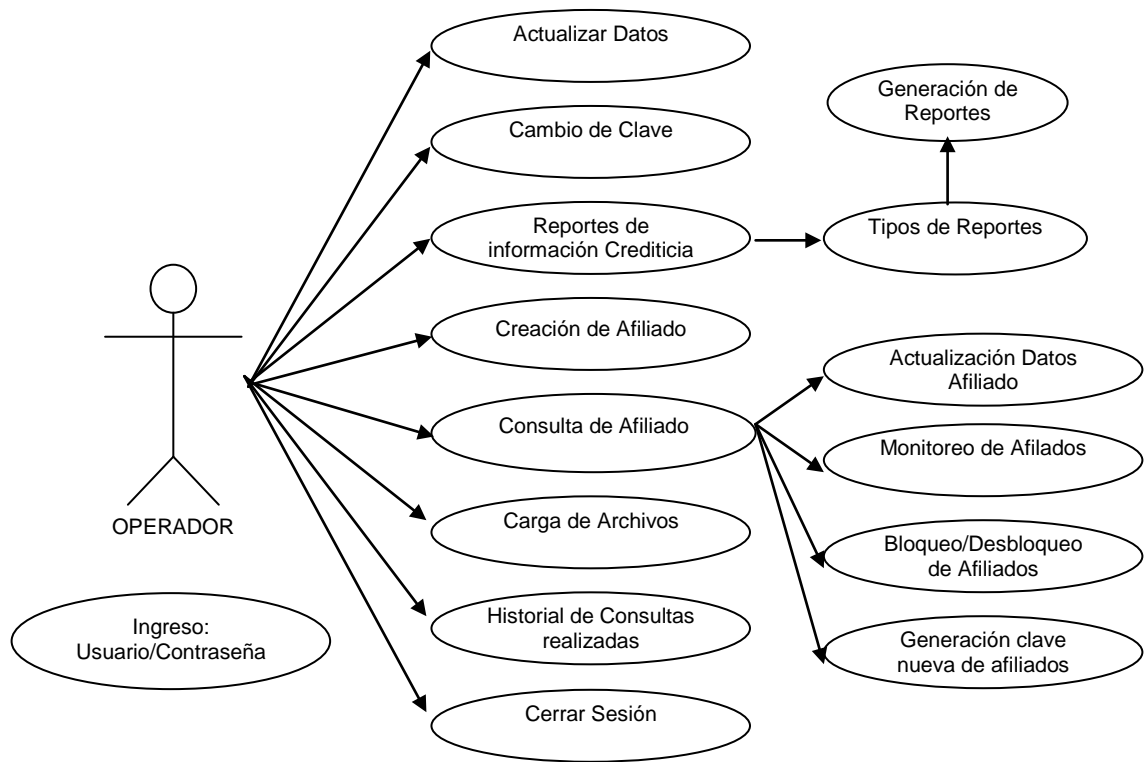


Figura 2.4: Caso de uso 1 – Usuario Operador

Escenario 1:

Nombre:	Inicio de Sesión
Actores:	Usuario Administrador, Usuario Operador
Descripción:	Permite inicio de sesión ingresando usuario/contraseña
Precondición:	Ninguna
Flujo Normal	<ol style="list-style-type: none"> 1. El actor digita su usuario y contraseña. 2. El Sistema verifica los datos ingresados. 3. Autoriza y autentifica según tipo de usuario (Operador o Administrador).
Flujo Alternativo 1:	<ol style="list-style-type: none"> 1. El sistema verifica datos, y son incorrectos. 2. Notifica al actor el error y brinda la opción de corregir.
Flujo Alternativo 2:	El actor no está registrado
Pos condiciones:	Ingreso al Sistema (Intranet)

Escenario 2:

Nombre:	Actualización de Datos
Actores:	Usuario Administrador, Usuario Operador
Descripción:	Permite actualizar datos personales de los usuarios que manejan el sistema de información crediticia; según su inicio de sesión puede ser Operador o Administrador
Precondición:	El actor debe haber iniciado su Sesión
Flujo Normal	<ol style="list-style-type: none"> 1. Modifica su información personal como teléfonos, dirección, sucursal, etc. 2. Se actualizan datos del usuario.
Flujo Alternativo 1:	<ol style="list-style-type: none"> 1. Datos ingresados no cumplen con estándares o tipos de datos. 2. Sistema solicita insertar nuevamente los datos erróneos.
Flujo Alternativo 2:	<ol style="list-style-type: none"> 1. El usuario desiste de modificar sus datos 2. No se actualizan datos.
Pos condiciones:	Confirmación por ventana emergente de guardar los datos

Escenario 3:

Nombre:	Cambio de Clave
Actores:	Usuario Administrador, Usuario Operador
Descripción:	Los actores cambiaran su clave en su primer ingreso al sistema y en el momento que consideren necesario.
Precondición:	El actor debe haber iniciado sesión.
Flujo Normal	<ol style="list-style-type: none"> 1. El actor digita contraseña anterior. 2. El actor digita contraseña nueva. 3. El actor confirma contraseña nueva. 4. Validación de contraseña anterior. 5. Validación de nueva contraseña que cumpla con los estándares de seguridad como longitud; valida la confirmación de claves. 5. Cambio de contraseña para inicio de sesión.
Flujo Alternativo:	<ol style="list-style-type: none"> 1. El actor ingresa erróneamente contraseñas. 2. Notifica al actor el error y brinda la opción de corregir.
Pos condiciones:	Confirmación para guardar los cambios.

Escenario 4:

Nombre:	Reportes de Información Crediticia
Actores:	Usuario Administrador, Usuario Operador
Descripción:	Los actores podrán consultar la información crediticia de los deudores que constan en las bases de datos, podrán generar reportes para ser impresos o guardados
Precondición:	El actor debe haber iniciado sesión.
Flujo Normal	<ol style="list-style-type: none"> 1. El sistema muestra las opciones de Reportes que pueden ser reportes personales, reportes corporativos o reportes por lotes. 2. El actor selecciona según el sujeto o modo de consulta o realizar. 3. El sistema muestra las opciones de tipo de reportes 4. El sistema según opción escogida solicita el ingreso de una identificación del sujeto (cedula, pasaporte, RUC) 5. Actor ingresa datos solicitados. 6. El reporte del sujeto o sujetos se presenta en pantalla 7. El sistema muestra las opciones imprimir o guardar reporte
Flujo Alternativo 1:	<ol style="list-style-type: none"> 1. El Actor seleccionó opciones de reporte y tipos de reporte 2. El ingresa un dato erróneo para la búsqueda 3. El sistema muestra la opción de volver a ingresar dato

Flujo Alternativo 2:	El actor desiste de la consulta de reportes y cancela.
Pos condiciones:	El sistema muestra las opciones imprimir, guardar reporte o salir

Escenario 5:

Nombre:	Creación de Usuario Afiliado
Actores:	Usuario Administrador, Usuario Operador
Descripción:	Los actores crearán un nuevo usuario Afiliado, le darán lo privilegios de afiliado según contratación de servicio e ingresaran los datos del afiliado que puede ser una persona natural o jurídica.
Precondición:	El actor debe haber iniciado sesión.
Flujo Normal	<ol style="list-style-type: none"> 1. El actor crea el usuario Afiliado, le asigna un nombre de usuario y una contraseña. 2. El actor otorga privilegios de Afiliado al usuario creado. 3. El actor ingresa los datos principales del afiliado que contrato lo servicios, tales como nombre, identificación, teléfonos, dirección, etc.

<p>Flujo Alternativo 1:</p>	<ol style="list-style-type: none"> 1. El usuario afiliado ya existe en la base de datos 2. El sistema presenta la opción de modificar o actualizar datos de ese usuario. 3. El sistema presenta el estado actual del usuario: activo o inactivo. 4. Si usuario afiliado está inactivo es por que está bloqueado pero existe, entonces sistema muestra la opción de “Bloqueo/Desbloqueo”
<p>Flujo Alternativo 2:</p>	<p>El actor desiste de la creación de nuevo afiliado y cancela.</p>
<p>Pos condiciones:</p>	<p>Confirmación para guardar cambios</p>

Escenario 6:

<p>Nombre:</p>	<p>Creación de Usuario Operador</p>
<p>Actores:</p>	<p>Usuario Administrador</p>
<p>Descripción:</p>	<p>El Administrador podrá crear usuarios operadores según las necesidades, les asignará los privilegios de operador e ingresara sus datos personales.</p>

Precondición:	El actor debe haber iniciado sesión como usuario administrador
Flujo Normal	<ol style="list-style-type: none"> 1. El actor crea el usuario operador y le asigna un nombre de usuario y una contraseña. 2. El actor otorga privilegios de operador. 3. El actor ingresa los datos principales del operador, tales como nombre, identificación, teléfonos, dirección, sucursal en la que labora, etc.
Flujo Alternativo:	El actor desiste de la creación de nuevo operador y cancela.
Pos condiciones:	Confirmación para guardar cambios

Escenario 7:

Nombre:	Consulta de Usuario Afiliado
Actores:	Usuario Administrador, Usuario Operador
Descripción:	Los actores podrán consultar los usuarios afiliados existentes, considerando también aquellos afiliados cuyo estado sea inactivo.
Precondición:	El actor debe haber iniciado sesión.

Flujo Normal	<ol style="list-style-type: none"> 1. El actor elige la opción de consultar afiliados. 2. El sistema muestra la opción de búsqueda que podrá hacerse por un filtro, como por ejemplo por estado o por identificación o todos. 3. El sistema muestra el listado de afiliados 4. El Actor elige un afiliado. 5. Se muestra datos del afiliado y monitoreo de los movimientos del mismo.
Flujo Alternativo:	El actor desiste de la consulta y cancela.
Pos condiciones:	Ninguna

Escenario 8:

Nombre:	Consulta de Usuario Operador
Actores:	Usuario Administrador
Descripción:	El Administrador podrá consultar los usuarios operadores existentes, considerando también aquellos que se encuentren en estado inactivo.
Precondición:	El actor debe haber iniciado sesión como administrador

Flujo Normal	<ol style="list-style-type: none"> 1. El actor elige la opción de consultar operadores. 2. El sistema muestra la opción de búsqueda que podrá hacerse por un filtro, como por ejemplo por estado o por identificación o todos. 3. El sistema muestra el listado de operadores. 4. El Actor elige un operador. 5. Se muestra datos del operador y monitoreo de los movimientos del mismo.
Flujo Alternativo 2:	El actor desiste de la consulta y cancela.
Pos condiciones:	Ninguna

Escenario 9:

Nombre:	Actualización de Datos de Usuario Afiliado
Actores:	Usuario Administrador, Usuario Operador
Descripción:	Los actores podrán actualizar los datos de los usuarios afiliados u operadores cuando consideren pertinente, ya depende de las definiciones de la empresa se lo hace periódicamente. Es

	importante mencionar q dentro de la sesión de usuario operador como de afiliado existe la opción de que actualicen sus datos cuando haya un cambio en los mismos.
Precondición:	El actor debe haber iniciado sesión. El actor ha elegido el usuario mediante la consulta de usuarios afiliados
Flujo Normal	<ol style="list-style-type: none"> 1. El actor elige la opción de actualizar datos de afiliados. 2. El sistema muestra los datos actuales del afiliado 3. El actor realiza los cambios. 4. Se envía una notificación al buzón del afiliado reportando los cambios.
Flujo Alternativo 2:	El actor desiste de la actualización y cancela.
Pos condiciones:	Confirmación para guardar los cambios.

Escenario 10:

Nombre:	Actualización de Datos de Usuario Operador
Actores:	Usuario Administrador

Descripción:	El Administrador podrá actualizar los datos de los operadores según sea necesario. En la sesión de usuario operador existe también la opción de que actualicen sus datos cuando haya un cambio en los mismos.
Precondición:	El actor debe haber iniciado sesión como administrador. El actor ha elegido el usuario mediante la consulta de usuarios operadores.
Flujo Normal	<ol style="list-style-type: none"> 1. El actor elige la opción de actualizar datos de operador. 2. El sistema muestra los datos actuales del operador 3. El actor realiza los cambios. 4. Se envía una notificación al operador reportando los cambios.
Flujo Alternativo 2:	El actor desiste de la actualización y cancela.
Pos condiciones:	Confirmación para guardar los cambios.

Escenario 11:

Nombre:	Bloqueo/Desbloqueo de Usuarios Afiliados
Actores:	Usuario Administrador, Usuario Operador
Descripción:	Los Usuarios afiliados serán bloqueados en caso que excedan el

	límite de consultas asignadas, por término de contrato etc. No elimina al usuario, sólo inhabilita el servicio de información crediticia. En consecuencia su estado será inactivo.
Precondición:	El actor debe haber iniciado sesión. El actor ha elegido el usuario mediante la consulta de usuarios afiliados.
Flujo Normal	1. El sistema muestra los dato del afiliado 2. El sistema muestra las opciones de bloqueo / Desbloqueo
Flujo Alternativo 2:	El actor desiste y cancela.
Pos condiciones:	Confirmación para proceder con el bloqueo/desbloqueo.

Escenario 12:

Nombre:	Bloqueo/Desbloqueo de Usuarios Operadores
Actores:	Usuario Administrador
Descripción:	El administrador podrá bloquear o desbloquear a los operadores, puede bloquearlos en caso que salgan temporalmente de su trabajo, vacaciones, o sanciones, etc; eso depende del

	administrador, y su estado pasará a inactivo. El desbloqueo será lo contrario, y se presentará cuando el operador se reintegra a sus actividades.
Precondición:	El actor debe haber iniciado sesión como administrador. El actor ha elegido el usuario mediante la consulta de usuarios operadores.
Flujo Normal	1. El sistema muestra los datos del operador. 2. El sistema muestra las opciones de bloqueo / Desbloqueo
Flujo Alternativo 2:	El actor desiste y cancela.
Pos condiciones:	Confirmación para proceder con el bloqueo/desbloqueo.

Escenario 13:

Nombre:	Eliminación de Usuarios Afiliados
Actores:	Usuario Administrador, Usuario Operador
Descripción:	El administrador podrá eliminar usuarios afiliados, según el caso. Según lógica de negocio, los datos de los afiliados no deben ser desechados, por que aun si dejan de contratar nuestros servicios,

	a futuro podríamos persuadirlos de hacerlo; esta opción podría ser utilizada en caso de cometer errores al momento de crear usuarios.
Precondición:	El actor debe haber iniciado sesión. El actor ha elegido el usuario mediante la consulta de usuarios afiliados.
Flujo Normal	1. El sistema muestra los datos del afiliado. 2. El sistema muestra la opción de eliminar afiliado.
Flujo Alternativo 2:	El actor desiste de la eliminación y cancela.
Pos condiciones:	Confirmación para proceder con la eliminación.

Escenario 14:

Nombre:	Eliminación de Usuarios Operador
Actores:	Usuario Administrador.
Descripción:	El administrador podrá eliminar usuarios operadores en caso que un operador se separe de la empresa, por errores al momento de la creación de usuario, etc. La decisión de la eliminación será de los administradores.

Precondición:	El actor debe haber iniciado sesión como administrador El actor ha elegido el usuario mediante la consulta de usuarios operadores.
Flujo Normal	1. El sistema muestra los datos del operador. 2. El sistema muestra la opción de eliminar operador.
Flujo Alternativo 2:	El actor desiste de la eliminación y cancela.
Pos condiciones:	Confirmación para proceder con la eliminación.

Escenario 15:

Nombre:	Carga de Archivos
Actores:	Usuario Administrador, Usuario Operador.
Descripción:	Los Actores deben encargarse de cargar los archivos con los datos de información crediticia que el Buró recibe mensualmente, estos datos se ingresarán o actualizarán en la base de datos del Buró.
Precondición:	El actor debe haber iniciado sesión El actor debe haber recibido el o los archivos planos con

	información crediticia de los sujetos deudores del País, esta información en el Ecuador será entregada por la Superintendencia de Bancos y Seguros.
Flujo Normal	<ol style="list-style-type: none"> 1. El sistema muestra las opciones de carga, que puede ser con un archivo en excel, bloc de notas, notepad o si son varios los archivos a cargar existe la opción de cargarlos comprimidos (.rar, .zip) 2. El sistema extrae los datos de los archivos y los almacena según correspondan en la Base de Datos
Flujo Alternativo 2:	El actor desiste de la carga de archivos y cancela.
Pos condiciones:	Mensaje de confirmación para cargar archivo.

Escenario 16:

Nombre:	Historial de Consultas
Actores:	Usuario Administrador, Usuario Operador.
Descripción:	Los Actores podrán visualizar las últimas consultas que han realizado a los deudores (Reportes Crediticios). Muestra los nombres e identificación de los sujetos, que serán un enlace para

	generar un nuevo reporte.
Precondición:	El actor debe haber iniciado sesión
Flujo Normal	<ol style="list-style-type: none"> 1. El sistema muestra las opciones para visualizar las consultas, puede ser: consultas realizadas ayer, consultas realizadas última semana, hace 15 días o ver todas las consultas del historial (un mes antes de la fecha actual) 2. El Actor elige un deudor del historial. 3. El sistema permite la opción de generar el reporte de ese deudor (Generación de Reportes)
Pos condiciones:	Ninguna

Caso de Uso 3: Usuario Afiliado

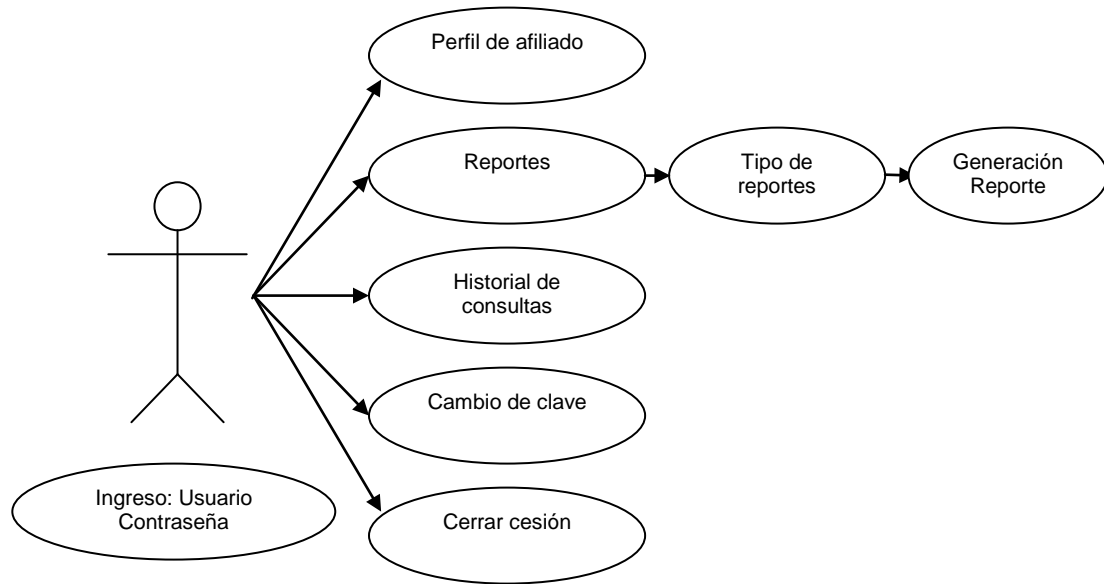


Figura 2.5: Caso de uso 3 – Usuario Afiliado

Escenario 1:

Nombre:	Ingreso de afiliado
Actores:	Afiliados(clientes)
Descripción:	Permite el ingreso del afiliado por medio de un usuario/contraseña
Precondición:	El afiliado debe haber ingresado a su sesión.
Flujo Normal	<ol style="list-style-type: none"> 1. El afiliado digita su usuario y contraseña. 2. El sistema verifica los datos ingresados. 3. Autoriza el ingreso al sistema al afiliado.
Flujo Alternativo 1:	<ol style="list-style-type: none"> 2. El sistema verifica datos, y son incorrectos. 2. Notifica al afiliado el error y brinda la opción de corregir.
Flujo Alternativo 2:	El afiliado no esta registrado.
Pos condiciones:	Ingreso al sistema internet(web).

Escenario 2:

Nombre:	Perfil de afiliado
Actores:	Afiliados(clientes)
Descripción:	Permite al afiliado consultar sus propios datos para verificar su nomina
Precondición:	El afiliado debe tener correcto sus datos.
Flujo Normal	<ol style="list-style-type: none"> 1. El afiliado tiene un mal dato. 2. el sistema le permitirá modificaciones. 3. Envío de datos al sistema interno para ser consultado.
Flujo Alternativo 1:	<ol style="list-style-type: none"> 3. los datos modificados fueron enviados al sistema interno. 2. notificación del usuario (administrador/operador) sobre su modificación.
Flujo Alternativo 2:	No permita cambio por replica.
Pos condiciones:	Confirmar modificación.

Escenario 3:

Nombre:	Reportes de Información Crediticia
Actores:	Afiliados(clientes)
Descripción:	Los afiliados podrán consultar la información crediticia de los deudores que constan en las bases de datos, podrán generar reportes para ser impresos, guardado o visualizarlo.
Precondición:	El afiliado debe haber iniciado sesión.
Flujo Normal	<ol style="list-style-type: none"> 1. El sistema muestra las opciones de Reportes que pueden ser reportes personales, reportes corporativos o reportes por lotes. 2. El afiliado selecciona según el sujeto o modo de consulta a realizar. 3. El sistema muestra las opciones de tipo de reportes: R1, R2, R3 4. El sistema según opción escogida solicita el ingreso de una identificación del deudor (cedula, pasaporte, RUC) 5. Afiliado ingresa datos solicitados. 6. El reporte del sujeto o sujetos se presenta en pantalla. 7. El sistema muestra las opciones imprimir o guardar reporte.

Flujo Alternativo 1:	<ol style="list-style-type: none"> 1. El Afiliado seleccionó opciones de reporte y tipos de reporte. 2. El ingresa un dato erróneo para la búsqueda. 3. El sistema muestra la opción de volver a ingresar dato.
Flujo Alternativo 2:	El afiliado desiste de la consulta de reportes y cancela.
Pos condiciones:	El sistema muestra las opciones imprimir, guardar reporte o salir.

Escenario 4:

Nombre:	Cambio de Clave
Actores:	Afiliado
Descripción:	Los Afiliados cambiaran su clave en su primer ingreso al sistema y en el momento que consideren necesario.
Precondición:	El Afiliado debe haber ingresado al sistema.
Flujo Normal	<ol style="list-style-type: none"> 1. El actor digita contraseña anterior. 2. El actor digita contraseña nueva. 3. El actor confirma contraseña nueva. 4. Validación de contraseña anterior. 5. Validación de nueva contraseña que cumpla con los

	estándares de seguridad como longitud; valida la confirmación de claves. 5. Cambio de contraseña para inicio de sesión.
Flujo Alternativo:	1. El actor ingresa erróneamente contraseñas. 2. Notifica al actor el error y brinda la opción de corregir.
Pos condiciones:	Confirmación para guardar los cambios.

Escenario 5:

Nombre:	Historial de Consultas
Actores:	Afiliado:
Descripción:	El afiliado podrá ver las 30 últimas consultas de los deudores, para la toma de decisión.
Precondición:	El afiliado debe haber ingresado al sistema.
Flujo Normal	1. El sistema muestra las 30 últimos deudores con sus respectivos nombres, apellidos y su identificación en un esquema fácil de entender.
Pos condiciones:	Ninguna

CAPÍTULO 3

CODIFICACION

A NIVEL DE PL-SQL

Demostraremos todos los procedimientos, funciones, paquetes, Job, triggers y java source realizados en pl-sql para este proyecto.

Funciones:

```
create or replace function BC_F_VERIFICA_ARCHIVO(PV_LINEA VARCHAR2)
return number is
```

```
VALIDA NUMBER:=1;
L_CONT NUMBER(3):=0;
LV_ENTIDAD ADMIN.BC_TITULAR.TIPO_TITULAR%TYPE;
LV_NOMBRE ADMIN.BC_TITULAR.NOMBRE%TYPE;
LV_APELLIDO ADMIN.BC_TITULAR.APELLIDO%TYPE;
LV_TIPO_IDENT ADMIN.BC_TITULAR.TIPO_IDENTIFICACION%TYPE;
LV_IDENTIFICACION ADMIN.BC_TITULAR.IDENTIFICACION%TYPE;
```

```

LV_MONEDA ADMIN.BC_DETALLE_CREDITO.TIPO_MONEDA%TYPE;
LN_CREDITO_APROBADO VARCHAR2(13);
LN_MONTO_ADEUDADO VARCHAR2(13);
LN_SALDO_NO_VENCIDO VARCHAR2(13);
LN_SALDO_VENCIDO VARCHAR2(13);
LD_FECHA_APERTURA ADMIN.BC_DETALLE_CREDITO.FECHA_APERTURA%TYPE;
LD_FECHA_VENCIMIENTO ADMIN.BC_PAGOS.FECHA_VENC%TYPE;
LN_S_30 VARCHAR2(13);
LN_S_31_60 VARCHAR2(13);
LN_S_61_90 VARCHAR2(13);
LN_S_91_120 VARCHAR2(13);
LN_S_MAS_120 VARCHAR2(13);
VL_NEW_LINE VARCHAR2(500):=PV_LINEA;

begin
  IF TRIM(VL_NEW_LINE) IS NULL THEN
    NULL;
  ELSE
    --VALIDA Q ENTIDAD SOLO SEA N O J
    LV_ENTIDAD :=
      TRIM(SUBSTR(VL_NEW_LINE,1,INSTR(VL_NEW_LINE,',' ,1,1)-1));
    IF LV_ENTIDAD<>'N' AND LV_ENTIDAD<>'J' THEN
      VALIDA:= 0;
    END IF;
    --VALIDA Q EXISTA UN NOMBRE
    LV_NOMBRE :=

    TRIM(SUBSTR(VL_NEW_LINE,INSTR(VL_NEW_LINE,',' ,1,1)+1,INSTR(VL_NEW_LINE,
    ',' ,1,2)-INSTR(VL_NEW_LINE,',' ,1,1)-1));
    IF LV_NOMBRE IS NULL THEN
      VALIDA:= 0;
    END IF;
    --VALIDA Q EXISTA UN APELLIDO
    LV_APELLIDO :=

    TRIM(SUBSTR(VL_NEW_LINE,INSTR(VL_NEW_LINE,',' ,1,2)+1,INSTR(VL_NEW_LINE,
    ',' ,1,3)-INSTR(VL_NEW_LINE,',' ,1,2)-1));
    IF LV_APELLIDO IS NULL THEN
      VALIDA:= 0;
    END IF;
    --VALIDA TIPOS DE IDENTIFICACION ACEPTADOS RUC - CED (CEDULA) -
    PAS(PASAPORTE)
    LV_TIPO_IDENT :=

    TRIM(SUBSTR(VL_NEW_LINE,INSTR(VL_NEW_LINE,',' ,1,3)+1,INSTR(VL_NEW_LINE,
    ',' ,1,4)-INSTR(VL_NEW_LINE,',' ,1,3)-1));
    IF LV_TIPO_IDENT<>'CED' AND LV_TIPO_IDENT<>'RUC' AND
    LV_TIPO_IDENT<>'PAS' THEN
      VALIDA:= 0;
    END IF;
  
```

```

--VALIDA IDENTIFICACIONES
LV_IDENTIFICACION :=

TRIM(SUBSTR(VL_NEW_LINE, INSTR(VL_NEW_LINE, ';', 1, 4)+1, INSTR(VL_NEW_LINE,
';', 1, 5)-INSTR(VL_NEW_LINE, ';', 1, 4)-1));
L_CONT:=LENGTH(LV_IDENTIFICACION);
IF LV_TIPO_IDENT='CED' THEN
  IF L_CONT<>10 THEN
    VALIDA:= 0;
  END IF;
  ELSIF LV_TIPO_IDENT='RUC' THEN
    IF L_CONT<>13 THEN
      VALIDA:= 0;
    END IF;
  END IF;
LV_MONEDA :=

TRIM(SUBSTR(VL_NEW_LINE, INSTR(VL_NEW_LINE, ';', 1, 9)+1, INSTR(VL_NEW_LINE,
';', 1, 10)-INSTR(VL_NEW_LINE, ';', 1, 9)-1));
IF LV_MONEDA<>'USD' AND LV_MONEDA<>'EUR' THEN
  VALIDA:= 0;
END IF;
-- VALIDA Q SEA SOLO NUMEROS
LN_CREDITO_APROBADO :=

TRIM(SUBSTR(VL_NEW_LINE, INSTR(VL_NEW_LINE, ';', 1, 10)+1, INSTR(VL_NEW_LINE
, ';', 1, 11)-INSTR(VL_NEW_LINE, ';', 1, 10)-1));
IF BC_F_NUMERO(LN_CREDITO_APROBADO)=0 THEN
  VALIDA:= 0;
END IF;
-- VALIDA Q SEA SOLO NUMEROS
LN_MONTO_ADEUDADO :=

TRIM(SUBSTR(VL_NEW_LINE, INSTR(VL_NEW_LINE, ';', 1, 11)+1, INSTR(VL_NEW_LINE
, ';', 1, 12)-INSTR(VL_NEW_LINE, ';', 1, 11)-1));
IF BC_F_NUMERO(LN_MONTO_ADEUDADO)=0 THEN
  VALIDA:= 0;
END IF;
-- VALIDA Q SEA SOLO NUMEROS
LN_SALDO_NO_VENCIDO :=

TRIM(SUBSTR(VL_NEW_LINE, INSTR(VL_NEW_LINE, ';', 1, 12)+1, INSTR(VL_NEW_LINE
, ';', 1, 13)-INSTR(VL_NEW_LINE, ';', 1, 12)-1));
IF BC_F_NUMERO(LN_SALDO_NO_VENCIDO)=0 THEN
  VALIDA:= 0;
END IF;
-- VALIDA Q SEA SOLO NUMEROS
LN_SALDO_VENCIDO :=

```

```

TRIM(SUBSTR(VL_NEW_LINE, INSTR(VL_NEW_LINE, ';' , 1, 13)+1, INSTR(VL_NEW_LINE
, ';' , 1, 14)-INSTR(VL_NEW_LINE, ';' , 1, 13)-1));
    IF BC_F_NUMERO(LN_SALDO_VENCIDO)=0 THEN
        VALIDA:= 0;
    END IF;
-- VALIDA Q NO CONTENGA LETRAS
LD_FECHA_APERTURA :=

TO_DATE(TRIM(SUBSTR(VL_NEW_LINE, INSTR(VL_NEW_LINE, ';' , 1, 14)+1, INSTR(VL_
NEW_LINE, ';' , 1, 15)-INSTR(VL_NEW_LINE, ';' , 1, 14)-1)), 'DD/MM/YY');
--     IF BC_F_NUMERO(LD_FECHA_APERTURA)=0 THEN
--         VALIDA:= 0;
--     END IF;
-- VALIDA Q NO CONTENGA LETRAS
LD_FECHA_VENCIMIENTO :=

TO_DATE(TRIM(SUBSTR(VL_NEW_LINE, INSTR(VL_NEW_LINE, ';' , 1, 15)+1, INSTR(VL_
NEW_LINE, ';' , 1, 16)-INSTR(VL_NEW_LINE, ';' , 1, 15)-1)), 'DD/MM/YYYY?');
--     IF BC_F_NUMERO(LD_FECHA_VENCIMIENTO)=0 THEN
--         VALIDA:= 0;
--     END IF;
--
--     LN_NUM_PAGOS :=
--
TO_NUMBER(TRIM(SUBSTR(VL_NEW_LINE, INSTR(VL_NEW_LINE, ';' , 1, 16)+1, INSTR(V
L_NEW_LINE, ';' , 1, 17)-INSTR(VL_NEW_LINE, ';' , 1, 16)-1)));
-- VALIDA Q SEA NUMERO
LN_S_30 :=

TRIM(SUBSTR(VL_NEW_LINE, INSTR(VL_NEW_LINE, ';' , 1, 17)+1, INSTR(VL_NEW_LINE
, ';' , 1, 18)-INSTR(VL_NEW_LINE, ';' , 1, 17)-1));
    IF BC_F_NUMERO(LN_S_30)=0 THEN
        VALIDA:= 0;
    END IF;
-- VALIDA Q SEA NUMERO
LN_S_31_60 :=

TRIM(SUBSTR(VL_NEW_LINE, INSTR(VL_NEW_LINE, ';' , 1, 18)+1, INSTR(VL_NEW_LINE
, ';' , 1, 19)-INSTR(VL_NEW_LINE, ';' , 1, 18)-1));
    IF BC_F_NUMERO(LN_S_31_60)=0 THEN
        VALIDA:= 0;
    END IF;
-- VALIDA Q SEA NUMERO
LN_S_61_90 :=

TRIM(SUBSTR(VL_NEW_LINE, INSTR(VL_NEW_LINE, ';' , 1, 19)+1, INSTR(VL_NEW_LINE
, ';' , 1, 20)-INSTR(VL_NEW_LINE, ';' , 1, 19)-1));
    IF BC_F_NUMERO(LN_S_61_90)=0 THEN
        VALIDA:= 0;

```

```

    END IF;
-- VALIDA Q SEA NUMERO
    LN_S_91_120 :=
    TRIM(SUBSTR(VL_NEW_LINE, INSTR(VL_NEW_LINE, ';', 1, 20)+1, INSTR(VL_NEW_LINE
, ';', 1, 21)-INSTR(VL_NEW_LINE, ';', 1, 20)-1));
    IF BC_F_NUMERO(LN_S_91_120)=0 THEN
        VALIDA:= 0;
    END IF;
-- VALIDA Q SEA NUMERO
    LN_S_MAS_120 :=
    TRIM(SUBSTR(VL_NEW_LINE, INSTR(VL_NEW_LINE, ';', 1, 21)+1, INSTR(VL_NEW_LINE
, ';', 1, 22)-INSTR(VL_NEW_LINE, ';', 1, 21)-1));
    IF BC_F_NUMERO(LN_S_MAS_120)=0 THEN
        VALIDA:= 0;
    END IF;
    END IF;
    RETURN VALIDA;
    DBMS_OUTPUT.PUT_LINE(VALIDA);
exception
when others then
    RETURN 0;
end BC_F_VERIFICA_ARCHIVO;

```

```

*****

```

VERIFICA ARCHIVO MASIVO

```

*****

```

```

create or replace function BC_F_VERIFICA_ARCHIVO_MASIVO(PV_LINEA
VARCHAR2) return number is

```

```

    VALIDA NUMBER:=1;
    L_CONT NUMBER(3):=0;
    L_CONT2 NUMBER(3):=0;
    LV_ENTIDAD ADMIN.BC_TITULAR.TIPO_TITULAR%TYPE;
    LV_NOMBRE ADMIN.BC_TITULAR.NOMBRE%TYPE;
    LV_APELLIDO ADMIN.BC_TITULAR.APELLIDO%TYPE;
    LV_TIPO_IDENT ADMIN.BC_TITULAR.TIPO_IDENTIFICACION%TYPE;
    LV_IDENTIFICACION ADMIN.BC_TITULAR.IDENTIFICACION%TYPE;
    LV_ACREEDOR ADMIN.BC_ACREEDOR.NOMBRE%TYPE;
    LV_IDENTIFICACION_ACREEDOR
ADMIN.BC_ACREEDOR.IDENTIFICACION%TYPE;
    LV_MONEDA ADMIN.BC_DETALLE_CREDITO.TIPO_MONEDA%TYPE;
    LN_CREDITO_APROBADO VARCHAR2(13);
    LN_MONTO_ADEUDADO VARCHAR2(13);
    LN_SALDO_NO_VENCIDO VARCHAR2(13);

```

```

LN_SALDO_VENCIDO VARCHAR2(13);
LD_FECHA_APERTURA ADMIN.BC_DETALLE_CREDITO.FECHA_APERTURA%TYPE;
LD_FECHA_VENCIMIENTO ADMIN.BC_PAGOS.FECHA_VENC%TYPE;
LN_S_30 VARCHAR2(13);
LN_S_31_60 VARCHAR2(13);
LN_S_61_90 VARCHAR2(13);
LN_S_91_120 VARCHAR2(13);
LN_S_MAS_120 VARCHAR2(13);
VL_NEW_LINE VARCHAR2(500):=PV_LINEA;

begin
  IF TRIM(VL_NEW_LINE) IS NULL THEN
    NULL;
  ELSE
    --VALIDA Q ENTIDAD SOLO SEA N O J
    LV_ENTIDAD :=
      TRIM(SUBSTR(VL_NEW_LINE,1, INSTR(VL_NEW_LINE, ';',1,1)-1));
    IF LV_ENTIDAD<>'N' AND LV_ENTIDAD<>'J' THEN
      VALIDA:= 0;
    END IF;
    --VALIDA Q EXISTA UN NOMBRE
    LV_NOMBRE :=

    TRIM(SUBSTR(VL_NEW_LINE, INSTR(VL_NEW_LINE, ';',1,1)+1, INSTR(VL_NEW_LINE,
    ';',1,2)-INSTR(VL_NEW_LINE, ';',1,1)-1));
    IF LV_NOMBRE IS NULL THEN
      VALIDA:= 0;
    END IF;
    --VALIDA Q EXISTA UN APELLIDO
    LV_APELLIDO :=

    TRIM(SUBSTR(VL_NEW_LINE, INSTR(VL_NEW_LINE, ';',1,2)+1, INSTR(VL_NEW_LINE,
    ';',1,3)-INSTR(VL_NEW_LINE, ';',1,2)-1));
    IF LV_APELLIDO IS NULL THEN
      VALIDA:= 0;
    END IF;
    --VALIDA TIPOS DE IDENTIFICACION ACEPTADOS RUC - CED (CEDULA) -
    PAS(PASAPORTE)
    LV_TIPO_IDENT :=

    TRIM(SUBSTR(VL_NEW_LINE, INSTR(VL_NEW_LINE, ';',1,3)+1, INSTR(VL_NEW_LINE,
    ';',1,4)-INSTR(VL_NEW_LINE, ';',1,3)-1));
    IF LV_TIPO_IDENT<>'CED' AND LV_TIPO_IDENT<>'RUC' AND
    LV_TIPO_IDENT<>'PAS' THEN
      VALIDA:= 0;
    END IF;
    --VALIDA IDENTIFICACIONES
    LV_IDENTIFICACION :=

    TRIM(SUBSTR(VL_NEW_LINE, INSTR(VL_NEW_LINE, ';',1,4)+1, INSTR(VL_NEW_LINE,

```



```

';',1,5)-INSTR(VL_NEW_LINE,';',1,4)-1));
  L_CONT:=LENGTH(LV_IDENTIFICACION);
  IF LV_TIPO_IDENT='CED' THEN
    IF L_CONT<>10 THEN
      VALIDA:= 0;
    END IF;
  ELSIF LV_TIPO_IDENT='RUC' THEN
    IF L_CONT<>13 THEN
      VALIDA:= 0;
    END IF;
  END IF;
LV_MONEDA :=
TRIM(SUBSTR(VL_NEW_LINE, INSTR(VL_NEW_LINE, ';', 1, 9)+1, INSTR(VL_NEW_LINE,
';', 1, 10)-INSTR(VL_NEW_LINE, ';', 1, 9)-1));
  IF LV_MONEDA<>'USD' AND LV_MONEDA<>'EUR' THEN
    VALIDA:= 0;
  END IF;
--VALIDA LA IDENTIFICACION DEL ACREEDOR
LV_IDENTIFICACION_ACREEDOR :=
TRIM(SUBSTR(VL_NEW_LINE, INSTR(VL_NEW_LINE, ';', 1, 6)+1, INSTR(VL_NEW_LINE,
';', 1, 7)-INSTR(VL_NEW_LINE, ';', 1, 6)-1));
  SELECT COUNT(*)
  INTO L_CONT2
  FROM ADMIN.BC_ACREEDOR
  WHERE IDENTIFICACION=LV_IDENTIFICACION_ACREEDOR;
  IF L_CONT2=0 THEN
    RETURN 2;
  END IF;
--END IF;
-- VALIDA Q SEA SOLO NUMEROS
LN_CREDITO_APROBADO :=
TRIM(SUBSTR(VL_NEW_LINE, INSTR(VL_NEW_LINE, ';', 1, 10)+1, INSTR(VL_NEW_LINE
, ';', 1, 11)-INSTR(VL_NEW_LINE, ';', 1, 10)-1));
  IF BC_F_NUMERO(LN_CREDITO_APROBADO)=0 THEN
    VALIDA:= 0;
  END IF;
-- VALIDA Q SEA SOLO NUMEROS
LN_MONTO_ADEUDADO :=
TRIM(SUBSTR(VL_NEW_LINE, INSTR(VL_NEW_LINE, ';', 1, 11)+1, INSTR(VL_NEW_LINE
, ';', 1, 12)-INSTR(VL_NEW_LINE, ';', 1, 11)-1));
  IF BC_F_NUMERO(LN_MONTO_ADEUDADO)=0 THEN
    VALIDA:= 0;
  END IF;
-- VALIDA Q SEA SOLO NUMEROS
LN_SALDO_NO_VENCIDO :=

```

```

TRIM(SUBSTR(VL_NEW_LINE, INSTR(VL_NEW_LINE, ';' , 1, 12)+1, INSTR(VL_NEW_LINE
, ';' , 1, 13)-INSTR(VL_NEW_LINE, ';' , 1, 12)-1));
    IF BC_F_NUMERO(LN_SALDO_NO_VENCIDO)=0 THEN
        VALIDA:= 0;
    END IF;
-- VALIDA Q SEA SOLO NUMEROS
LN_SALDO_VENCIDO :=

TRIM(SUBSTR(VL_NEW_LINE, INSTR(VL_NEW_LINE, ';' , 1, 13)+1, INSTR(VL_NEW_LINE
, ';' , 1, 14)-INSTR(VL_NEW_LINE, ';' , 1, 13)-1));
    IF BC_F_NUMERO(LN_SALDO_VENCIDO)=0 THEN
        VALIDA:= 0;
    END IF;
-- VALIDA Q NO CONTENGA LETRAS
LD_FECHA_APERTURA :=

TO_DATE(TRIM(SUBSTR(VL_NEW_LINE, INSTR(VL_NEW_LINE, ';' , 1, 14)+1, INSTR(VL_
NEW_LINE, ';' , 1, 15)-INSTR(VL_NEW_LINE, ';' , 1, 14)-1)), 'DD/MM/YY');
-- IF BC_F_NUMERO(LD_FECHA_APERTURA)=0 THEN
--     VALIDA:= 0;
-- END IF;
-- VALIDA Q NO CONTENGA LETRAS
LD_FECHA_VENCIMIENTO :=

TO_DATE(TRIM(SUBSTR(VL_NEW_LINE, INSTR(VL_NEW_LINE, ';' , 1, 15)+1, INSTR(VL_
NEW_LINE, ';' , 1, 16)-INSTR(VL_NEW_LINE, ';' , 1, 15)-1)), 'DD/MM/YYYY?');
-- IF BC_F_NUMERO(LD_FECHA_VENCIMIENTO)=0 THEN
--     VALIDA:= 0;
-- END IF;
--
-- LN_NUM_PAGOS :=
--
TO_NUMBER(TRIM(SUBSTR(VL_NEW_LINE, INSTR(VL_NEW_LINE, ';' , 1, 16)+1, INSTR(V
L_NEW_LINE, ';' , 1, 17)-INSTR(VL_NEW_LINE, ';' , 1, 16)-1)));
-- VALIDA Q SEA NUMERO
LN_S_30 :=

TRIM(SUBSTR(VL_NEW_LINE, INSTR(VL_NEW_LINE, ';' , 1, 17)+1, INSTR(VL_NEW_LINE
, ';' , 1, 18)-INSTR(VL_NEW_LINE, ';' , 1, 17)-1));
    IF BC_F_NUMERO(LN_S_30)=0 THEN
        VALIDA:= 0;
    END IF;
-- VALIDA Q SEA NUMERO
LN_S_31_60 :=

TRIM(SUBSTR(VL_NEW_LINE, INSTR(VL_NEW_LINE, ';' , 1, 18)+1, INSTR(VL_NEW_LINE
, ';' , 1, 19)-INSTR(VL_NEW_LINE, ';' , 1, 18)-1));
    IF BC_F_NUMERO(LN_S_31_60)=0 THEN

```

```

        VALIDA:= 0;
    END IF;
-- VALIDA Q SEA NUMERO
    LN_S_61_90 :=
    TRIM(SUBSTR(VL_NEW_LINE, INSTR(VL_NEW_LINE, ';' , 1, 19)+1, INSTR(VL_NEW_LINE
, ';' , 1, 20)-INSTR(VL_NEW_LINE, ';' , 1, 19)-1));
    IF BC_F_NUMERO(LN_S_61_90)=0 THEN
        VALIDA:= 0;
    END IF;
-- VALIDA Q SEA NUMERO
    LN_S_91_120 :=
    TRIM(SUBSTR(VL_NEW_LINE, INSTR(VL_NEW_LINE, ';' , 1, 20)+1, INSTR(VL_NEW_LINE
, ';' , 1, 21)-INSTR(VL_NEW_LINE, ';' , 1, 20)-1));
    IF BC_F_NUMERO(LN_S_91_120)=0 THEN
        VALIDA:= 0;
    END IF;
-- VALIDA Q SEA NUMERO
    LN_S_MAS_120 :=
    TRIM(SUBSTR(VL_NEW_LINE, INSTR(VL_NEW_LINE, ';' , 1, 21)+1, INSTR(VL_NEW_LINE
, ';' , 1, 22)-INSTR(VL_NEW_LINE, ';' , 1, 21)-1));
    IF BC_F_NUMERO(LN_S_MAS_120)=0 THEN
        VALIDA:= 0;
    END IF;
    END IF;
    RETURN VALIDA;
    DBMS_OUTPUT.PUT_LINE(VALIDA);
exception
    when others then
        RETURN 0;
end BC_F_VERIFICA_ARCHIVO_MASIVO;

```

VERIFICA RUC

```

create or replace function BC_F_VERIFICA_RUC(PV_RUC VARCHAR2) return
NUMBER is

    dato varchar2(13):=PV_RUC;

```

```
tmp_1 varchar2(2);  
tmp_2 number;  
acum number:=0;  
ultimo number;  
verifica number;  
  
lv_1 number;  
lv_2 number;  
lv_3 number;  
lv_4 number;  
lv_5 number;  
lv_6 number;  
lv_7 number;  
lv_8 number;  
lv_9 number;  
  
begin  
  
verifica:=to_number(SUBSTR(dato,10,1));  
lv_1:=to_number(SUBSTR(dato,9,1));  
lv_2:=to_number(SUBSTR(dato,8,1));  
lv_3:=to_number(SUBSTR(dato,7,1));  
lv_4:=to_number(SUBSTR(dato,6,1));  
lv_5:=to_number(SUBSTR(dato,5,1));  
lv_6:=to_number(SUBSTR(dato,4,1));  
lv_7:=to_number(SUBSTR(dato,3,1));  
lv_8:=to_number(SUBSTR(dato,2,1));
```

```
lv_9:=to_number(SUBSTR(dato,1,1));

if lv_7<6 then
  if lv_1*2>9 then
    tmp_1:=to_char(lv_1*2);
    tmp_2:=to_number(SUBSTR(tmp_1,1,1))+ to_number(SUBSTR(tmp_1,2,1));
    acum:=acum+tmp_2;
  else
    acum:=acum+(lv_1*2);
  end if;

  acum:=acum+lv_2;

  if lv_3*2>9 then
    tmp_1:=to_char(lv_3*2);
    tmp_2:=to_number(SUBSTR(tmp_1,1,1))+ to_number(SUBSTR(tmp_1,2,1));
    acum:=acum+tmp_2;
  else
    acum:=acum+(lv_3*2);
  end if;

  acum:=acum+lv_4;

  if lv_5*2>9 then
    tmp_1:=to_char(lv_5*2);
```

```
tmp_2:=to_number(SUBSTR(tmp_1,1,1))+ to_number(SUBSTR(tmp_1,2,1));
acum:=acum+tmp_2;
    else
        acum:=acum+(lv_5*2);
end if;

acum:=acum+lv_6;

if lv_7*2>9 then
tmp_1:=to_char(lv_7*2);
tmp_2:=to_number(SUBSTR(tmp_1,1,1))+ to_number(SUBSTR(tmp_1,2,1));
acum:=acum+tmp_2;
    else
        acum:=acum+(lv_7*2);
end if;

acum:=acum+lv_8;

if lv_9*2>9 then
tmp_1:=to_char(lv_9*2);
tmp_2:=to_number(SUBSTR(tmp_1,1,1))+ to_number(SUBSTR(tmp_1,2,1));
acum:=acum+tmp_2;
    else
        acum:=acum+(lv_9*2);
end if;

ultimo:=mod(acum,10);
```

```
if (10-ultimo)= verifica then
    dbms_output.put_line('RUC OK');
    RETURN 1;
else
    dbms_output.put_line('RUC incorrecto');
    RETURN 0;
end if;

elsif lv_7=9 then
    acum := ((lv_9*4)+(lv_8*3)+(lv_7*2)+(lv_6*7)+(lv_5*6)
            +(lv_4*5)+(lv_3*4)+(lv_2*3)+(lv_1*2));
    ultimo:=mod(acum,11);
    if ultimo = 0 or (11-ultimo)= verifica then
        dbms_output.put_line('RUC OK');
        RETURN 1;
    else
        dbms_output.put_line('RUC incorrecto');
        RETURN 0;
    end if;
elsif lv_7=6 then
    acum := ((lv_9*3)+(lv_8*2)+(lv_6*7)+(lv_5*6)
            +(lv_4*5)+(lv_3*4)+(lv_2*3)+(lv_1*2));
    ultimo:=mod(acum,11);
    if ultimo=0 or (11-ultimo)= verifica then
        dbms_output.put_line('RUC OK');
```

```

        RETURN 1;

    else

        dbms_output.put_line('RUC incorrecto');

    RETURN 0;

    end if;

end if;

end;

*****

VERIFICA CEDULA

*****

create or replace function BC_VERIFICA_CED(PV_CEDULA VARCHAR2) return
NUMBER is

    dato varchar2(10):=PV_CEDULA;

    tmp_1 varchar2(2);

    tmp_2 number;

    acum number:=0;

    ultimo number;

    verifica number;

    lv_1 number;

    lv_2 number;

    lv_3 number;

    lv_4 number;

    lv_5 number;

    lv_6 number;

```



```
lv_7 number;

lv_8 number;

lv_9 number;

begin

verifica:=to_number(SUBSTR(dato,10));

lv_1:=to_number(SUBSTR(dato,9,1));

lv_2:=to_number(SUBSTR(dato,8,1));

lv_3:=to_number(SUBSTR(dato,7,1));

lv_4:=to_number(SUBSTR(dato,6,1));

lv_5:=to_number(SUBSTR(dato,5,1));

lv_6:=to_number(SUBSTR(dato,4,1));

lv_7:=to_number(SUBSTR(dato,3,1));

lv_8:=to_number(SUBSTR(dato,2,1));

lv_9:=to_number(SUBSTR(dato,1,1));

if lv_1*2>9 then

tmp_1:=to_char(lv_1*2);

tmp_2:=to_number(SUBSTR(tmp_1,1,1))+ to_number(SUBSTR(tmp_1,2,1));

acum:=acum+tmp_2;

else

acum:=acum+(lv_1*2);

end if;

acum:=acum+lv_2;
```

```
if lv_3*2>9 then
tmp_1:=to_char(lv_3*2);
tmp_2:=to_number(SUBSTR(tmp_1,1,1))+ to_number(SUBSTR(tmp_1,2,1));
acum:=acum+tmp_2;
    else
        acum:=acum+(lv_3*2);
end if;
acum:=acum+lv_4;
if lv_5*2>9 then
tmp_1:=to_char(lv_5*2);
tmp_2:=to_number(SUBSTR(tmp_1,1,1))+ to_number(SUBSTR(tmp_1,2,1));
acum:=acum+tmp_2;
    else
        acum:=acum+(lv_5*2);
end if;
acum:=acum+lv_6;

if lv_7*2>9 then
tmp_1:=to_char(lv_7*2);
tmp_2:=to_number(SUBSTR(tmp_1,1,1))+ to_number(SUBSTR(tmp_1,2,1));
acum:=acum+tmp_2;
    else
        acum:=acum+(lv_7*2);
end if;
acum:=acum+lv_8;
```

```

if lv_9*2>9 then
tmp_1:=to_char(lv_9*2);
tmp_2:=to_number(SUBSTR(tmp_1,1,1))+ to_number(SUBSTR(tmp_1,2,1));
acum:=acum+tmp_2;
    else
        acum:=acum+(lv_9*2);
end if;

ultimo:=mod(acum,10);

if (10-ultimo)= verifica then
    dbms_output.put_line('CEDULA OK');
    RETURN 1;
else
    dbms_output.put_line('CEDULA incorrecta');
    RETURN 0;
end if;

end;

*****

ENCRYPTA TEXTO

*****

create or replace function F_ENCRYPTA_TEXT(var_dato in
varchar2,var_usuario in varchar2) RETURN NUMBER IS

```

```
RESULTADO NUMBER;

var_valor_encriptado LONG default NULL ;var_datol LONG default NULL;

var_llave varchar2(4000):='100100101';

var_dato_base varchar2(4000);

BEGIN

var_datol := rpad( var_dato, (trunc(length(var_dato)/8)+1)*8, chr(0));

dbms_obfuscation_toolkit.DESEncrypt
( input_string => var_datol,
key_string => var_llave,
encrypted_string=> var_valor_encriptado );

select clave into var_dato_base from bc_usuario where
usuario=var_usuario;

if utl_raw.cast_to_raw(var_valor_encriptado)=var_dato_base then
RESULTADO:=1;
else
resultado:=0;
end if;

RETURN(RESULTADO);

END;
```

PROCEDIMIENTOS:

```

CREATE OR REPLACE PROCEDURE BC_CALIFICA_AL_CARGAR IS
  x number;
  cursor pagos is select * from bc_pagos;
begin
  for regis in pagos LOOP
  x:=round(sysdate-regis.fecha_venc);
  if x>90 then
    update bc_pagos set calif_mes='D', estado='I'
      where id_titular=regis.id_titular and
            id_credito=regis.id_credito and
            num_pago=regis.num_pago;
  elsif x>60 then
    update bc_pagos set calif_mes='C', estado='I'
      where id_titular=regis.id_titular and
            id_credito=regis.id_credito and
            num_pago=regis.num_pago;
  elsif x>30 then
    update bc_pagos set calif_mes='B', estado='I'
      where id_titular=regis.id_titular and
            id_credito=regis.id_credito and
            num_pago=regis.num_pago;
  end if;
  end loop;
  commit;
end;
*****

```

ALTER USUARIO

```

*****

create or replace procedure BC_P_ALTER_USUARIO(PV_USUARIO IN VARCHAR2,
PV_CLAVE IN VARCHAR2,
PV_ERROR OUT VARCHAR2) is
begin
  EXECUTE IMMEDIATE 'ALTER USER ' || PV_USUARIO || ' IDENTIFIED BY
' || PV_CLAVE;
EXCEPTION
  WHEN OTHERS THEN

```

```

    PV_ERROR := SQLERRM;
end BC_P_ALTER_USUARIO;

```

```

*****

```

CARGA MASIVA

```

*****

```

```

create or replace procedure BC_P_CARGA_MASIVA_MERGE(PV_AÑO VARCHAR2,
                                                    PV_MES VARCHAR2,
                                                    PV_LINEA VARCHAR2)

```

```

IS

```

```

    LV_ENTIDAD ADMIN.BC_TITULAR.TIPO_TITULAR%TYPE;

```

```

    LV_NOMBRE ADMIN.BC_TITULAR.NOMBRE%TYPE;

```

```

    LV_APELLIDO ADMIN.BC_TITULAR.APELLIDO%TYPE;

```

```

    LV_RAZON_SOCIAL ADMIN.BC_TITULAR.RAZON_SOCIAL%TYPE;

```

```

    LV_NOM_COMERCIAL ADMIN.BC_TITULAR.NOMBRE_COMERCIAL%TYPE;

```

```

    LV_TIPO_IDENT ADMIN.BC_TITULAR.TIPO_IDENTIFICACION%TYPE;

```

```

    LV_IDENTIFICACION ADMIN.BC_TITULAR.IDENTIFICACION%TYPE;

```

```

    LV_ACREEDOR ADMIN.BC_ACREEDOR.NOMBRE%TYPE;

```

```

    LV_IDENTIFICACION_ACREEDOR

```

```

ADMIN.BC_ACREEDOR.IDENTIFICACION%TYPE;

```

```

    LV_NUM_CREDITO ADMIN.BC_CREDITO.ID_CREDITO%TYPE;

```

```

    LV_TIPO_CREDITO VARCHAR2(50);

```

```

    LV_MONEDA ADMIN.BC_DETALLE_CREDITO.TIPO_MONEDA%TYPE;

```

```

    LN_CREDITO_APROBADO ADMIN.BC_CREDITO.MONTO_APROBADO%TYPE;

```

```

    LN_MONTO_ADEUDADO ADMIN.BC_DETALLE_CREDITO.TOTAL_ADEUDADO%TYPE;

```

```

LN_SALDO_NO_VENCIDO ADMIN.BC_DETALLE_CREDITO.MONTO_VENCER%TYPE;
LN_SALDO_VENCIDO ADMIN.BC_DETALLE_CREDITO.MONTO_VENCIDO%TYPE;
LD_FECHA_APERTURA ADMIN.BC_DETALLE_CREDITO.FECHA_APERTURA%TYPE;
LD_FECHA_VENCIMIENTO ADMIN.BC_PAGOS.FECHA_VENC%TYPE;
LN_NUM_PAGOS ADMIN.BC_CREDITO.PLAZO_MESES%TYPE;

LN_S_30 NUMBER(11,2);
LN_S_31_60 NUMBER(11,2);
LN_S_61_90 NUMBER(11,2);
LN_S_91_120 NUMBER(11,2);
LN_S_MAS_120 NUMBER(11,2);

LN_ID_TIT ADMIN.BC_TITULAR.ID_TITULAR%TYPE:=0;
LN_ID_DET ADMIN.BC_DETALLE_CREDITO.ID_DETALLE_CRED%TYPE:=0;
LN_ID_PAG ADMIN.BC_PAGOS.ID_PAGO%TYPE:=0;

VL_NEW_LINE VARCHAR2(500):=PV_LINEA;

CONT1 NUMBER(3):=0; --PARA VERIFICAR SI EXISTE UN TITULAR O
CREDITO YA REGISTRADO

CONT2 NUMBER(3):=0; -- PARA VERIFICAR SI EXISTE DETALLE DE UN
TITULAR O NO

LN_PAGO_NUMERO ADMIN.BC_PAGOS.NUM_PAGO%TYPE:=0;

begin

IF TRIM(VL_NEW_LINE) IS NULL THEN

    NULL;

ELSE

LV_ENTIDAD :=

    TRIM(SUBSTR(VL_NEW_LINE,1, INSTR(VL_NEW_LINE, ';', 1, 1)-1));

```

LV_NOMBRE :=

TRIM(SUBSTR(VL_NEW_LINE, INSTR(VL_NEW_LINE, ';' , 1, 1)+1, INSTR(VL_NEW_LINE, ';' , 1, 2)-INSTR(VL_NEW_LINE, ';' , 1, 1)-1));

LV_APELLIDO :=

TRIM(SUBSTR(VL_NEW_LINE, INSTR(VL_NEW_LINE, ';' , 1, 2)+1, INSTR(VL_NEW_LINE, ';' , 1, 3)-INSTR(VL_NEW_LINE, ';' , 1, 2)-1));

LV_TIPO_IDENT :=

TRIM(SUBSTR(VL_NEW_LINE, INSTR(VL_NEW_LINE, ';' , 1, 3)+1, INSTR(VL_NEW_LINE, ';' , 1, 4)-INSTR(VL_NEW_LINE, ';' , 1, 3)-1));

LV_IDENTIFICACION :=

TRIM(SUBSTR(VL_NEW_LINE, INSTR(VL_NEW_LINE, ';' , 1, 4)+1, INSTR(VL_NEW_LINE, ';' , 1, 5)-INSTR(VL_NEW_LINE, ';' , 1, 4)-1));

LV_ACREEDOR :=

TRIM(SUBSTR(VL_NEW_LINE, INSTR(VL_NEW_LINE, ';' , 1, 5)+1, INSTR(VL_NEW_LINE, ';' , 1, 6)-INSTR(VL_NEW_LINE, ';' , 1, 5)-1));

LV_IDENTIFICACION_ACREEDOR :=

TRIM(SUBSTR(VL_NEW_LINE, INSTR(VL_NEW_LINE, ';' , 1, 6)+1, INSTR(VL_NEW_LINE, ';' , 1, 7)-INSTR(VL_NEW_LINE, ';' , 1, 6)-1));

LV_NUM_CREDITO :=


```

TRIM(SUBSTR(VL_NEW_LINE, INSTR(VL_NEW_LINE, ';', 1, 7)+1, INSTR(VL_NEW_LINE,
';', 1, 8)-INSTR(VL_NEW_LINE, ';', 1, 7)-1));

```

```

LV_TIPO_CREDITO :=

```

```

TRIM(SUBSTR(VL_NEW_LINE, INSTR(VL_NEW_LINE, ';', 1, 8)+1, INSTR(VL_NEW_LINE,
';', 1, 9)-INSTR(VL_NEW_LINE, ';', 1, 8)-1));

```

```

LV_MONEDA :=

```

```

TRIM(SUBSTR(VL_NEW_LINE, INSTR(VL_NEW_LINE, ';', 1, 9)+1, INSTR(VL_NEW_LINE,
';', 1, 10)-INSTR(VL_NEW_LINE, ';', 1, 9)-1));

```

```

LN_CREDITO_APROBADO :=

```

```

TO_NUMBER(TRIM(SUBSTR(VL_NEW_LINE, INSTR(VL_NEW_LINE, ';', 1, 10)+1, INSTR(V
L_NEW_LINE, ';', 1, 11)-INSTR(VL_NEW_LINE, ';', 1, 10)-1)));

```

```

LN_MONTO_ADEUDADO :=

```

```

TO_NUMBER(TRIM(SUBSTR(VL_NEW_LINE, INSTR(VL_NEW_LINE, ';', 1, 11)+1, INSTR(V
L_NEW_LINE, ';', 1, 12)-INSTR(VL_NEW_LINE, ';', 1, 11)-1)));

```

```

LN_SALDO_NO_VENCIDO :=

```

```

TO_NUMBER(TRIM(SUBSTR(VL_NEW_LINE, INSTR(VL_NEW_LINE, ';', 1, 12)+1, INSTR(V
L_NEW_LINE, ';', 1, 13)-INSTR(VL_NEW_LINE, ';', 1, 12)-1)));

```

```

LN_SALDO_VENCIDO :=

```

```
TO_NUMBER( TRIM( SUBSTR( VL_NEW_LINE, INSTR( VL_NEW_LINE, ';' , 1, 13) + 1, INSTR( V
L_NEW_LINE, ';' , 1, 14) - INSTR( VL_NEW_LINE, ';' , 1, 13) - 1 ) ) );
```

```
LD_FECHA_APERTURA :=
```

```
TO_DATE( TRIM( SUBSTR( VL_NEW_LINE, INSTR( VL_NEW_LINE, ';' , 1, 14) + 1, INSTR( VL_
NEW_LINE, ';' , 1, 15) - INSTR( VL_NEW_LINE, ';' , 1, 14) - 1 ) ) , 'DD/MM/YY' );
```

```
LD_FECHA_VENCIMIENTO :=
```

```
TO_DATE( TRIM( SUBSTR( VL_NEW_LINE, INSTR( VL_NEW_LINE, ';' , 1, 15) + 1, INSTR( VL_
NEW_LINE, ';' , 1, 16) - INSTR( VL_NEW_LINE, ';' , 1, 15) - 1 ) ) , 'DD/MM/YYYY?' );
```

```
LN_NUM_PAGOS :=
```

```
TO_NUMBER( TRIM( SUBSTR( VL_NEW_LINE, INSTR( VL_NEW_LINE, ';' , 1, 16) + 1, INSTR( V
L_NEW_LINE, ';' , 1, 17) - INSTR( VL_NEW_LINE, ';' , 1, 16) - 1 ) ) );
```

```
LN_S_30 :=
```

```
TO_NUMBER( TRIM( SUBSTR( VL_NEW_LINE, INSTR( VL_NEW_LINE, ';' , 1, 17) + 1, INSTR( V
L_NEW_LINE, ';' , 1, 18) - INSTR( VL_NEW_LINE, ';' , 1, 17) - 1 ) ) );
```

```
LN_S_31_60 :=
```

```
TO_NUMBER( TRIM( SUBSTR( VL_NEW_LINE, INSTR( VL_NEW_LINE, ';' , 1, 18) + 1, INSTR( V
L_NEW_LINE, ';' , 1, 19) - INSTR( VL_NEW_LINE, ';' , 1, 18) - 1 ) ) );
```

```
LN_S_61_90 :=
```

```
TO_NUMBER( TRIM( SUBSTR( VL_NEW_LINE, INSTR( VL_NEW_LINE, ';' , 1, 19) + 1, INSTR( V
```

```
L_NEW_LINE, ';' , 1, 20) - INSTR(VL_NEW_LINE, ';' , 1, 19) - 1));
```

```
LN_S_91_120 :=
```

```
TO_NUMBER( TRIM( SUBSTR( VL_NEW_LINE, INSTR( VL_NEW_LINE, ';' , 1, 20) + 1, INSTR( V
L_NEW_LINE, ';' , 1, 21) - INSTR( VL_NEW_LINE, ';' , 1, 20) - 1) ));
```

```
LN_S_MAS_120 :=
```

```
TO_NUMBER( TRIM( SUBSTR( VL_NEW_LINE, INSTR( VL_NEW_LINE, ';' , 1, 21) + 1, INSTR( V
L_NEW_LINE, ';' , 1, 22) - INSTR( VL_NEW_LINE, ';' , 1, 21) - 1) ));
```

```
LV_RAZON_SOCIAL :=
```

```
TRIM( SUBSTR( VL_NEW_LINE, INSTR( VL_NEW_LINE, ';' , 1, 22) + 1, INSTR( VL_NEW_LINE
, ';' , 1, 23) - INSTR( VL_NEW_LINE, ';' , 1, 22) - 1) ));
```

```
LV_NOM_COMERCIAL :=
```

```
TRIM( SUBSTR( VL_NEW_LINE, INSTR( VL_NEW_LINE, ';' , 1, 23) + 1, INSTR( VL_NEW_LINE
, ';' , 1, 24) - INSTR( VL_NEW_LINE, ';' , 1, 23) - 1) ));
```

```
IF (LV_ENTIDAD IS NOT NULL AND LV_NOMBRE IS NOT NULL AND
LV_APELLIDO IS NOT NULL AND LV_TIPO_IDENT IS NOT NULL AND
LV_IDENTIFICACION IS NOT NULL AND LV_NUM_CREDITO IS NOT NULL
AND
LV_ACREEDOR IS NOT NULL AND LV_IDENTIFICACION_ACREEDOR IS NOT
NULL AND
```

```

LV_MONEDA IS NOT NULL AND LN_MONTO_ADEUDADO IS NOT NULL) THEN

DBMS_OUTPUT.PUT_LINE( 'DATOS COMPLETOS' );

--##### INSERCION A TABLA BC_TITULAR CON DATOS DEL ARCHIVO
Y VALORES POR DEFECTO #####

IF LV_ENTIDAD='N' THEN

--PERSONAS NATURALES

SELECT COUNT(*) INTO CONT1 FROM ADMIN.BC_TITULAR WHERE
IDENTIFICACION=LV_IDENTIFICACION;

IF CONT1=0 THEN

SELECT NVL(MAX(ID_TITULAR), 0)+1

INTO LN_ID_TIT

FROM ADMIN.BC_TITULAR;

INSERT INTO ADMIN.BC_TITULAR(ID_TITULAR,

TIPO_TITULAR,

TIPO_IDENTIFICACION,

IDENTIFICACION,

NOMBRE,

APELLIDO,

FECHA_INGRESO,

ESTADO,

HABIL_CTA_CTE_CH,

ID_CALIFICACION)

VALUES(LN_ID_TIT,

LV_ENTIDAD,

```

```
        LV_TIPO_IDENT,  
        LV_IDENTIFICACION,  
        LV_NOMBRE,  
        LV_APELLIDO,  
        SYSDATE,  
        'A' ,  
        'SI' ,  
        'A' );  
  
ELSE  
  
    SELECT ID_TITULAR  
    INTO LN_ID_TIT  
    FROM ADMIN.BC_TITULAR  
    WHERE IDENTIFICACION=LV_IDENTIFICACION;  
  
    UPDATE ADMIN.BC_TITULAR  
    SET NOMBRE=LV_NOMBRE,  
        APELLIDO=LV_APELLIDO,  
        ESTADO='A' ,  
        HABIL_CTA_CTE_CH='SI'--,  
        --ID_CALIFICACION='A'  
    WHERE IDENTIFICACION=LV_IDENTIFICACION;  
  
END IF;  
  
CONT1:=0;  
  
ELSE  
  
    --PERSONAS JURIDICAS
```

```
SELECT COUNT(*) INTO CONT1 FROM ADMIN.BC_TITULAR WHERE
IDENTIFICACION=LV_IDENTIFICACION;

IF CONT1=0 THEN

SELECT NVL(MAX(ID_TITULAR),0)+1

INTO LN_ID_TIT

FROM ADMIN.BC_TITULAR;

INSERT INTO ADMIN.BC_TITULAR(ID_TITULAR,

TIPO_TITULAR,

TIPO_IDENTIFICACION,

IDENTIFICACION,

NOMBRE,

APELLIDO,

NOMBRE_COMERCIAL,

RAZON_SOCIAL,

FECHA_INGRESO,

ESTADO,

HABIL_CTA_CTE_CH,

ID_CALIFICACION)

VALUES(LN_ID_TIT,

LV_ENTIDAD,

LV_TIPO_IDENT,

LV_IDENTIFICACION,

LV_NOMBRE,

LV_APELLIDO,

LV_NOM_COMERCIAL,
```

```
        LV_RAZON_SOCIAL,
        SYSDATE,
        'A' ,
        'SI' ,
        'A' );
ELSE
    SELECT ID_TITULAR
    INTO LN_ID_TIT
    FROM ADMIN.BC_TITULAR
    WHERE IDENTIFICACION=LV_IDENTIFICACION;

    UPDATE ADMIN.BC_TITULAR
    SET NOMBRE=LV_NOMBRE,
        APELLIDO=LV_APELLIDO,
        NOMBRE_COMERCIAL=LV_NOM_COMERCIAL,
        RAZON_SOCIAL=LV_RAZON_SOCIAL,
        ESTADO='A' ,
        HABIL_CTA_CTE_CH='SI'--,
        --ID_CALIFICACION='A'

    WHERE IDENTIFICACION=LV_IDENTIFICACION;

END IF;

CONT1:=0;

END IF;

--##### INSERCIÓN A TABLA BC_CREDITO CON DATOS DEL ARCHIVO
```

Y VALORES POR DEFECTO #####

```

SELECT COUNT(*) INTO CONT1 FROM ADMIN.BC_CREDITO
WHERE ID_CREDITO=LV_NUM_CREDITO AND
IDENT_ACREEDOR=LV_IDENTIFICACION_ACREEDOR; --AND ID_TITULAR=LN_ID_TIT;

IF CONT1=0 THEN

INSERT INTO ADMIN.BC_CREDITO(ID_CREDITO,
                                ID_TIPO_CRED,
                                IDENT_ACREEDOR,
                                ID_TITULAR,
                                FECHA_ULT_ACT,
                                MONTO_APROBADO,
                                PLAZO_MESES)

VALUES(LV_NUM_CREDITO,
        LV_TIPO_CREDITO,
        LV_IDENTIFICACION_ACREEDOR,
        LN_ID_TIT,
        SYSDATE,
        LN_CREDITO_APROBADO,
        LN_NUM_PAGOS);

```

--##### INSERCION A TABLA BC_DETALLE_CREDITO CON DATOS DEL ARCHIVO Y VALORES POR DEFECTO #####

```

SELECT NVL(MAX(ID_DETALLE_CRED),0)+1 INTO LN_ID_DET
FROM ADMIN.BC_DETALLE_CREDITO
WHERE ID_CREDITO=LV_NUM_CREDITO AND
IDENT_ACREEDOR=LV_IDENTIFICACION_ACREEDOR AND ID_TITULAR=LN_ID_TIT;

```



```
INSERT INTO
ADMIN.BC_DETALLE_CREDITO( ID_DETALLE_CRED, ID_CREDITO,
IDENT_ACREEDOR, ID_TITULAR, FECHA_APERTURA, FECHA_REPORTADO,
TIPO_MONEDA, TOTAL_ADEUDADO, MONTO_VENCER, MONTO_VENCIDO,
CALIF_GENERAL, ESTADO, SALDO_30, SALDO_31_60, SALDO_61_90,
SALDO_91_120, SALDO_MAS_120)
VALUES( LN_ID_DET,
LV_NUM_CREDITO,
LV_IDENTIFICACION_ACREEDOR,
LN_ID_TIT,
LD_FECHA_APERTURA,
SYSDATE,
LV_MONEDA,
LN_MONTO_ADEUDADO,
LN_SALDO_NO_VENCIDO,
LN_SALDO_VENCIDO,
'A',
'A',
LN_S_30,
LN_S_31_60,
LN_S_61_90,
LN_S_91_120,
LN_S_MAS_120);
```

ELSE

*-- SI YA EXISTE CREDITO SE ACTUALIZAN DATOS EN TABLA CREDITO Y TABLA
DETALLE CREDITO*

-- UPDATE CREDITO

UPDATE ADMIN.BC_CREDITO

SET FECHA_ULT_ACT=**SYSDATE**,

MONTO_APROBADO=LN_CREDITO_APROBADO,

PLAZO_MESES=LN_NUM_PAGOS

WHERE ID_CREDITO=LV_NUM_CREDITO

AND IDENT_ACREEDOR=LV_IDENTIFICACION_ACREEDOR;

--AND ID_TITULAR=LN_ID_TIT;

--UPDATE DETALLE_CREDITO

UPDATE ADMIN.BC_DETALLE_CREDITO

SET TIPO_MONEDA=LV_MONEDA,

TOTAL_ADEUDADO=LN_MONTO_ADEUDADO,

MONTO_VENCER=LN_SALDO_NO_VENCIDO,

MONTO_VENCIDO=LN_SALDO_VENCIDO,

-- CALIF_GENERAL=' A' ,

ESTADO='A' ,

SALDO_30=LN_S_30,

SALDO_31_60=LN_S_31_60,

SALDO_61_90=LN_S_61_90,

SALDO_91_120=LN_S_91_120,

SALDO_MAS_120=LN_S_MAS_120

WHERE ID_CREDITO=LV_NUM_CREDITO

```

        AND IDENT_ACREEDOR=LV_IDENTIFICACION_ACREEDOR

        AND ID_TITULAR=LN_ID_TIT;

    END IF;

    CONT1:=0;

--##### INSERCION A TABLA BC_PAGOS CON DATOS DEL ARCHIVO Y
VALORES POR DEFECTO #####

    SELECT NVL(MAX(NUM_PAGO),0)+1

    INTO LN_PAGO_NUMERO

    FROM ADMIN.BC_PAGOS WHERE ID_CREDITO=LV_NUM_CREDITO AND

IDENT_ACREEDOR=LV_IDENTIFICACION_ACREEDOR AND ID_TITULAR=LN_ID_TIT;

    SELECT NVL(MAX(ID_PAGO),0)+1

    INTO LN_ID_PAG

    FROM ADMIN.BC_PAGOS WHERE ID_CREDITO=LV_NUM_CREDITO AND

IDENT_ACREEDOR=LV_IDENTIFICACION_ACREEDOR AND ID_TITULAR=LN_ID_TIT;

    INSERT INTO ADMIN.BC_PAGOS(ID_PAGO,

                                IDENT_ACREEDOR,

                                ID_TITULAR,

                                ID_CREDITO,

                                NUM_PAGO,

                                FECHA_VENC,

                                ESTADO,

                                AÑO,

                                MES,

                                CALIF_MES,

```

```

                SALDO_DEUDA,
                FECHA_CARGA)

VALUES (LN_ID_PAG,
        LV_IDENTIFICACION_ACREEDOR,
        LN_ID_TIT,
        LV_NUM_CREDITO,
        LN_PAGO_NUMERO,
        LD_FECHA_VENCIMIENTO,
        'P',
        PV_AÑO,
        PV_MES,
        'A',
        LN_MONTO_ADEUDADO,
        SYSDATE);

    END IF;

END IF;

--      EXCEPTION
--      WHEN OTHERS THEN

end BC_P_CARGA_MASIVA_MERGE;

*****

CARGA

*****

create or replace procedure BC_P_CARGA_MERGE(PV_AÑOVARCHAR2, PV_MES
VARCHAR2, PV_LINEA VARCHAR2, PV_ACREEDOR VARCHAR2)
IS

```

```

LV_ENTIDAD ADMIN.BC_TITULAR.TIPO_TITULAR%TYPE;
LV_NOMBRE ADMIN.BC_TITULAR.NOMBRE%TYPE;
LV_APELLIDO ADMIN.BC_TITULAR.APELLIDO%TYPE;
LV_RAZON_SOCIAL ADMIN.BC_TITULAR.RAZON_SOCIAL%TYPE;
LV_NOM_COMERCIAL ADMIN.BC_TITULAR.NOMBRE_COMERCIAL%TYPE;
LV_TIPO_IDENT ADMIN.BC_TITULAR.TIPO_IDENTIFICACION%TYPE;
LV_IDENTIFICACION ADMIN.BC_TITULAR.IDENTIFICACION%TYPE;
LV_DIRECCION VARCHAR2(60);
LV_TELEFONO VARCHAR2(30);
LV_NUM_CREDITO ADMIN.BC_CREDITO.ID_CREDITO%TYPE;
LV_TIPO_CREDITO VARCHAR2(50);
LV_MONEDA ADMIN.BC_DETALLE_CREDITO.TIPO_MONEDA%TYPE;
LN_CREDITO_APROBADO ADMIN.BC_CREDITO.MONTO_APROBADO%TYPE;
LN_MONTO_ADEUDADO ADMIN.BC_DETALLE_CREDITO.TOTAL_ADEUDADO%TYPE;
LN_SALDO_NO_VENCIDO ADMIN.BC_DETALLE_CREDITO.MONTO_VENCER%TYPE;
LN_SALDO_VENCIDO ADMIN.BC_DETALLE_CREDITO.MONTO_VENCIDO%TYPE;
LD_FECHA_APERTURA ADMIN.BC_DETALLE_CREDITO.FECHA_APERTURA%TYPE;
LD_FECHA_VENCIMIENTO ADMIN.BC_PAGOS.FECHA_VENC%TYPE;
LN_NUM_PAGOS ADMIN.BC_CREDITO.PLAZO_MESES%TYPE;
LN_S_30 NUMBER(11,2);
LN_S_31_60 NUMBER(11,2);
LN_S_61_90 NUMBER(11,2);
LN_S_91_120 NUMBER(11,2);
LN_S_MAS_120 NUMBER(11,2);
LN_ID_TIT ADMIN.BC_TITULAR.ID_TITULAR%TYPE:=0;
LN_ID_DET ADMIN.BC_DETALLE_CREDITO.ID_DETALLE_CRED%TYPE:=0;
LN_ID_PAG ADMIN.BC_PAGOS.ID_PAGO%TYPE:=0;
VL_NEW_LINE VARCHAR2(500):=PV_LINEA;
CONT1 NUMBER(3):=0; --PARA VERIFICAR SI EXISTE UN TITULAR O
CREDITO YA REGISTRADO
CONT2 NUMBER(3):=0; -- PARA VERIFICAR SI EXISTE DETALLE DE UN
TITULAR O NO
LN_PAGO_NUMERO ADMIN.BC_PAGOS.NUM_PAGO%TYPE:=0;
begin
  IF TRIM(VL_NEW_LINE) IS NULL THEN
    NULL;
  ELSE
    LV_ENTIDAD :=
      TRIM(SUBSTR(VL_NEW_LINE,1,INSTR(VL_NEW_LINE,';',1,1)-1));
    LV_NOMBRE :=
      TRIM(SUBSTR(VL_NEW_LINE,INSTR(VL_NEW_LINE,';',1,1)+1,INSTR(VL_NEW_LINE,
';',1,2)-INSTR(VL_NEW_LINE,';',1,1)-1));
    LV_APELLIDO :=
      TRIM(SUBSTR(VL_NEW_LINE,INSTR(VL_NEW_LINE,';',1,2)+1,INSTR(VL_NEW_LINE,
';',1,3)-INSTR(VL_NEW_LINE,';',1,2)-1));
    LV_TIPO_IDENT :=

```

```

TRIM(SUBSTR(VL_NEW_LINE, INSTR(VL_NEW_LINE, ';' , 1, 3)+1, INSTR(VL_NEW_LINE,
';' , 1, 4)-INSTR(VL_NEW_LINE, ';' , 1, 3)-1));
LV_IDENTIFICACION :=

TRIM(SUBSTR(VL_NEW_LINE, INSTR(VL_NEW_LINE, ';' , 1, 4)+1, INSTR(VL_NEW_LINE,
';' , 1, 5)-INSTR(VL_NEW_LINE, ';' , 1, 4)-1));
LV_DIRECCION :=

TRIM(SUBSTR(VL_NEW_LINE, INSTR(VL_NEW_LINE, ';' , 1, 5)+1, INSTR(VL_NEW_LINE,
';' , 1, 6)-INSTR(VL_NEW_LINE, ';' , 1, 5)-1));
LV_TELEFONO :=

TRIM(SUBSTR(VL_NEW_LINE, INSTR(VL_NEW_LINE, ';' , 1, 6)+1, INSTR(VL_NEW_LINE,
';' , 1, 7)-INSTR(VL_NEW_LINE, ';' , 1, 6)-1));
LV_NUM_CREDITO :=

TRIM(SUBSTR(VL_NEW_LINE, INSTR(VL_NEW_LINE, ';' , 1, 7)+1, INSTR(VL_NEW_LINE,
';' , 1, 8)-INSTR(VL_NEW_LINE, ';' , 1, 7)-1));
LV_TIPO_CREDITO :=

TRIM(SUBSTR(VL_NEW_LINE, INSTR(VL_NEW_LINE, ';' , 1, 8)+1, INSTR(VL_NEW_LINE,
';' , 1, 9)-INSTR(VL_NEW_LINE, ';' , 1, 8)-1));
LV_MONEDA :=

TRIM(SUBSTR(VL_NEW_LINE, INSTR(VL_NEW_LINE, ';' , 1, 9)+1, INSTR(VL_NEW_LINE,
';' , 1, 10)-INSTR(VL_NEW_LINE, ';' , 1, 9)-1));
LN_CREDITO_APROBADO :=

TO_NUMBER(TRIM(SUBSTR(VL_NEW_LINE, INSTR(VL_NEW_LINE, ';' , 1, 10)+1, INSTR(V
L_NEW_LINE, ';' , 1, 11)-INSTR(VL_NEW_LINE, ';' , 1, 10)-1)));
LN_MONTO_ADEUDADO :=

TO_NUMBER(TRIM(SUBSTR(VL_NEW_LINE, INSTR(VL_NEW_LINE, ';' , 1, 11)+1, INSTR(V
L_NEW_LINE, ';' , 1, 12)-INSTR(VL_NEW_LINE, ';' , 1, 11)-1)));
LN_SALDO_NO_VENCIDO :=

TO_NUMBER(TRIM(SUBSTR(VL_NEW_LINE, INSTR(VL_NEW_LINE, ';' , 1, 12)+1, INSTR(V
L_NEW_LINE, ';' , 1, 13)-INSTR(VL_NEW_LINE, ';' , 1, 12)-1)));
LN_SALDO_VENCIDO :=

TO_NUMBER(TRIM(SUBSTR(VL_NEW_LINE, INSTR(VL_NEW_LINE, ';' , 1, 13)+1, INSTR(V
L_NEW_LINE, ';' , 1, 14)-INSTR(VL_NEW_LINE, ';' , 1, 13)-1)));
LD_FECHA_APERTURA :=

TO_DATE(TRIM(SUBSTR(VL_NEW_LINE, INSTR(VL_NEW_LINE, ';' , 1, 14)+1, INSTR(VL_
NEW_LINE, ';' , 1, 15)-INSTR(VL_NEW_LINE, ';' , 1, 14)-1)), 'DD/MM/YY');
LD_FECHA_VENCIMIENTO :=

TO_DATE(TRIM(SUBSTR(VL_NEW_LINE, INSTR(VL_NEW_LINE, ';' , 1, 15)+1, INSTR(VL_

```

```

NEW_LINE, ';' , 1, 16) - INSTR(VL_NEW_LINE, ';' , 1, 15) - 1) , 'DD/MM/YYYY?');
LN_NUM_PAGOS :=

TO_NUMBER( TRIM( SUBSTR( VL_NEW_LINE, INSTR( VL_NEW_LINE, ';' , 1, 16) + 1, INSTR( V
L_NEW_LINE, ';' , 1, 17) - INSTR( VL_NEW_LINE, ';' , 1, 16) - 1) ) );
LN_S_30 :=

TO_NUMBER( TRIM( SUBSTR( VL_NEW_LINE, INSTR( VL_NEW_LINE, ';' , 1, 17) + 1, INSTR( V
L_NEW_LINE, ';' , 1, 18) - INSTR( VL_NEW_LINE, ';' , 1, 17) - 1) ) );
LN_S_31_60 :=

TO_NUMBER( TRIM( SUBSTR( VL_NEW_LINE, INSTR( VL_NEW_LINE, ';' , 1, 18) + 1, INSTR( V
L_NEW_LINE, ';' , 1, 19) - INSTR( VL_NEW_LINE, ';' , 1, 18) - 1) ) );
LN_S_61_90 :=

TO_NUMBER( TRIM( SUBSTR( VL_NEW_LINE, INSTR( VL_NEW_LINE, ';' , 1, 19) + 1, INSTR( V
L_NEW_LINE, ';' , 1, 20) - INSTR( VL_NEW_LINE, ';' , 1, 19) - 1) ) );
LN_S_91_120 :=

TO_NUMBER( TRIM( SUBSTR( VL_NEW_LINE, INSTR( VL_NEW_LINE, ';' , 1, 20) + 1, INSTR( V
L_NEW_LINE, ';' , 1, 21) - INSTR( VL_NEW_LINE, ';' , 1, 20) - 1) ) );
LN_S_MAS_120 :=

TO_NUMBER( TRIM( SUBSTR( VL_NEW_LINE, INSTR( VL_NEW_LINE, ';' , 1, 21) + 1, INSTR( V
L_NEW_LINE, ';' , 1, 22) - INSTR( VL_NEW_LINE, ';' , 1, 21) - 1) ) );
LV_RAZON_SOCIAL :=

TRIM( SUBSTR( VL_NEW_LINE, INSTR( VL_NEW_LINE, ';' , 1, 22) + 1, INSTR( VL_NEW_LINE
, ';' , 1, 23) - INSTR( VL_NEW_LINE, ';' , 1, 22) - 1) ) );
LV_NOM_COMERCIAL :=

TRIM( SUBSTR( VL_NEW_LINE, INSTR( VL_NEW_LINE, ';' , 1, 23) + 1, INSTR( VL_NEW_LINE
, ';' , 1, 24) - INSTR( VL_NEW_LINE, ';' , 1, 23) - 1) ) );

IF (LV_ENTIDAD IS NOT NULL AND LV_NOMBRE IS NOT NULL AND
LV_APELLIDO IS NOT NULL AND LV_TIPO_IDENT IS NOT NULL AND
LV_IDENTIFICACION IS NOT NULL AND LV_NUM_CREDITO IS NOT NULL
AND
LV_MONEDA IS NOT NULL AND LN_MONTO_ADEUDADO IS NOT NULL) THEN
DBMS_OUTPUT.PUT_LINE( 'DATOS COMPLETOS' );
--##### INSERCION A TABLA BC_TITULAR CON DATOS DEL ARCHIVO
Y VALORES POR DEFECTO #####
IF LV_ENTIDAD='N' THEN
--PERSONAS NATURALES
SELECT COUNT(*) INTO CONT1 FROM ADMIN.BC_TITULAR WHERE
IDENTIFICACION=LV_IDENTIFICACION;
IF CONT1=0 THEN
SELECT NVL( MAX( ID_TITULAR) , 0) + 1

```

```

INTO LN_ID_TIT
FROM ADMIN.BC_TITULAR;
INSERT INTO ADMIN.BC_TITULAR(ID_TITULAR,
                             TIPO_TITULAR,
                             TIPO_IDENTIFICACION,
                             IDENTIFICACION,
                             NOMBRE,
                             APELLIDO,
                             FECHA_INGRESO,
                             ESTADO,
                             HABIL_CTA_CTE_CH,
                             ID_CALIFICACION)
VALUES(LN_ID_TIT,
       LV_ENTIDAD,
       LV_TIPO_IDENT,
       LV_IDENTIFICACION,
       LV_NOMBRE,
       LV_APELLIDO,
       SYSDATE,
       'A',
       'SI',
       'A');
ELSE
SELECT ID_TITULAR
INTO LN_ID_TIT
FROM ADMIN.BC_TITULAR
WHERE IDENTIFICACION=LV_IDENTIFICACION;

UPDATE ADMIN.BC_TITULAR
SET NOMBRE=LV_NOMBRE,
    APELLIDO=LV_APELLIDO,
    ESTADO='A',
    HABIL_CTA_CTE_CH='SI'--,
    --ID_CALIFICACION='A'
WHERE IDENTIFICACION=LV_IDENTIFICACION;
END IF;
CONT1:=0;
ELSE
--PERSONAS JURIDICAS
SELECT COUNT(*) INTO CONT1 FROM ADMIN.BC_TITULAR WHERE
IDENTIFICACION=LV_IDENTIFICACION;
IF CONT1=0 THEN
SELECT NVL(MAX(ID_TITULAR),0)+1
INTO LN_ID_TIT
FROM ADMIN.BC_TITULAR;
INSERT INTO ADMIN.BC_TITULAR(ID_TITULAR,
                             TIPO_TITULAR,
                             TIPO_IDENTIFICACION,
                             IDENTIFICACION,

```



```

NOMBRE,
APELLIDO,
NOMBRE_COMERCIAL,
RAZON_SOCIAL,
FECHA_INGRESO,
ESTADO,
HABIL_CTA_CTE_CH,
ID_CALIFICACION)

VALUES(LN_ID_TIT,
       LV_ENTIDAD,
       LV_TIPO_IDENT,
       LV_IDENTIFICACION,
       LV_NOMBRE,
       LV_APELLIDO,
       LV_NOM_COMERCIAL,
       LV_RAZON_SOCIAL,
       SYSDATE,
       'A',
       'SI',
       'A');

ELSE
SELECT ID_TITULAR
INTO LN_ID_TIT
FROM ADMIN.BC_TITULAR
WHERE IDENTIFICACION=LV_IDENTIFICACION;

UPDATE ADMIN.BC_TITULAR
SET NOMBRE=LV_NOMBRE,
    APELLIDO=LV_APELLIDO,
    NOMBRE_COMERCIAL=LV_NOM_COMERCIAL,
    RAZON_SOCIAL=LV_RAZON_SOCIAL,
    ESTADO='A',
    HABIL_CTA_CTE_CH='SI'--,
--      ID_CALIFICACION='A'
WHERE IDENTIFICACION=LV_IDENTIFICACION;
END IF;
CONT1:=0;
END IF;
--#####INSERCION A TABLA BC_DETALLE_TITULAR
SELECT COUNT(*) INTO CONT2
FROM BC_DETALLE_TITULAR
WHERE ID_TITULAR=LN_ID_TIT;
IF CONT2=0 AND (LV_DIRECCION IS NOT NULL OR LV_TELEFONO IS
NOT NULL) THEN
INSERT INTO
BC_DETALLE_TITULAR( ID_TITULAR,DOMICILIO,TELEFONO)
VALUES(LN_ID_TIT, LV_DIRECCION, LV_TELEFONO);
ELSE
IF LV_DIRECCION IS NOT NULL AND LV_TELEFONO IS NOT NULL

```

```

THEN
        UPDATE BC_DETALLE_TITULAR
        SET DOMICILIO=LV_DIRECCION,
            TELEFONO=LV_TELEFONO
        WHERE ID_TITULAR=LN_ID_TIT;
    ELSIF LV_DIRECCION IS NULL AND LV_TELEFONO IS NOT NULL
THEN
        UPDATE BC_DETALLE_TITULAR
        SET TELEFONO=LV_TELEFONO
        WHERE ID_TITULAR=LN_ID_TIT;
    ELSIF LV_DIRECCION IS NOT NULL AND LV_TELEFONO IS NULL
THEN
        UPDATE BC_DETALLE_TITULAR
        SET DOMICILIO=LV_DIRECCION
        WHERE ID_TITULAR=LN_ID_TIT;
    END IF;
END IF;

--##### INSERCION A TABLA BC_CREDITO CON DATOS DEL ARCHIVO
Y VALORES POR DEFECTO #####
        SELECT COUNT(*) INTO CONT1 FROM ADMIN.BC_CREDITO
        WHERE ID_CREDITO=LV_NUM_CREDITO AND IDENT_ACREEDOR=PV_ACREEDOR;
--AND ID_TITULAR=LN_ID_TIT;
        IF CONT1=0 THEN
            INSERT INTO ADMIN.BC_CREDITO(ID_CREDITO,
                                           ID_TIPO_CRED,
                                           IDENT_ACREEDOR,
                                           ID_TITULAR,
                                           RESPONSABILIDAD,
                                           FECHA_ULT_ACT,
                                           MONTO_APROBADO,
                                           PLAZO_MESES)

            VALUES(LV_NUM_CREDITO,
                   LV_TIPO_CREDITO,
                   PV_ACREEDOR,
                   LN_ID_TIT,
                   'TITULAR',
                   SYSDATE,
                   LN_CREDITO_APROBADO,
                   LN_NUM_PAGOS);

--##### INSERCION A TABLA BC_DETALLE_CREDITO CON DATOS DEL
ARCHIVO Y VALORES POR DEFECTO #####
        SELECT NVL(MAX(ID_DETALLE_CRED),0)+1 INTO LN_ID_DET
        FROM ADMIN.BC_DETALLE_CREDITO
        WHERE ID_CREDITO=LV_NUM_CREDITO AND IDENT_ACREEDOR=PV_ACREEDOR
AND ID_TITULAR=LN_ID_TIT;

        INSERT INTO
ADMIN.BC_DETALLE_CREDITO(ID_DETALLE_CRED, ID_CREDITO,

```

```
IDENT_ACREEDOR, ID_TITULAR,FECHA_APERTURA,FECHA_REPORTADO, TIPO_MONEDA,
TOTAL_ADEUDADO, MONTO_VENCER, MONTO_VENCIDO, CALIF_GENERAL, ESTADO,
SALDO_30, SALDO_31_60, SALDO_61_90, SALDO_91_120, SALDO_MAS_120)
```

```
VALUES (LN_ID_DET,
        LV_NUM_CREDITO,
        PV_ACREEDOR,
        LN_ID_TIT,
        LD_FECHA_APERTURA,
        SYSDATE,
        LV_MONEDA,
        LN_MONTO_ADEUDADO,
        LN_SALDO_NO_VENCIDO,
        LN_SALDO_VENCIDO,
        'A',
        'A',
        LN_S_30,
        LN_S_31_60,
        LN_S_61_90,
        LN_S_91_120,
        LN_S_MAS_120);
```

```
ELSE
```

```
-- SI YA EXISTE CREDITO SE ACTUALIZAN DATOS EN TABLA CREDITO Y TABLA
DETALLE CREDITO
```

```
-- UPDATE CREDITO
```

```
UPDATE ADMIN.BC_CREDITO
SET FECHA_ULT_ACT=SYSDATE,
    MONTO_APROBADO=LN_CREDITO_APROBADO,
    PLAZO_MESES=LN_NUM_PAGOS
WHERE ID_CREDITO=LV_NUM_CREDITO
AND IDENT_ACREEDOR=PV_ACREEDOR;
```

```
-- AND ID_TITULAR=LN_ID_TIT;
```

```
--UPDATE DETALLE_CREDITO
```

```
UPDATE ADMIN.BC_DETALLE_CREDITO
SET TIPO_MONEDA=LV_MONEDA,
    TOTAL_ADEUDADO=LN_MONTO_ADEUDADO,
    MONTO_VENCER=LN_SALDO_NO_VENCIDO,
    MONTO_VENCIDO=LN_SALDO_VENCIDO,
    --CALIF_GENERAL='A',
    ESTADO='A',
    SALDO_30=LN_S_30,
    SALDO_31_60=LN_S_31_60,
    SALDO_61_90=LN_S_61_90,
    SALDO_91_120=LN_S_91_120,
    SALDO_MAS_120=LN_S_MAS_120
WHERE ID_CREDITO=LV_NUM_CREDITO
AND IDENT_ACREEDOR=PV_ACREEDOR
AND ID_TITULAR=LN_ID_TIT;
```

```

        END IF;
        CONT1:=0;
--##### INSERCIÓN A TABLA BC_PAGOS CON DATOS DEL ARCHIVO Y
VALORES POR DEFECTO #####
        SELECT NVL(MAX(NUM_PAGO),0)+1
        INTO LN_PAGO_NUMERO
        FROM ADMIN.BC_PAGOS WHERE ID_CREDITO=LV_NUM_CREDITO AND
IDENT_ACREEDOR=PV_ACREEDOR AND ID_TITULAR=LN_ID_TIT;

        SELECT NVL(MAX(ID_PAGO),0)+1
        INTO LN_ID_PAG
        FROM ADMIN.BC_PAGOS WHERE ID_CREDITO=LV_NUM_CREDITO AND
IDENT_ACREEDOR=PV_ACREEDOR AND ID_TITULAR=LN_ID_TIT;
        INSERT INTO ADMIN.BC_PAGOS (ID_PAGO,
                                IDENT_ACREEDOR,
                                ID_TITULAR,
                                ID_CREDITO,
                                NUM_PAGO,
                                FECHA_VENC,
                                ESTADO,
                                AÑO,
                                MES,
                                CALIF_MES,
                                SALDO_DEUDA,
                                FECHA_CARGA)
        VALUES (LN_ID_PAG,
                PV_ACREEDOR,
                LN_ID_TIT,
                LV_NUM_CREDITO,
                LN_PAGO_NUMERO,
                LD_FECHA_VENCIMIENTO,
                'P',
                PV_AÑO,
                PV_MES,
                'A',
                LN_MONTO_ADEUDADO,
                SYSDATE);
    END IF;
END IF;
--      EXCEPTION
--      WHEN OTHERS THEN
end BC_P_CARGA_MERGE;

```

```

*****

```

CREAR USUARIO

```

*****

```

```

create or replace procedure BC_P_CREA_USUARIO(PV_USUARIO IN VARCHAR2,
PV_CLAVE IN VARCHAR2, PV_ERROR OUT VARCHAR2) is
begin
    EXECUTE IMMEDIATE 'CREATE USER '||PV_USUARIO||' IDENTIFIED BY
' ||PV_CLAVE;
    EXECUTE IMMEDIATE 'GRANT CONNECT TO '||PV_USUARIO||' WITH ADMIN
OPTION';
    EXECUTE IMMEDIATE 'GRANT CREATE SESSION TO '||PV_USUARIO||' WITH
ADMIN OPTION';
    EXECUTE IMMEDIATE 'GRANT RESOURCE TO '||PV_USUARIO||' WITH ADMIN
OPTION';
    EXECUTE IMMEDIATE 'GRANT CREATE USER TO '||PV_USUARIO||' WITH ADMIN
OPTION';
    EXECUTE IMMEDIATE 'GRANT ALTER USER TO '||PV_USUARIO||' WITH ADMIN
OPTION'; --*****
    EXECUTE IMMEDIATE 'GRANT SELECT ANY TABLE TO '||PV_USUARIO||' WITH
ADMIN OPTION';
    EXECUTE IMMEDIATE 'GRANT SELECT ANY SEQUENCE TO '||PV_USUARIO||' WITH
ADMIN OPTION';
    EXECUTE IMMEDIATE 'GRANT SELECT ANY TRANSACTION TO '||PV_USUARIO||'
WITH ADMIN OPTION';
    EXECUTE IMMEDIATE 'GRANT INSERT ANY TABLE TO '||PV_USUARIO||' WITH
ADMIN OPTION'; --*****
    EXECUTE IMMEDIATE 'GRANT UPDATE ANY TABLE TO '||PV_USUARIO||' WITH
ADMIN OPTION'; --*****
    EXECUTE IMMEDIATE 'GRANT EXECUTE ANY PROCEDURE TO '||PV_USUARIO||'
WITH ADMIN OPTION';
    EXECUTE IMMEDIATE 'GRANT EXECUTE ANY PROGRAM TO '||PV_USUARIO||' WITH
ADMIN OPTION';
    EXECUTE IMMEDIATE 'GRANT EXECUTE ANY TYPE TO '||PV_USUARIO||' WITH
ADMIN OPTION'; --*****

    EXECUTE IMMEDIATE 'GRANT SELECT ON ADMIN.BC_MENU_ADMINISTRADOR TO
' ||PV_USUARIO||' WITH GRANT OPTION';
    EXECUTE IMMEDIATE 'GRANT SELECT ON ADMIN.BC_USUARIO TO
' ||PV_USUARIO||' WITH GRANT OPTION';
    EXECUTE IMMEDIATE 'GRANT UPDATE ON ADMIN.BC_USUARIO TO
' ||PV_USUARIO||' WITH GRANT OPTION';
    EXECUTE IMMEDIATE 'GRANT INSERT ON ADMIN.BC_USUARIO TO
' ||PV_USUARIO||' WITH GRANT OPTION';
    --EXECUTE IMMEDIATE ('GRANT ANY PRIVILEGE TO' ||PV_USUARIO);

    EXECUTE IMMEDIATE 'GRANT ALL ON ADMIN.BC_USUARIO TO '||PV_USUARIO||'
WITH GRANT OPTION';
    EXECUTE IMMEDIATE 'GRANT SELECT ON ADMIN.BC_SUCURSALES TO
' ||PV_USUARIO||' WITH GRANT OPTION';

    EXECUTE IMMEDIATE 'GRANT EXECUTE ON ADMIN.BC_F_NUMERO TO
' ||PV_USUARIO||' WITH GRANT OPTION';

```

```

EXECUTE IMMEDIATE 'GRANT EXECUTE ON ADMIN.BC_F_VERIFICA_ARCHIVO TO
'|PV_USUARIO|' WITH GRANT OPTION';
EXECUTE IMMEDIATE 'GRANT EXECUTE ON
ADMIN.BC_F_VERIFICA_ARCHIVO_MASIVO TO '|PV_USUARIO|' WITH GRANT
OPTION';
EXECUTE IMMEDIATE 'GRANT EXECUTE ON ADMIN.BC_F_VERIFICA_RUC TO
'|PV_USUARIO|' WITH GRANT OPTION';
EXECUTE IMMEDIATE 'GRANT EXECUTE ON ADMIN.BC_VERIFICA_CED TO
'|PV_USUARIO|' WITH GRANT OPTION';
EXECUTE IMMEDIATE 'GRANT EXECUTE ON ADMIN.BC_VERIFICA_RUC TO
'|PV_USUARIO|' WITH GRANT OPTION';

EXCEPTION
WHEN OTHERS THEN
PV_ERROR:=SQLERRM;
end BC_P_CREA_USUARIO;

```

GUARDA ARCHIVO

```

create or replace procedure BC_P_GUARDA_ARCHIVO(PV_RUTA VARCHAR2,
PN_AFILIADO VARCHAR2, PN_ESTADO NUMBER) is
LF_BFILE BFILE;
ARCHIVO VARCHAR2(100);
BEGIN
ARCHIVO:=SUBSTR(PV_RUTA, INSTR(PV_RUTA, '\', -1, 1)+1);
--CARGAMOS EL ARCHIVO EN EL APUNTADOR.. CREAMOS ARCHIVO
LF_BFILE:=BFILENAME('DBCRECURSO', ARCHIVO);
-- PARA ABRIR EL ARCHIVO
--ESTAMOS GRABANDO EN MEMORIA EL CONTENIDO DEL ARCHIVO
DBMS_LOB.FILEOPEN(LF_BFILE);
DBMS_OUTPUT.PUT_LINE(DBMS_LOB.getlength(LF_BFILE));
--ACTUALIZAMOS EL CAMPO
INSERT INTO ADMIN.BC_ARCHIVOS_CARGADOS(FECHA_CARGA, USUARIO,
AFILIADO, ARCHIVO, ESTADO)
VALUES (SYSDATE, USER, PN_AFILIADO, LF_BFILE, PN_ESTADO);
--CERRAMOS ARCHIVO
DBMS_LOB.FILECLOSE(LF_BFILE);
END;

```

MESSAGE MAIL

```

create or replace procedure BC_P_MESSAG_MAIL(mess1 out varchar2, mess2
out varchar2) is
  line varchar2(500):='empty';
  vSfile utl_file.file_type;

BEGIN
  vSfile := utl_file.fopen('SENDEMAILDIR', 'mail.log', 'R');
  IF utl_file.is_open(vSfile) THEN
    LOOP
      utl_file.get_line(vSfile, line, 500 );
    END LOOP;
  END IF;
  EXCEPTION
    WHEN others THEN
      utl_file.fclose(vSfile);
mess2:=substr(line, 1,16);
mess1:=substr(line, 39,30);

end BC_P_MESSAG_MAIL;

```

SEND MAIL

```

create or replace procedure BC_SEND_MAIL(lv_mail in archar2, asunto in
varchar2, cuerpo in varchar2) is
begin
  --SET SERVEROUTPUT ON;
  --CALL DBMS_JAVA.SET_OUTPUT(1000000);
host_command(p_command =>'D:\FORMAS\sendemail\sendEmail -f
infocreditosa@gmail.com -t '|lv_mail ||' -s smtp.gmail.com:587 -xu

```

```
infocreditosa@gmail.com -xp oraclecisc -u '|| asunto||' -m '|| cuerpo);
end BC_SEND_MAIL;
```

```
*****
```

FILE TITULAR

```
*****
```

```
CREATE OR REPLACE PROCEDURE FILE_TITULAR (PV_IDENTIFICACION VARCHAR2,
PV_ERROR OUT VARCHAR2 ) IS
REPORTEFILE UTL_FILE.FILE_TYPE;
```

```
CURSOR USUARIOSCUR IS
```

```
  SELECT C.ID_CREDITO CRED,
         C.FECHA_ULT_ACT FECHA,
         A.NOMBRE ACREED,
         C.RESPONSABILIDAD RESP,
         DC.CALIF_GENERAL CALIF,
         ID_TIPO_CRED TIPO,
         DC.MONTO_VENCER VENCER,
         DC.MONTO_VENCIDO VENCIDO
```

```
FROM BC_CREDITO C, BC_DETALLE_CREDITO DC, BC_ACREEDOR A, BC_TITULAR T
WHERE T.ID_TITULAR=C.ID_TITULAR AND
      T.ID_TITULAR=DC.ID_TITULAR AND
      T.IDENTIFICACION=PV_IDENTIFICACION AND
      C.ID_CREDITO=DC.ID_CREDITO;
ARCH_NAME VARCHAR2(3);
```

```
BEGIN
```

```
  SELECT TIPO_IDENTIFICACION INTO ARCH_NAME FROM BC_TITULAR WHERE
IDENTIFICACION=PV_IDENTIFICACION;
```

```
  -- OPEN FILE FOR WRITING
```

```
  REPORTEFILE :=
  UTL_FILE.FOPEN('FILES_DIR',ARCH_NAME || '_' || PV_IDENTIFICACION || TO_CHAR(SY
SDATE, 'DD-MM-YYYY') || '.TXT', 'W');
```

```
  -- TAKE ONE RECORD FROM CURSOR
```

```
  FOR REC IN USUARIOSCUR
  LOOP
    UTL_FILE.PUT_LINE(REPORTEFILE,
REC.CRED || ';' || REC.FECHA || ';' || REC.ACREED || ';' || REC.RESP || ';' || REC.CALI
F || ';' || REC.TIPO || ';' || REC.VENCER || ';' || REC.VENCIDO);  -- WRITE INTO
FILE
```



```

END LOOP;

UTL_FILE.FCLOSE(REPORTEFILE);  -- CLOSE FILE

EXCEPTION
  WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE( 'ERROR -->' || SQLERRM);
END;

*****

HOST COMMAND

*****

CREATE OR REPLACE PROCEDURE host_command (p_command IN VARCHAR2)
AS LANGUAGE JAVA
NAME 'Host.executeCommand (java.lang.String)';

*****

REP_CREDITOS

*****

CREATE OR REPLACE PROCEDURE REP_CREDITOS
(P_FILEDIR IN VARCHAR2, P_FILENAME IN VARCHAR2)
IS
V_FILEHANDLE UTL_FILE.FILE_TYPE;
CURSOR TIT_INFO IS
  SELECT ID_TITULAR, NOMBRE, APELLIDO, FECHA_INGRESO, HABIL_CTA_CTE_CH
  FROM BC_TITULAR;
V_NEWTIT BC_TITULAR.ID_TITULAR%TYPE;
V_OLDTIT BC_TITULAR.ID_TITULAR%TYPE;

BEGIN
V_FILEHANDLE:= UTL_FILE.FOPEN(P_FILEDIR, P_FILENAME, 'W');
UTL_FILE.PUTF(V_FILEHANDLE, 'REPORTE TITULARES: GENERADO EN %s \n',
SYSDATE);
UTL_FILE.NEW_LINE(V_FILEHANDLE);
FOR V_TITULAR_REC IN TIT_INFO LOOP
  V_NEWTIT := V_TITULAR_REC.ID_TITULAR;
IF V_NEWTIT <> V_OLDTIT THEN
  UTL_FILE.PUTF(V_FILEHANDLE, 'NOMBRE : %s\n',
V_TITULAR_REC.ID_TITULAR);

```

```

END IF;
UTL_FILE.PUTF(V_FILEHANDLE, 'TITULAR: %s DEBE %s \n',
V_TITULAR_REC.NOMBRE, V_TITULAR_REC.APELLIDO);
V_OLDTIT:=V_NEWTIT;
END LOOP;

UTL_FILE.PUT_LINE(V_FILEHANDLE, '* * * END OF REPORT * * *');
UTL_FILE.FCLOSE(V_FILEHANDLE);

EXCEPTION
WHEN UTL_FILE.INVALID_FILEHANDLE THEN
  RAISE_APPLICATION_ERROR(-20001, 'INVALID FILE. ');
WHEN UTL_FILE.WRITE_ERROR THEN
  RAISE_APPLICATION_ERROR(-20002, 'UNABLE TO WRITE TO FILE. ');
END;

```

RP_USUARIOS

```

CREATE OR REPLACE PROCEDURE RP_USUARIOS IS

USUARIOSFILE UTL_FILE.FILE_TYPE;

-- TAKE ALL JOB TITLES FROM JOBS

CURSOR USUARIOSCUR IS

  SELECT NOMBRE FROM BC_USUARIO;

BEGIN

  -- OPEN FILE FOR WRITING

  USUARIOSFILE := UTL_FILE.FOPEN('FILESDIR', 'USERS.TXT', 'W');

  -- TAKE ONE RECORD FROM CURSOR

FOR REC IN USUARIOSCUR

LOOP

  UTL_FILE.PUT_LINE(USUARIOSFILE, REC.NOMBRE); -- WRITE INTO FILE

END LOOP;

```

```
UTL_FILE.FCLOSE(USUARIOSFILE);  -- CLOSE FILE
```

EXCEPTION

WHEN OTHERS THEN

```
DBMS_OUTPUT.PUT_LINE( 'ERROR -->' || SQLERRM);
```

END;

PAQUETES:

create or replace package BC_CALIFICA_K_AL_CARGAR **is**

```
-- Author   : ADMINISTRADOR
-- Created  : 16/02/2010 21:37:38
-- Purpose  : ACTUALIZAR CALIFICACIONES INSTANTANEAMENTE
```

```
PROCEDURE BC_P_CAL_PAGOS;
PROCEDURE BC_P_CAL_CREDITO;
PROCEDURE BC_P_CAL;
```

end BC_CALIFICA_K_AL_CARGAR;

BC_K_CALIFICA_AL_CARGAR

create or replace package BC_K_CALIFICA_AL_CARGAR **is**

```
-- Author   : ADMINISTRADOR
-- Created  : 16/02/2010 21:37:38
-- Purpose  : ACTUALIZAR CALIFICACIONES INSTANTANEAMENTE
```

```

PROCEDURE BC_P_CAL_PAGOS;

PROCEDURE BC_P_CAL_CREDITO;

PROCEDURE BC_P_CAL;

end BC_K_CALIFICA_AL_CARGAR;

*****

BC_K_NOTIFICA

*****

create or replace package BC_K_NOTIFICA is

  -- Author   : ADMINISTRADOR
  -- Created  : 06/02/2010 21:30:56
  -- Purpose  :

  PROCEDURE BC_P_NOTIFICA_AF(PV_USUARIO IN VARCHAR2, PV_ASUNTO IN
  VARCHAR2, PV_CUERPO IN VARCHAR2, PV_ERROR OUT VARCHAR2);
  PROCEDURE BC_P_NOTIFICA_US(PV_USUARIO IN VARCHAR2, PV_ASUNTO IN
  VARCHAR2, PV_CUERPO IN VARCHAR2, PV_ERROR OUT VARCHAR2);
  PROCEDURE BC_P_MOVIMIENTO_US(PV_USUARIO IN VARCHAR2, PV_DETALLE IN
  VARCHAR2);

end BC_K_NOTIFICA;

*****

USUARIO_FETCHER

*****

create or replace package USUARIO_FETCHER as
FUNCTION get_USER(USUARIO_ID IN NUMBER) RETURN USUARIO_REC;
END;

*****

USUARIO_FETCHER2

```

```

create or replace package USUARIO_FETCHER2 as
FUNCTION get_USER(USUARIO_ID IN NUMBER) RETURN USUARIO_REC;
PROCEDURE RP_USUARIOS;
END;

```

PAQUETE BODIES

```

create or replace package body BC_K_CALIFICA_AL_CARGAR is

```

```

PROCEDURE BC_P_CAL_PAGOS IS

```

```

    x number;
    cursor pagos is select * from bc_pagos;
    begin
    for regis in pagos LOOP
    x:=round(sysdate-regis.fecha_venc);
    if x>90 then
        update bc_pagos set calif_mes='D', estado='I'
            where id_titular=regis.id_titular and
                id_credito=regis.id_credito and
                num_pago=regis.num_pago;
    elsif x>60 then
        update bc_pagos set calif_mes='C', estado='I'
            where id_titular=regis.id_titular and
                id_credito=regis.id_credito and
                num_pago=regis.num_pago;
    elsif x>30 then
        update bc_pagos set calif_mes='B', estado='I'
            where id_titular=regis.id_titular and
                id_credito=regis.id_credito and
                num_pago=regis.num_pago;
    end if;
    end loop;
    commit;
end BC_P_CAL_PAGOS;

```

```

-----
PROCEDURE BC_P_CAL_CREDITO IS

```

```

cursor C_PROMCRED is select a.id_credito, avg(c.valor_referencial)as
prom
                                from bc_pagos a, bc_calificacion c
                                where a.calif_mes=c.id_calificacion
                                group by a.id_credito
                                order by id_credito;
rPromCred C_PROMCRED%rowtype;
promedio char(1);
begin
    open C_PROMCRED;
    loop

```

```

fetch C_PROMCRED into rPromCred;
if rPromCred.prom > 499 then
    promedio:='A';
elsif
    rPromCred.prom >= 400 then
    promedio := 'B';
elsif
    rPromCred.prom >= 300 then
    promedio:='C';
elsif
    rPromCred.prom >= 200 then
    promedio:='D';
elsif
    rPromCred.prom < 200 then
    promedio:='E';
end if;
dbms_output.put_line(rPromCred.id_credito);
dbms_output.put_line(promedio);
update bc_detalle_credito set calif_general = promedio
where bc_detalle_credito.id_credito=rPromCred.id_credito;
--next_record ;
exit when C_PROMCRED%NOTFOUND;
end loop;
close C_PROMCRED;
commit;
END BC_P_CAL_CREDITO;

-----
PROCEDURE BC_P_CAL IS
cursor C_PROMCRED IS select b.id_titular, avg(c.valor_referencial)as
prom
from bc_detalle_credito a, bc_titular b,
bc_calificacion c
where a.calif_general=c.id_calificacion and
a.id_titular = b.id_titular
group by b.id_titular;
rPromCred C_PROMCRED%rowtype;
promedio char(1);
BEGIN
open C_PROMCRED;
loop
fetch C_PROMCRED into rPromCred;
if rPromCred.prom > 499 then
    promedio:='A';
elsif
    rPromCred.prom >= 400 then
    promedio := 'B';
elsif
    rPromCred.prom >= 300 then

```

```

        promedio:='C';
    elsif
        rPromCred.prom >= 200 then
            promedio:='D';
    elsif
        rPromCred.prom < 200 then
            promedio:='E';
    end if;
--dbms_output.put_line(rPromCred.id_credito);
dbms_output.put_line(promedio);
update bc_titular set id_calificacion = promedio
    where bc_titular.id_titular=rPromCred.id_titular;
--next_record ;
exit when C_PROMCRED%NOTFOUND;
end loop;
close C_PROMCRED;
commit;
END BC_P_CAL;

```

```
end BC_K_CALIFICA_AL_CARGAR;
```

```
*****
```

BC_K_NOTIFICA

```
*****
```

```

create or replace package body BC_K_NOTIFICA is
--DM Notifica al afiliado en su mail algun movimiento o creacion en su
cuenta
-- Los correos le llegan de parte de INFOCREDITOSA@GMAIL.COM
    PROCEDURE BC_P_NOTIFICA_AF(PV_USUARIO IN VARCHAR2,
                                PV_ASUNTO IN VARCHAR2,
                                PV_CUERPO IN VARCHAR2,
                                PV_ERROR OUT VARCHAR2) IS

    LV_MAIL ADMIN.BC_AFILIADO.EMAIL%TYPE;
    BEGIN
        SELECT EMAIL
        INTO LV_MAIL
        FROM BC_AFILIADO
        WHERE USUARIO=PV_USUARIO;

        ADMIN.BC_SEND_MAIL(LV_MAIL,PV_ASUNTO,PV_CUERPO);
    EXCEPTION
    WHEN OTHERS THEN
        PV_ERROR:='Error en BC_K_NOTIFICA '||sqlerrm;
    END BC_P_NOTIFICA_AF;

```

--DM Notifica al usuario en su mail algun movimiento o creacion en su cuenta

-- Los correos le llegan de parte de INFOCREDITOSA@GMAIL.COM

```
PROCEDURE BC_P_NOTIFICA_US(PV_USUARIO IN VARCHAR2,
                           PV_ASUNTO IN VARCHAR2,
                           PV_CUERPO IN VARCHAR2,
                           PV_ERROR OUT VARCHAR2) IS
```

```
LV_MAIL ADMIN.BC_USUARIO.EMAIL%TYPE;
```

```
BEGIN
```

```
  SELECT EMAIL
  INTO LV_MAIL
  FROM BC_USUARIO
  WHERE USUARIO=PV_USUARIO;
```

```
  ADMIN.BC_SEND_MAIL(LV_MAIL, PV_ASUNTO, PV_CUERPO);
```

```
  EXCEPTION
```

```
  WHEN OTHERS THEN
```

```
    PV_ERROR:='Error en BC_K_NOTIFICA ' || sqlerrm;
```

```
END BC_P_NOTIFICA_US;
```

--Registra movimiento de los Usuarios en la tabla bc_movimientos_usuario

```
PROCEDURE BC_P_MOVIMIENTO_US(PV_USUARIO IN VARCHAR2, PV_DETALLE IN VARCHAR2) IS
```

```
LV_ID_US ADMIN.BC_USUARIO.ID_USUARIO%TYPE;
```

```
NO_MOVIMIENTO ADMIN.BC_MOVIMIENTO_USUARIO.ID_MOVIMIENTO%TYPE;
```

```
BEGIN
```

```
  SELECT ID_USUARIO
  INTO LV_ID_US
  FROM ADMIN.BC_USUARIO
  WHERE USUARIO=PV_USUARIO;
```

```
  SELECT NVL(MAX(ID_MOVIMIENTO), 0)+1
  INTO NO_MOVIMIENTO
  FROM ADMIN.BC_MOVIMIENTO_USUARIO;
```

```
  INSERT INTO ADMIN.BC_MOVIMIENTO_USUARIO
```

```
  VALUES(NO_MOVIMIENTO,
         LV_ID_US,
         USER,
         USERENV('TERMINAL'),
         PV_DETALLE,
         SYSDATE,
         TO_CHAR(SYSDATE, 'HH24:MI:SS'),
         TO_CHAR(SYSDATE, 'HH24:MI:SS'),
         SYS_CONTEXT('USERENV', 'IP_ADDRESS', 15));
```

```
  COMMIT;
```



```

    END BC_P_MOVIMIENTO_US;
end BC_K_NOTIFICA;

```

```

*****

```

USUARIO_FETCHER

```

*****

```

```

create or replace package body USUARIO_FETCHER as
FUNCTION get_USER(USUARIO_ID IN NUMBER) RETURN USUARIO_REC IS
USER_FOUND BC_USUARIO%rowtype;
USER_RTN USUARIO_REC;
BEGIN
SELECT *
INTO USER_FOUND
FROM BC_USUARIO
WHERE BC_USUARIO.ID_USUARIO=USUARIO_ID;

USER_RTN := USUARIO_REC
(
USER_FOUND.ID_USUARIO,
USER_FOUND.TIPO_USUARIO,
USER_FOUND.TIPO_IDENTIFICACION,
USER_FOUND.IDENTIFICACION,
USER_FOUND.NOMBRE,
USER_FOUND.APELLIDO,
USER_FOUND.ID_SUCURSAL,
USER_FOUND.USUARIO,
USER_FOUND.CLAVE,
USER_FOUND.FECHA_INGRESO,
USER_FOUND.FECHA_SALIDA,
USER_FOUND.EMAIL,
USER_FOUND.ESTADO
);

RETURN USER_RTN;
END;

END;

```

```

*****

```

USUARIO_FETCHER2

```

create or replace package body USUARIO_FETCHER2 as
FUNCTION get_USER(USUARIO_ID IN NUMBER) RETURN USUARIO_REC IS
USER_FOUND BC_USUARIO%rowtype;
USER_RTN USUARIO_REC;
BEGIN
SELECT *
INTO USER_FOUND
FROM BC_USUARIO
WHERE BC_USUARIO.ID_USUARIO=USUARIO_ID;

USER_RTN := USUARIO_REC
(
USER_FOUND.ID_USUARIO,
USER_FOUND.TIPO_USUARIO,
USER_FOUND.TIPO_IDENTIFICACION,
USER_FOUND.IDENTIFICACION,
USER_FOUND.NOMBRE,
USER_FOUND.APELLIDO,
USER_FOUND.ID_SUCURSAL,
USER_FOUND.USUARIO,
USER_FOUND.CLAVE,
USER_FOUND.FECHA_INGRESO,
USER_FOUND.FECHA_SALIDA,
USER_FOUND.EMAIL,
USER_FOUND.ESTADO
);

RETURN USER_RTN;
END;

PROCEDURE RP_USUARIOS IS
USUARIOSFILE UTL_FILE.FILE_TYPE;
-- TAKE ALL JOB TITLES FROM JOBS
CURSOR USUARIOSCUR IS
SELECT NOMBRE FROM BC_USUARIO;
BEGIN
-- OPEN FILE FOR WRITING
USUARIOSFILE := UTL_FILE.FOPEN('FILESDIR', 'USERS.TXT', 'W');

-- TAKE ONE RECORD FROM CURSOR
FOR REC IN USUARIOSCUR
LOOP
    UTL_FILE.PUT_LINE(USUARIOSFILE, REC.NOMBRE); -- WRITE INTO FILE
END LOOP;

```

```

    UTL_FILE.FCLOSE(USUARIOSFILE);  -- CLOSE FILE

EXCEPTION
  WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE( 'ERROR -->' || SQLERRM);
END;
END;

```

TRIGGERS:

```

create or replace trigger BC_TR_SEND_MAIL_CONT
  after insert on bc_mail_content

declare
  cursor last_reg is
    select * from BC_mail_content
    where id_mails in (select nvl(max(id_mails),0) from
bc_mail_content);
  registro last_reg%rowtype;
  lv_mail varchar2(100);
  asunto varchar2(100) ;
  cuerpo varchar2(3000) ;

begin
  open last_reg;
  fetch last_reg into registro;
  lv_mail := registro.remitente;
  asunto := registro.asunto;
  cuerpo:= registro.mensaje;
  admin.bc_send_mail(lv_mail, asunto, cuerpo);
  close last_reg;
end BC_TR_SEND_MAIL_CONT;

```

```

*****

```

BC_TR_SEND_MAIL_CONT_AF

```

*****

```

```

create or replace trigger BC_TR_SEND_MAIL_CONT_AF
  after insert on BC_MAIL_CONTENT_AF

```

```

declare
  cursor last_reg is
    select * from BC_MAIL_CONTENT_AF
      where id_mails in (select nvl(max(id_mails),0) from
BC_MAIL_CONTENT_AF);
registro last_reg%rowtype;
lv_mail varchar2(100);
asunto varchar2(100) ;
cuerpo varchar2(3000) ;

begin
open last_reg;
fetch last_reg into registro;
  lv_mail := registro.receptor;
  asunto := registro.emisor || '-' || registro.asunto;
  cuerpo:= registro.mensaje;
  admin.bc_send_mail(lv_mail, asunto, cuerpo);
close last_reg;
end BC_TR_SEND_MAIL_CONT;

```

JAVA SOURCES:

create or replace and compile java source named host as

```

import java.io.*;
public class Host {
  public static void executeCommand(String command) {
    try {
      String[] finalCommand;
      if (isWindows()) {
        finalCommand = new String[4];
        // Use the appropriate path for your windows version.
        finalCommand[0] = "C:\\windows\\system32\\cmd.exe"; // Windows
XP/2003
        finalCommand[1] = "/y";
        finalCommand[2] = "/c";
        finalCommand[3] = command;
      }
      else {
        finalCommand = new String[3];
        finalCommand[0] = "/bin/sh";
        finalCommand[1] = "-c";
        finalCommand[2] = command;
      }

      final Process pr = Runtime.getRuntime().exec(finalCommand);
      pr.waitFor();
    }
  }
}

```

```

new Thread(new Runnable(){
    public void run() {
        BufferedReader br_in = null;
        try {
            br_in = new BufferedReader(new
InputStreamReader(pr.getInputStream()));
            String buff = null;
            while ((buff = br_in.readLine()) != null) {
                System.out.println("Process out :" + buff);
                try {Thread.sleep(100);} catch(Exception e) {}
            }
            br_in.close();
        }
        catch (IOException ioe) {
            System.out.println("Exception caught printing process
output.");
            ioe.printStackTrace();
        }
        finally {
            try {
                br_in.close();
            } catch (Exception ex) {}
        }
    }
}).start();

new Thread(new Runnable(){
    public void run() {
        BufferedReader br_err = null;
        try {
            br_err = new BufferedReader(new
InputStreamReader(pr.getErrorStream()));
            String buff = null;
            while ((buff = br_err.readLine()) != null) {
                System.out.println("Process err :" + buff);
                try {Thread.sleep(100);} catch(Exception e) {}
            }
            br_err.close();
        }
        catch (IOException ioe) {
            System.out.println("Exception caught printing process
error.");
            ioe.printStackTrace();
        }
        finally {
            try {
                br_err.close();
            } catch (Exception ex) {}
        }
    }
}

```

```

    }
  }).start();
}
catch (Exception ex) {
  System.out.println(ex.getLocalizedMessage());
}
}

public static boolean isWindows() {
  if (System.getProperty("os.name").toLowerCase().indexOf("windows")
  != -1)
    return true;
  else
    return false;
}
};

```

JOBS:

```

begin
  sys.dbms_job.submit(job => :job,
    what => 'DECLARE
  cursor C_PROMCRED is select a.id_credito, avg(c.valor_referencial)as
prom
                                from bc_pagos a, bc_calificacion c
                                where a.calif_mes=c.id_calificacion
                                group by a.id_credito
                                order by id_credito;
rPromCred C_PROMCRED%rowtype;
promedio char(1);
begin
  open C_PROMCRED;
  loop
  fetch C_PROMCRED into rPromCred;
  if rPromCred.prom > 499 then
    promedio:='A';
  elsif
    rPromCred.prom >= 400 then
    promedio := 'B';
  elsif
    rPromCred.prom >= 300 then
    promedio:='C';
  elsif
    rPromCred.prom >= 200 then

```

```

        promedio:='D';
    elsif
        rPromCred.prom < 200 then
            promedio:='E';
        end if;
        dbms_output.put_line(rPromCred.id_credito);
dbms_output.put_line(promedio);
        update bc_detalle_credito set calif_general = promedio
        where bc_detalle_credito.id_credito=rPromCred.id_credito;
--next_record ;
exit when C_PROMCRED%NOTFOUND;
end loop;
close C_PROMCRED;
commit;
END;',
        next_date => to_date('18-03-2010 03:26:02', 'dd-
mm-yyyy hh24:mi:ss'),
        interval => 'sysdate+(1/24)');
    commit;
end;
/

```

22

```

begin
    sys.dbms_job.submit(job => :job,
        what => 'Declare
x number;
cursor pagos is select * from bc_pagos;
begin
for regis in pagos LOOP
x:=round(sysdate-regis.fecha_venc);
if x>90 then
    update bc_pagos set calif_mes='D', estado='I'
        where id_titular=regis.id_titular and
            id_credito=regis.id_credito and
            num_pago=regis.num_pago;
elseif x>60 then
    update bc_pagos set calif_mes='C', estado='I'
        where id_titular=regis.id_titular and
            id_credito=regis.id_credito and
            num_pago=regis.num_pago;
elseif x>30 then
    update bc_pagos set calif_mes='B', estado='I'
        where id_titular=regis.id_titular and

```

```

        id_credito=regis.id_credito and
        num_pago=regis.num_pago;
    end if;
    end loop;
    commit;
end;',
        next_date => to_date('18-03-2010 03:26:02', 'dd-
mm-yyyy hh24:mi:ss'),
        interval => 'sysdate+(1/24)');
    commit;
end;
/

```

```

*****

```

23

```

*****

```

```

begin
    sys.dbms_job.submit(job => :job,
        what => 'declare
cursor C_PROMCRED IS select b.id_titular, avg(c.valor_referencial)as
prom
        from bc_detalle_credito a, bc_titular b,
bc_calificacion c
        where a.calif_general=c.id_calificacion and
a.id_titular = b.id_titular
        group by b.id_titular;
rPromCred C_PROMCRED%rowtype;
promedio char(1);
BEGIN
    open C_PROMCRED;
    loop
    fetch C_PROMCRED into rPromCred;
    if rPromCred.prom > 499 then
        promedio:='A';
    elsif
        rPromCred.prom >= 400 then
        promedio := 'B';
    elsif
        rPromCred.prom >= 300 then
        promedio:='C';
    elsif
        rPromCred.prom >= 200 then
        promedio:='D';
    elsif
        rPromCred.prom < 200 then
        promedio:='E';
    end if;

```



```

--dbms_output.put_line(rPromCred.id_credito);
dbms_output.put_line(promedio);
update bc_titular set id_calificacion = promedio
      where bc_titular.id_titular=rPromCred.id_titular;
--next_record ;
exit when C_PROMCRED%NOTFOUND;
end loop;
close C_PROMCRED;
commit;
END;',
                                next_date => to_date('18-03-2010 03:26:02', 'dd-
mm-yyyy hh24:mi:ss'),
                                interval => 'sysdate+(1/24)');
    commit;
end;
/

```

A NIVEL DE JDEVELOPER

CONEXIÓN A LA BASE DE DATOS --

```

try
{
    DriverManager.registerDriver (new oracle.jdbc.driver.OracleDriver());
    unaCon = DriverManager.getConnection
        ("jdbc:oracle:thin:@192.168.0.101:1521:credito","admin","admin");
}catch (Throwable e){ System.out.println("conexion fail");return false; }

```

CLASE AFILIADOBD

```

-----
----  VALIDARINGRESO  ----
-----

public boolean validarIngreso(String usuario, String clave)
{
    boolean resul=false;
    Connection unaCon=null;

        --      CONEXIÓN A BASE DE DATOS      --

    try{
        String clave_des;
        CallableStatement cs = null;
        cs = unaCon.prepareCall("{ call BC_DESENC_CLAVE_AF(?,?) }");
        cs.setString(1,usuario);
        cs.registerOutParameter(2,OracleTypes.VARCHAR);
        cs.execute();
        clave_des=cs.getString(2);
    }
}

```

```

        if(clave_des.equalsIgnoreCase(clave.toUpperCase()))
            resul=true;
        else
            resul=false;

        cs.close();
        unaCon.close();
    }catch (Throwable e) { return false; }
    return resul;
}

-----
----  GUARDARDATOS  ----
-----

public Afiliado guardarDatos(String usuario)
{
    boolean resul=false;
    Connection unaCon=null;
    ingreso_app.view.Afiliado a_session=new ingreso_app.view.Afiliado();

        --          CONEXIÓN A BASE DE DATOS          --

    String sql="SELECT USUARIO, TIPO_AFILIADO, TIPO_IDENTIFICACION, IDENTIFICACION, NOMBRE,
APELLIDO, RAZON_SOCIAL, NOMBRE_COMERCIAL, CLAVE,
TELEFONO,FAX, ID_PE, CIUDAD, DIRECCION, ESTADO_CIVIL, EMAIL, EMAIL2, to_char(FECHA_AFILIACION, 'd
d/mm/yyyy') FECHA, ESTADO FROM bc_afiliado WHERE usuario=upper(?)";

    try{
        PreparedStatement ps=unaCon.prepareStatement(sql);
        ps.setString(1,usuario);
        ResultSet rs=ps.executeQuery();

        if(rs.next())
        {
            a_session.setUsuario(rs.getString("USUARIO"));
            a_session.setTipo_afiliado(rs.getString("TIPO_AFILIADO"));
            a_session.setTipo_identificacion(rs.getString("TIPO_IDENTIFICACION"));
            a_session.setIdentificacion(rs.getString("IDENTIFICACION"));
            a_session.setNombre(rs.getString("NOMBRE"));
            a_session.setApellido(rs.getString("APELLIDO"));
            a_session.setRazon_social(rs.getString("RAZON_SOCIAL"));
            a_session.setNombre_com(rs.getString("NOMBRE_COMERCIAL"));
            a_session.setClave(rs.getString("CLAVE"));
            a_session.setTelf1(rs.getString("TELEFONO"));
            a_session.setTelf2(rs.getString("FAX"));
            a_session.setProvincia(rs.getString("ID_PE"));
            a_session.setCiudad(rs.getString("CIUDAD"));
            a_session.setDireccion(rs.getString("DIRECCION"));
            a_session.setEstado_civil(rs.getString("ESTADO_CIVIL"));
            a_session.setEmail1(rs.getString("EMAIL"));
            a_session.setEmail2(rs.getString("EMAIL2"));
            a_session.setFecha_afil(rs.getString("FECHA"));
        }
        unaCon.close();
    }catch (Throwable e) { return a_session; }
    return a_session;
}

```

```

-----
---- ACTUALIZARDATOS ----
-----

public boolean actualizarDatos(Afiliado unAfiliado)
{
    boolean resul=false;
    Connection unaCon=null;
    ingreso_app.view.Afiliado a_session=new ingreso_app.view.Afiliado();
    String sql=null;

        --          CONEXIÓN A BASE DE DATOS          --

    if(unAfiliado.getTipo_afiliado().equals("N")){
        sql="update BC_AFILIADO set estado_civil='"+unAfiliado.getEstado_civil()+"',
telefono='"+unAfiliado.getTelf1()+"', fax='"+unAfiliado.getTelf2()+"',
direccion='"+unAfiliado.getDireccion()+"', email='"+unAfiliado.getEmail1()+"',
email2='"+unAfiliado.getEmail2()+"', ciudad='"+unAfiliado.getCiudad()+"' where
usuario='"+unAfiliado.getUsuario()+"'";
    }
    else if(unAfiliado.getTipo_afiliado().equals("J"))
    {
        sql="update BC_AFILIADO set telefono='"+unAfiliado.getTelf1()+"',
fax='"+unAfiliado.getTelf2()+"', direccion='"+unAfiliado.getDireccion()+"',
email='"+unAfiliado.getEmail1()+"', email2='"+unAfiliado.getEmail2()+"',
ciudad='"+unAfiliado.getCiudad()+"' where usuario='"+unAfiliado.getUsuario()+"'";
    }

    try{
        Statement st=unaCon.createStatement();
        int row=st.executeUpdate(sql);
        if(row>0)
            resul=true;
        else
            resul=false;
        unaCon.close();
    }
    catch (Throwable e) { return false; }
    return resul;
}

```

```

-----
---- CAMBIARCLAVE ----
-----

public boolean cambiarClave(String usuario, String new_psw, String old_psw)
{
    boolean resul=false;
    Connection unaCon=null;
    String sql=null;

        --          CONEXIÓN A BASE DE DATOS          --

    sql="update BC_AFILIADO set clave='"+new_psw+"' where usuario='"+usuario+"'";

    try{
        Statement st=unaCon.createStatement();
        int row=st.executeUpdate(sql);
    }
}

```

```

        if(row>0)
            resul=true;
        else
            resul=false;
            unaCon.close();
    }
    catch (Throwable e) { return false; }
    return resul;
}

-----
----  VERIFICACLAVE  ----
-----

public int verificaClave(String usuario, String new_psw, String old_psw)
{
    int resul=0;
    int largo=0,i,cont=0;
    if(new_psw.toLowerCase().equals(usuario.toLowerCase()))
        return 1;
    else{
        if(new_psw.length()<6)
            return 2;
        else{
            if(new_psw.toLowerCase().equals("123456"))|||new_psw.toLowerCase().equals("welcome")||ne
w_psw.toLowerCase().equals("computer"))
                return 3;
            else
            {
                largo=new_psw.length();
                for(i=0;i<largo;i++)
                {
                    if(new_psw.charAt(i)==' ')
                        return 4;
                }
            }
        }
    }
    return resul;
}

-----
----  INGRESOCONSULTA  ----
-----

public void ingresoConsulta(String usuario,String identificacion)
{
    boolean resul=false;
    Connection unaCon=null;

        --          CONEXIÓN A BASE DE DATOS          --

    String sql="insert into bc_consulta_afiliado
values(BC_SC_ING_CON_AFI.NEXTVAL,'"+usuario+"','"+identificacion+"',to_date(SYSDATE,'dd/m
m/rrrr'),TO_CHAR(SYSDATE,'HH24:MI:SS'),SYS_CONTEXT('USERENV','IP_ADDRESS',15))";

    try{
        Statement st=unaCon.createStatement();
        int row=st.executeUpdate(sql);
    }
}

```

```

        if(row>0)
            resul=true;
        else
            resul=false;
            unaCon.close();
    }
    catch (Throwable e) { return ; }
    return;
}
}

```

CLASE AFILIADODB

```

-----
---- CONSULTAHoy -----
-----

public Vector consultaHoy(String usuario)
{
    boolean resul;
    Connection unaCon=null;
    Vector v=new Vector();

    -- CONEXIÓN A BASE DE DATOS --

    String sql="select
af.id_consulta,af.identificacion_titular,t.razon_social||'/'||t.apellido||'
' ||t.nombre,to_char(af.fecha_consulta,'dd/mm/yyyy'),af.hora_consulta,af.ip_consulta from
bc_consulta_afiliado af,bc_titular t where af.usuario_afiliado=UPPER('+usuario+') and
af.identificacion_titular=t.identificacion and
to_date(af.fecha_consulta,'dd/mm/yyyy')=to_date(sysdate,'dd/mm/yyyy') order by
id_consulta";
    try{
        PreparedStatement ps=unaCon.prepareStatement(sql);
        ResultSet rs=ps.executeQuery();
        while(rs.next()){
            Consulta unaConsulta=new Consulta();
            unaConsulta.setId_consult(rs.getInt(1));
            unaConsulta.setIdent_tit(rs.getString(2));
            unaConsulta.setNombre(rs.getString(3));
            unaConsulta.setFecha_consult(rs.getString(4));
            unaConsulta.setHora_consult(rs.getString(5));
            unaConsulta.setIp_consult(rs.getString(6));
            v.addElement(unaConsulta);
        }
        unaCon.close();
        return v;
    }
    catch (Throwable e) { return null; }
}

```

```

-----
----- CONSULTAAYER -----
-----

public Vector consultaAyer(String usuario)
{
    boolean resul;
    Connection unaCon=null;
    Vector v=new Vector();

    -- CONEXIÓN A BASE DE DATOS --

    String sql="select
af.id_consulta,af.identificacion_titular,t.razon_social||'/'||t.apellido||'
' ||t.nombre,to_char(af.fecha_consulta,'dd/mm/yyyy'),af.hora_consulta,af.ip_consulta from
bc_consulta_afiliado af,bc_titular t where af.usuario_afiliado=UPPER('+usuario+') and
af.identificacion_titular=t.identificacion and
to_date(af.fecha_consulta,'dd/mm/yyyy')=to_date(sysdate-1,'dd/mm/yyyy') order by
id_consulta";
    try{
        PreparedStatement ps=unaCon.prepareStatement(sql);
        ResultSet rs=ps.executeQuery();
        while(rs.next()){
            Consulta unaConsulta=new Consulta();
            unaConsulta.setId_consult(rs.getInt(1));
            unaConsulta.setIdent_tit(rs.getString(2));
            unaConsulta.setNombre(rs.getString(3));
            unaConsulta.setFecha_consult(rs.getString(4));
            unaConsulta.setHora_consult(rs.getString(5));
            unaConsulta.setIp_consult(rs.getString(6));
            v.addElement(unaConsulta);
        }
        unaCon.close();
        return v;
    }
    catch (Throwable e) { return null; }
}

-----
-- CONSULTAPERIODO_DH ---
-----

public Vector consultaPeriodo_DH(String usuario, String f_desde, String f_hasta)
{
    boolean resul;
    Connection unaCon=null;
    Vector v=new Vector();

    -- CONEXIÓN A BASE DE DATOS --

    String sql="select
af.id_consulta,af.identificacion_titular,t.razon_social||'/'||t.apellido||'
' ||t.nombre,to_char(af.fecha_consulta,'dd/mm/yyyy'),af.hora_consulta,af.ip_consulta from
bc_consulta_afiliado af,bc_titular t where af.usuario_afiliado=UPPER('+usuario+') and
af.identificacion_titular=t.identificacion and
to_date(af.fecha_consulta,'dd/mm/yyyy')>=to_date('+f_desde+', 'dd/mm/yyyy') and
to_date(af.fecha_consulta,'dd/mm/yyyy')<=to_date('+f_hasta+', 'dd/mm/yyyy') order by
af.fecha_consulta";
    try{
        PreparedStatement ps=unaCon.prepareStatement(sql);

```

```

        ResultSet rs=ps.executeQuery();
        while(rs.next()){
        Consulta unaConsulta=new Consulta();
        unaConsulta.setId_consult(rs.getInt(1));
        unaConsulta.setIdent_tit(rs.getString(2));
        unaConsulta.setNombre(rs.getString(3));
        unaConsulta.setFecha_consult(rs.getString(4));
        unaConsulta.setHora_consult(rs.getString(5));
        unaConsulta.setIp_consult(rs.getString(6));
        v.addElement(unaConsulta);
        }
        unaCon.close();
        return v;
    }
    catch (Throwable e) { return null; }
}

-----
---- GETDAY ----
-----

public String getDay()
{
    boolean resul=false;
    Connection unaCon=null;
    String dia="00";

        -- CONEXIÓN A BASE DE DATOS --

    String sql="select to_char(sysdate,'dd') dia from dual";

    try{
        PreparedStatement ps=unaCon.prepareStatement(sql);
        ResultSet rs=ps.executeQuery();
        while(rs.next()){
            dia=rs.getString(1);
        }
        unaCon.close();
    }catch (Throwable e) { return dia; }
    return dia;
}

-----
---- GETMES ----
-----

public String getMes()
{
    boolean resul=false;
    Connection unaCon=null;
    String mes="00";
    try
    {
        DriverManager.registerDriver (new oracle.jdbc.driver.OracleDriver());
        unaCon = DriverManager.getConnection
            ("jdbc:oracle:thin:@192.168.0.101:1521:credito","admin","admin");
    }catch (Throwable e){ return mes; }

    String sql="select to_char(sysdate,'mm') mes from dual";
}

```

```

    try{
        PreparedStatement ps=unaCon.prepareStatement(sql);
        ResultSet rs=ps.executeQuery();
        while(rs.next()){
            mes=rs.getString(1);
        }
        unaCon.close();
    }catch (Throwable e) { return mes; }
    return mes;
}

-----
---- GETANIO ---
-----

public String getAnio()
{
    boolean resul=false;
    Connection unaCon=null;
    String anio="0000";

    --          CONEXIÓN A BASE DE DATOS          --

    String sql="select to_char(sysdate,'rrrr') aniop from dual";
    try{
        PreparedStatement ps=unaCon.prepareStatement(sql);
        ResultSet rs=ps.executeQuery();
        while(rs.next()){
            anio=rs.getString(1);
        }
        unaCon.close();
    }catch (Throwable e) { return anio; }
    return anio;
}

```

CLASE MAILBD

```

-----
-- ENVIOMAILAF ---
-----

public void envioMailAf(String pv_emisor_de, String pv_asunto, String pv_mensaje)
{
    boolean resul=false;
    Connection unaCon=null;

    --          CONEXIÓN A BASE DE DATOS          --

    String sql=
    "insert into BC_MAIL_CONTENT_AF(ID_MAILS,
                                EMISOR,
                                RECEPTOR,
                                ASUNTO,
                                MENSAJE,
                                FECHA,
                                HORA)
    values(BC_SC_MAIL_CONT_AFI.NEXTVAL,
          '"+pv_emisor_de+"',
          'infocreditosa@gmail.com',

```



```

        '"+pv_asunto+"',
        '"+pv_mensaje+"',
        to_date(SYSDATE,'dd/mm/yyyy'),
        TO_CHAR(SYSDATE,'HH24:MM:SS'))";

    try{
        Statement st=unaCon.createStatement();
        int row=st.executeUpdate(sql);
        if(row>0)
            resul=true;
        else
            resul=false;
        unaCon.close();
    }
    catch (Throwable e) { return ; }
    return;
}

```

PAQUETE: RPT_TITULAR – WEB SERVICE

create or replace package body RPT_TITULAR as

FUNCTION GET_CREDITOS(PV_IDENTIFICACION VARCHAR2) RETURN CREDTS_TITULAR_ARRAY AS

```

l_data CREDTS_TITULAR_ARRAY := CREDTS_TITULAR_ARRAY();
CURSOR c_emp IS
SELECT ID_CREDITO
FROM BC_CREDITO C, BC_TITULAR T
WHERE C.ID_TITULAR=T.ID_TITULAR
AND T.IDENTIFICACION=PV_IDENTIFICACION;

```

BEGIN

```

FOR emp_rec IN c_emp LOOP
    l_data.extend;
    l_data(l_data.count) := emp_rec.ID_CREDITO;
END LOOP;

```

RETURN L_DATA;

END GET_CREDITOS;

FUNCTION GET_TITULAR(PV_IDENTIFICACION IN VARCHAR2) RETURN TITULAR_REC IS

```

TITULAR_ENCONTRADO BC_TITULAR%rowtype;
TITULAR_RTN TITULAR_REC;

```

BEGIN

```

SELECT *
INTO TITULAR_ENCONTRADO
FROM BC_TITULAR
WHERE BC_TITULAR.IDENTIFICACION=PV_IDENTIFICACION;

```

TITULAR_RTN := TITULAR_REC

```

(
    TITULAR_ENCONTRADO.TIPO_TITULAR,
    TITULAR_ENCONTRADO.TIPO_IDENTIFICACION,
    TITULAR_ENCONTRADO.NOMBRE,
    TITULAR_ENCONTRADO.APELLIDO,
    TITULAR_ENCONTRADO.RAZON_SOCIAL,
    TITULAR_ENCONTRADO.NOMBRE_COMERCIAL,
    TITULAR_ENCONTRADO.ID_CALIFICACION,

```

```

    TITULAR_ENCONTRADO.HABIL_CTA_CTE_CH
);

```

```

RETURN TITULAR_RTN;
END GET_TITULAR;

```

```

FUNCTION GET_DETALLE_CREDITO(PV_IDENTIFICACION IN VARCHAR2, PV_ID_CREDITO VARCHAR2)
RETURN CREDITO_REC IS

```

```

    CREDITO_RTN CREDITO_REC;

```

```

    VC_NOMBRE                VARCHAR2(50);
    VC_ID_CREDITO            VARCHAR2(15);
    VC_RESPONSABILIDAD       VARCHAR2(50);
    VC_MONEDA                VARCHAR2(10);
    VN_MONTO_APROBADO        NUMBER(11,2);
    VN_PLAZO_MESES           NUMBER(3);
    VN_TOTAL_ADEUDADO        NUMBER(11,2);
    VN_MONTO_VENCIDO         NUMBER(11,2);
    VN_MONTO_VENCER          NUMBER(11,2);
    VC_CALIFICACION_GENERAL  CHAR(1);
    VD_FECHA_ULT_ACT         DATE;
    VC_ESTADO                CHAR(1);

```

```

BEGIN

```

```

SELECT A.NOMBRE,
       C.ID_CREDITO,
       C.RESPONSABILIDAD,
       DC.TIPO_MONEDA,
       C.MONTO_APROBADO,
       C.PLAZO_MESES,
       DC.TOTAL_ADEUDADO,
       DC.MONTO_VENCIDO,
       DC.MONTO_VENCER,
       DC.CALIF_GENERAL,
       C.FECHA_ULT_ACT,
       DC.ESTADO
INTO   VC_NOMBRE,
       VC_ID_CREDITO,
       VC_RESPONSABILIDAD,
       VC_MONEDA,
       VN_MONTO_APROBADO,
       VN_PLAZO_MESES,
       VN_TOTAL_ADEUDADO,
       VN_MONTO_VENCIDO,
       VN_MONTO_VENCER,
       VC_CALIFICACION_GENERAL,
       VD_FECHA_ULT_ACT,
       VC_ESTADO
FROM   BC_ACREEDOR A,
       BC_CREDITO C,
       BC_DETALLE_CREDITO DC,
       BC_TITULAR T
WHERE  A.IDENTIFICACION=C.IDENT_ACREEDOR
AND    A.IDENTIFICACION=DC.IDENT_ACREEDOR
AND    C.ID_CREDITO=DC.ID_CREDITO
AND    C.ID_TITULAR=T.ID_TITULAR
AND    DC.ID_TITULAR=T.ID_TITULAR
AND    T.IDENTIFICACION=PV_IDENTIFICACION
AND    C.ID_CREDITO=PV_ID_CREDITO;

```

```

CREDITO_RTN := CREDITO_REC
(
    VC_NOMBRE,
    VC_ID_CREDITO,
    VC_RESPONSABILIDAD,
    VC_MONEDA,
    VN_MONTO_APROBADO,
    VN_PLAZO_MESES,
    VN_TOTAL_ADEUDADO,
    VN_MONTO_VENCIDO,
    VN_MONTO_VENCER,
    VC_CALIFICACION_GENERAL,
    VD_FECHA_ULT_ACT,
    VC_ESTADO
);

RETURN CREDITO_RTN;
END GET_DETALLE_CREDITO;

FUNCTION GET_PAGOS(PV_ID_CREDITO VARCHAR2, PV_IDENTIFICACION VARCHAR2) RETURN
PAGOS_CREDITO_ARRAY
AS
    l_data PAGOS_CREDITO_ARRAY := PAGOS_CREDITO_ARRAY();
    CURSOR c_emp IS
        SELECT NUM_PAGO
        FROM BC_PAGOS P, BC_TITULAR T
        WHERE P.ID_TITULAR=T.ID_TITULAR
            AND T.IDENTIFICACION=PV_IDENTIFICACION
            AND P.ID_CREDITO=PV_ID_CREDITO;

    BEGIN
        FOR emp_rec IN c_emp LOOP
            l_data.extend;
            l_data(l_data.count) := emp_rec.NUM_PAGO;
        END LOOP;

        RETURN L_DATA;
    END GET_PAGOS;

FUNCTION GET_DETALLE_PAGO(PV_IDENTIFICACION IN VARCHAR2, PV_ID_CREDITO VARCHAR2,
PV_NUM_PAGO NUMBER) RETURN PAGOS_REC IS

PAGOS_RTN PAGOS_REC;

VC_NOMBRE                VARCHAR2(50);
VC_ID_CREDITO            VARCHAR2(15);
VC_MONEDA                VARCHAR2(10);
VN_NUM_PAGO              NUMBER(2);
VD_FECHA_VENC            DATE;
VC_ESTADO                CHAR(1);
VC_AÑO                   VARCHAR2(4);
VC_MES                   VARCHAR2(10);
VC_CALIFICACION          CHAR(1);

BEGIN

SELECT A.NOMBRE,
       C.ID_CREDITO,
       DC.TIPO_MONEDA,
       P.NUM_PAGO,
       P.FECHA_VENC,

```

```

        P.ESTADO,
        P.AÑO,
        P.MES,
        P.CALIF_MES
INTO   VC_NOMBRE,
        VC_ID_CREDITO,
        VC_MONEDA,
        VN_NUM_PAGO,
        VD_FECHA_VENC,
        VC_ESTADO,
        VC_AÑO,
        VC_MES,
        VC_CALIFICACION
FROM   BC_ACREEDOR A,
        BC_CREDITO C,
        BC_DETALLE_CREDITO DC,
        BC_TITULAR T,
        BC_PAGOS P
WHERE  A.IDENTIFICACION=C.IDENT_ACREEDOR
AND    A.IDENTIFICACION=DC.IDENT_ACREEDOR
AND    C.ID_CREDITO=DC.ID_CREDITO
AND    C.ID_TITULAR=T.ID_TITULAR
AND    DC.ID_TITULAR=T.ID_TITULAR
AND    T.IDENTIFICACION=PV_IDENTIFICACION
AND    C.ID_CREDITO=PV_ID_CREDITO
AND    P.ID_TITULAR=C.ID_TITULAR
AND    P.ID_TITULAR=DC.ID_TITULAR
AND    P.ID_CREDITO=C.ID_CREDITO
AND    P.NUM_PAGO=PV_NUM_PAGO;

PAGOS_RTN := PAGOS_REC
(
    VC_NOMBRE,
    VC_ID_CREDITO,
    VC_MONEDA,
    VN_NUM_PAGO,
    VD_FECHA_VENC,
    VC_ESTADO,
    VC_AÑO,
    VC_MES,
    VC_CALIFICACION
);

RETURN PAGOS_RTN;
END GET_DETALLE_PAGO;

PROCEDURE FILE_TITULAR (PV_IDENTIFICACION VARCHAR2, PV_ERROR OUT VARCHAR2 ) IS
REPORTEFILE UTL_FILE.FILE_TYPE;

CURSOR USUARIOCUR IS
SELECT  C.ID_CREDITO CRED,
        C.FECHA_ULT_ACT FECHA,
        A.NOMBRE ACREED,
        C.RESPONSABILIDAD RESP,
        DC.CALIF_GENERAL CALIF,
        ID_TIPO_CRED TIPO,
        DC.MONTO_VENCER VENCER,
        DC.MONTO_VENCIDO VENCIDO
FROM    BC_CREDITO C, BC_DETALLE_CREDITO DC, BC_ACREEDOR A, BC_TITULAR T
WHERE   T.ID_TITULAR=C.ID_TITULAR
AND     T.ID_TITULAR=DC.ID_TITULAR

```

```

        AND T.IDENTIFICACION=PV_IDENTIFICACION
        AND C.ID_CREDITO=DC.ID_CREDITO;

ARCH_NAME VARCHAR2(3);
BEGIN
SELECT TIPO_IDENTIFICACION
INTO ARCH_NAME
FROM BC_TITULAR
WHERE IDENTIFICACION=PV_IDENTIFICACION;

-- OPEN FILE FOR WRITING
REPORTEFILE := UTL_FILE.FOPEN('RPTFILES',ARCH_NAME||'_'||PV_IDENTIFICACION||'
'||TO_CHAR(SYSDATE,'DD-MM-YYYY')||'.TXT','W');

-- TAKE ONE RECORD FROM CURSOR
FOR REC IN USUARIOSCUR
LOOP
    UTL_FILE.PUT_LINE(REPORTEFILE,
REC.CRED||';'||REC.FECHA||';'||REC.ACREED||';'||REC.RESP||';'||REC.CALIF||';'||REC.TIPO||
';'||REC.VENCER||';'||REC.VENCIDO); -- WRITE INTO FILE
END LOOP;

UTL_FILE.FCLOSE(REPORTEFILE); -- CLOSE FILE

EXCEPTION
WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('ERROR -->' || SQLERRM);
END;

END RPT_TITULAR;

```

ANEXOS

ANEXO 1 LEY DE BURÓS DE INFORMACION CREDITICIA

LEY DE BURÓS DE INFORMACIÓN CREDITICIA NO. 13

PUBLICADA EN REGISTRO OFICIAL 127 DE 18 DE OCTUBRE DEL 2005.

EL CONGRESO NACIONAL

CONSIDERANDO:

Que el artículo 81 de la Constitución Política de la República, establece que el Estado arantizará el derecho a acceder a fuentes de información; a buscar, recibir, conocer y difundir información objetiva, veraz, plural, oportuna y sin censura previa, de los acontecimientos de interés general, que preserve los valores de la comunidad;

Que es necesario dictar una ley que permita a las instituciones del sistema financiero, a las del sistema de seguros privados, a las del sistema nacional de seguridad social y a las del sector real de la economía, contar con información completa que les permita tomar adecuadas decisiones de riesgo, con la finalidad de proteger los intereses de los depositantes y del público en general;

Que es fundamental proteger el derecho de los titulares de la información de crédito respecto a que ésta sea correcta y veraz, y, en adición, que no lesione su derecho constitucional a la intimidad personal o familiar; y, En ejercicio de sus facultades constitucionales y legales expide la siguiente:

LEY DE BURÓS DE INFORMACIÓN CREDITICIA TÍTULO I DE LOS BURÓS DE INFORMACIÓN CREDITICIA

Art. 1.- Esta Ley tiene por objeto regular la constitución, organización, funcionamiento y extinción de los burós de información crediticia, cuya actividad exclusiva será la prestación de los servicios de referencia crediticia.

Art. 2.- Para efectos de aplicación de esta Ley, se entenderá como:

Burós de información crediticia (burós).- Son las sociedades anónimas cuyo objeto social exclusivo es la prestación de servicios de referencias crediticias del titular de la información crediticia.

Titular de la información crediticia.- Es la persona, natural o jurídica, a la que se refiere la información de riesgos crediticios.

Fuentes de información.- Son las personas que, debido a sus actividades poseen información de riesgos crediticios.

Cliente de los burós de información crediticia.- Es toda persona legalmente autorizada que contrata con los burós la prestación de servicios de referencias crediticias.

Información prohibida.- Es aquella que, por lesionar el derecho a la intimidad personal o familiar garantizado por la Constitución Política de la República, los burós no pueden recolectar, almacenar, organizar, interconectar en sus bases de datos o, en general, incluir en un reporte de riesgos.

Base de datos.- Es el conjunto de información de riesgos crediticios, administrada por los burós, cualquiera que sea la forma o modalidad de su creación, organización, almacenamiento, sistematización, seguridades o acceso.

Información de Riesgos Crediticios.- Es aquella relacionada con obligaciones o antecedentes financieros, comerciales, contractuales, de seguros privados y de seguridad social, de una persona natural o jurídica, pública o privada, que sirva para identificarla adecuadamente y determinar sus niveles de endeudamiento y en general de riesgos crediticios.

Art. 3.- Los servicios de referencias crediticias, sólo podrán ser prestados por los burós autorizados para operar por la Superintendencia de Bancos y Seguros.

Los burós se constituirán como sociedades anónimas, cuyo objeto social exclusivo debe ser la prestación de servicios de referencias crediticias. En su denominación se incluirá obligatoriamente la frase: "Buró de Información Crediticia".

La aprobación de la constitución de los burós, que conlleva el permiso de operación, corresponde a la Superintendencia de Bancos y Seguros, a cuyo control y vigilancia estarán sometidos en forma exclusiva.

Para aprobar la constitución de un buró la Superintendencia de Bancos y Seguros calificará la idoneidad, responsabilidad y solvencia de los accionistas, haciendo uso de las mismas normas legales y reglamentarias aplicables a la calificación que hace de los accionistas de instituciones financieras.

Igual calificación sobre idoneidad, responsabilidad y solvencia será necesaria para la Transferencia de acciones de un buró, en forma previa a su inscripción en el Libro de Acciones y Accionistas, siendo aplicables, para este efecto, las disposiciones de la Ley General de Instituciones del Sistema Financiero. Los clientes de un buró y las instituciones del sistema financiero sujetas al control de la Superintendencia de Bancos y Seguros, no podrán directa ni indirectamente, ni a ningún título, ser accionistas, socios o miembros de los burós de información crediticia.

Art. 4.- El capital social mínimo requerido para la constitución de los burós, deberá estar íntegramente suscrito y pagado al momento de su constitución. Su cuantía será fijada por resolución general de la Superintendencia de Bancos y Seguros; pero en ningún caso será inferior a cien mil dólares de los Estados Unidos de América.

TÍTULO II DEL MANEJO DE LA INFORMACIÓN CREDITICIA

Art. 5.- La información de riesgos que obtengan y mantengan los burós tendrá por exclusiva finalidad destinarla a la prestación del servicio de referencias crediticias y deberán mantenerla en el país.

La información histórica crediticia requerida sobre personas naturales y jurídicas, no podrá exceder de 6 años, por tanto, a los burós de información crediticia les está prohibido expresamente recabar y proporcionar información anterior a este límite.

Sólo con el conocimiento pleno y la autorización previa del titular de la información crediticia, en cada operación, los burós de crédito podrán obtener y mantener en sus archivos la nueva información crediticia distinta de aquella proveniente de la Central de Riesgos. En este caso, los clientes de los burós pondrán en conocimiento de los titulares de la información crediticia, lo siguiente:

- a) La existencia de las bases de datos que administran los burós, su finalidad y los potenciales destinatarios de la información;
- b) La identidad y dirección de los burós que recepcen la información;
- c) Las posibles consecuencias del uso de la información; y,
- d) Los derechos que les asisten.

El buró de crédito que obtenga y archive esa información, con la simple solicitud del titular de la información y sin ningún otro trámite, obligatoriamente, deberá entregársela tantas y cuantas veces la requiera, de forma irrestricta y totalmente gratuita.

La información crediticia será lícita, exacta y veraz, de forma tal que responda a la situación real de su titular en determinado momento. En cada reporte los burós deberán especificar la fecha a la que corresponde la información.

Los titulares de información crediticia pueden proporcionar directamente a los burós su propia información, en cuyo caso los burós deberán informarles previamente lo señalado en las letras a), b), c) y d) de este artículo.

La información proveniente de la Central de Riesgos, no requiere autorización.

Art. 6.- Los burós solo podrán recolectar, acopiar, almacenar, actualizar, grabar, organizar, sistematizar, elaborar, seleccionar, confrontar, interconectar en sus bases de datos, información referente al riesgo crediticio.

En consecuencia, no podrán manejar la siguiente información:

- a) Aquella que, por afectar el derecho a la intimidad personal o familiar, lesione las garantías previstas en los numerales 8, 11 y 21 del artículo 23 de la Constitución Política de la República, a través de la difusión de características físicas, morales o emocionales de una persona o cualquier otra información relacionada con circunstancias de su vida afectiva o familiar, hábitos personales y de consumo, ideologías, opiniones políticas, creencias o convicciones religiosas, estados de salud físico o psicológico, vida sexual o información genética; así como toda violación a las garantías previstas por las leyes, tratados y convenios internacionales; y,

- b) La información que de conformidad con la Ley General de Instituciones del Sistema Financiero, se encuentre protegida por el sigilo bancario, así como la información del patrimonio personal y familiar, las cuales solo pueden ser entregadas por expresa orden judicial.

El buró no podrá recolectar, procesar o difundir la información prohibida expresamente en este artículo, aunque cuente con la autorización del titular de la información; en todo caso, quien se considere afectado por la violación del presente artículo podrá iniciar las acciones civiles y penales a que hubiere lugar.

Art. 7.- Los burós sólo podrán prestar servicios de referencias crediticias a clientes debidamente identificados.

Solo podrán ser clientes de los burós de información crediticia:

- a) Las instituciones controladas por la Superintendencia de Bancos y Seguros;
- b) Las personas jurídicas, empresas, fundaciones y otras sociedades legalmente

autorizadas y que otorguen crédito; y,

c) Las personas naturales que se dediquen a actividades económicas, que cuenten con el Registro Único de Contribuyentes actualizado y que otorguen crédito.

Los burós no podrán comercializar a título universal sus bases de datos ni entregar toda la información crediticia contenida en las mismas, ni podrán dar a conocer esta información por medios de comunicación colectiva tales como radio, prensa, televisión u otros medios.

TÍTULO III

DE LA DEFENSA DE LOS TITULARES DE LA INFORMACIÓN CREDITICIA

Art. 8.- Los clientes de los burós y cualquier otra persona que por diversas causas lleguen a tener acceso a reportes emitidos por los burós (incluyendo a funcionarios, empleados, agentes, entre otros), deberán obligatoriamente guardar confidencialidad sobre la información contenida en ellos, siendo prohibido utilizarla para fines distintos del análisis de riesgo crediticio.

Quien empleare o divulgare indebidamente la información contenida en un reporte de crédito o alterare la información proporcionada por la fuente, estará sujeto a las sanciones establecidas en el artículo 201 del Código Penal, sin perjuicio de las acciones y responsabilidades civiles a las que hubiere lugar.

Art. 9.- El titular de la información crediticia tendrá derecho a:

a) Conocer si en la base de datos de un buró existe información sobre sí mismo y acceder a ella sin restricción alguna; y,

b) Exigir de la fuente de información crediticia, la rectificación de la información ilegal, inexacta o errónea y comunicarla al buró para que éste, de ser el caso, la rectifique.

Dentro del plazo de quince días desde la presentación de la solicitud, las fuentes de información crediticia obligatoriamente la resolverán, por escrito, admitiéndola o rechazándola motivadamente y poniendo en conocimiento de los burós autorizados para operar. Hasta tanto, sin perjuicio de continuar incluyéndola en los reportes de riesgos que emitan, los burós anunciarán que la información materia de la solicitud está siendo revisada a pedido del titular.

Si se concluye que la información materia de impugnación del titular es ilegal, inexacta o errónea, el buró, por cuenta de la fuente de información crediticia, inmediatamente enviará comunicaciones rectificatorias a todos quienes hubieren recibido reportes conteniéndola.

Art. 10.- Los burós y las fuentes de información crediticia serán legalmente responsables por los daños ocasionados al titular como consecuencia de la transmisión de información ilegal, inexacta o errónea y, por tanto, no estarán exonerados alegando ausencia de dolo o de culpa.

La responsabilidad de las fuentes es entregar información a los burós de manera exacta y legal; la responsabilidad de los burós es reportarla sin alteración o modificación alguna.

Sin perjuicio de lo anterior, en los procesos promovidos contra los burós, éstos podrán pedir que se cite también con la demanda a la o las fuentes de las que hubieren obtenido la información crediticia materia del proceso, siguiendo el procedimiento establecido en el artículo 94 del Código de Procedimiento Civil.

También responderán por los daños causados al titular de la información crediticia, quienes utilicen dolosa o culposamente informaciones o reportes provenientes de los

burós. El afectado podrá demandar indemnización, cuando la información errónea no ha sido rectificadora por los burós.

TÍTULO IV DE LAS INFRACCIONES ADMINISTRATIVAS

Art. 11.- Corresponde a la Superintendencia de Bancos y Seguros imponer sanciones administrativas y disponer medidas correctivas a los burós que infrinjan las disposiciones de esta Ley y las normas emitidas por la Junta Bancaria.

Art. 12.- Son infracciones administrativas:

a) Cualquier violación de las prohibiciones contenidas en el artículo 6 de esta Ley; y, b) El desconocimiento de cualquiera de los derechos de los titulares de la información crediticia determinados en esta Ley.

Art. 13.- En caso de que un buró violare las disposiciones de esta Ley o de la normativa expedida por la Junta Bancaria o no acatare las instrucciones de la Superintendencia de Bancos y Seguros, ésta impondrá las siguientes sanciones:

a) Multas de entre tres mil y veinte mil dólares por cada caso;
b) En caso de reincidencia, suspensión por hasta seis meses del permiso de operación; y,
c) Si es nuevamente reincidente, cancelación del permiso de operación, lo que conlleva la orden de disolver y liquidar al respectivo buró.

DISPOSICIONES GENERALES

PRIMERA.- Las personas naturales y jurídicas, empresas, fundaciones y otras sociedades vinculadas a las instituciones controladas por la Superintendencia de Bancos y Seguros, no podrán directa ni indirectamente ni a ningún título, ser accionistas, socios o miembros de los burós de información crediticia.

SEGUNDA.- Cuando la Superintendencia de Bancos y Seguros tuviere noticias de que otras personas naturales o jurídicas, distintas a los burós, han realizado o están realizando operaciones reservadas a éstos, efectuará las investigaciones del caso de acuerdo a lo establecido en el artículo 121 de la Ley General de Instituciones del Sistema Financiero, y de ser necesario ordenará la suspensión inmediata de dichas operaciones, para cuyo efecto podrá contar con el auxilio de la fuerza pública. El Superintendente de Bancos y Seguros pondrá los hechos en conocimiento del Ministerio Público.

TERCERA.- Corresponde a la Junta Bancaria dictar las normas de carácter general para la organización, funcionamiento, control y demás aspectos relacionados con los burós, incluyendo su liquidación.

CUARTA.- En todo lo no previsto en esta Ley, regirán supletoriamente la Ley General de Instituciones del Sistema Financiero, Ley de Compañías y otras leyes aplicables.

DISPOSICIONES TRANSITORIAS:

PRIMERA.- Los burós de información crediticia creados al amparo de las normas emitidas por la Junta Bancaria, deberán ajustar sus estatutos sociales, su organización y funcionamiento a lo previsto en esta Ley, en el plazo de noventa días contados desde su publicación en el Registro Oficial; caso contrario, la Superintendencia de Bancos y Seguros ordenará de oficio su disolución y liquidación.

SEGUNDA.- Los clientes de los burós de información crediticia que actualmente sean accionistas de los mismos, deberán efectuar la respectiva desinversión dentro de los ciento ochenta días contados desde la publicación de esta Ley en el Registro Oficial.

Art. FINAL.- La presente Ley entrará en vigencia a partir de su publicación en el Registro Oficial.

Dada, en la ciudad de San Francisco de Quito, Distrito Metropolitano, en la Sala de Sesiones del Congreso Nacional, a los veinte y nueve días del mes de septiembre del año dos mil cinco.

BIBLIOGRAFIA

www.roseindia.net

www.zonaoracle.com

www.forosdelweb.com

www.lawebdelprogramador.com

www.openssl.org

[http://www.microsoft.com/downloads/details.aspx?familyid=9B2DA534-](http://www.microsoft.com/downloads/details.aspx?familyid=9B2DA534-3E03-4391-8A4D-074B9F2BC1BF&displaylang=en)

[3E03-4391-8A4D-074B9F2BC1BF&displaylang=en](http://www.microsoft.com/downloads/details.aspx?familyid=9B2DA534-3E03-4391-8A4D-074B9F2BC1BF&displaylang=en)

ANEXO # 2 FORMATO DE ARCHIVO CARGA

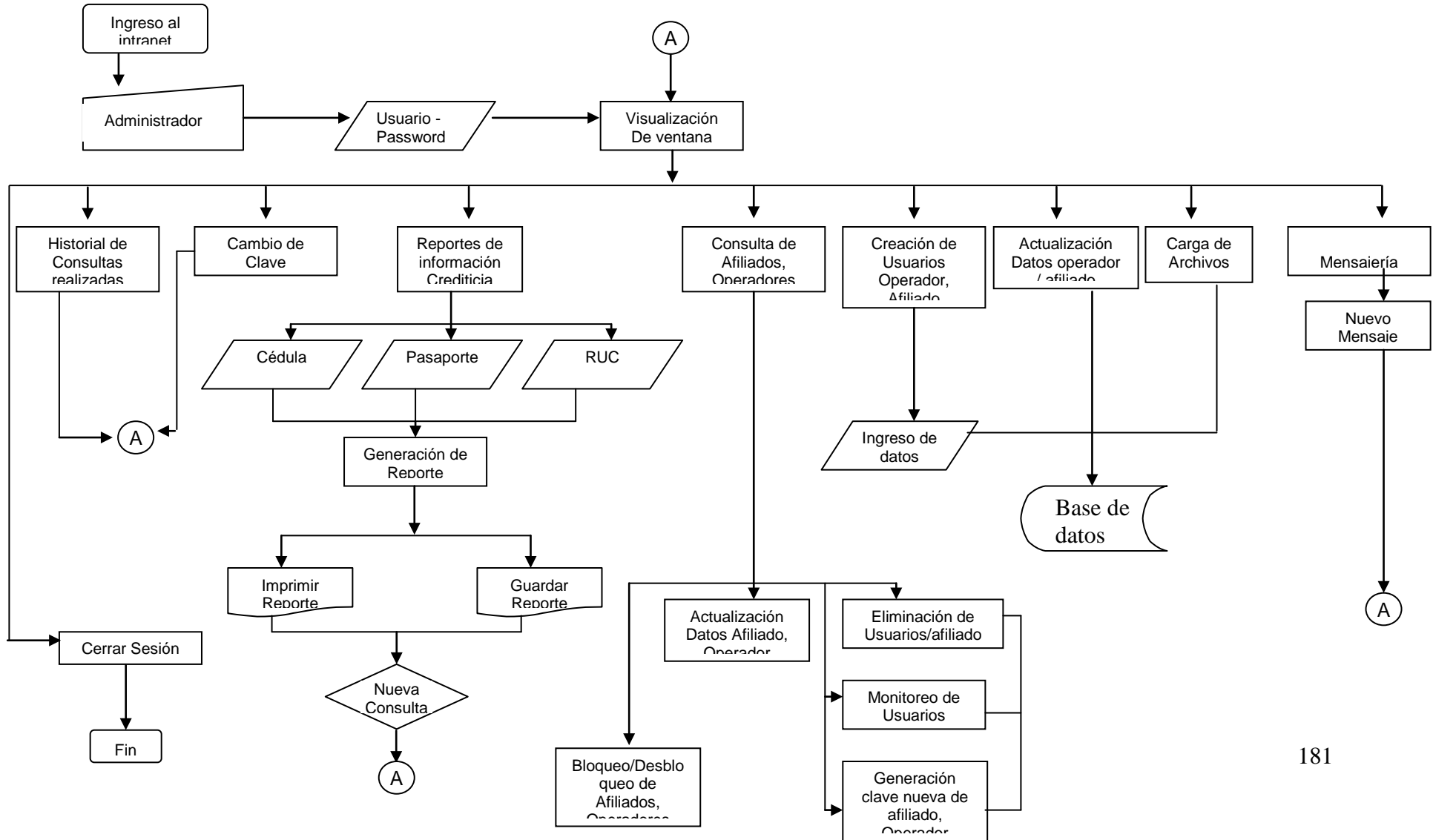
Formato de Archivo de carga para una persona natural

Entidad	NOMBRE	APELLIDO	TIPO IDENTIFICACION	CEDULA / R.U.C.	OTORGANTE	IDENTIFICACION OTORGANTE	CREDITO #	TIPO DE CREDITO	MONEDA	CREDITO APROBADO	MONTO ADEUDADO	SALDO NO VENCIDO	SALDO TOTAL VENCIDO	FECHA APERTURA	FECHA VENCIMIENTO	# PAGOS	SALDO HASTA 30 DIAS	SALDO DE 31-60 DIAS	SALDO DE 61-90 DIAS	SALDO DE 91-120 DIAS	SALDO DE MAS DE 120 DIAS
N J	PEDRO JOSÉ LIPARI S.A.	ALVARADO SÁNCHEZ	CED	120096874-9 0991293019001	Banco Pichincha				USD		1500,00	300,00	0,00			(EN MESES)					

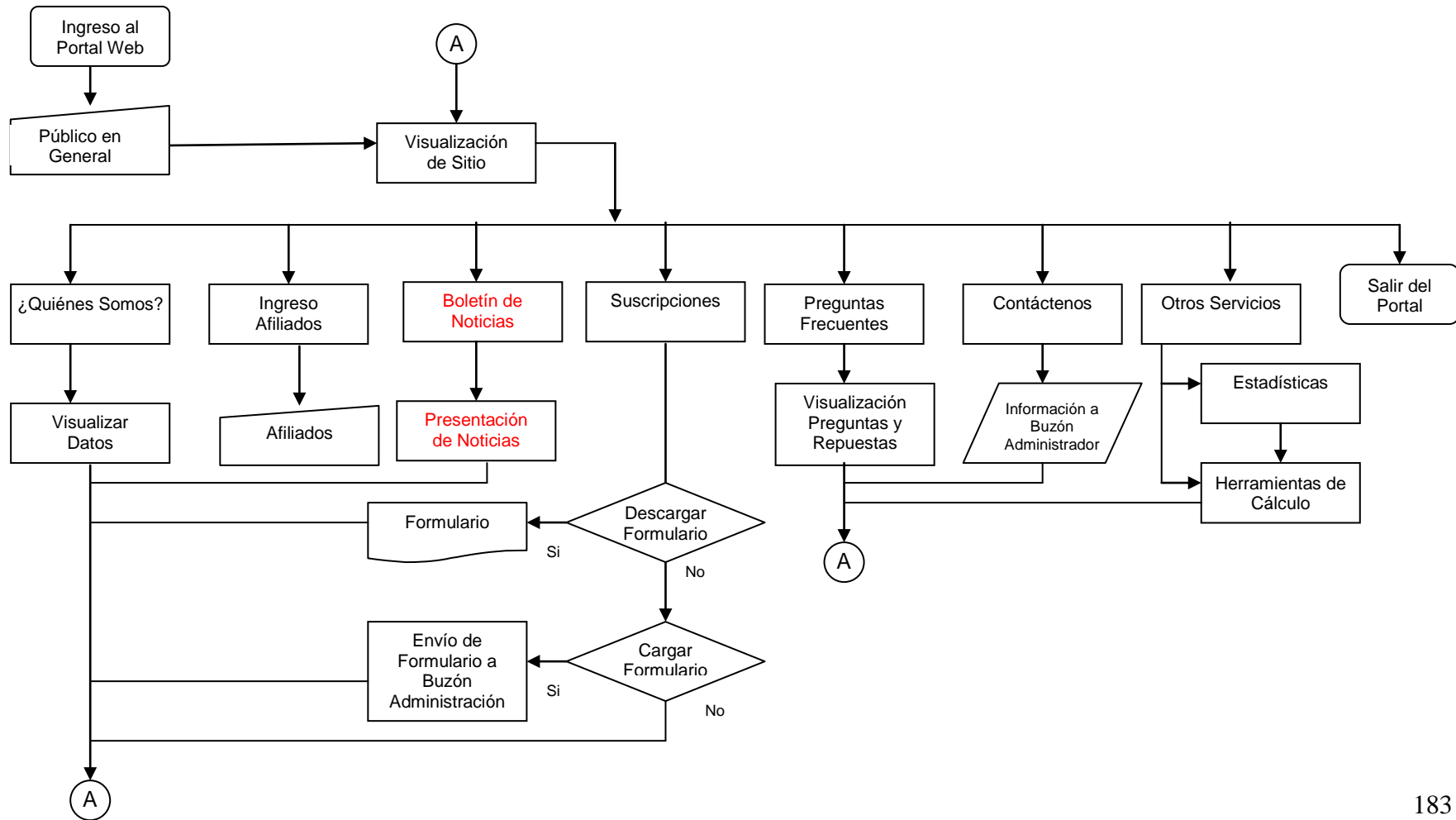
Campo	Descripcion	Observaciones
ENTIDAD	Tipo de persona	N para persona Natural, J para persona Juridica
NOMBRE	Nombre de Persona	
APELLIDO	Apellido de Persona	
TIPO IDENTIFICACION		Ced para cedula, Pas para Pasaporte y RUC
CEDULA / R.U.C.	Numero de Identificacion	
OTORGANTE	Nombre de Emisor de Credito	
IDENTIFICACION OTORGANTE	Codigo de emisor de credito	
CREDITO #	Identificacion de credito	
TIPO DE CREDITO	Tipo de Credito	Consumo, Comercial, diferido, hipotecario, rotativo, Pyme
MONEDA	Tipo de moneda a utilizarse	
CREDITO APROBADO	Monto Total aprobado	
MONTO ADEUDADO	Monto adeudado actualmente	
SALDO NO VENCIDO	Saldo no vencido	
SALDO TOTAL VENCIDO	Saldo vencido	
FECHA APERTURA	Fecha de apertura	
FECHA VENCIMIENTO	Fecha de vencimiento	
# PAGOS	Numero de pago correspondiente	
SALDO HASTA 30 DIAS	Saldo a 30 dias	
SALDO DE 31-60 DIAS	Saldo a 60 dias	
SALDO DE 61-90 DIAS	Saldo a 90 dias	
SALDO DE 91-120 DIAS	Saldo a 120 dias	
SALDO DE MAS DE 120 DIAS	Saldo superior a 120 dias	

ANEXO 5:

DIAGRAMA DE FLUJO: USUARIO ADMINISTRADOR



ANEXO 7: DIAGRAMA DE FLUJO: PÚBLICO EN GENERAL



ANEXO 8:

DIAGRAMA DE FLUJO: USUARIO AFILIADO

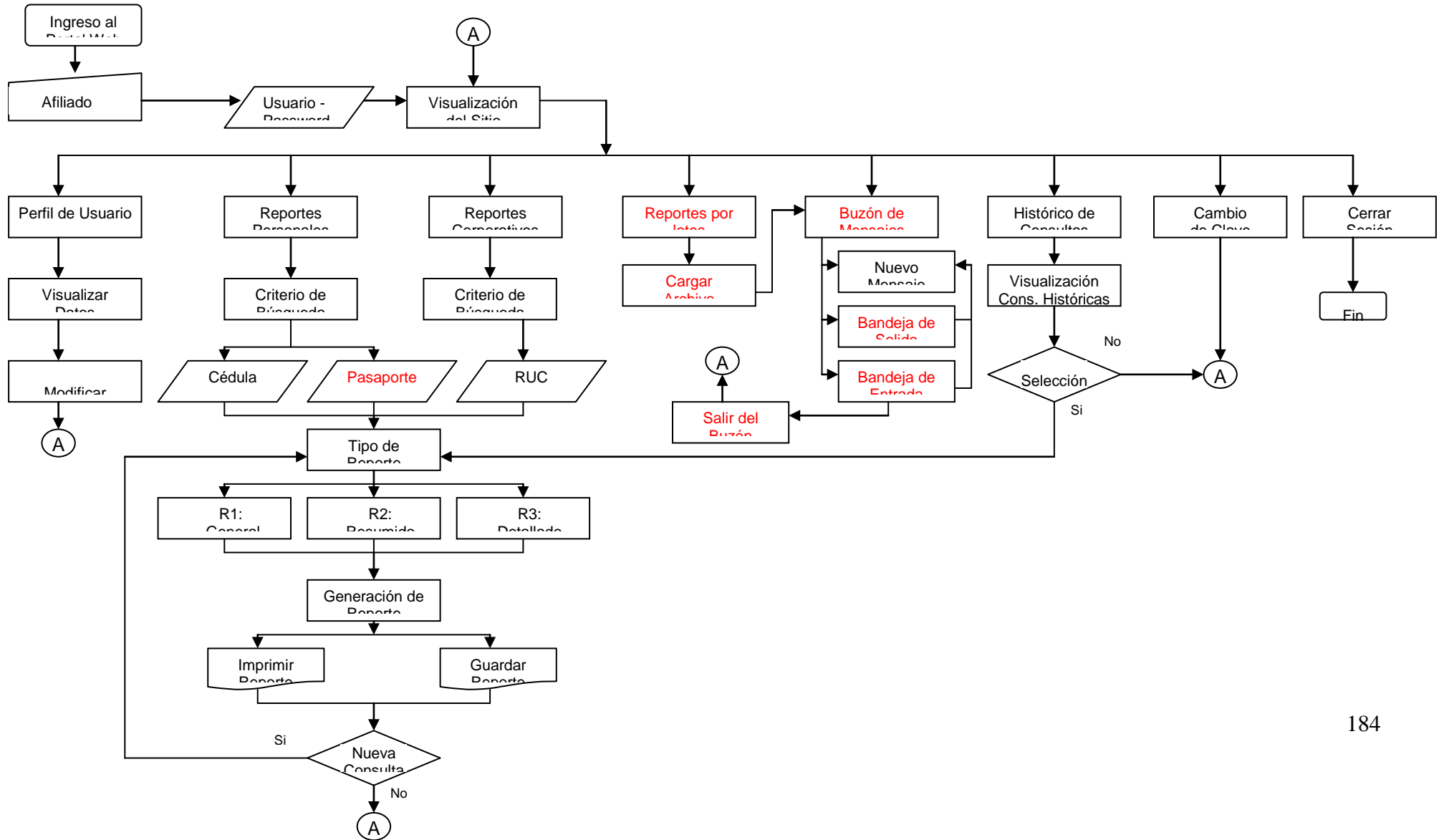


TABLA	CAMPO	TIPO DE DATO	DESCRIPCION
BC_ACREEDOR			Tabla donde se registra informacion de los acreedores con que trabaja el sistema
BC_ACREEDOR	id_acreedor	number(5)	Identificador de acreedor asignada por Infocredito
BC_ACREEDOR	id_tipo_ac	varchar2(20)	Registra cooperativas, Mutualistas, Bancos, Sociedad financieras, Seguros privados, Tarjeta de credito y afiliados
BC_ACREEDOR	nombre	varchar2(100)	Nombre comercial
BC_ACREEDOR	servicio	varchar2(50)	
BC_ACREEDOR	identificacion	varchar2(15)	Numero de Ruc del acreedor
BC_ACREEDOR	id_pe	number(4)	Codigo de pais
BC_ACREEDOR	id_ciudad	varchar2(4)	codigo de ciudad
BC_ACREEDOR	direccion	varchar2(60)	Direccion de oficina matriz

BC_AFILIADO			Tabla donde se registra informacion de los afiliados del sistema
BC_AFILIADO	id_afiliado	number(10)	Codigo del afiliado asignado por Infocredito
BC_AFILIADO	tipo_afiliado	char(1)	Tipo de afiliado J--> Juridico N--> Natural
BC_AFILIADO	tipo_identificacion	varchar2(3)	Tipo de Identificacion CED-PAS-RUC
BC_AFILIADO	identificacion	varchar2(15)	Numero de identificacion del afiliado
BC_AFILIADO	nombre	varchar2(25)	Nombre del afiliado
BC_AFILIADO	apellido	varchar2(25)	Apellido del afiliado
BC_AFILIADO	razon_social	varchar2(70)	Razon Social del afiliado (Opcional)
BC_AFILIADO	nombre_comercial	varchar2(70)	Nombre comercial del afiliado (Opcional)
BC_AFILIADO	usuario	varchar2(15)	Usuario asignado por Infocredito
BC_AFILIADO	clave	varchar2(30)	Clave utilizada para ingresar al sistema (Campo encriptado)
BC_AFILIADO	telefono	varchar2(30)	Telefono del afiliado
BC_AFILIADO	fax	varchar2(30)	Numero de fax
BC_AFILIADO	id_pe	number(4)	Codigo de pais
BC_AFILIADO	ciudad	varchar2(40)	Ciudad de residencia
BC_AFILIADO	direccion	varchar2(60)	Direccion del afiliado
BC_AFILIADO	estado_civil	char(1)	Estado civil
BC_AFILIADO	email	varchar2(40)	direccion electronica
BC_AFILIADO	email2	varchar2(40)	direccion electronica alternativa (Opcional)
BC_AFILIADO	fecha_afiliacion	date	Fecha de afiliacion
BC_AFILIADO	estado	char(1)	Estado A --> Activo I--> Inactivo

BC_CALIFICACION			Tabla de referencia para calificaciones
BC_CALIFICACION	id_calificacion	char(1)	Calificacion (A-B-C-D-E)
BC_CALIFICACION	descripcion	varchar2(40)	Detalle de calificacion
BC_CALIFICACION	valor_referencial	number(3)	Valor aritmetico asignado a cada calificacion

BC_SUCURSAL			Tabla que registra informacion de las sucursales de las empresas registradas en Infoc
BC_SUCURSAL	id_sucursal	number(4)	
BC_SUCURSAL	id_empresa	number(4)	
BC_SUCURSAL	id_pe	number(4)	
BC_SUCURSAL	id_ciudad	number(4)	
BC_SUCURSAL	nombre	varchar2(70)	
BC_SUCURSAL	fecha_apertura	date	
BC_SUCURSAL	direccion	varchar2(60)	
BC_SUCURSAL	telefono	varchar2(30)	
BC_SUCURSAL	fax	varchar2(30)	
BC_SUCURSAL	email	varchar2(40)	
BC_SUCURSAL	estado	char(1)	

BC_MAIL_CONTENT			Tabla que registra informacion generada desde buzon de sugerencias de sitio web
BC_MAIL_CONTENT	id_mails	number	identificador de registro
BC_MAIL_CONTENT	emisor	varchar2(100)	email de persona que realizo sugerencia
BC_MAIL_CONTENT	remite	varchar2(100)	
BC_MAIL_CONTENT	asunto	varchar2(100)	Titulo del mensaje
BC_MAIL_CONTENT	mensaje	varchar2(3000)	Cuerpo del mensaje
BC_MAIL_CONTENT	fecha	date	Fecha de envio
BC_MAIL_CONTENT	hora	varchar2(12)	Hora de envio

BC_MAIL_CONTENT_AF			Tabla que registra comunicaciones con afiliados de sus operaciones del sistema
BC_MAIL_CONTENT	id_mails	number	identificador de registro
BC_MAIL_CONTENT	emisor	varchar2(100)	email de Infocredito
BC_MAIL_CONTENT	remite	varchar2(100)	email de afiliado
BC_MAIL_CONTENT	copias	varchar2(500)	email alternativo
BC_MAIL_CONTENT	asunto	varchar2(100)	Titulo del mail
BC_MAIL_CONTENT	mensaje	varchar2(3000)	Cuerpo del mensaje
BC_MAIL_CONTENT	fecha	date	Fecha de envio
BC_MAIL_CONTENT	hora	varchar2(12)	Hora de envio

BC_EMPRESA			Tabla que registra parametros de empresas que se afilian a Infocredito
BC_EMPRESA	id_empresa	number(4)	
BC_EMPRESA	identificacion	varchar2(15)	
BC_EMPRESA	nombre_comercial	varchar2(80)	
BC_EMPRESA	razon_social	varchar2(80)	

TABLA	CAMPO	TIPO DE DATO	DESCRIPCION
BC_EMPRESA	fecha_apertura	date	
BC_EMPRESA	id_pe	number(4)	
BC_EMPRESA	id_ciudad	number(5)	
BC_EMPRESA	direccion	varchar2(70)	
BC_EMPRESA	telefono	varchar2(30)	
BC_EMPRESA	fax	varchar2(30)	
BC_EMPRESA	email	varchar2(40)	
BC_EMPRESA	estado	char(1)	
BC_CONSULTA_AFILIADO			Tabla que registra las consultas realizadas por cada afiliado
BC_CONSULTA_AFIL	id_consulta	number(10)	
BC_CONSULTA_AFIL	usuario_afiliado	varchar2(15)	
BC_CONSULTA_AFIL	identificacion_titular	varchar2(15)	
BC_CONSULTA_AFIL	fecha_consulta	date	
BC_CONSULTA_AFIL	hora_consulta	varchar2(12)	
BC_CONSULTA_AFIL	ip_consulta	varchar2(15)	
BC_DETALLE_CREDITO			Tabla que registra datos de creditos asignados a las personas del historial
BC_DETALLE_CREDI	id_detalle_cred	number(5)	
BC_DETALLE_CREDI	id_credito	varchar2(15)	
BC_DETALLE_CREDI	ident_acreedor	varchar2(15)	
BC_DETALLE_CREDI	id_titular	number(10)	
BC_DETALLE_CREDI	fecha_apertura	date	
BC_DETALLE_CREDI	fecha_reportado	date	
BC_DETALLE_CREDI	tipo_moneda	varchar2(10)	
BC_DETALLE_CREDI	total_adeudado	number(11,2)	
BC_DETALLE_CREDI	monto_vencer	number(11,2)	
BC_DETALLE_CREDI	monto_vencido	number(11,2)	
BC_DETALLE_CREDI	morosidad	number(3)	
BC_DETALLE_CREDI	calif_general	char(1)	
BC_DETALLE_CREDI	estado	char(1)	
BC_DETALLE_CREDI	saldo_30	number(11,2)	
BC_DETALLE_CREDI	saldo_31_60	number(11,2)	
BC_DETALLE_CREDI	saldo_61_90	number(11,2)	
BC_DETALLE_CREDI	saldo_91_120	number(11,2)	
BC_DETALLE_CREDI	saldo_mas_120	number(11,2)	
BC PAGOS			Tabla que registra los pagos realizados por los beneficiarios de los creditos
BC PAGOS	id_pago	number(5)	
BC PAGOS	ident_acreedor	varchar2(15)	
BC PAGOS	id_titular	number(5)	
BC PAGOS	id_credito	varchar2(15)	
BC PAGOS	num_pago	number(2)	
BC PAGOS	fecha_venc	date	
BC PAGOS	estado	char(1)	
BC PAGOS	año	varchar2(4)	
BC PAGOS	mes	varchar2(10)	
BC PAGOS	calif_mes	char(1)	
BC PAGOS	saldo_deuda	number(11,2)	
BC PAGOS	fecha_carga	date	
BC CREDITO			Tabla que registra datos de los creditos asignados a los beneficiarios
BC CREDITO	id_credito	varchar2(15)	
BC CREDITO	id_tipo_cred	varchar2(50)	
BC CREDITO	ident_acreedor	varchar2(15)	
BC CREDITO	id_titular	number(10)	
BC CREDITO	responsabilidad	varchar2(20)	
BC CREDITO	fecha_ult_act	date	
BC CREDITO	monto_aprobado	number(11,2)	
BC CREDITO	plazo_meses	number(3)	
BC CREDITO	tasa_interes	number(2)	
BC MOVIMIENTO_USUARIO			Tabla que registra las operaciones que registra un uysuario dentro del sistema
BC MOVIMIENTO_US	id_movimiento	number(5)	
BC MOVIMIENTO_US	id_usuario	number(5)	
BC MOVIMIENTO_US	usuario_bd	varchar2(15)	
BC MOVIMIENTO_US	usuario_so	varchar2(30)	
BC MOVIMIENTO_US	detalle_movimiento	varchar2(100)	
BC MOVIMIENTO_US	fecha_movimiento	date	
BC MOVIMIENTO_US	hora_inicio	varchar2(12)	
BC MOVIMIENTO_US	hora_fin	varchar2(12)	
BC MOVIMIENTO_US	ip_movimiento	varchar2(15)	
BC PROVINCIA_ESTADO			Tabla de detalle provincias
BC PROVINCIA_EST	id_pe	number(4)	

TABLA	CAMPO	TIPO DE DATO	DESCRIPCION
BC_PROVINCIA_EST	id_pais	number(4)	
BC_PROVINCIA_EST	nombre	varchar2(60)	
BC_PAIS			Tabla de detalles de paises
BC_PAIS	id_pais	number(4)	
BC_PAIS	nombre	varchar2(60)	
BC_CIUADAD			Tabla de nombres de ciudades
BC_CIUADAD	id_ciudad	number(5)	
BC_CIUADAD	id_pe	number(4)	
BC_CIUADAD	nombre	varchar2(40)	
BC_TITULAR			Tabla que registra datos de los titulares de cada credito
BC_TITULAR	id_titular	number(10)	
BC_TITULAR	tipo_titular	char(1)	
BC_TITULAR	tipo_identificacion	varchar2(3)	
BC_TITULAR	identificacion	varchar2(15)	
BC_TITULAR	nombre	varchar2(25)	
BC_TITULAR	apellido	varchar2(25)	
BC_TITULAR	razon_social	varchar2(70)	
BC_TITULAR	nombre_comercial	varchar2(70)	
BC_TITULAR	fecha_ingreso	date	
BC_TITULAR	estado	char(1)	
BC_TITULAR	habil_cta_cte_ch	char(2)	
BC_TITULAR	id_calificacion	char(1)	
BC_TIPO_ACREEDOR			
BC_TIPO_ACREEDOR	id_tipo_ac	number(2)	
BC_TIPO_ACREEDOR	tipo_ac	varchar2(20)	
BC_TIPO_ACREEDOR	descripcion	varchar2(50)	
BC_TIPO_CREDITO			Tabla que registra descripcion de los tipos de creditos
BC_TIPO_CREDITO	id_tipo_cred	number(2)	
BC_TIPO_CREDITO	descripcion	varchar2(50)	
BC_DETALLE_TITULAR			Tabla que registra detalles adicionales delos titulares de creditos
BC_DETALLE_TITULAR	id_titular	number(10)	
BC_DETALLE_TITULAR	domicilio	varchar2(60)	
BC_DETALLE_TITULAR	trabajo	varchar2(60)	
BC_DETALLE_TITULAR	telefono	varchar2(30)	
BC_DETALLE_TITULAR	declarativa_titular	varchar2(500)	
BC_DETALLE_TITULAR	fecha_declarativa	date	
BC_DETALLE_TITULAR	asunto_declarativa	varchar2(60)	
BC_DETALLE_TITULAR	aviso	varchar2(60)	
BC_DETALLE_TITULAR	descripcion_aviso	varchar2(500)	
BC_DETALLE_TITULAR	observaciones	varchar2(500)	
BC_USUARIO			Tabla que registra datos de los usuarios operadores y administradores del sistema
BC_USUARIO	id_usuario	number(5)	
BC_USUARIO	tipo_usuario	varchar2(4)	
BC_USUARIO	tipo_identificacion	varchar2(3)	
BC_USUARIO	identificacion	varchar2(15)	
BC_USUARIO	nombre	varchar2(25)	
BC_USUARIO	apellido	varchar2(25)	
BC_USUARIO	id_sucursal	number(4)	
BC_USUARIO	usuario	varchar2(15)	
BC_USUARIO	clave	varchar2(4000)	
BC_USUARIO	fecha_ingreso	date	
BC_USUARIO	fecha_salida	date	
BC_USUARIO	email	varchar2(40)	
BC_USUARIO	estado	char(1)	
BC_CONSULTA_USUARIO			Tabla que registra las consultas realizadas por los susuarios del sistema
BC_CONSULTA_USU	id_consulta	number(10)	
BC_CONSULTA_USU	usuario_usuario	varchar2(15)	
BC_CONSULTA_USU	identificacion_titular	varchar2(15)	
BC_CONSULTA_USU	fecha_consulta	date	
BC_CONSULTA_USU	hora_consulta	varchar2(12)	
BC_CONSULTA_USU	ip_consulta	varchar2(15)	