



Universidad Austral de Chile

Facultad de Ciencias de la Ingeniería
Escuela de Ingeniería Civil en Informática

MODELO DE CURSE CENTRALIZADO DE OPERACIONES DE CRÉDITO MEDIANTE DIGITALIZACIÓN DE IMÁGENES

Tesis para optar al Título de:
Ingeniero Civil en Informática

Profesor Patrocinante:
Sr. Martín Solar Monsalves
Docente Instituto de Informática

CLAUDIA JACQUELYN MONTECINO QUEZADA
ENRIQUE ALEJANDRO MOREIRA HANSEN

VALDIVIA – CHILE

2006

Valdivia, 31 de Mayo de 2006

De : Martín Gonzalo Solar Monsalves
A : Director Escuela Ingeniería Civil en Informática
Ref.: Informe Calificación Trabajo de Titulación

Nombre Trabajo de Titulación:

"MODELO DE CURSE CENTRALIZADO DE OPERACIONES DE CRÉDITO MEDIANTE DIGITALIZACIÓN DE IMÁGENES."

Nombre Alumnos:

Claudia Jacquelyn Montecino Quezada.
Enrique Alejandro Moreira Hansen.

Evaluación:

Cumplimiento del objetivo propuesto	7.0
Satisfacción de alguna necesidad	7.0
Aplicación del método científico	6.0
Interpretación de los datos y obtención de conclusiones	7.0
Originalidad	7.0
Aplicación de criterios de análisis y diseño	6.5
Perspectivas del trabajo	6.8
Coherencia y rigurosidad lógica	7.0
Precisión del lenguaje técnico en la exposición, composición, redacción e ilustración	6.8
Nota Final	6.8

Sin otro particular, atte.:



Martín Solar Monsalves

Fecha : Lunes 5 de Junio de 2006

DE : Claudia P. Cabezas Aravena (Jefe de Proyecto BancoEstado)

Para : DIRECTOR ESCUELA INGENIERÍA CIVIL EN INFORMÁTICA

MOTIVO: Evaluación del Proyecto de Tesis.

INFORME TRABAJO DE TITULACIÓN

Nombre Trabajo de Titulación:

"Modelo de Curse Centralizado de Operaciones de Crédito mediante Digitalización de Imágenes".

Alumnos:

Claudia Jacquelyn Montecino Quezada

Enrique Alejandro Moreira Hansen

Nota: 7.0 (Siete)

FUNDAMENTO DE LA NOTA: Me parece un muy trabajo, donde se cumplieron los objetivos planteados, además de ser un muy buen aporte en cuanto a la metodología utilizada y a la solución implementada. Además el equipo de trabajo demuestra un amplio dominio sobre el tema.


Atte.
Claudia Cabezas Aravena
Rut: 12.051.907-7

Valdivia, 12 Junio del 2006

DE : Miguelina Vega Rosales

Profesor Instituto Informática

A: Dirección Escuela Ingeniería Civil en Informática

Informo a usted que el Proyecto de Título “ MODELO DE CURSE CENTRALIZADO DE OPERACIONES DE CRÉDITOS MEDIANTE DIGITALIZACIÓN DE IMÁGENES ”, presentado por la señorita Claudia Montecino Quezada y el señor Enrique Moreira Hansen, cumple con el objetivo general propuesto, que es contar con un Sistema y el soporte tecnológico adecuado que permita administrar, hacer seguimiento y controlar el flujo de procesos asociados al curse y visación centralizado de operaciones con documentos digitalizados (imágenes).

La metodología de trabajo y el lenguaje utilizado es el adecuado, sin embargo, en el trabajo los alumnos no mostraron la capacidad de síntesis. Por lo anteriormente expuesto., califico este proyecto de título con nota 6,8 (seis, ocho).

Atentamente



Miguelina Vega R.

AGRADECIMIENTOS

Este trabajo se lo dedico a mis padres, por su apoyo incondicional, su confianza y su infinito cariño, todo lo que soy se lo debo a ellos.

Quisiera agradecer a mi hermano Julito, por su compañía y ayuda. A Víctor, mi amigo incondicional, por su cancioncita ...

Y a mí amado esposo Joaquín, por estar conmigo a partir de ahora y en adelante.

Claudia

A la memoria de mi Madre

A la memoria de mi Padre

Sin su motivación, esfuerzo y apoyo, no estaría en esta instancia. Para ellos todo mi amor y agradecimiento, siempre estarán en mi corazón.

Enrique

En conjunto, quisiéramos agradecer a todas las personas de la escuela de Ingeniería Civil en Informática, especialmente al Profesor Martin Solar, a la Profesora Sra. Miguelina Vega y a la Sra. Juanita Paredes por su constante paciencia, apoyo y disposición. Agradecer también a Claudia Cabezas, por su paciencia y confianza en este proyecto.

INDICE

INDICE	2
INDICE DE TABLAS Y FIGURAS	4
RESUMEN	5
SUMMARY	6
CAPITULO 1: INTRODUCCIÓN	7
ANTECEDENTES GENERALES	7
DEFINICIONES BASICAS PARA EL PROCESO DE DESARROLLO	10
CAPITULO 2: DEFINICIÓN DE OBJETIVOS	33
OBJETIVO GENERAL	33
OBJETIVOS ESPECÍFICOS.	33
CAPÍTULO 3: ANÁLISIS Y DISEÑO DE SISTEMA INFORMÁTICO	34
ESTADO ACTUAL	34
DESCRIPCIÓN DE LA METODOLOGÍA	35
LEVANTAMIENTO CASOS DE USO EN SUCURSALES	43
Caso Uso Solicitud Apertura Operación de Crédito	44
Caso Uso Creación Operación en GDI	45
Caso Uso Envío Operación al CPN	46
Caso Uso Recepción de Operación desde el CPN	47
LEVANTAMIENTO CASOS DE USO EN CPN	48
Caso Uso Asignación Operación en el CPN	48
Caso Uso Control Operaciones en el CPN	49
Caso Uso Devolución Operación a Sucursales	49
MODELO DE CURSE PARA SUCURSALES Y CPN	50
CAPITULO 4: MODELAMIENTO DEL SISTEMA INFORMATICO	52
CONSIDERACIONES INICIALES	52
De la Arquitectura	52
Del Tamaño de los objetos	52
De las herramientas a utilizar	53
De la implementación de la Capa de Negocio	54
DISEÑO DE MODELO DE DATOS	55
DISEÑO DE MODELO DE OBJETOS	58
Capa de Datos	58
Capa de Negocios	62
MODELAMIENTO DE ROLES EN SUCURSALES	66
MODELAMIENTO DE ROLES EN CPN	67
CAPITULO 5: CONSTRUCCION DEL SISTEMA INFORMATICO	68
CONSTRUCCIÓN DE PROTOTIPO	68
Prototipo Sucursales	70
Prototipo CPN	74

VALIDACIONES Y ESPECIFICACIONES DEL SISTEMA.....	77
Encargado Correspondencia	77
Jefe de Operaciones	81
Supervisor CPN.....	85
Administrativo CPN.....	87
DESARROLLO DE OBJETOS COM+	88
DESARROLLO DE CAPA DE PRESENTACIÓN.....	92
INTEGRACIÓN DE COM+ A CAPA DE PRESENTACIÓN	93
POBLAMIENTO DE DATOS.....	94
INTEGRACIÓN DE LA SOLUCIÓN CON SERVICIO DE DIGITALIZACIÓN DE IMÁGENES.	98
Definición Proceso.....	99
Explicación del Proceso.....	101
PREPARACIÓN DE AMBIENTE DE PRUEBAS	104
<i>CAPITULO 6: IMPLANTACIÓN DE LA SOLUCIÓN INFORMÁTICA EN PRODUCCIÓN.....</i>	<i>140</i>
PUESTA EN PRODUCCIÓN DE LA SOLUCIÓN	140
IMPLANTAR LA SOLUCIÓN EN SUCURSALES PILOTO Y CPN.....	144
SEGUIMIENTO DEL PROCESO DE CURSE IMPLANTADO.....	152
<i>CAPITULO 7: ESTABLECIMIENTO DE NUEVOS PARÁMETROS DE MEDICIÓN.....</i>	<i>156</i>
RECOPIACIÓN DE INFORMACIÓN SOBRE EL FUNCIONAMIENTO DEL SISTEMA.....	158
ESTUDIO DE MEJORAS AL SISTEMA INFORMÁTICO.	163
<i>CAPITULO 8: CONCLUSIONES</i>	<i>167</i>
<i>BIBLIOGRAFÍA</i>	<i>170</i>
DOCUMENTOS NORMATIVA BANCOESTADO	171
REFERENCIAS INTERNET.....	172
<i>APENDICE 1: GLOSARIO.....</i>	<i>174</i>
<i>APENDICE 2: ABREVIATURAS UTILIZADAS.</i>	<i>175</i>
<i>ANEXO 1: EXTRACTO CODIGO FUENTE</i>	<i>177</i>
<i>ANEXO 2: DICCIONARIO DATOS.....</i>	<i>210</i>
<i>ANEXO 3: INFORME PRUEBA CONTROLADA PUNTA ARENAS - CPN</i>	<i>216</i>

INDICE DE TABLAS Y FIGURAS

FIGURA 1. 1 APLICACIÓN MONOLÍTICA 11

FIGURA 1. 2 APLICACIÓN MONOLÍTICA CON OBJETOS 12

FIGURA 1. 3 APLICACIÓN CLIENTE – SERVIDOR..... 13

FIGURA 1. 4 MODELO DE DESARROLLO DE MÚLTIPLES CAPAS 16

FIGURA 1. 5 CLASES Y OBJETOS 23

FIGURA 1. 6 USO DE CLASES DESDE DISTINTAS APLICACIONES..... 25

FIGURA 1. 7 USO DE CLASES MEDIANTE COMPONENTES 26

FIGURA 1. 8 ESQUEMA DE SERVIDOR FUERA DE PROCESO..... 28

FIGURA 1. 9 ESQUEMA DE SERVIDOR DENTRO DE PROCESO 29

FIGURA 1. 10 ESQUEMA DE SERVIDOR DENTRO DE PROCESO PARA VARIOS PROCESOS 29

FIGURA 3. 1 DESARROLLO ORIENTADO A PROTOTIPOS..... 39

FIGURA 3. 2 SOLICITUD APERTURA CRÉDITO..... 44

FIGURA 3. 3 CREACIÓN OPERACIÓN DE CURSE 45

FIGURA 3. 4 ENVÍO OPERACIÓN A CPN 46

FIGURA 3. 5 RECEPCIÓN OPERACIONES EN CPN..... 47

FIGURA 3. 6 ASIGNACIÓN DE OPERACIONES 48

FIGURA 3. 7 CONTROL DE OPERACIONES 49

FIGURA 3. 8 DEVOLUCIÓN OPERACIONES A SUCURSAL 49

FIGURA 3. 9 FLUJO CURSE DE OPERACIONES EN SUCURSALES 50

FIGURA 3. 10 FLUJO CURSE OPERACIONES EN CPN..... 51

FIGURA 4. 1 MODELO DE DATOS RELACIONAL..... 57

FIGURA 4. 2 MODELO OBJETOS COM PRINCIPAL 58

FIGURA 4. 3 MODELO OBJETOS COM PRINCIPAL DE CONSULTAS 59

FIGURA 4. 4 MODELO OBJETOS COM PRINCIPAL - ADMINISTRADOR..... 61

FIGURA 4. 5 MODELO OBJETOS COM PRINCIPAL NEGOCIOS..... 62

FIGURA 4. 6 MODELO OBJETOS COM PRINCIPAL NEGOCIOS - CONSULTAS 63

FIGURA 4. 7 MODELO OBJETOS COM PRINCIPAL NEGOCIOS - ADMINISTRACIÓN 64

FIGURA 4. 8 ROLES DEFINIDOS EN SUCURSAL 66

FIGURA 4. 9 ROLES DEFINIDOS EN CPN 67

FIGURA 5. 1 INGRESO OPERACIÓN EN SUCURSAL 71

FIGURA 5. 2 INGRESO OPERACIÓN EN SUCURSAL - IMÁGENES..... 71

FIGURA 5. 3 CERTIFICACIÓN DE OPERACIONES 73

FIGURA 5. 4 CERTIFICACIÓN DE OPERACIONES – RECHAZAR OPERACIONES 73

FIGURA 5. 5 RESUMEN DE OPERACIONES A ASIGNAR 74

FIGURA 5. 6 RESUMEN DE OPERACIONES A CONTROLAR 75

FIGURA 5. 7 RESUMEN DE OPERACIONES PARA CONTROL EXPOST 76

FIGURA 5. 8 PANTALLA DE INGRESO DE IMÁGENES 102

FIGURA 5. 9 DESPLIEGUE DE IMÁGENES INGRESADAS PARA UNA OPERACIÓN 102

FIGURA 5. 10 DESPLIEGUE DE IMÁGENES CONSULTADAS..... 103

TABLA 5. 1 ACTIVIDADES A CONTROLAR POR EL ÁREA DE TESTING 105

TABLA 5. 2 PLAN DE PRUEBAS PERFIL ESCANEADOR (ENCARGADO CORRESPONDENCIA) 109

TABLA 5. 3 PLAN DE PRUEBAS PERFIL JEFE DE OPERACIONES 115

TABLA 5. 4 PLAN DE PRUEBAS PERFIL SUPERVISOR CPN. 123

TABLA 5. 5 PLAN DE PRUEBAS DEL PERFIL ADMINISTRATIVO CPN 131

TABLA 6. 1 ACTIVIDADES A REALIZAR POR EL IMPLANTADOR EN SUCURSALES 146

TABLA 6. 2 ACTIVIDADES A REALIZAR POR EL TÉCNICO IBM..... 147

TABLA 6. 3 PREGUNTAS FRECUENTES PARA GDI - MAU..... 152

TABLA 7. 1 SUCURSALES IMPLEMENTADAS CON GDI 156

TABLA 7. 2 ENCUESTA A USUARIOS DE GDI..... 158

TABLA 7. 3 TIEMPOS INVOLUCRADOS EN EL PROCESO DE CURSE DE OPERACIONES ANTES Y DESPUÉS DEL GDI..... 159

TABLA 7. 4 TIEMPOS DE RESPUESTA EN CPN ANTES Y DESPUÉS DE IMPLEMENTARSE EL GDI..... 160

TABLA 7. 5 VOLÚMENES DE OPERACIONES CURSADAS CON GDI 161

RESUMEN

Este proyecto de Tesis se origina a raíz de la necesidad presente en BancoEstado de contar con una herramienta para agilizar el curso de operaciones de crédito para las sucursales que geográficamente se encuentran mas alejadas. Esta necesidad surge ya que al encontrarse las operaciones de curso de operaciones de crédito centralizadas en el Centro de Procesos Nacional (CPN), en el caso de las sucursales más alejadas, se agregaba un tiempo innecesario por concepto de traslado de la documentación física al CPN.

El “modelo de curso centralizado de operaciones de crédito mediante digitalización de imágenes” al desarrollarse al interior de BancoEstado, ha de utilizar y ceñirse estrictamente a las normativas establecidas al interior del Banco para los desarrollos en la arquitectura Windows DNA. Estas normas indican el uso de componentes COM+, paginas ASP, Visual Basic, Visual Basic Scripting Edition y la integración con el sistema de privilegios del Banco. Además, para la construcción de la aplicación se utilizará un modelo de desarrollo que mezcla el desarrollo en cascada con el desarrollo por prototipos.

A continuación, se presenta el diseño, modelamiento y construcción de un modelo de curso de operaciones de crédito al interior de BancoEstado, utilizando la arquitectura Windows DNA 2000 y obedeciendo las normativas pertinentes para este tipo de desarrollos al interior de BancoEstado.

SUMMARY

This Thesis Project raises from the need present in BancoEstado to count with a tool to speed the road that a Credit operation dispatch need to accomplish to be complete, this is especially important to the branch offices that are more distant in a geographic sense. This need raises because BancoEstado was centralize this Credit Operations in its “National Process Center” (from its Spanish name “Centro de Procesos Nacional” a.k.a. CPN) and from this the branch offices that are more far to CPN its Credit Operation adds an unnecessary time because of the moving of the physical documentation to the CPN.

When the “Model of Centralized Credit operations dispatch using images digitalized” has developed inside BancoEstado, it needs to use and be in strictly to the rules and methods established in BancoEstado for the Windows DNA development projects. This rules indicates the use of COM+ components, ASP pages, Visual Basic, Visual Basic Scripting Edition and the integration with the privileges system of the Bank. Moreover, for the application develop is used a development method that is a mixture of the cascade method and the prototype method.

Following, is presented the design, the modeling and the construction of a Model of Credit Operations dispatch within BancoEstado, using the Windows DNA 2000 architecture and obeying the related rules for this kind of develops inside BancoEstado.

CAPITULO 1: INTRODUCCIÓN

ANTECEDENTES GENERALES

BancoEstado, ofrece a sus clientes distintos productos, los cuales proveen soluciones a diversas necesidades de los clientes en diferentes ámbitos de aplicación.

En particular, y dentro del ámbito de esta Tesis, el Banco ofrece a sus clientes los productos:

- Créditos de Consumo
- Líneas de Crédito (asociadas a Cuentas Corrientes)
- Tarjetas de Crédito

Al solicitar el cliente alguno de estos productos, se genera un proceso llamado "Curse de Operaciones de Crédito". Actualmente, este proceso demora cerca de 10 días hábiles.

El Banco, está empeñado en aumentar el nivel de bancarización¹ del país, y de esta forma entregar más y mejores productos a la mayor cantidad de clientes posibles con la mayor cantidad de puntos de atención a lo largo del país.

Gracias a la modernización que ha experimentado el Banco desde fines de los años 90, este objetivo se ha ido cumpliendo progresivamente en distintos productos, por un lado aumentando la cobertura geográfica, el número de puntos de atención a clientes, como también aumentando considerablemente el número de operaciones cursadas.

A medida que el número de operaciones cursadas aumenta con las distintas campañas, se hace necesario el proveer de mecanismos más rápidos de respuesta a los clientes.

¹ Entregar servicios bancarios de calidad a los clientes de todos los sectores del país.

Es en este contexto que nace el proyecto del “Centro de Procesos Nacional” (CPN) , el cual a través de la centralización de las operaciones, la disponibilidad de personal altamente capacitado, herramientas específicas y la posibilidad de trabajar en la modalidad 7x24, disponibiliza el trabajo con mayores volúmenes de operaciones, aumentando las ventas de productos bancarios. Esta centralización permite que los ejecutivos de las sucursales se centren más en el proceso de venta de productos, que en el proceso de curso de operaciones de crédito.

Sin embargo, existe en el proceso diseñado el problema que dada la distancia relativamente grande entre el CPN y las sucursales más extremas o lejanas del país, la llegada de los documentos al CPN, puede demorar demasiado tiempo, retrasando de esta forma el curso de las operaciones al cliente.

Es en este contexto que nace el proyecto GDI, el cual busca definir una serie de sucursales que presentan una criticidad especial, para las cuales se proveerá un sistema informático que asegure el curso de sus operaciones de crédito en un plazo de 1 día. Para ello, se enviará de forma inmediata al CPN la documentación asociada a una Operación de Crédito para las sucursales definidas como críticas a través del GDI.

Este proyecto, nace con ciertas definiciones básicas:

- Se implementará inicialmente en un grupo de sucursales piloto (3) para posteriormente ampliar su base de implementación al total de sucursales definidas como críticas (44).
- Debe ser soportado por la tecnología disponible en este momento en el Banco, sin ajustes al Software existente y sin una inversión relevante en Hardware.
- El proceso debe proveer controles manuales e independizando el control del flujo respecto del proceso de curso.
- Han de respetarse las disposiciones establecidas en el Banco tanto para el desarrollo mismo como para la arquitectura de la solución y los procedimientos de pruebas y puesta en producción.

Se definirán de este modo, dos entidades básicas para el proceso de Curse: Sucursal y CPN, cuyas funcionalidades generales son:

Sucursal

- Creación de una operación, especificándose datos demográficos que identifican una operación de crédito y el cliente que la solicita.
- Digitalización de los documentos asociados a una operación.
- Consulta de documentos digitalizados.
- Envío de imágenes asociadas a una operación.

CPN

- Consulta de operaciones.
- Asignación de operaciones para proceso de Control de Operaciones, identificándose un cambio de estado en la operación.
- Impresión de imágenes.
- Cambio de estado de operación al estado definido como final.

Los roles establecidos en el curse de operaciones son:

Sucursales

- Escaneador(Encargado Correspondencia)
- Asistente de Negocios
- Jefe de Operaciones

CPN

- Supervisor de CPN
- Analista de Control Operacional.

Por otro lado, los resultados que se espera obtener con este sistema son:

- Registro y seguimiento de operaciones vía imágenes, contemplando cadena completa de procesos desde la sucursal hasta la visación de la operación y control expost² de la carpeta física asociada a la operación.
- Identificación y distribución de las operaciones a los roles establecidos según sean los estados de la operación en el flujo operativo.
- Generación de reportes de las operaciones cursadas.
- Seguimiento mediante mensajería del ciclo de una operación.
- Aumento del curso de operaciones.
- Agilización por parte del cliente de los trámites asociados a un curso de crédito, línea de crédito o tarjetas de crédito.

Se reserva la opción de implementar nuevas funcionalidades, modificar tecnología utilizada según sean los requerimientos del usuario y de la plataforma tecnológica del Banco.

Este proyecto de tesis contempla descripciones y especificaciones asociadas netamente al Sistema de Gestión de Imágenes (GDI), reservándose el derecho de mencionar otros sistemas sin entrar en detalle de éstos, esto se debe a las características confidenciales y estratégicas que estos sistemas tienen para el Banco.

DEFINICIONES BASICAS PARA EL PROCESO DE DESARROLLO

En BancoEstado, el desarrollo de aplicaciones actualmente se hace utilizando el estándar Windows DNA 2000. Pero, ¿Como se ha llegado al desarrollo con esta versión de Windows

² Control de documentación versus documentación en carpeta física.

DNA?, ¿Cómo se pasó desde las aplicaciones Cliente Servidor a DNA? y ¿Por qué se hace este cambio?

El primer paso importante en la evolución de la arquitectura de las aplicaciones es el paso de las aplicaciones de carácter Monolítico a aplicaciones Cliente Servidor.

Las aplicaciones monolíticas (ver figura 1.1) contenían en sí mismas todo lo necesario para su ejecución. De esta forma, ellas no necesitaban recursos externos a ella para su ejecución, pero tampoco podían ofrecer ni utilizar servicios a otras aplicaciones de una forma dinámica y cooperativa. Claramente, esta aproximación al desarrollo de aplicaciones facilitaba en cierta forma la vida a los desarrolladores, pues al ejecutarse el programa solo en un Sistema Operativo determinado no existían problemas de portabilidad, por otro lado no existían problemas de conectividad pues no existía conectividad con otras máquinas. Además, la seguridad asociada a este tipo de aplicaciones era relativamente simple pues normalmente se trataba de aplicaciones monousuario. Finalmente, al no ser necesaria la interacción con otras aplicaciones ni la interoperabilidad con otras plataformas, era permitido el uso de formatos propietarios para el almacenamiento de los datos.

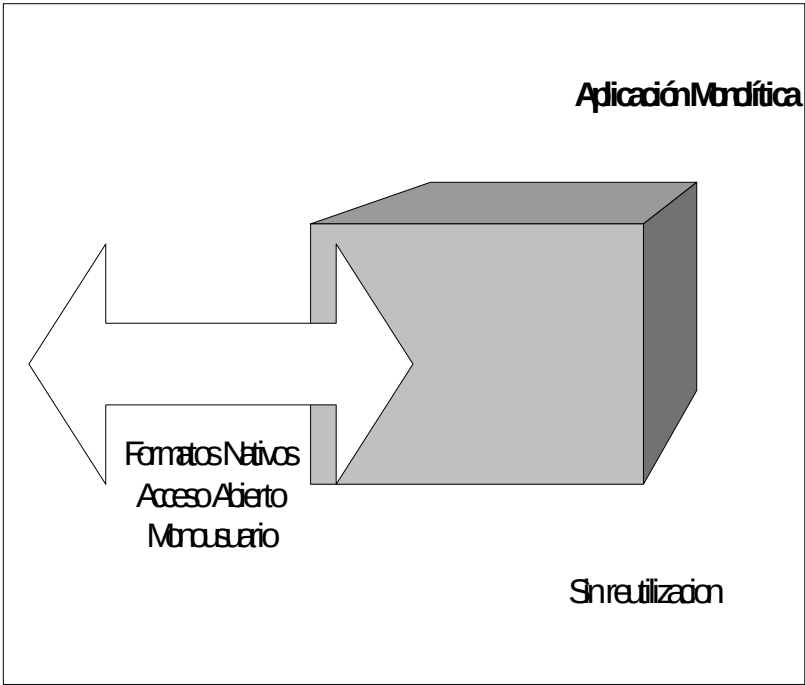


Figura 1. 1 Aplicación Monolítica

Con el tiempo, la complejidad de los programas creció, y apareció evidentemente la necesidad de reutilizar en algo al menos los esfuerzos realizados en desarrollos anteriores. Las aplicaciones monolíticas no proveían de herramientas para la reutilización de código, ni tampoco era posible delegar tareas más complejas en una máquina más potente que sobre la cual se ejecutaba la aplicación. Con el tiempo, se hizo clara la necesidad de reutilizar los esfuerzos previos de desarrollo. Esto dio origen a la programación orientada a objetos (OOP), donde las funcionalidades podían ser implementadas y luego de probadas, encapsuladas en una entidad y posteriormente ser reutilizadas donde quiera que la misma funcionalidad fuese necesaria nuevamente. Lamentablemente, el nivel de reutilización de código era normalmente en cuanto a reutilizar el código fuente o en el caso de objetos previamente compilados a través de enlazamientos durante la compilación. De esta forma, dentro de la aplicación, los bloques comenzaron a cooperar (ver figura 1.2), pero desde el punto de vista exterior las aplicaciones seguían siendo monolíticas pues no proveían de formas de utilización de las rutinas y procedimientos a entes externos a la misma aplicación.

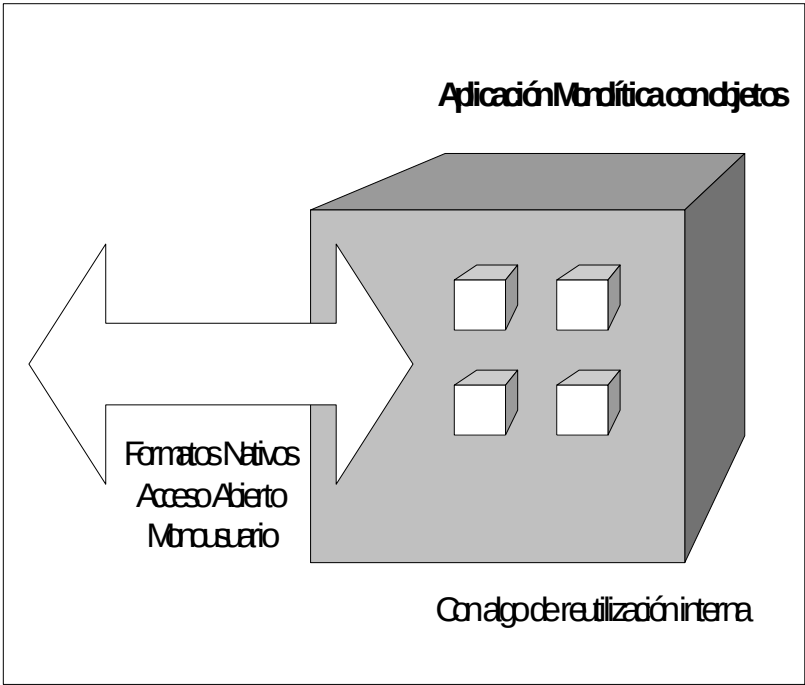


Figura 1. 2 Aplicación Monolítica con Objetos

Por otro lado, se empezaron a desarrollar sistemas de administración de datos de forma centralizada. Inherentemente, las bases de datos constituyen un recurso de uso centralizado. Todos los usuarios necesitan ver la misma versión de los datos que necesitan. Por otro lado, si estos usuarios son varios y están distribuidos espacialmente en ubicaciones distintas, se hace necesario acceder a estos datos a través de una red. De esta forma, nace la arquitectura Cliente Servidor (ver figura 1.3), permitiendo: la compartición de recursos escasos y costosos pero poderosos tanto de hardware como de software en el servidor para asegurar confiabilidad y disponibilidad; minimizar los recursos en el cliente pues la implementación de las consultas como de las transacciones residen y son administradas en el servidor de datos por lo cual es posible que se implementen diferentes reglas para diferentes aplicaciones.

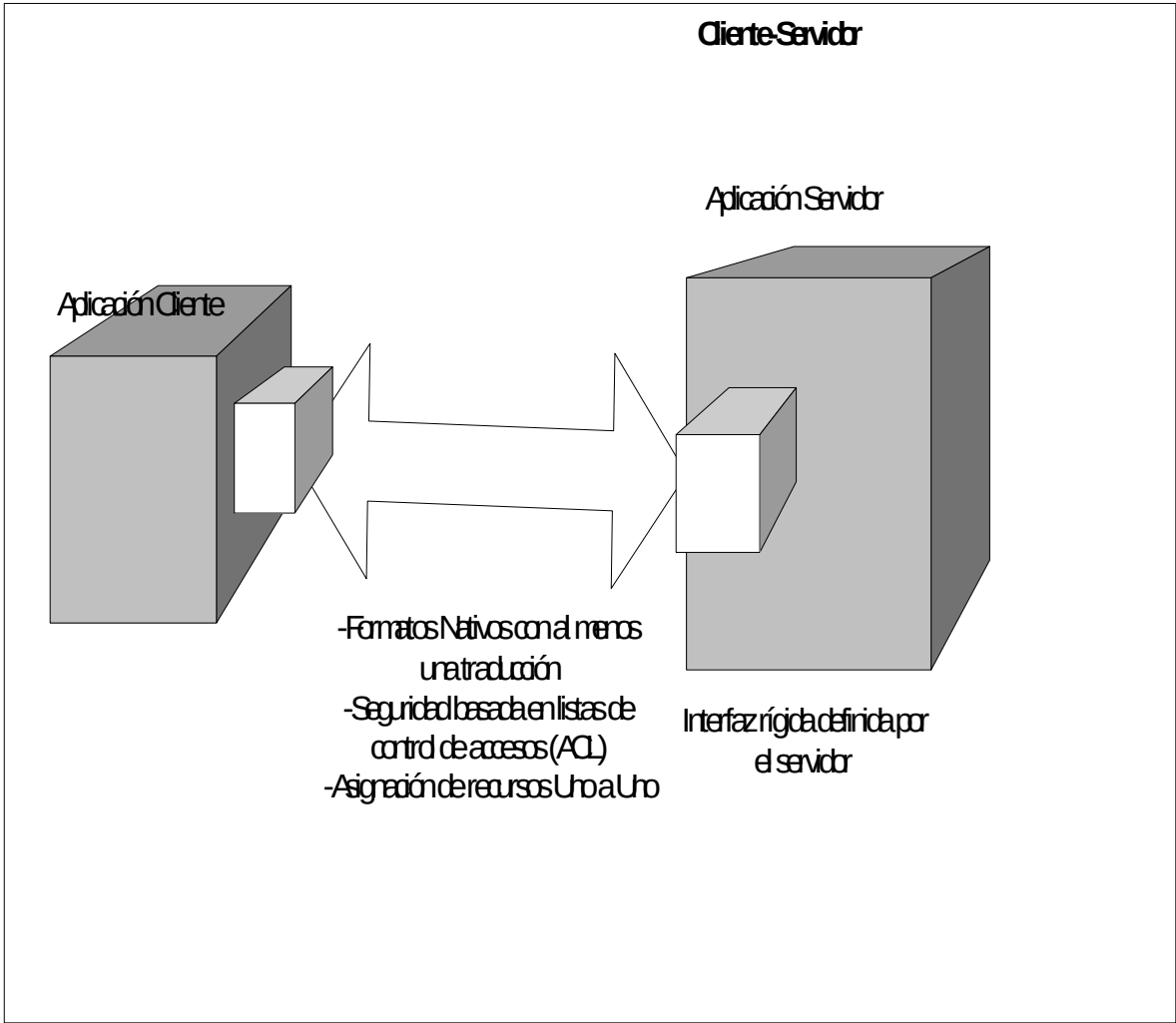


Figura 1. 3 Aplicación Cliente – Servidor

Como se dijo anteriormente, al moverse de las aplicaciones de tipo monolítico a las del tipo Cliente Servidor, aparece naturalmente el problema de la conectividad. Además, el tema de los diferentes tipos de datos utilizados también se hace presente al permitirse la interacción de diferentes plataformas (por ejemplo al acceder a servidores ejecutando un motor de base de datos sobre UNIX desde clientes desarrollados para ser ejecutados desde Windows) de esta forma es necesario realizar una transformación de los datos entre el cliente y el servidor naciendo de esta forma las interfaces. Estándares como el SQL en parte resuelven esta problemática permitiendo a los diferentes proveedores convenir en un estándar tanto en la consulta como en la respuesta a los requerimientos. Sin embargo, debido a la multiplicidad de formatos propietarios de los proveedores de motores de datos, aun es necesario lograr la conectividad entre el motor de datos y los diversos clientes. La seguridad en este caso está firmemente ligada a las características y número de posibles clientes. Idealmente, deberíamos establecer reglas de seguridad para cada cliente, pero para grandes aplicaciones este enfoque dificulta mucho su administración, y nos lleva al concepto de la utilización de los recursos en el servidor. Cada cliente activo utiliza recursos como conexiones, memoria usada por la sesión, transacciones, etc. De esta forma el servidor de datos crecerá en recursos a medida que la carga de clientes también aumente lo cual se hace cada vez más difícil de mantener a medida que las aplicaciones salen del ámbito departamental y pasan a ser de uso más intenso y crítico en un ambiente de carácter corporativo. Finalmente, desde el punto de vista de la reutilización de desarrollos anteriores el modelo Cliente Servidor mejora solamente ciertos aspectos ya que provee de una herramienta para reutilizar los temas concernientes a los datos, pero deja aun en el lado del cliente gran parte de la lógica del negocio subyacente. Para subsanar este predicamento y dada la potencia que empezaron a tener los motores de datos, poco a poco los procedimientos almacenados y los triggers³ comenzaron a ser cada vez más pequeños programas en si mismos, incluyendo cada vez más procesamiento que no necesariamente estaba relacionado con la integridad de los datos sino más bien con reglas y lógica relativa al negocio subyacente, es decir unidades de

³ Trozo de código a ejecutar en el caso de la ocurrencia de un evento en la Base de Datos.

procesamiento o algoritmos que representaban algún concepto o idea de importancia para la organización en lo relativo a los datos. Como estas reglas de negocio eran probablemente necesarias de ser usadas nuevamente una y otra vez en distintas partes del desarrollo o aun más en desarrollos distintos, surgió la necesidad de implementarlos de una sola vez, de forma robusta y que sean manejados de forma centralizada en un servidor ad-hoc. Como estos pequeños programas no necesariamente estaban relacionados con la integridad de los datos era relativamente claro que no necesariamente debían de ser implementados en un motor de datos con herramientas y lenguajes orientados al manejo de datos. De este proceso de separación de funcionalidades nació el modelo de desarrollo en múltiples capas.

En este modelo el cual muchas veces es conocido como Modelo de desarrollo en tres capas (ver figura 1.4), los clientes aún se preocupan de la presentación de los datos y reciben los datos del usuario en lo que se conoce como la **capa de presentación**. Los datos siguen residiendo en uno o más servidores en la **capa de datos** quedando en esta capa solamente los procesos y aplicaciones relacionados con el acceso a los datos y a la mantención de la integridad de los datos. Las reglas de negocio se ubican en la **capa de lógica de aplicación** también conocida como **capa de servicios de negocio** o **capa media**. Si bien es cierto que los procedimientos almacenados a veces son utilizados para implementar y para asegurar procedimientos o lógica del negocio esto puede llevar a mejoras sustanciales de velocidad, esto rompe con el concepto más purista de desarrollo en múltiples capas, donde la capa de datos está destinada estrictamente a los datos. Además, el término n-capas proviene del hecho de que la capa de lógica de aplicación muchas veces es subdividida en varias capas lógicas dedicadas a funciones específicas.

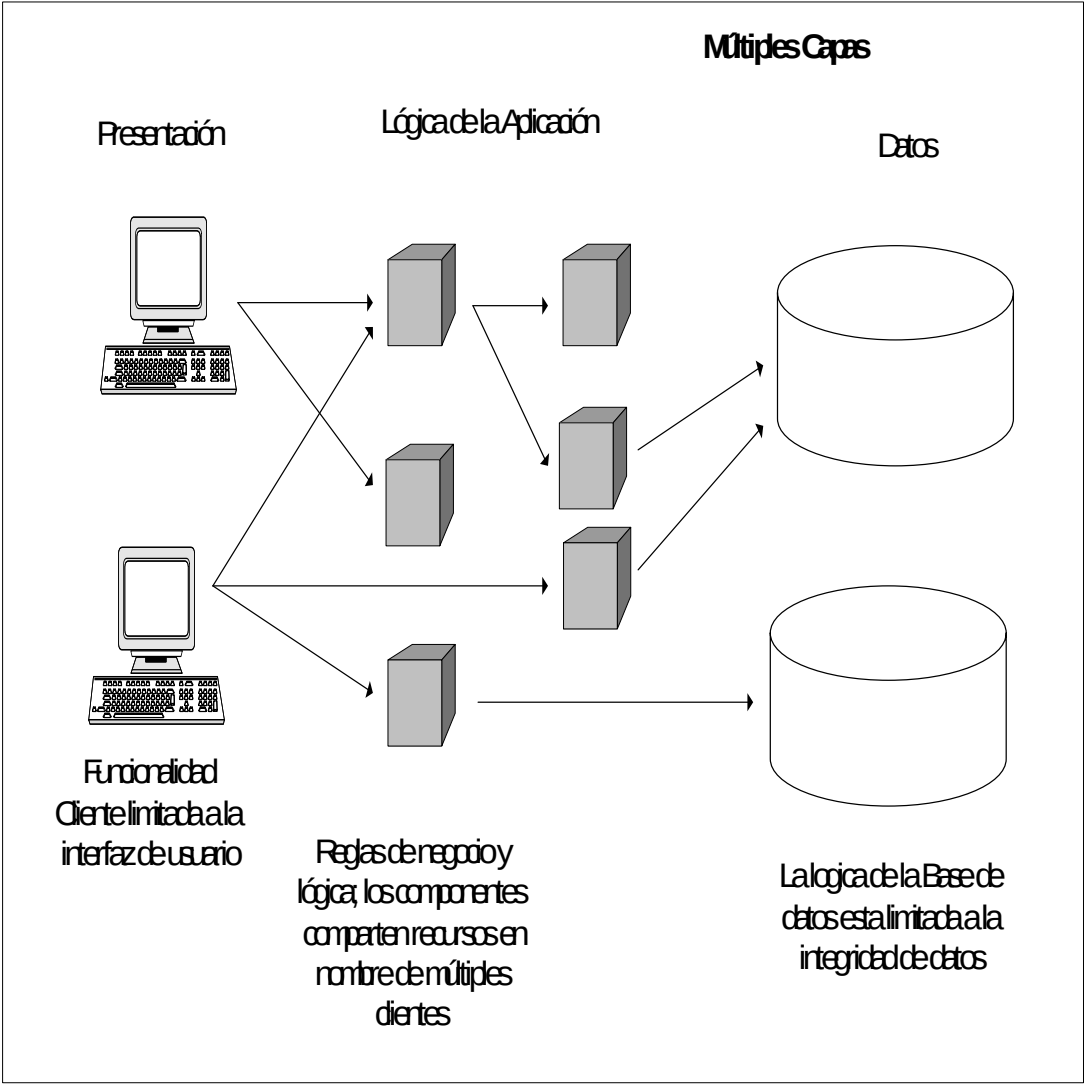


Figura 1. 4 Modelo de desarrollo de Múltiples Capas

Dividir una tarea en una o más capas trae consigo los siguientes beneficios:

- Separación de la presentación de las funcionalidades de la aplicación.
 - o Se hace más fácil cambiar la presentación visual de forma separada de las funcionalidades que ya están probadas.
 - o La máquina destinada a ser cliente, sólo requiere poder ejecutar las tareas relacionadas con el despliegue de la aplicación dentro del browser. La parte compleja de cálculo es trasladada al servidor.
- Posibilidad de optimizar cada capa de forma separada para el escalamiento y el desempeño.

- o Por ejemplo, se puede destinar el esfuerzo de potenciamiento a la capa de datos, o a la de lógica de la aplicación, manteniendo los clientes en el mínimo posible de potencia.
- Existe la posibilidad (aunque a veces bastante limitada) de desarrollos en paralelo durante la etapa de desarrollo.
 - o Definiendo interfaces claras entre las capas, es posible lograr cierto grado de paralelismo en el desarrollo.
- Posibilita la reutilización de módulos funcionales.
 - o Nuevamente las interfaces han de ser limpias y útiles para los otros desarrolladores. Los diseñadores han de tener en mente que otros programas usarán sus componentes.
- Se puede seleccionar la plataforma apropiada para cada tarea.

Existen varias propuestas de arquitectura de múltiples capas (DNA, J2EE, WebSphere, etc.), las diferencias entre estas radican por un lado en los Sistemas Operativos sobre los cuales se ejecutan y la tecnología de objetos distribuidos que se propone (COM+, Corba). En el caso de BancoEstado, se ha optado por utilizar Windows DNA.

¿Qué es Windows DNA?, ¿Cómo se desarrolla en múltiples capas con Windows DNA? y ¿Que herramientas/servicios dan soporte al desarrollo en múltiples capas en Windows DNA?

DNA y la arquitectura de múltiples capas

Servicios de Presentación

Si bien es cierto, las herramientas para Windows ofrecieron tempranamente ayudas para el desarrollo de interfaces de usuario (Microsoft Foundation Classes, Windows Forms), con la llegada del web, las aplicaciones comenzaron a ser normalmente desarrolladas con esta plataforma en vista. Principalmente esto se debió a la promesa de dejar de lado el clásico

problema de la distribución de versiones. En el caso del desarrollo basado en Web, la única versión de la aplicación es la que reside en el servidor, y los clientes acceden todos a esta versión usando sus browsers. Es cierto que en el caso de aplicaciones de carácter empresarial se presenta el problema del manejo de versiones entre los servidores, pero es un tema mucho más manejable que manejar distintas versiones de programas clientes y su distribución coordinada dentro de la organización. Por otro lado, al existir distintos productos para explorar o ver las páginas web (Netscape Navigator, Mozilla, Internet Explorer, Opera, etc.) se presenta el problema del manejo de distintas versiones del software para ser visto por los distintos navegadores y/o la aplicación web ha de ser lo suficientemente robusta como para “adaptarse” al navegador con el cual se está accediendo en un momento dado. Este predicamento es salvado fácilmente en el caso de las aplicaciones desarrolladas al interior de las empresas pues se puede hacer todo el desarrollo pensando en un único tipo de navegador. En el caso de BancoEstado al venir Internet Explorer incluido con el Sistema Operativo Windows, la elección fue natural.

Por otro lado, junto con el navegador para acceder a las páginas web que constituirán la aplicación es necesario contar con un servidor que provea dichas páginas. Además, como se desea que la lógica de presentación resida en el servidor web, es necesario que este tipo de servidor provea de ejecución de código de lado del servidor.

En el caso de Windows DNA, se propone el desarrollo de páginas interactivas utilizando VBScript, las cuales serán desplegadas utilizando Internet Explorer como navegador.

Capa de Lógica de Aplicación

En DNA, esta capa está compuesta por componentes de software que no cuentan con una representación visual, son del tipo “sin interfaz de usuario”, sin embargo, estos componentes de software han de ser capaces de comunicarse con las otras capas. El modelo de tecnología de componentes de Windows es COM+ (como sucesor de COM).

COM+ nos permite desarrollar pequeñas piezas de funcionalidad que pueden dejarse disponibles en un servidor y ser usadas desde cualquier cliente que cuente con COM+. Esto es muy útil cuando deseamos utilizar un componente tipo desde distintas aplicaciones, además de permitir que estas aplicaciones estén escritas en lenguajes distintos. De esta forma el mismo componente puede ser accedido desde aplicaciones escritas en VBScript, o desde Visual Basic o C++. Una vez que se ha configurado el componente, este está disponible para cualquier aplicación que cuente con los permisos de seguridad adecuados.

Así, la capa de lógica de aplicación en Windows DNA, consiste en muchos componentes COM+ que son capaces de interactuar con servidores web en la capa de presentación. Además, típicamente estos componentes COM+, utilizan a otros componentes COM+ como ADO para acceder a la capa de datos.

Por otro lado, junto con la división de las funcionalidades en distintos componentes, debido a que las aplicaciones de carácter empresarial tienden a tener un gran número de usuarios y a necesitar estar disponibles todo el tiempo es que surge la necesidad de manejar los recursos disponibles, disponibilizar los componentes y aislar las eventuales fallas para que no involucren a toda la aplicación o peor aun a todos los componentes residentes en un servidor determinado. Además es necesario proveer de capacidades de procesamiento de transacciones distribuidas. En Windows DNA, esto es provisto por Transaction Server.

Servicios de Datos

En Windows DNA, los servicios de datos son provistos por el servidor relacional de Microsoft, SQL Server. Además, estos datos son accesibles fácilmente al utilizar ActiveX Data Objects (ADO).

Windows DNA 2000

Originalmente, Windows DNA se implementó en BancoEstado utilizando las herramientas:

- Windows NT 4.0 Server: Sistema Operativo Base para los servidores.

- Windows NT 4.0 Workstation: Sistema Operativo para las estaciones de trabajo.
- MTS 4.0 e IIS 4.0: Servidor de transacciones distribuidas y servidor Web.
- SQL Server 6.5: Servidor de datos.
- Visual Studio 6.0: Como herramienta de desarrollo.

Sin embargo, con el advenimiento de Windows 2000, esta arquitectura fue modificada como sigue:

- Windows 2000 Server incluyendo sus servicios nativos:
 - o IIS 5.0
 - o COM+
 - o Seguridad integrada.
 - o Manejo de carga.
 - o Directorio Activo.
- Visual Studio 6.0 como herramienta de desarrollo.
- SQL Server 7.0 y posteriormente SQL Server 2000.

¿Qué es COM+?

COM es un acrónimo de Component Object Model, COM+ es la evolución de COM para Windows 2000. COM es una especificación que está basada en un tipo de estándar binario para la reutilización a través de interfaces. Esto significa que los componentes (que son los bloques o porciones de código fuente precompilado) escritos para COM pueden ser reutilizados sin dependencia del lenguaje en el cual fueron creados. De esta forma no importa si un componente fue creado usando Visual Basic, C++ o incluso Cobol para ser utilizado por cualquier lenguaje de programación que sea capaz de acceder a componentes COM (como Visual Basic, Visual Basic Scripting Edition o C++). Lo único importante en este sentido es que este componente cumpla con la especificación COM.

Las librerías de COM+

COM está implementado a través de código al nivel del Sistema Operativo en lo que se conoce como “Librería COM” utilizando librerías de enlace dinámico (dll's). La librería COM consiste en varias funciones API's que permiten que los componentes desarrollados para y con COM logren comunicarse, es decir, las librerías de COM son las encargadas de proveer la infraestructura para ubicar y activar estos componentes.

Ventajas de COM+

- COM+ Promueve el desarrollo basado en componentes.
 - o Antes del desarrollo basado en componentes, el método más usado para el desarrollo de aplicaciones era el llamado “método procedural”. El desarrollo basado en componentes provee la habilidad de utilizar componentes pre empaquetados, eventualmente de proveedores externos a la organización. Además, provee la reutilización de código dentro de una aplicación.
- COM+ promueve la reutilización de código.
 - o En el desarrollo clásico basado en clases, se podía utilizar una clase y compilarla junto con el programa que la utilizaba y cualquier reutilización de código implicaba copiar y pegar porciones de código a otros proyectos. En el caso de los componentes COM, están diseñados para ser utilizados separadamente de las aplicaciones que los utilizan. De esta manera los componentes COM pueden ser utilizados por numerosas aplicaciones sin problemas.
- COM+ promueve la programación orientada a objetos (OOP).
 - o COM se ha diseñado con la OOP en mente pues provee las tres mayores características de la OOP, a saber: encapsulación (que permite ocultar los detalles de implementación de un objeto), polimorfismo (el cual nos permite que un objeto desarrolle distintos comportamientos) y herencia (la cual nos

permite la reutilización de las clases existentes para crear objetos nuevos y eventualmente más especializados).

- COM+ provee los mecanismos necesarios a los componentes COM para comunicarse unos con otros.
 - o Normalmente los componentes de software que no son COM no se comunicaban muy eficientemente con componentes escritos en lenguajes diferentes. Pero COM provee una solución a este predicamento ya que desde el punto de vista del componente o programa que utiliza un componente COM, COM es neutral al lenguaje de implementación del componente a utilizar. Como resultado, podemos tener un mix de componentes COM desarrollados en distintos lenguajes.
- COM+ provee los medios para acceder a diferentes componentes a través de una red.
 - o Los componentes COM son independientes de su ubicación. Esto significa que un componente puede residir en una máquina y utilizar otro componente que reside en una máquina distinta de la primera, utilizando una red. En otras palabras, las aplicaciones que utilizan COM pueden acceder y compartir componentes COM sin importar donde estos residen. Lo importante aquí es que la aplicación que utiliza los componentes no se preocupa ella misma con los detalles de donde residen los componentes que ha de utilizar, pues esa es una misión de COM.

Objetos y Clases

Objetos

Los objetos son abstracciones basadas en código que representan elementos del mundo real e interacciones entre estos. Cada objeto tiene tres características básicas: **identidad**, **comportamiento** y **estado**. La identidad es simplemente designar al objeto con un nombre único

para que pueda se distinguido de los otros objetos. El comportamiento de un objeto está basado en el conjunto de métodos que pueden ser llamados para consulta o manipular el estado del objeto. El estado del objeto son los datos que están correlacionados con un objeto en particular.

Clases

Un componente puede consistir de una o más clases. Como los objetos son abstracciones de elementos del mundo real y de sus relaciones, entonces para cada objeto ha de haber una clase que defina tal objeto. Una clase es simplemente una plantilla, desde el cual se crea el objeto (ver figura 1.5). Como una clase es una plantilla, cobra sentido pensar que muchos objetos están basados en una clase.

La palabra clase es bastante descriptiva. Por ejemplo, pensemos en dos objetos, una orden de compra y una factura, ambos objetos podrían ser instancias de la clase Documento. Cada Factura u orden de compra puede tener su propios objetos, pero todos los objetos deben ser creados en base a la misma plantilla, la clase Documento.

Por ejemplo, en Visual Basic, definimos clases utilizando módulos de clase. Entonces podemos crear objetos basados en un modulo de clase. Siendo cada objeto una instancia de una clase.

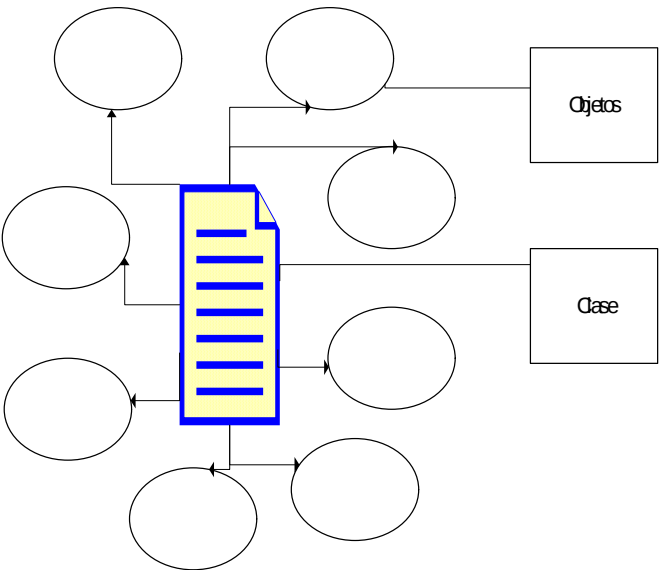


Figura 1. 5 Clases y Objetos

Siguiendo con el ejemplo de Visual Basic, los módulos de clase son parecidos a los formularios estándar, pero tienen una sección de declaraciones seguidas de una serie de subrutinas del tipo **Sub**, **Function** o **Property**. La diferencia principal radica en que las variables y el código en un módulo de clase solo pueden ser usadas creando una instancia de esa clase (un objeto).

Ahora bien, durante la ejecución de un programa, cuando se instancia un objeto desde una clase, un bloque de memoria contiguo es asignado a las variables del objeto. De esta forma la identidad del objeto es la dirección de memoria de este objeto. Su comportamiento es controlado por los métodos definidos para el que pueden ser llamados para consultar o modificar la data asociada al objeto, finalmente, el estado del objeto son los datos contenidos en esa dirección de memoria asociada al objeto.

Componentes COM+

Hasta hace poco, si queríamos por ejemplo, el código para el objeto Cliente dentro de un programa, teníamos que copiar el código completo desde otro programa en nuestra aplicación. Podíamos reutilizar el código, pero si algo cambiaba en el código fuente, debíamos encontrar todos los programas donde este trozo de código estaba incluido, y cambiarlos para que contengan los cambios al programa original.

El problema era que al existir al menos dos copias del mismo código fuente, no existía una manera simple de mantener estas copias sincronizadas. De muchas formas, volvíamos a donde comenzamos, es decir a reutilizar el código a través del copiar y pegar (ver figura 1.6).

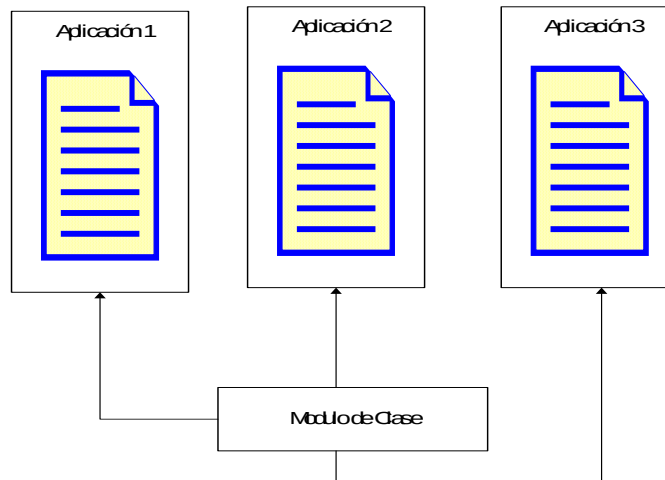


Figura 1. 6 Uso de clases desde distintas aplicaciones

Es aquí donde nace el diseño orientado a componentes. Los componentes proveen una mejor forma de reutilizar los objetos.

Ahora bien, los términos “objeto” y “componente” se usan a veces de manera indistinta, pero denotan cosas distintas y existen importantes diferencias entre ellos:

- Los objetos son creados a partir de clases. Las clases están hechas de código fuente que define la data de los objetos y las rutinas que son necesarias para manipular esa data.
- Los componentes son código binario precompilado. Como los componentes son precompilados, ellos son independientes del lenguaje en que han sido implementados.

Con los componentes en estado binario, podemos alcanzar un nuevo nivel de reutilización de código en nuestras aplicaciones. Volviendo a nuestro ejemplo de la clase *Cliente*, podemos agregar el código fuente en un componente en vez de ponerlo en cada aplicación individual. Las aplicaciones pueden entonces usar el objeto directamente desde el componente (ver figura 1.7).

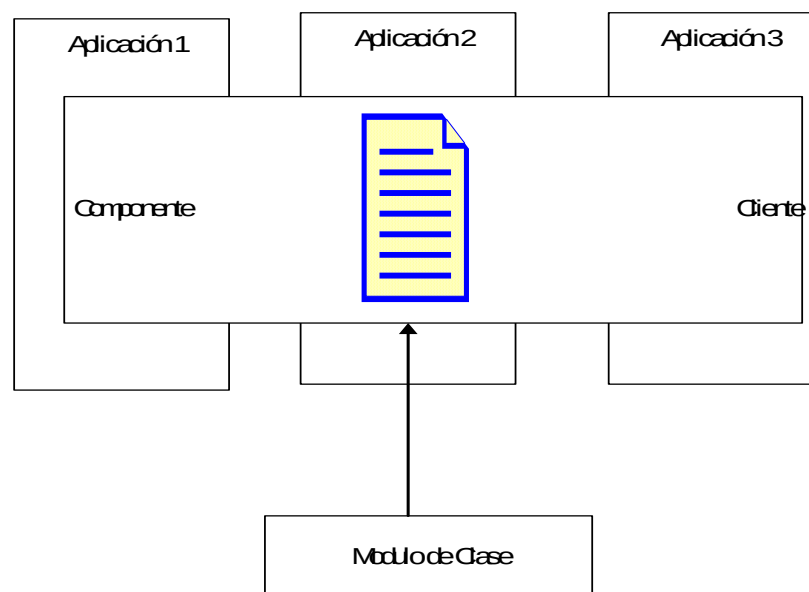


Figura 1. 7 Uso de clases mediante componentes

Ahora obtenemos la misma funcionalidad de cuando copiábamos el código fuente en cada aplicación, pero ahora para hacer mantenimiento de ese código fuente, solo debemos modificar en un solo lugar, en el componente.

Tanto las clases como los componentes COM+ proveen alguna forma de servicios. Las clases proveen servicios a una sola aplicación pues no pueden comunicarse con otras aplicaciones, mientras que los componentes COM+ proveen servicios a numerosas aplicaciones a nivel de sistemas porque ellas pueden comunicarse con otras aplicaciones.

Cientes y Servidores

En general, un cliente requiere algún tipo de servicios de un servidor. Así el trabajo del servidor es hacer algún tipo de tarea a solicitud del cliente. A gran escala, esto puede ser tan simple como una aplicación común de bases de datos donde el cliente representa la interfaz de usuario que hace los requerimientos a la base de datos (el servidor) para varias tareas tales como almacenamiento de información, manipulación de datos y obtención de datos.

Asumiendo que la aplicación de bases de datos que hemos mencionado esta construido con COM+, podemos dividir las distintas funcionalidades en partes más pequeñas (componentes

COM+). Los componentes COM+ también siguen el principio cliente-servidor. Sin embargo, determinar si un componente COM+ es un cliente o un servidor puede llegar a ser confuso porque puede cumplir ambos roles. Cuando una aplicación cliente hace un requerimiento a un componente COM+, el componente COM+ actúa como un servidor (para el cliente). Cuando un componente COM+ hace un requerimiento a otro componente COM+, el componente COM+ que realiza la llamada es considerado el cliente mientras que el componente que al cual es realizada la llamada es considerado el servidor. En palabras simples, la razón por la cual un componente COM+ puede desempeñar ambos roles es porque cada componente COM+ expone lo que se conoce como interfaz. Estas interfaces nos llevan a distintas funciones que están disponibles para ser llamadas por los clientes.

Tipos de Componentes de Servidor

Veamos como los componentes se comportan en diferentes ambientes.

El ambiente COM+ nos provee con varias opciones cuando creamos un componente:

- Servidor fuera de proceso (out of process server)
- Servidor dentro del proceso (in process server)

Servidor Fuera de proceso

Un servidor fuera de proceso es un componente que es efectivamente un programa separado que contiene objetos. Este tipo de servidor siempre se ejecutará en su propio espacio de proceso, y será independiente de otros programas.

Los servidores fuera de proceso son aplicaciones independientes que además proveen acceso a sus objetos internos a través de COM+, incluso podrían ser diseñados para sólo exponer sus objetos a otras aplicaciones.

Estos servidores proveen acceso a sus objetos para facilitar a otros programas por ejemplo la creación de macros o utilizar alguna otra forma de extensión de funcionalidades. Por ejemplo,

servidores fuera de proceso son Microsoft Word, Excel, SAP, etc. que permiten a otras aplicaciones acceder a sus objetos a través de COM+.

Otra razón para crear servidores fuera de proceso es porque un proceso fuera de servidor se ejecuta dentro de su propio proceso (no dentro del proceso que realiza la llamada). Esto se ilustra en el figura 1.8 que muestra a varios clientes interactuando con objetos que se ejecutan en el mismo servidor fuera de proceso.

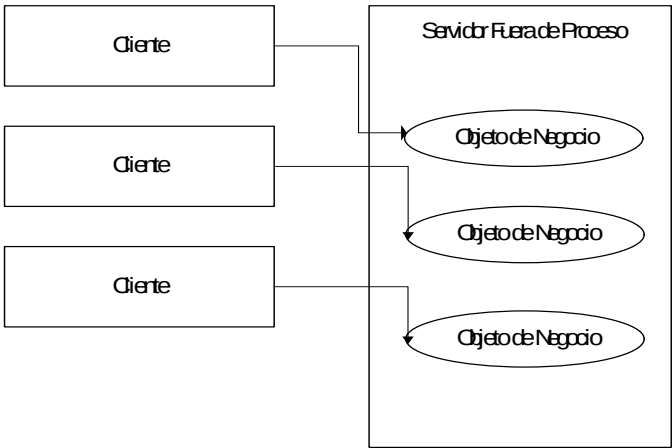


Figura 1. 8 Esquema de servidor fuera de proceso

Existe un tema de performance implícito en este tipo de componentes. Como la aplicación cliente se comunica con los objetos en un proceso se genera un poco de sobrecarga. COM+ pasa entre el cliente y el servidor en este caso, y maneja todas las comunicaciones entre ellos. Aunque eso es positivo pues nos ahorra mucho tiempo de programación y esfuerzos de desarrollo, tiende a enlentecer las cosas porque se genera trabajo extra en cuanto al manejo de las comunicaciones entre procesos.

En general los servidores fuera de proceso son útiles cuando queremos realizar algo realmente fuera de lo ordinario, o cuando tratamos de hacer uso de objetos de una aplicación preexistente. De otra forma lo que normalmente se usa son los más comunes servidores dentro de proceso.

Servidores dentro de proceso

Un servidor dentro de proceso es un componente donde nuestros objetos se ejecutan dentro del proceso cliente. Cada cliente esencialmente obtiene su propia copia del componente y de esta forma una copia de todos nuestros objetos (ver figura 1.9).

Normalmente, este es el tipo más común de servidores COM+ y de componentes COM+. Los servidores dentro de proceso no tienen un proceso asignado por sí mismos, pues siempre se ejecutan dentro del contexto de otro proceso.

En el caso donde un componente actúa directamente como servidor de una aplicación cliente, tenemos una mejora de performance. Existe sólo una pequeña sobrecarga al tener código cliente interactuando directamente con un objeto que se está ejecutando dentro del mismo proceso.

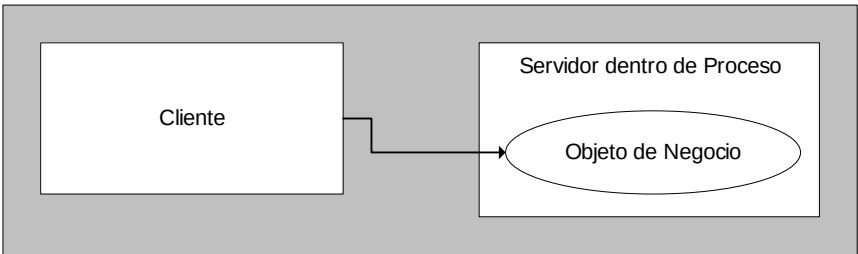


Figura 1. 9 Esquema de servidor dentro de proceso

Cuando el componente dentro de proceso actúa como servidor de varios otros procesos tales como aplicaciones COM+, las cosas son un poco diferentes. En este caso, el cliente debe comunicarse entre los procesos para interactuar con el objeto. De hecho, esto es bastante similar a los servidores fuera de proceso (ver figura 1.10).

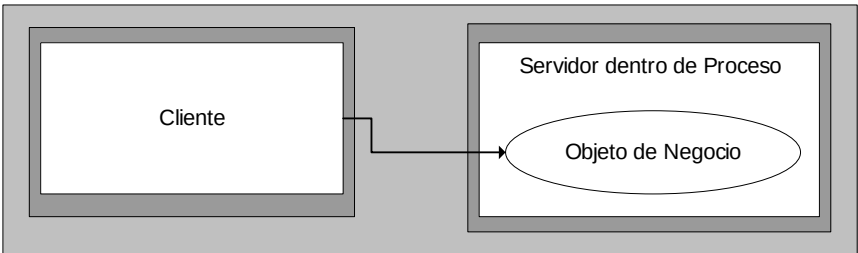


Figura 1. 10 Esquema de Servidor dentro de proceso para varios procesos

Las principales ventajas de ejecutar tales componentes en una aplicación COM+ son estabilidad y mayor manejabilidad.

Ganamos algo de estabilidad (al menos en teoría) si un objeto al hacer que su proceso deje de funcionar no haga que deje de funcionar la aplicación cliente, un ejemplo es el comportamiento del mismo ASP (visto como un componente COM+) dentro de un Windows 2000. Sin embargo, podría dejar de funcionar el proceso que hace de servidor, gatillando una reacción en cadena con los otros procesos que están actuando en ese momento como clientes del componente.

También ganamos algo de manejabilidad cuando por ejemplo ASP actúa como un servidor dentro de proceso directamente dentro de su proceso. De fallar un componente que usa ASP como servidor, esa componente queda bloqueada mientras ASP es reiniciado. Obviamente esto no es práctico en entornos del tipo de 7x24.

Si ejecutamos el servidor dentro de proceso en otro proceso, tal como una aplicación COM+, podemos desbloquear el componente reiniciando solamente el componente cliente, dejando que ASP se siga ejecutando dentro de IIS. Aunque esto aun significa que debemos detener nuestra aplicación para por ejemplo actualizar un componente, evitamos detener el servidor Web completo.

Interfaces el estándar binario de COM+

Las ventajas que COM+ provee para el desarrollo de aplicaciones distribuidas es provisto a través del uso de interfaces.

Mencionamos previamente que los componentes COM+ pueden ser escritos en cualquier lenguaje. Si todos los componentes COM+ fuesen escritos exclusivamente en el mismo lenguaje de programación, entonces la comunicación entre los objetos no sería difícil. Sin embargo, cuando usamos componentes escritos en diferentes lenguajes de programación, necesitamos una manera para que estos diferentes componentes puedan comunicarse. Las interfaces de COM+ proveen un mecanismo universal para que cada componente de COM+ pueda

comunicarse en un lenguaje que sea entendido por todos los otros componentes escritos según el estándar COM+, independientemente del lenguaje en que fueron escritos.

Las interfaces de COM+ no son otra cosa que un grupo o colección de funciones públicas que permiten a los componentes COM+ interactuar unos con otros. En otras palabras, una interfaz es simplemente una lista de métodos que un componente COM+ debe soportar. Más aun, es importante entender que una interfaz no es ni una clase ni un objeto. Mientras que una clase puede ser instanciada para crear un objeto COM+, una interfaz COM+ no puede ser instanciada.

Cada interfaz constituye una especie de contrato de enlace entre un objeto COM+ y un objeto que ofrece una interfaz. Estos contratos (las interfaces) como cualquier contrato nunca pueden ser cambiados. En otras palabras, se dice que las interfaces de COM+ son inmutables ya que no pueden contar con versiones. Esto significa que una vez que se ha puesto en producción un componente COM+ y consiguientemente su interfaz, esta no debe modificarse nunca.

Sin embargo, no ha de malentenderse esa regla, pues no significa que no se pueda realizar ningún cambio a un objeto COM+. Por el contrario, podemos cambiar la lógica del negocio siempre y cuando no se interfiera con las propiedades y métodos originales.

Además, COM+ nos facilita la vida permitiendo más de una interfaz a la vez. Las nuevas interfaces se pueden añadir para soportar nuevas funcionalidades mientras se conserven las interfaces originales intactas. Esto significa que las aplicaciones más antiguas seguirán funcionando usando las interfaces originales del componente mientras que las aplicaciones más actualizadas pueden utilizar las nuevas interfaces.

GUID's y el registro de Windows

Para los componentes COM+, se utiliza un número identificador llamado GUID (Global Unique Identifier) el cual es un valor único de 128 bits que identifica cada componente COM+ y cada interfaz COM+ sin duplicaciones en cualquier parte del mundo. Esta garantía es obligatoria para el desarrollador de aplicaciones como para el usuario de la aplicación pues de esta forma se

asegura que los componentes COM+ nunca se conectarán a otros componentes de forma equivocada debido a GUID equivocados.

Los GUID se generan tomando el valor de la MAC Address de la tarjeta de red del PC donde se está ejecutando Windows (el valor de la MAC Address es único para cada tarjeta de red) y el valor de la hora y la fecha del sistema del desarrollador. Si el PC en cuestión no tuviese tarjeta de red, se toma un valor aleatorio con la fecha y hora del PC. De esta forma, las posibilidades para que dos GUID sean iguales son bajísimas.

En la jerga de COM+, a veces se habla de CLSID (Class Identifier) en lugar de GUID. Esto ocurre cuando nos referimos a GUIDs que se refieren a clases COM+. De la misma forma, cuando nos referimos a interfaces COM+, se habla de IID (Interfase Identifier).

Anteriormente se mencionó que COM+ era independiente de la ubicación de los componentes, lo cual significa que un componente COM+ puede residir en cualquier lugar en el computador o incluso en cualquier otro computador conectado mediante una red al computador donde está el componente que lo solicita. Antes que un objeto pueda ser creado, el entorno de COM+ debe ubicar el componente COM+. El entorno de COM+ puede ubicar el componente COM+ usando el registro de Windows. Para identificar un componente COM+, es que se utilizan los CLSID los cuales recordemos que son una particularización de los GUIDs.

Finalmente en este trabajo se encuentra la bibliografía, referencias electrónicas, apéndices (abreviaturas y glosario) y anexos empleados en este proyecto de tesis. En los anexos se incluye un extracto del código fuente y el diccionario de datos de la base de datos relacional creada para este proyecto.

CAPITULO 2: DEFINICIÓN DE OBJETIVOS

OBJETIVO GENERAL

Contar con un Sistema y el soporte tecnológico adecuado que permita administrar, hacer seguimiento y controlar el flujo de procesos asociados al curse y visación centralizado de operaciones con documentos digitalizados (imágenes).

OBJETIVOS ESPECÍFICOS.⁴

1. Analizar, Diseñar y Modelar Sistema Informático para realizar Curse de Operaciones de Crédito (Responsable: Claudia Montecino Q).
2. Implementar la metodología de desarrollo basada en prototipos integrando COM+, utilizando las normas existentes en BancoEstado (Responsable: Enrique Moreira H.).
3. Contrastar las funcionalidades del sistema informático desarrollado en un ambiente distribuido (Responsable: Enrique Moreira H.).
4. Implantar la solución informática en Sucursales Piloto, para luego estudiar su comportamiento (Responsable: Claudia Montecino Q.).
5. Disminuir los tiempos de respuesta del proceso de curse de operaciones (Responsable: Claudia Montecino Q.).
6. Establecer nuevos parámetros de medición con el sistema implementado en el total de sucursales definidas en primera versión de sistema informático (Responsable: Claudia Montecino Q.).

⁴ En este proyecto han participado dos personas debido a la necesidad de contar con un equipo multidisciplinario que incluyera la participación de un integrante con conocimientos del negocio y normativa interna (Claudia Montecino Q.) y otro integrante con conocimientos de la metodología y arquitectura (Enrique Moreira H.). Sin embargo ambos integrantes estuvieron en cada etapa del proyecto.

CAPÍTULO 3: ANÁLISIS Y DISEÑO DE SISTEMA INFORMÁTICO

En este capítulo revisaremos el estado actual del proceso de curse de operaciones de crédito en sucursales, dado este análisis explicaremos sus falencias en un escenario donde el volumen de los datos es mayor como sucede en el periodo de campañas.

Se explicará la metodología a utilizar y el porqué de su uso.

A continuación en base a los requerimientos se realizará el levantamiento de casos de uso en sucursales y CPN, modelándolo mediante UML.

Para finalizar este capítulo se entregarán las especificaciones del diseño de modelo de curse tanto para sucursales como para el CPN.

ESTADO ACTUAL

En la actualidad se cuenta con el sistema de gestión de correspondencia que cubre en parte la necesidad de digitalizar imágenes, enviar operaciones y recibo en CPN, este sistema tiene por finalidad controlar y distribuir la correspondencia entre las diferentes áreas del Banco, definiéndose tres roles:

- ❖ Supervisor
- ❖ Secretaria
- ❖ Destinatario

Dado que este sistema fue creado con otros fines no incluye parámetros asociados al proceso de course de operaciones de crédito, omitiéndose en la actualidad datos fundamentales para la recepción y visación en el CPN.

El Banco cuenta con un servicio destinado a la digitalización de imágenes; el cual consiste en una interfaz gráfica que permite capturar imágenes desde un escáner e ingresarlas a una Base de Datos Relacional, permitiéndose además la consulta de dichas imágenes.

Para otros sistemas asociados a este servicio como las aplicaciones que utiliza el Centro de Contacto Lota (CCL), se permite además el adjuntar archivos de tipo Word, Excel, texto, mensajes de correo electrónico (msg), que también son almacenados en la base de datos.

Lo indicado anteriormente resuelve en sucursales el problema de hacer llegar al CPN los documentos digitalizados, pero deben recurrir a envíos de correo electrónico para especificar los datos que el sistema actual no soporta de forma nativa. Esto no mejora los tiempos de respuesta al encontrarse las sucursales en periodo de campañas, produciéndose cuellos de botella al momento de ingresar imágenes y preparar el correo electrónico con la información faltante.

El uso del correo electrónico del Banco no es sustentable para grandes volúmenes de datos, puesto que se produce desorden, traslape y pérdida de datos, al no contar con una marca de seguimiento correcta que permita gestionar al CPN sobre los correos que vienen desde sucursales para course y visación de operaciones de crédito, separándolas del resto de sus correspondencia.

DESCRIPCIÓN DE LA METODOLOGÍA

Se utilizará la metodología de desarrollo por prototipos, con este tipo de diseño se armará una maqueta del Software. Con esto se persigue corregir las fallas que pueda tener el software previo

a su construcción. Este prototipo mostrará las pantallas y listados, los cuales serán vistas y discutidas con y por el usuario hasta llegar a un acuerdo en forma y cantidad, para luego proceder a la construcción del software.

Una definición de un prototipo de software sería: un modelo del comportamiento del sistema que puede ser usado para entenderlo completamente o ciertos aspectos de él y así clarificar los requerimientos. Un prototipo es una representación de un sistema, aunque no es un sistema completo, posee las características del sistema final o parte de ellas.

Las fases que comprende el método de desarrollo orientado a prototipos serían:

Investigación preliminar. Las metas principales de esta fase son: determinar el problema y su ámbito, la importancia y sus efectos potenciales sobre la organización por una parte y, por otro lado, identificar una idea general de la solución para realizar un estudio de factibilidad que determine la factibilidad de una solución software.

Definición de los requerimientos del sistema. El objetivo de esta etapa es registrar todos los requerimientos y deseos que los usuarios tienen en relación al proyecto bajo desarrollo. Esta etapa es la más importante de todo el ciclo de vida, es aquí donde el desarrollador determina los requisitos mediante la construcción, demostración y retroalimentaciones del prototipo. Por lo mismo esta etapa será revisada con más detalle luego de esta descripción.

Diseño técnico. Durante la construcción del prototipo, el desarrollador ha obviado el diseño detallado. El sistema debe ser entonces rediseñado y documentado según los estándares de la organización y para ayudar a las mantenciones futuras. Esta fase de diseño técnico tiene dos etapas: por un lado, la producción de una documentación de diseño que especifica y describe la estructura del software, el control de flujo, las interfaces de usuario y las funciones y, como segunda etapa, la producción de todo lo requerido para promover cualquier mantención futura del software.

Programación y prueba. Es donde los cambios identificados en el diseño técnico son implementados y probados para asegurar la corrección y completitud de los mismos con respecto a los requerimientos.

Operación y mantención. La instalación del sistema en ambiente de producción, en este caso, resulta de menor complejidad, ya que se supone que los usuarios han trabajado con el sistema al hacer las pruebas de los prototipos. Además, la mantención también debería ser una fase menos importante, ya que se supone que el refinamiento del prototipo permitiría una mejor claridad en los requerimientos, por lo cual las mantenciones perfectivas se reducirían. Si eventualmente se requiriese una mantención entonces el proceso de prototipado es repetido y se definirá un nuevo conjunto de requerimientos.

La fase más importante corresponde a la definición de requerimientos, la cual corresponde a un proceso que busca aproximar las visiones del usuario y del desarrollador mediante sucesivas iteraciones. La definición de requerimientos consiste de cinco etapas entre dos de las cuales se establece un ciclo iterativo:

Análisis grueso y especificación. El propósito de esta subfase es desarrollar un diseño básico para el prototipo inicial.

Diseño y construcción. El objetivo de esta subfase es obtener un prototipo inicial. El desarrollador debe concentrarse en construir un sistema con la máxima funcionalidad, poniendo énfasis en la interfaz del usuario.

Evaluación. Esta etapa tiene dos propósitos: extraer a los usuarios la especificación de los requerimientos adicionales del sistema y verificar que el prototipo desarrollado lo haya sido en concordancia con la definición de requerimientos del sistema. Si los usuarios identifican fallas en el prototipo, entonces el desarrollador simplemente corrige el prototipo antes de la siguiente evaluación. El prototipo es repetidamente modificado y evaluado hasta que todos los requerimientos del sistema han sido satisfechos. El proceso de evaluación puede ser dividido en

cuatro pasos separados: preparación, demostración, uso del prototipo y discusión de comentarios. En esta fase se decide si el prototipo es aceptado o modificado.

Modificación. Esto ocurre cuando la definición de requerimientos del sistema es alterada en la subfase de evaluación. El desarrollador entonces debe modificar el prototipo de acuerdo a los comentarios hechos por los usuarios.

Término. Una vez que se ha desarrollado un prototipo estable y completo, es necesario ponerse de acuerdo en relación a aspectos de calidad y de representación del sistema.

En la figura 3.1 se puede ver un esquema que ilustra las etapas se realizan en el desarrollo por prototipos, es en ella donde se utiliza el prototipado, ya que permite entregar al usuario lo que sería una visión de la solución final en etapas tempranas del desarrollo, reduciendo tempranamente los costos de especificaciones erróneas.

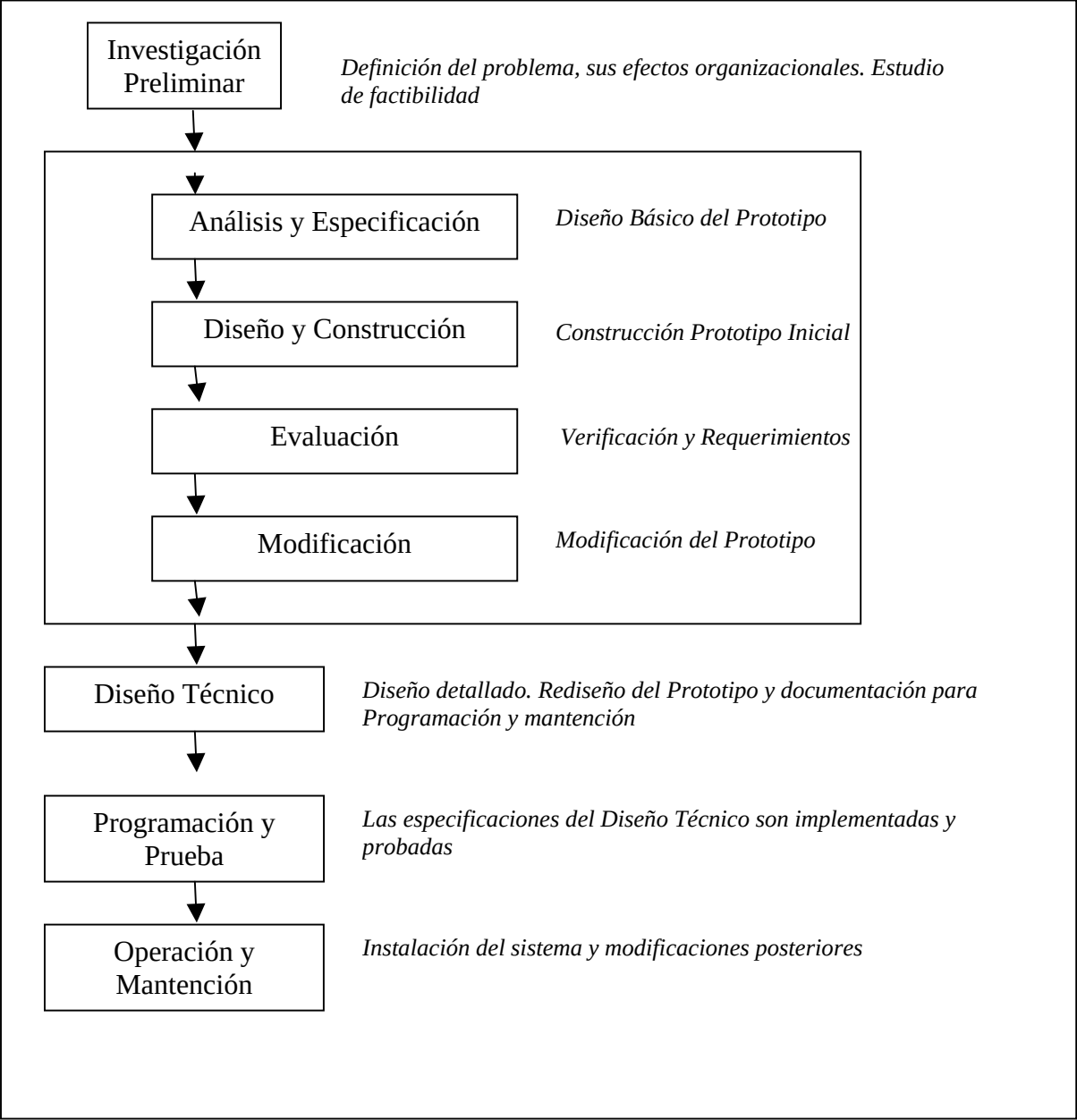


Figura 3. 1 Desarrollo Orientado a Prototipos

Las ventajas de un enfoque de desarrollo orientado a prototipos están dadas por: reducción de la incertidumbre y del riesgo, reducción de tiempo y de costos, incrementos en la aceptación del nuevo sistema, mejoras en la administración de proyectos, mejoras en la comunicación entre desarrolladores y clientes.

Si bien, el desarrollo orientado a prototipos tiene considerables ventajas, también presenta desventajas como: la dependencia de las herramientas de software para el éxito ya que la necesidad de disminución de incertidumbre depende de las iteraciones del prototipo, entre más

iteraciones exista mejor y esto último se logra mediante el uso de mejores herramientas lo que hace a este proceso dependiente de las mismas. También, no es posible aplicar la metodología a todos los proyectos de software y, finalmente, la mala interpretación que pueden hacer los usuarios del prototipo, al cual pueden confundir con el sistema terminado.

No se puede desconocer que la fase de definición de requerimientos se ha perfeccionado en dos aspectos importantes: primero se han aproximado las visiones del usuario y el desarrollador, lo cual representa el beneficio de establecer una base común de comunicación; también, el hacer explícita la posibilidad de iterar sobre estos dominios permitiría que la convergencia de los mismos sea una posibilidad cierta.

Pero lo anterior no asegura el éxito, por ejemplo, ¿Qué certeza existe en que esta iteración sea en la dirección correcta y lleve a la convergencia de los dominios?; no se puede desconocer las diferencias que existen entre la prueba de un prototipo de software en la fase de definición de requerimientos y el uso del mismo ya como un producto terminado. Para explicar esto, podemos hablar de dos dominios en el usuario, uno que es el que se establece cuando se prueba el prototipo y otro, distinto por cierto, el que ocurre cuando el usuario hace uso del software en ambiente de producción.

Por último, el proceso de iteración para que sea efectivo debería ser infinito, lo que lo hace poco efectivo. Es decir, mediante este método acercamos la problemática de los usuarios a los dominios de los desarrolladores y vice versa, pero no sería posible lograr un pareamiento uno a uno entre estos dominios, lo que sería el ideal.

Una vez aprobado el prototipo por el usuario, cualquier modificación posterior constituye un cambio sobre el acuerdo inicial (Control de Cambios).

Los motivos por los cuales se optó por una metodología de desarrollo basada en prototipos incrementales, se debe principalmente a que al ser el GDI un producto que no tiene precedentes

al interior del Banco, el usuario final no tiene los requerimientos iniciales claros hasta interactuar con el software. Es por esto que se hace necesario entregar al usuario interfaces y procedimientos basados en prototipos para familiarizar al usuario y así tener un mayor grado de retroalimentación con él. Si bien es cierto los objetivos del software fueron inicialmente definidos y acotados, aún no se conocía algunos elementos ni se tenía una idea global muy clara, por lo que la activa participación del usuario en etapas iniciales y finales, como captura y análisis de requerimientos; diseño y en las pruebas fue fundamental.

Junto con el equipo de desarrollo, trabajará en esta etapa del proyecto el usuario definido por la Subgerencia de Procesos, dependiente de la Gerencia División Operaciones y Sistemas (GDOS). No nos corresponde entregar nombres de terceros en este proyecto, sólo las áreas involucradas.

Aun tratándose de un prototipo debe ser diseñado y desarrollado según las normativas existentes en el Banco, definidas por el área de Soporte al Desarrollo y Modelamiento (SDM).

La normativa exigida en cuanto a prototipos hace referencia a:

- ❖ Utilización de hojas de estilos (css, xsl), de acuerdo a las definiciones de la interfaz SSF.
- ❖ Control de versiones (código fuente), usando la herramienta CCCHarvest.
- ❖ Manejo de una estructura de directorios orientada a los perfiles que tendrá la aplicación.
- ❖ Creación de Perfiles para las pruebas de usuario, los que serán utilizados posteriormente en el sistema final. Esto debe hacerse en conjunto con el área de Seguridad de la información, utilizando el alias dado al sistema (GDI) y los roles y cargos que tendrán asociado dicho perfil.

El prototipo desarrollado se validará realizando una instalación en formato de piloto en un grupo reducido de sucursales, las cuales corresponden a Arica, Iquique y Punta Arenas. Los usuarios

responsables de la validación del prototipo en sucursales serán definidos por la subgerencia de procesos, así como también los usuarios encargados de dichas pruebas en el CPN.

Al contar con la aprobación de los usuarios encargados de las pruebas piloto sobre el prototipo ya sea en sucursales y en CPN se procederá a la construcción de la versión final del software, considerando reuniones periódicas de avance, posibles cambios y mejoras a lo validado en la etapa anterior.

Para la elaboración del prototipo se utilizará modelamiento UML, donde en cada reunión de avance con el usuario líder de la subgerencia de Ingeniería de Procesos se presentarán los casos de uso para cada uno de los posibles escenarios que tenga el GDI tanto en sucursales como en CPN.

El Lenguaje de Modelamiento Unificado (UML - Unified Modeling Language) es un lenguaje gráfico para visualizar, especificar y documentar cada una de las partes que comprende el desarrollo de software. UML entrega una forma de modelar cosas conceptuales como lo son procesos de negocio y funciones de sistema, además de cosas concretas como lo son escribir clases en un lenguaje determinado, esquemas de base de datos y componentes de software reusables.

Uno de los objetivos principales de la creación de UML fue posibilitar el intercambio de modelos entre las distintas herramientas CASE orientadas a objetos del mercado. Para ello era necesario definir una notación y semántica común.

En la especificación del UML podemos comprobar que una de las partes que lo componen es un metamodelo formal. Un metamodelo es un modelo que define el lenguaje para expresar otros modelos. Un modelo en OOP es una abstracción cerrada semánticamente de un sistema y un sistema es una colección de unidades conectadas que son organizadas para realizar un propósito específico. Un sistema puede ser descrito por uno o más modelos, posiblemente desde distintos puntos de vista.

El UML es una técnica de modelado de objetos y como tal supone una abstracción de un sistema para llegar a construirlo en términos concretos. El modelado no es más que la construcción de un modelo a partir de una especificación.

Un modelo es una abstracción de algo, que se elabora para comprender ese algo antes de construirlo. El modelo omite detalles que no resultan esenciales para la comprensión del original y por lo tanto facilita dicha comprensión.

Una de las herramientas que provee UML es el Diagrama de Casos de Uso, el cual muestra la relación entre los actores y los casos de uso del sistema. Representa la funcionalidad que ofrece el sistema en lo que se refiere a su interacción externa. En el diagrama de casos de uso se representa también el sistema como una caja rectangular con el nombre en su interior. Los casos de uso están en el interior de la caja del sistema, y los actores fuera, y cada actor está unido a los casos de uso en los que participa mediante una línea.

LEVANTAMIENTO CASOS DE USO EN SUCURSALES

A continuación mediante los diagramas de casos de uso se mostrará el proceso que ocurre en sucursales desde la creación de una operación.

Caso Uso Solicitud Apertura Operación de Crédito

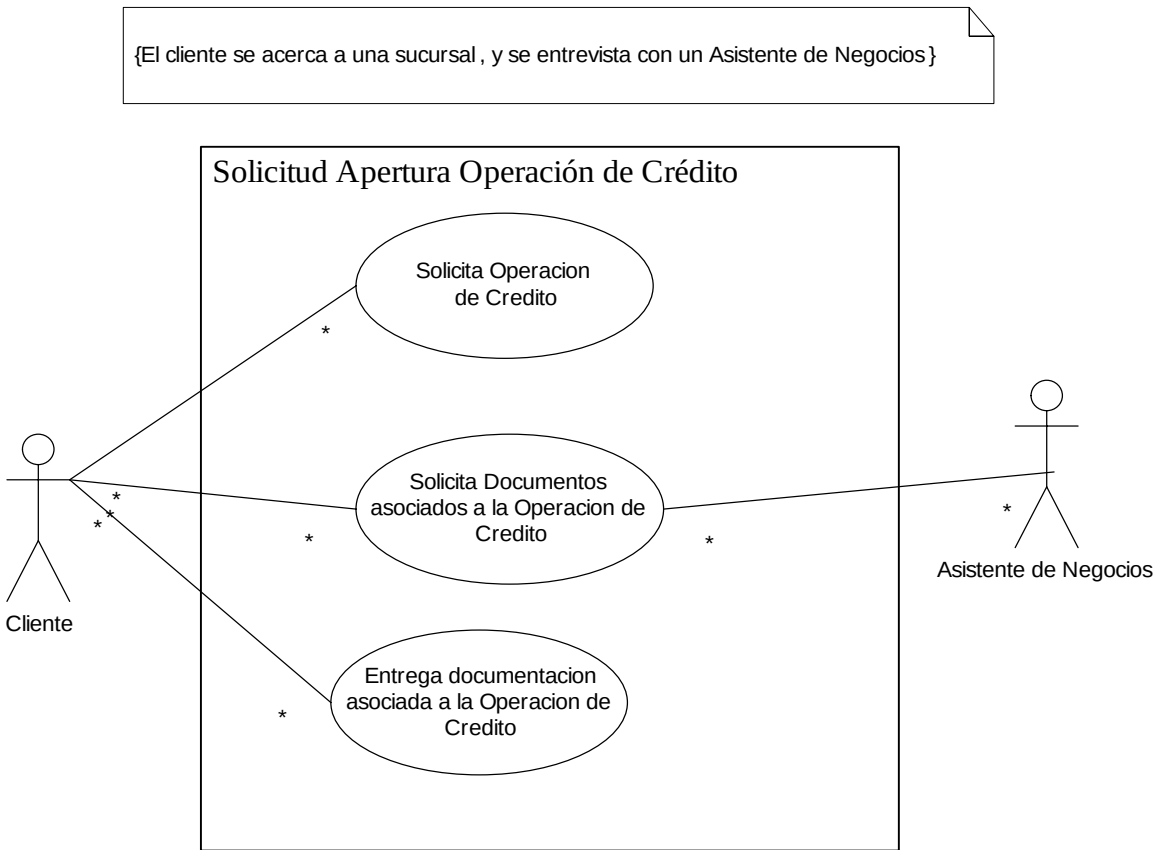


Figura 3. 2 Solicitud Apertura Crédito

La figura 3.2 muestra el flujo que sigue la apertura de una operación de crédito, ya sea para Línea de Crédito, Tarjetas de Crédito o Créditos de Consumo.

Son dos personajes los que interactúan en esta etapa, el cliente y el Asistente de Negocios que atenderá su solicitud.

Caso Uso Creación Operación en GDI

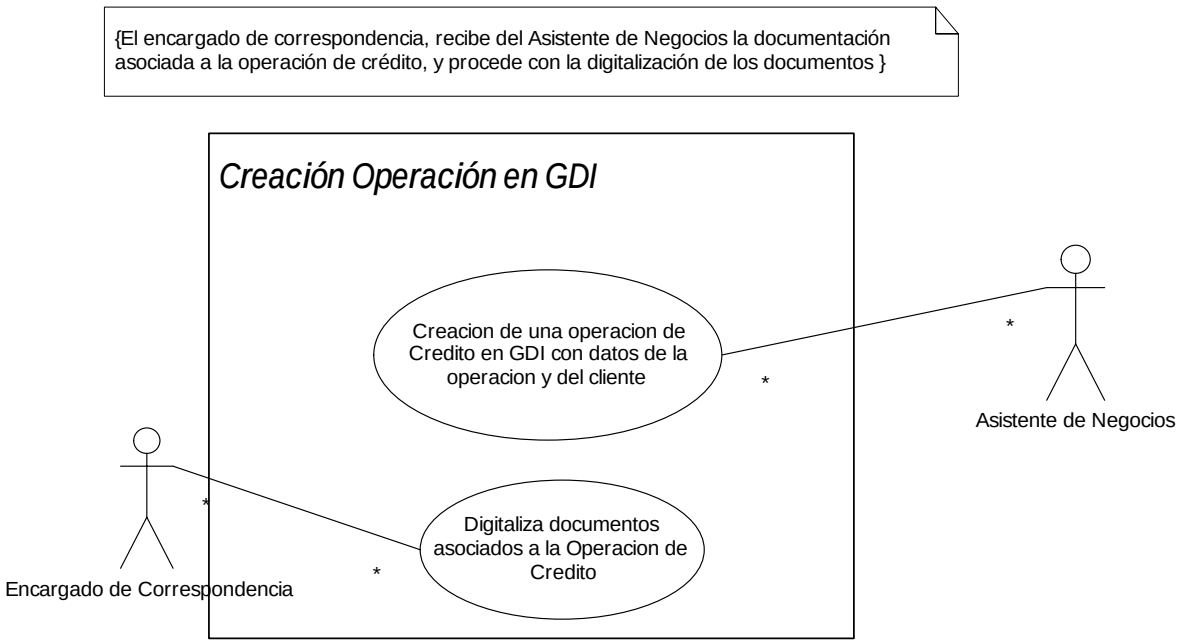


Figura 3. 3 Creación Operación de curse

En la figura 3.3 se representa el flujo que sigue una operación al ser ingresada al GDI.

Una vez que el Asistente de Negocios tiene en su poder la documentación requerida por el Banco para crear una operación de crédito, éste se la hace llegar al Encargado de Correspondencia, encargado de digitalizar y completar vía electrónica la información entregada por el cliente.

Caso Uso Envío Operación al CPN

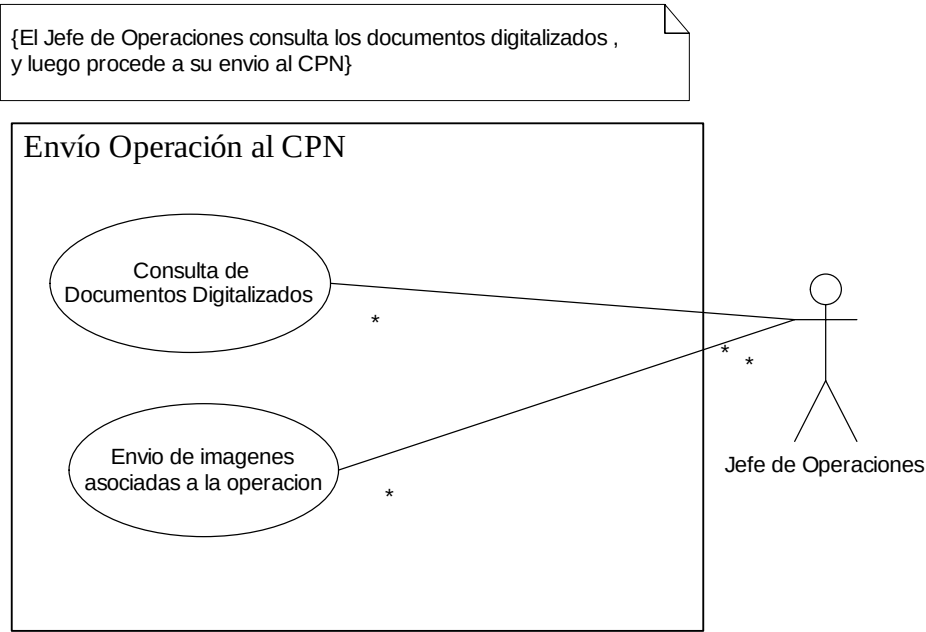


Figura 3. 4 Envío Operación a CPN

En la figura 3.4 podemos ver una vez que el encargado de correspondencia ha ingresado todos los datos de la operación es tomado por el jefe de Operaciones el cual valida los datos y los envía a PCN para el Control, en caso de haber errores le comunica al Encargado de correspondencia o lo rechaza, evento que gatillará un mail al Asistente de negocios que abrió la operación de curse.

Caso Uso Recepción de Operación desde el CPN

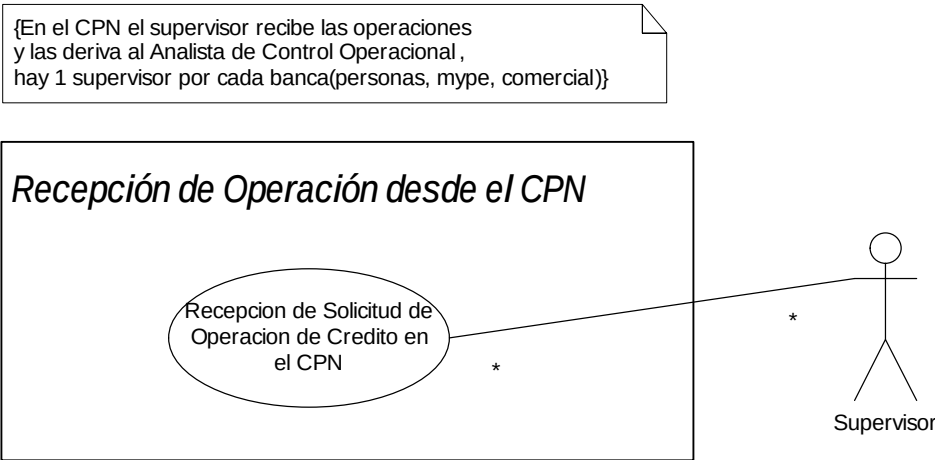


Figura 3. 5 Recepción Operaciones en CPN

En CPN las operaciones son recepcionadas por alguno de los Supervisores a Cargo (figura 3.5), quien dependiendo de la Banca desde donde provienen las operaciones las re-dirigen a algún Administrativo para su Control y Aprobación.

LEVANTAMIENTO CASOS DE USO EN CPN

Caso Uso Asignación Operación en el CPN

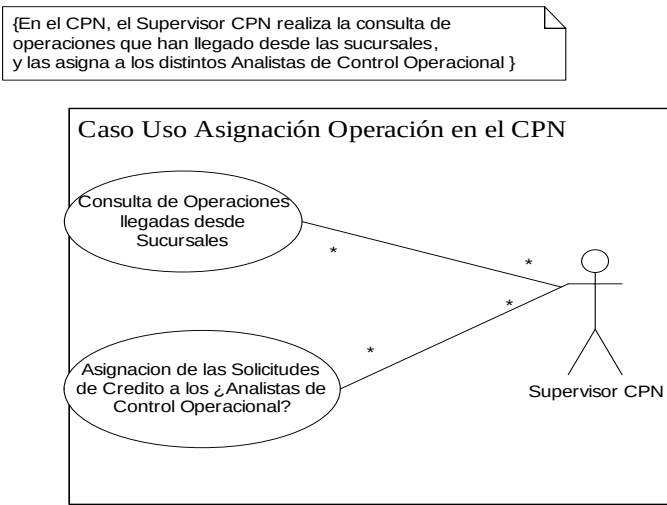


Figura 3. 6 Asignación de Operaciones

El Supervisor que toma la operación verifica la carga de trabajo de l administrativos y les asigna según sea la Banca la operación de curse (figura 3.6).

Caso Uso Control Operaciones en el CPN

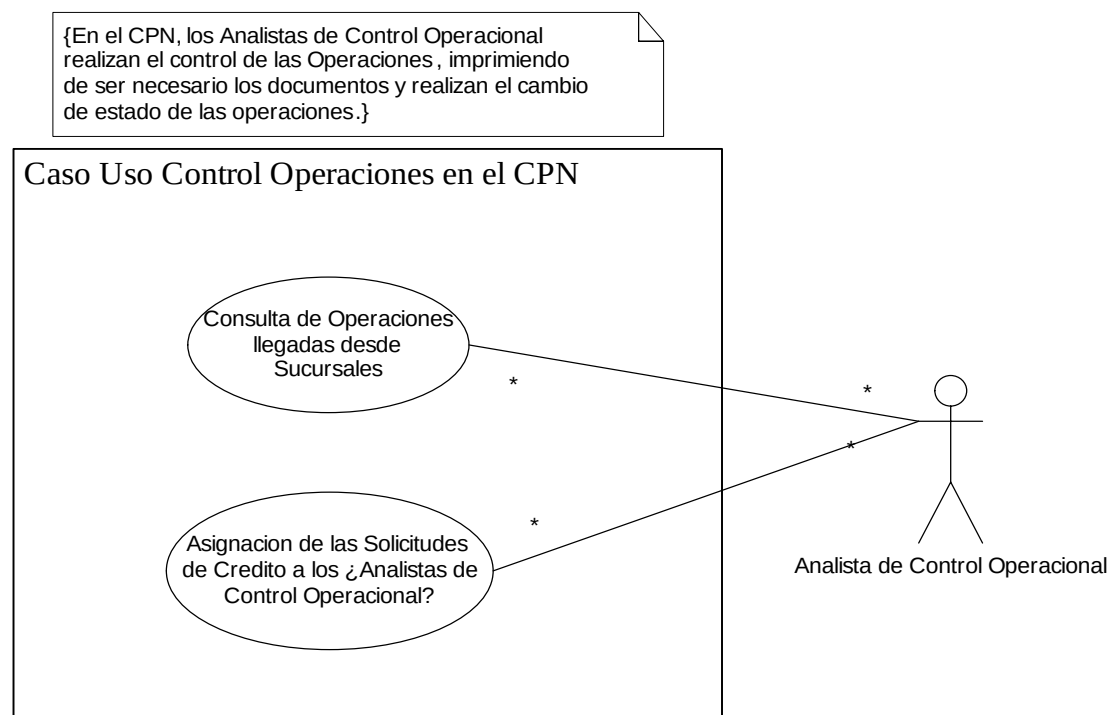


Figura 3. 7 Control de Operaciones

Los analistas de control de operaciones son los encargados de realizar el curse de la operación de crédito (figura 3.7), para ello validan la información ingresada con que enviada por carpeta física.

Caso Uso Devolución Operación a Sucursales

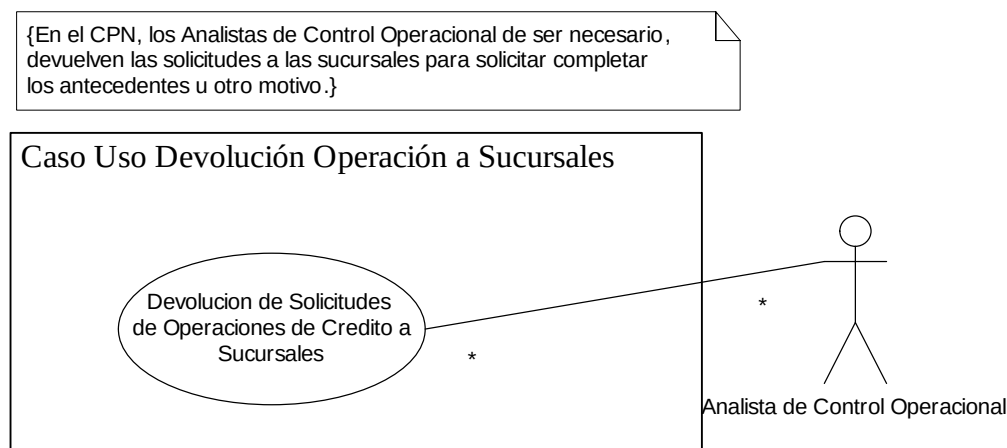


Figura 3. 8 Devolución Operaciones a Sucursal

En caso de existir algún error en la operación, esta es devuelta a la sucursal de origen (figura 3.8), pudiendo modificar los datos con problemas y enviando nuevamente dicha operación sin perder los datos asociados a este.

MODELO DE CURSE PARA SUCURSALES Y CPN

A continuación se ilustrarán el modelo de flujo de curse de operaciones para sucursales (figura 3.9) y para CPN (figura.10).

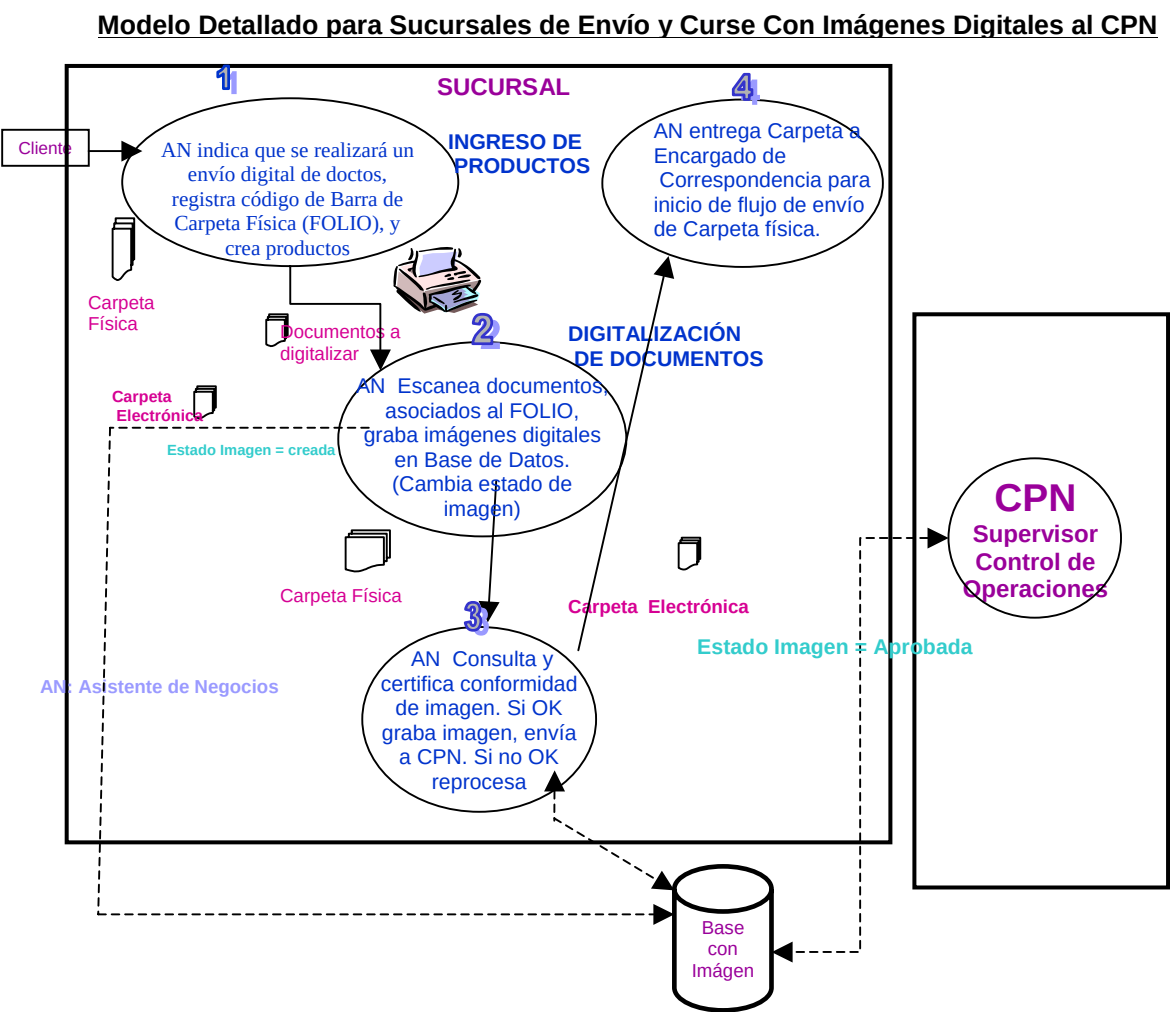


Figura 3. 9 Flujo Curse de Operaciones en Sucursales

Modelo Detallado Propuesto de Envío y Curse Con Imágenes Digitales

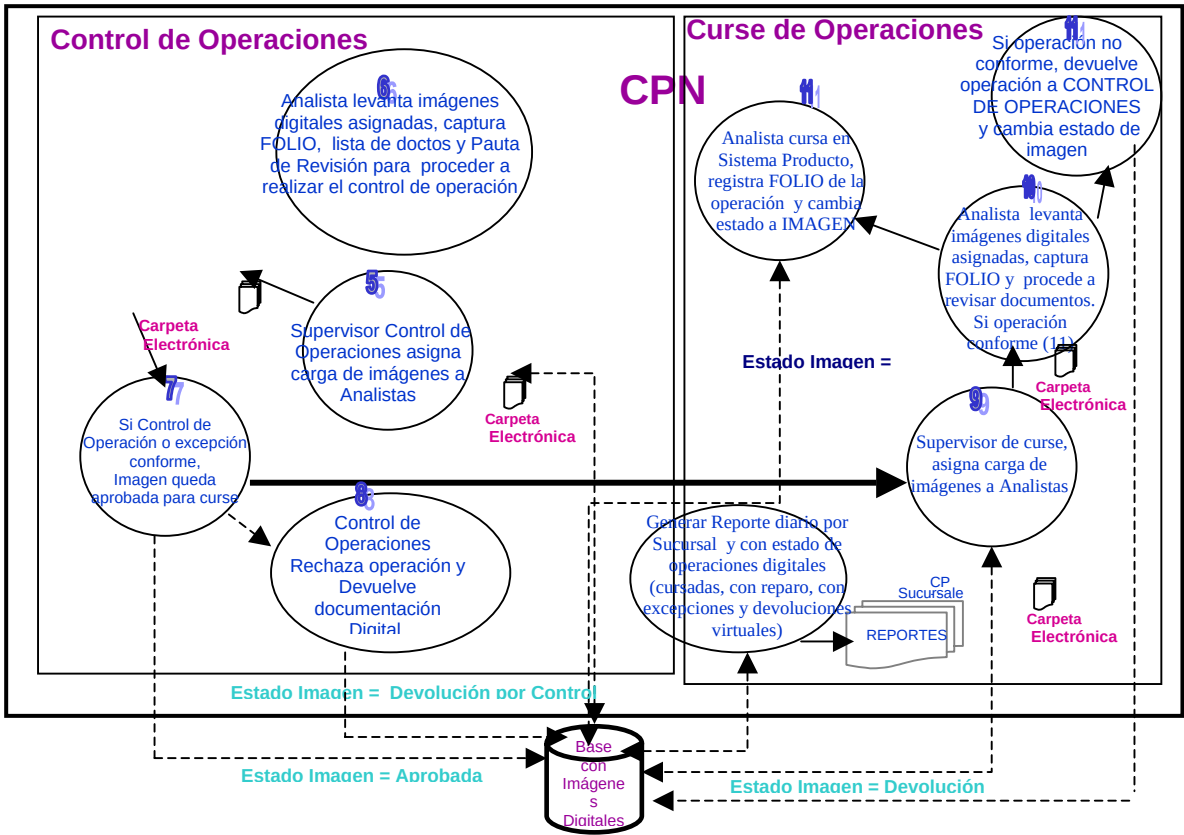


Figura 3. 10 Flujo Curse Operaciones en CPN

CAPITULO 4: MODELAMIENTO DEL SISTEMA INFORMATICO

Para modelar y diseñar la solución informática que soporte los requerimientos solicitados por el cliente (Subgerencia de Procesos), serán seguidas las siguientes reglas impuestas por el área de Soporte al Desarrollo y Modelamiento, en cuanto a desarrollo de aplicaciones Web DNA:

CONSIDERACIONES INICIALES

De la Arquitectura

El desarrollo debe hacerse siguiendo el estándar DNA, donde:

- La capa de presentación es desarrollada sobre Internet Information Server (IIS), particularmente con HTML, DHTML, XML, VB Script y Java Script.
- Capa de Negocio y Datos, se utiliza Componentes COM.
- En la capa de datos se utilizará SQL Server. Los accesos a la Base de datos serán a través de conexiones desde los objetos COM, los cuales gatillan procedimientos almacenados en la Base de datos.

Del Tamaño de los objetos

- Para el desarrollo de la capa de presentación, se tienen:

- o Páginas WEB, HTML, DHTML (con imágenes de cualquier tipo), tamaño máximo permitido 5 KB (se mide a partir del código fuente de la página, considerando las imágenes e “includes” existentes).
- o Páginas ASP o Aplicativas, ídem al anterior incluyendo código de scripting y datos retornados; tamaño máximo permitido 50 KB (se mide considerando el tráfico entre dos request HTTP).
- o Se ha considerado un caso particular para el sistema de Curse de Operaciones de Crédito, considerando un peso aproximado de 150 KB.

De las herramientas a utilizar

Según las fases de desarrollo las herramientas a utilizar son las siguientes:

	Visión	Planificación y diseño	Desarrollo
Visual Modeller o Rational Rose		Diseño Conceptual Diseño Lógico Diseño físico	Diseño Físico
CCC Harvest	Versiones de documentos	Versiones de documentos Versiones repository	Versiones de documentos Versiones de código fuente
Visual Basic		Prototipos capa intermedia	Código capa intermedia
Visual Interdev	Prototipos	Prototipos capa de presentación	Código capa de presentación
IIS	Documentación interna	Documentación interna	Documentación interna
Componentes corporativos			

De la implementación de la Capa de Negocio

Se debe considerar lo siguiente:

- Evitar pasar objetos por referencia COM+. El paso de objetos por referencia, genera requerimientos a la red de comunicaciones, pues es necesario muchos viajes de ida y vuelta para mapear las áreas de memoria involucradas. Esto atenta contra la performance de la arquitectura, y en algunos casos genera estados inconsistentes de información.

Lo anterior aplica también al paso de variables de otro tipo.

- Usar Early-Binding para la creación de Objetos. Todos los objetos COM+ deben ser creados en modalidad “Early-Binding”, es decir, todo lo referente al objeto se conoce antes del tiempo de ejecución. Por lo tanto, se debe usar “Referencias” a los componentes y se debe usar el método Create Object().
- Evitar guardar objetos en el Share Property Manager. El SPM está diseñado para almacenar pequeñas piezas de información, especialmente para esquemas en que existe una única escritura y muchas lecturas. Los objetos almacenados en SPM están sujetos al modelo multithreading, lo que genera cuellos de botella en el acceso por construcción del servicio COM+.
- Usar el modelo de programación sin estados para el servidor. El manejo de estados atenta contra la re usabilidad de los componentes y la performance en el servidor de aplicaciones.
- No levantar eventos desde un objeto COM+. Si es necesario comunicar al cliente algún tipo de información, no usar eventos (RaiseEvent()). Esto hace que el método no maneje los thread en modo seguro, y puede ocasionar problemas en una transacción.

- No compartir variables globales. Visual Basic 6.0 almacena de cierta manera el valor de una variable cuando se opera en modo multi thread. Lo cual significa que una variable global escrita para COM+ tendrá diferente valor en cada instancia del objeto. Al menos que por azar se ejecuten en un mismo thread.
- Guardar estados en base de datos. Dado que el modelo de programación es sin estados, y las necesidades del negocio pueden forzar el almacenamiento de datos sensibles al contexto, se recomienda guardar estos estados en un repositorio de base de datos central a la granja de servidores.
- No declarar variables con As New. Esto causa que cada referencia al objeto sea más lenta, alargando el tiempo de vida de los objetos.
- Declarar parámetros por valor (ByVal) siempre que sea posible. Recomendación para evitar exceso de tráfico en la red o paginación debido al intercambio de punteros.

DISEÑO DE MODELO DE DATOS

Dentro de los estándares establecidos para el modelamiento de una Base de datos se indican:

- Uso de tripletas indicadas en el catálogo de abreviaturas del área de Modelamiento. Las tripletas deben ser representativas del concepto a describir.
- Presentación de Modelo lógico y físico.
- Uso de la Herramienta Erwin 3.2 para modelar.
- Documentación de todos los campos y tablas del modelo.
- Definición de índices para optimizar las búsquedas.

- Identificar con nombre distintivo cada una de las relaciones físicas entre las tablas del modelo.

A continuación se mostrará parte del modelo lógico implementado para este sistema (figura 4.1).

El modelo físico es el que se implementa, pero por seguridad no es posible entregar su estructura.

DISEÑO DE MODELO DE OBJETOS

A continuación se presentará el diagrama de objetos obtenido con la herramienta Visual Modeler, donde se presenta la capa de negocios y de datos (figuras 4.2 a 4.7).

Capa de Datos

COM Principal

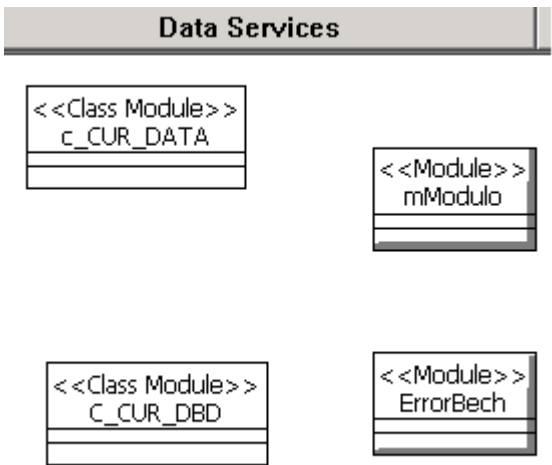


Figura 4. 2 Modelo Objetos COM Principal

La componente Principal de GDI consta de 2 clases y dos módulos, para los cuales se entregará una descripción de su funcionalidad:

- ❖ C_CUR_DATA: esta clase es la responsable de realizar las consultas a la base de datos del GDI, ya sea para realizar procesos de ingreso, actualización, eliminación y consultas. Las funciones que tiene esta clase no pueden ser explicadas ni mostradas por normas de seguridad del Banco.
- ❖ C_CUR_DBF: esta clase es la encargada de establecer la conexión a la base de datos, utilizando la componente COM corporativa XregQuery.
- ❖ mModulo: este módulo se encarga de inicializar variables.

- ❖ mErrorBech: este módulo debe ser incluido en cada proyecto que utilice COM+ y consta de funciones que reportan tanto al visor de eventos como a un archivo txt los errores detectados durante la ejecución de una instrucción.

Esta componente tiene como referencias:

- ❖ xRegQuery, COM corporativa, que realiza la conexión a la base de datos, mediante el uso de Shared Property Manager.
- ❖ ADO 2.6 (Microsoft Active Data Objects).
- ❖ COMSVCS (COM+ Services Type Library).
- ❖ StrCat, componente corporativa, utilizada al trabajar con concatenación de string⁵.

COM Consultas

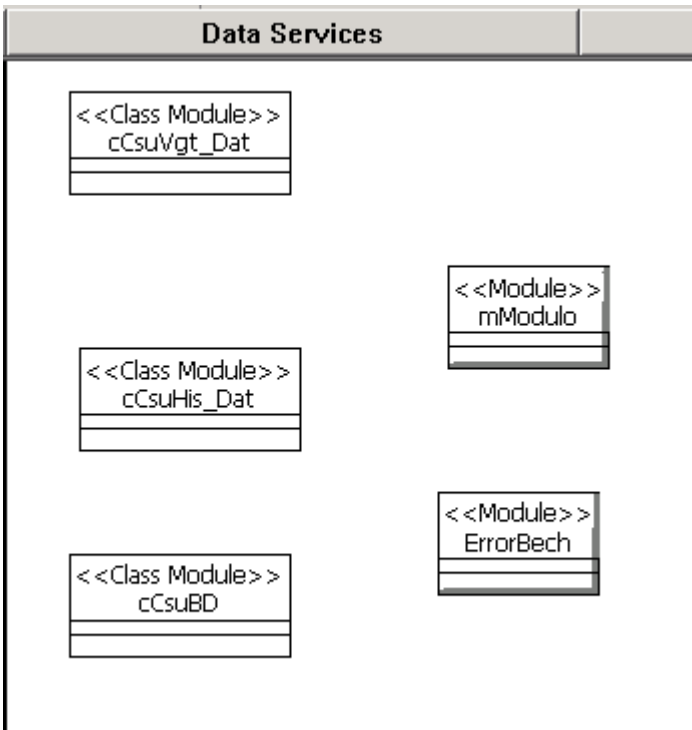


Figura 4. 3 Modelo Objetos COM Principal de Consultas

⁵ Cadena de caracteres.

La COM de Consultas, realiza las consultas a la base GDI para el módulo de Consultas que tiene cada perfil, debe proporcionar consultas:

- ❖ Vigentes
- ❖ Históricas

Está compuesto por:

- ❖ cCsuVgt_Dat: clase encargada de todas las consultas vigentes.
- ❖ cCsuHis_Dat: clase encargada de todas las consultas históricas.
- ❖ mModulo: este módulo se encarga de inicializar variables.
- ❖ mErrorBech: este módulo debe ser incluido en cada proyecto que utilice COM y consta de funciones que reportan tanto al visor de eventos como a un archivo txt los errores detectados durante la ejecución de una instrucción.

Tiene como referencias:

- ❖ xRegQuery, COM corporativa, que realiza la conexión a la base de datos, mediante el uso de Shared property Manager.
- ❖ ADO 2.6 (Microsoft Active Data Objects).
- ❖ COMSVCS (COM+ Services Type Library).

COM Administracion

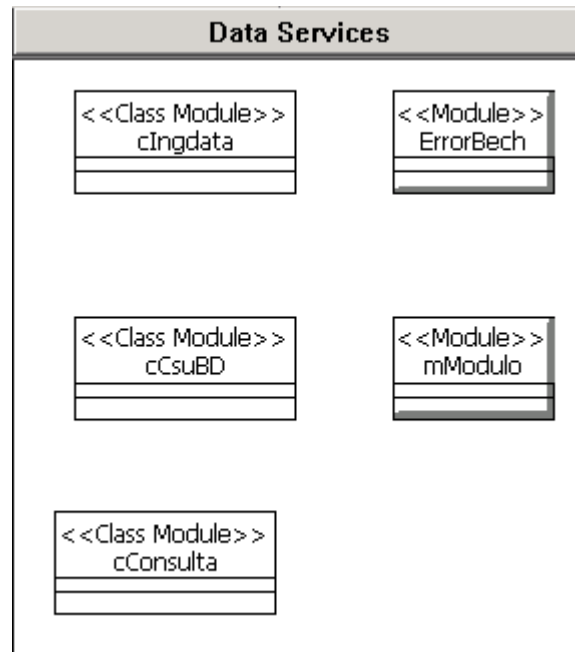


Figura 4. 4 Modelo Objetos COM Principal - Administrador

Esta COM se encarga de la administración de los datos en GDI, atendiendo:

- ❖ Ingreso, eliminación y actualización de estado.
- ❖ Ingreso, eliminación y actualización de productos.
- ❖ Ingreso, eliminación y actualización de oficinas
- ❖ Ingreso, eliminación y actualización de motivos de rechazo.

Sus referencias son:

- ❖ xRegQuery, COM corporativa, que realiza la conexión a la base de datos, mediante el uso de Shared property Manager.
- ❖ ADO 2.6 (Microsoft Active Data Objects).
- ❖ COMSVCS (COM+ Services Type Library).

Capa de Negocios

COM Principal

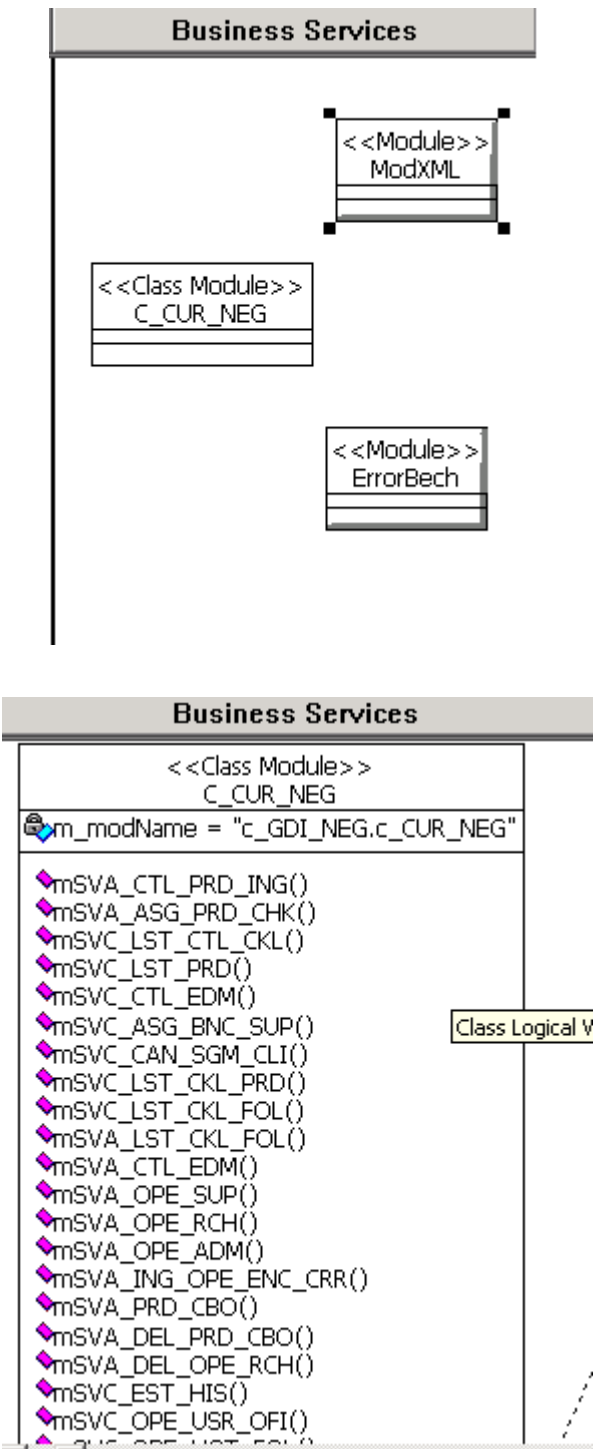


Figura 4. 5 Modelo Objetos COM Principal Negocios

Para esta COM se han mostrado algunas de sus funciones, consta de una clase y 2 módulos, la clase esta encargada de recibir desde la capa de presentación la información necesaria, depurarla, y enviársela a la componente de datos.

El módulo modXML se encarga de que para cada retorno de la componente de datos se prepare un string de salida, de ahí que cada uno de los métodos que tiene la componente retorne string.

Sus referencias son:

- ❖ C_GDI_DAT_COM
- ❖ xRegQuery, COM corporativa, que realiza la conexión a la base de datos, mediante el uso de Shared property Manager.
- ❖ ADO 2.6 (Microsoft Active Data Objects).
- ❖ COMSVCS (COM+ Services Type Library).
- ❖ StrCat, componente corporativa, utilizada al trabajar con concatenación de string.

COM Consultas

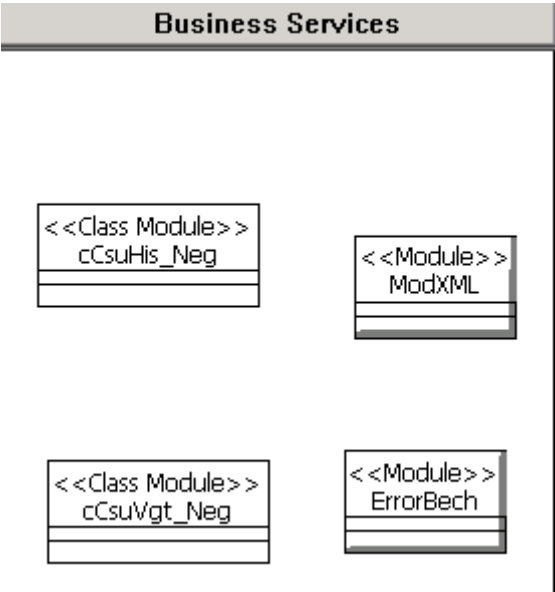


Figura 4. 6 Modelo Objetos COM Principal Negocios - Consultas

Al igual que la COM de negocios principal esta se encarga de pasarle información a la componente de datos y una vez retornados los datos son convertidos en un string XML.

Sus referencias son:

- ❖ GDI_DAT_Csu
- ❖ xRegQuery, COM corporativa, que realiza la conexión a la base de datos, mediante el uso de Shared property Manager.
- ❖ ADO 2.6 (Microsoft Active Data Objects).
- ❖ COMSVCS (COM+ Services Type Library).
- ❖ StrCat, componente corporativa, utilizada al trabajar con concatenación de string.

COM Administración

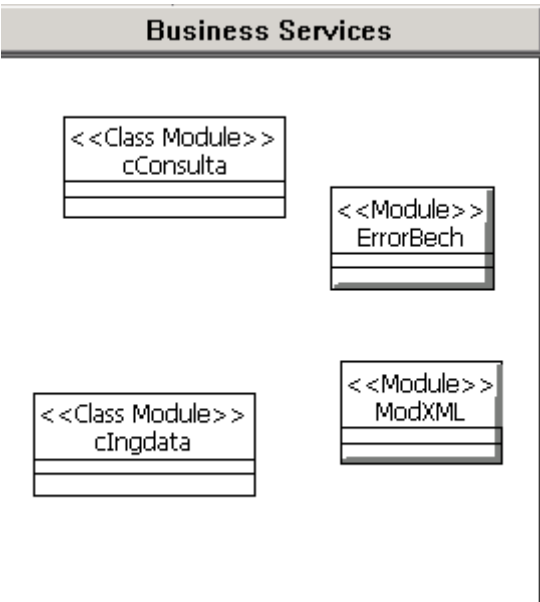


Figura 4. 7 Modelo Objetos COM Principal Negocios - Administración

Lo mismo ocurre con esta COM, donde todas sus salidas son string XML.

Sus referencias son:

- ❖ GDI_DAT_Mantenedor
- ❖ xRegQuery, COM corporativa, que realiza la conexión a la base de datos, mediante el uso de Shared property Manager.
- ❖ ADO 2.6 (Microsoft Active Data Objects).
- ❖ COMSVCS (COM+ Services Type Library).
- ❖ StrCat, componente corporativa, utilizada al trabajar con concatenación de string.

Se han establecido ciertas normas para la construcción de COM las cuales se ven reflejadas en los diagramas presentados:

1. Las COM de datos deben identificarse después del Alias asignado a la aplicación por DAT o NEG según sea el caso.
2. Se deben utilizar rutinas comunes en todas las COM para el manejo de errores, módulo mErrorBech.
3. Para la conexión en la Base de datos debe utilizarse xRegQuery.
4. El acceso a la base de datos es mediante Store Procedures (procedimientos almacenados), no se permiten consultas directas a la base.
5. Los procedimientos almacenados deben estar con las abreviaturas disponibles en el catálogo de abreviaturas que tiene el área de Modelamiento de Datos.
6. Los procedimientos almacenados deben estar claramente explicados en su interior, indicando:
 - ❖ Autor
 - ❖ Fecha de creación
 - ❖ Objetivos
7. Se trabaja con recordset desconectados.

MODELAMIENTO DE ROLES EN SUCURSALES

La figura 4.8 indica el modelo que se planteó para los roles y sus tareas en sucursales:

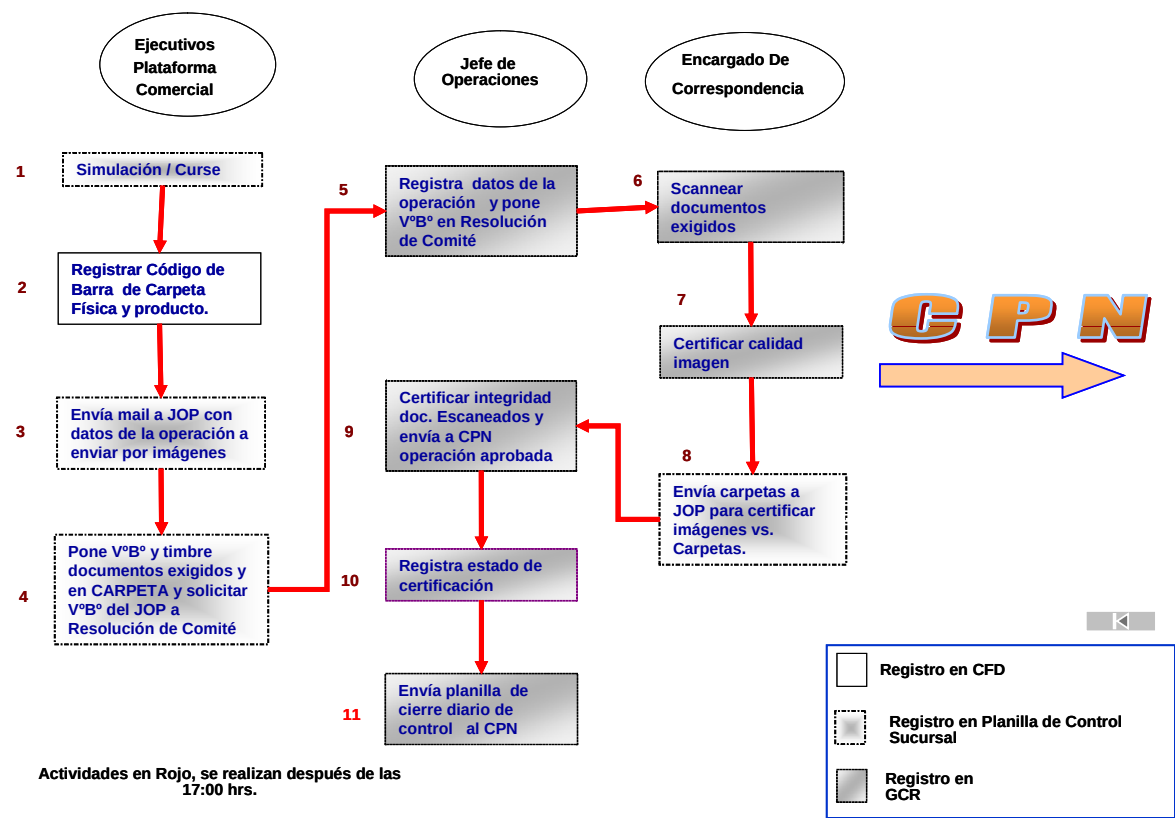


Figura 4. 8 Roles definidos en Sucursal

De los roles aquí identificados, los correspondientes a Jefe de Operaciones y Encargado de Correspondencia son los que interactúan directamente con el sistema GDI, el tercero es el que origina la operación de Curse de crédito pero para efectos de roles funcionales del sistema sólo se inicia el proceso cuando el encargado de correspondencia escanea los documento e ingresa los datos requeridos para el envío de la operación.

MODELAMIENTO DE ROLES EN CPN

La figura 4.9 indica el modelo que se planteó para los roles y sus tareas en CPN:

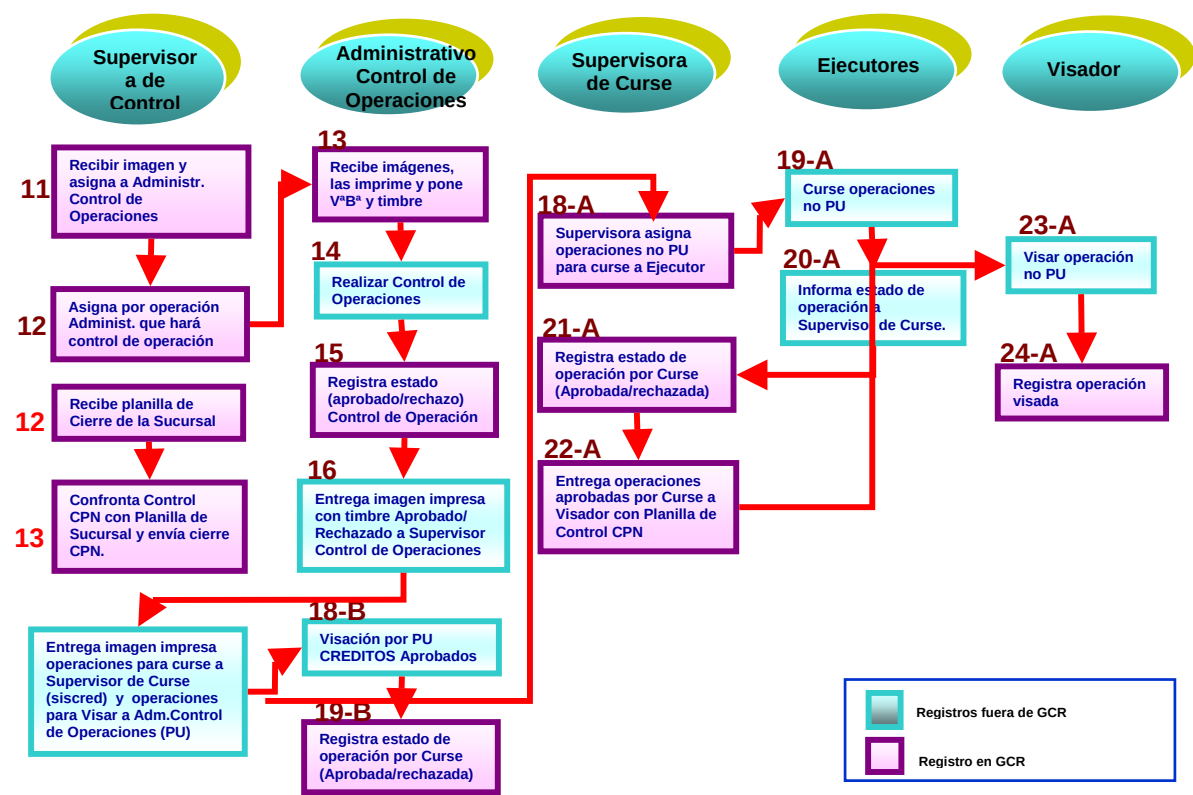


Figura 4. 9 Roles definidos en CPN

De los roles aquí identificados, sólo los correspondientes a Supervisor de Control de operaciones y Administrativo Control de Operaciones tiene incidencia directa en el sistema GDI, los otros pertenecen al flujo de trabajo pero no se le han concedido atributos en el sistema.

CAPITULO 5: CONSTRUCCION DEL SISTEMA INFORMATICO

CONSTRUCCIÓN DE PROTOTIPO

La construcción del software debe ser un proceso en cascada donde el diseño ideal debe preceder a la codificación. En el desarrollo de programas computacionales según se nos ha enseñado, existen dos etapas claramente definidas: el diseño lógico y el diseño físico.

En el modelo en Cascada, también denominado “clásico”⁶, los procedimientos de la metodología se ordenan en pasos o etapas, las cuales deberán ser seguidas bajo un enfoque secuencial de análisis, diseño y desarrollo. Creado a partir del modelo convencional de “línea de producción” de la ingeniería clásica, este modelo es el más aplicado en el desarrollo de Software.

Las etapas del modelo en cascada se pueden definir en:

- Análisis y definición de Requerimientos.
- Diseño de sistema y software.
- Implementación y prueba de unidades.
- Integración y prueba del sistema.
- Operación y mantenimiento.

Este modelo, ampliamente difundido en el ámbito de las tecnologías de la información para el desarrollo de software, permite visualizar en términos generales y como fases claramente diferenciadas las diferentes etapas por las que el desarrollo de un sistema de información debiera ir transcurriendo en el tiempo.

⁶ Véase: [Pressman 93], [Barros] y otros

«El modelo de cascada es el primero en establecer el proceso de desarrollo como la ejecución de un conjunto de actividades. En su concepción básica, cada una de las actividades generan como salidas productos y modelos que son utilizados como entradas para el proceso subsiguiente. Lo cual supone que una actividad debe terminarse (por lo menos, en algún grado) para empezar la siguiente».

Para el caso de nuestro desarrollo el principal inconveniente que tiene este modelo - dificultad para incorporar cambios después que el proceso parte - se remediará mediante la construcción de prototipos, los cuales serán mostrados a los usuarios y validados con ellos, antes de la etapa de desarrollo del software y su posterior implementación. Cualquier cambio será considerado para la etapa de mantención y evolución del software (Control de Cambios).

La construcción de prototipos semi funcionales es de gran ayuda al momento de probar el comportamiento real de las ideas del diseño y de los requerimientos del usuario. Lo que se persigue con este diseño es dejar claramente establecidas las fases por las cuales pasará el software antes de llegar al usuario final.

La idea es armar una maqueta del software, para poder corregir las fallas que pueda tener este, previo a su construcción. El prototipo muestra en dibujo las pantallas y listados, el usuario las ve y las discute hasta llegar a un acuerdo en forma y cantidad, luego se procede a construir el software, puesto que una vez aprobado el prototipo por el usuario, cualquier modificación o cambio posterior, ambas partes (usuario y el equipo de desarrollo) reconocen que es un cambio a lo acordado. En esta metodología aparecen las etapas: Diseño de Prototipo, Construcción de Prototipo y Refinamiento de Prototipo, entre las etapas Diseño Lógico y Diseño Físico de la metodología tradicional.

Los prototipos son modelos (no necesariamente productos de software) que permiten estudiar y probar aspectos específicos del producto final (en este caso el producto de software). Bajo este modelo, se planifica la aplicación de las diferentes herramientas, para producir elementos de

pruebas específicas (interfaz de usuario, mantenedores, procesos) que deberán ser presentados al usuario y confirmados por éste.

Se definirán dos prototipos, uno llamado PROTOTIPO SUCURSAL que tendrá las funcionalidades asociadas a los roles que participan en la Sucursal para el proceso y el otro llamado PROTOTIPO CPN, que contendrá todos los roles que participan en el CPN para el proceso.

Prototipo Sucursales

1. Perfil Encargado de Correspondencia (Escaneador):

INGRESAR OPERACIONES: Este menú permite registrar una operación a ser cursada vía imágenes, ingresando los datos de la operación y adjuntando las imágenes (figura 5.1).

Atributos de una operación:

- ❖ Folio interno.
- ❖ Código de barra de carpeta física.
- ❖ Sucursal origen (Sucursal que gestiona la operación puede ser la Sucursal Centralizadora o la Satélite).
- ❖ Asistente de Negocios de la Sucursal Centralizadora o Satélite.
- ❖ Fono Asistente de negocios de la Sucursal Centralizadora o Satélite.
- ❖ Nombre del Cliente.
- ❖ Rut cliente
- ❖ Tipo de producto.
- ❖ Monto bruto de la operación.
- ❖ Estado operación.

- ❖ Jefe de operación que certifica la operación en la Sucursal Centralizadora:



PLATAFORMA UNIVERSAL

desde 1855

CLIENTES

CARTERA

GESTIÓN

GESTIÓN

Gestion Imágenes

Ingresar Operaciones

Resumen Operaciones

Consultar Operaciones

< GERENCIA DE INFORMATICA >

Ingresar Operaciones

Folio Interno Nº 559

CODIGO BARRA CARPETA

SUCURSAL ORIGEN

ASISTENTE NEGOCIO

FONO ASISTENTE

TIPO PRODUCTO

RUT CLIENTE

NOMBRE CLIENTE

TIPO MONEDA/MONTO BRUTO

FECHA INGRESO OPERACION

ESTADO

FECHA ESTADO

HORA ESTADO

JEFE OPERACIONES

SANTIAGO PRINCIPAL

CAMPBELL APABLAZA ROWLAND MAURICIO

0

\$

0

19-01-2004

EN INGRESO

19-01-2004

06:03:07 PM

Adjuntar Imágenes

Grabar

Ingresar

Detalle

Figura 5. 1 Ingreso Operación en Sucursal

ADJUNTAR IMÁGENES: En esta interfaz el usuario debe ingresar las imágenes escaneadas con anterioridad o en el momento de crear la operación (figura 5.2).

GESTIÓN

Gestion Imagenes

Ingresar Operaciones

Resumen Operaciones

Consultar Operaciones

desde 1855

Adjuntar Documentos

NºFolio: 559

IMAGEN ANEXADA

Examinar...

Seleccionar Documento en PC

Capturar en Escáner y Grabar

Grabar Documento Seleccionado

Ver Documentos Asociados

Figura 5. 2 Ingreso Operación en Sucursal - Imágenes

2. Perfil Jefe de Operaciones (JOP):

CERTIFICAR OPERACIONES: Este menú permite certificar una operación que ha sido creada por el Escaneador de la Sucursal y se encuentra en estado INGRESADA (figura 5.3).

Los atributos de la operación que deben aparecer para poder certificar una operación son:

- ❖ Fecha ingreso operación.
- ❖ Folio interno.
- ❖ Código de Barra de Carpeta.
- ❖ Estado operación.
- ❖ Fecha del Estado.
- ❖ Sucursal Origen :
- ❖ Asistente de Negocios.
- ❖ Nombre del Cliente
- ❖ Monto operación.
- ❖ Tipo producto.

Figura 5. 3 Certificación de Operaciones

Al ver el detalle de una operación se presenta la opción de aprobar o rechazar la certificación (figura 5.4).

BancoEstado PLATAFORMA UNIVERSAL																		
CLIENTES			CARTERA			GESTIÓN												
Gestión																		
Gestión Imágenes																		
Certificar Operaciones																		
Consultar Operaciones																		
Resumen Diario																		
desde 1855																		
< GERENCIA DE INFORMATICA > Folio Interno Nº 549																		
Certificar Operaciones																		
CODIGO BARRA CARPETA			549															
SUCURSAL ORIGEN			SANTIAGO PRINCIPAL															
TIPO PRODUCTO			CREDITO AUTOMOTRIZ															
MONEDA/MONTO BRUTO			\$ 5800															
RUT CLIENTE			12496817 8															
NOMBRE CLIENTE			SDHFGASFHASFHU															
ASISTENTE NEGOCIO			CAMPBELL APABLAZA ROWLAND MAURICIO															
FONO ASISTENTE			0															
JEFE OPERACIONES			0															
FECHA INGRESO OPERACION			16-01-2004															
FECHA ESTADO			16-01-2004															
ESTADO			INGRESADA															
HORA ESTADO			03:29:06 AM															
☐ Certificación Aprobada ☐ Certificación Rechazada																		
Grabar			Enviar			Detalle												

Figura 5. 4 Certificación de Operaciones – Rechazar Operaciones

Prototipo CPN

Una vez que el jefe de operaciones aprueba la operación desde la sucursal aparece automáticamente en pantalla del supervisor para ser asignada a algún administrativo.

1. Supervisor CPN (SupCPN):

CONTROL DE OPERACIONES: Este menú permite asignar las operaciones que se encuentran en estado ENVIADA a los Administrativos de Control de Operaciones (figura 5.5).



PLATAFORMA UNIVERSAL

CLIENTES

CARTERA

GESTIÓN

GESTIÓN

Gestion Imágenes

Resumen Operaciones

Operaciones por Asignar

Rechazos por Confirmación

Por Visar Siscred

Asignadas sin Resolver

Cierre Ciclo

Consultar Operaciones

Resumen Diario

desde 1855

< GERENCIA DE INFORMATICA >

Control de Operaciones

Listado Operaciones, Páginas[1]

Fecha Ingreso	Folio Interno	Código Barra	Estado Operación	Fecha Estado	Sucursal Origen	Asistente de Negocio	RUT Cliente	Monto Operación	Tipo Producto
16/01/2004	553	553	ENVIADA SUP	20/01/2004	PUNTA ARENAS 18 DE SEPTIEM	Elga Benitez Moya	12051907-7	UF150.000,00	Crédito Reajutable para Estudiantes Universitarios 14 CU
19/01/2004	556	20040119	ENVIADA SUP	20/01/2004	SANTIAGO PRINCIPAL	CAMPBELL APABLAZA ROWLAND MAURICIO	12496817-8	\$5.000.000,00	CRED 2
20/01/2004	570	7857587	ENVIADA SUP	20/01/2004	PUERTO NATALES	RODRIGUEZ GARRIDO PATRICIO	22222222-2	\$111,00	COMBO1
21/01/2004	595	547854	ENVIADA SUP	21/01/2004	PUNTA ARENAS	Danny Frias Bruna	12051907-7	UF150.000,00	Crédito Reajutable para Estudiantes Universitarios 14 CU

Figura 5. 5 Resumen de Operaciones a Asignar

Este menú se divide en el siguiente sub-menú:

- ❖ Operaciones por Asignar: el supervisor asigna según operación a los administrativos a cargo
- ❖ Rechazos por Confirmar: el supervisor confirma las operaciones rechazadas de los administrativos de CPN, a modo de controlar los criterios utilizados.

- ❖ Curse / Visación
- ❖ Control Ex – Post
- ❖ Control de Operaciones: se despliegan todas las operaciones asignadas al administrativo.
- ❖ Curse / Visación: En esta etapa el administrativo marca el estado de visación de la operación, esta se divide en dos estados (Visado Aprobado o Visado Rechazado) y depende del resultado que entregue al visar la operación en PU.
- ❖ Control Ex – Post: este control se realiza una vez que llega la carpeta física al administrativo CPN, el menú en pantalla es el que se muestra en la figura 5.7:

BancoEstado

desde 1855

PLATAFORMA UNIVERSAL

CLIENTES

CARTERA

GESTIÓN

GESTIÓN
 Gestión Imágenes
 Operaciones Asignadas
 Control de Operaciones
 Curse / Visación
 Control Expost
 Consultar Operaciones

< GERENCIA DE INFORMATICA >
Operaciones Control Expost

Listado Operaciones, Páginas[1]

Fecha Ingreso	Folio Interno	Código Barra	Estado Operación	Fecha Estado	Sucursal Origen	Asistente de Negocio	Rut Cliente	Monto Operación	Tipo Producto	Nro Días	Días Atrasados
10/01/2004	458	77777	VISADO APROBADO	13/01/2004	PUNTA ARENAS	Danny Frias Bruna	14154032-7	\$ 3.265,00	CREDITO AUTOMOTRIZ	0	8
12/01/2004	483	22222222	VISADO APROBADO	13/01/2004	PUNTA ARENAS	Danny Frias Bruna	22222222-2	\$ 2.222,00	CREDIT INTERNO	0	8
12/01/2004	484	22222222	VISADO APROBADO	13/01/2004	PUNTA ARENAS	Danny Frias Bruna	22222222-2	\$ 2.222,00	CRED 2	0	8
15/01/2004	517	2222	VISADO APROBADO	15/01/2004	PUNTA ARENAS	Danny Frias Bruna	11111111-1	\$ 2.222,00	CREDITAZO	0	6
15/01/2004	518	33333333	VISADO APROBADO	22/01/2004	PUNTA ARENAS	Danny Frias Bruna	11111111-1	\$ 2.222,00	CRED 3	0	0
15/01/2004	528	222222	VISADO APROBADO	15/01/2004	PUNTA ARENAS	Danny Frias Bruna	22222222-2	\$ 2.222,00	CRED 3	0	6
20/01/2004	574	23424234	VISADO APROBADO	22/01/2004	PUNTA ARENAS	Danny Frias Bruna	77777777-7	\$ 444.444.444,00	CREDITAZO	0	0

desde 1855

Figura 5. 7 Resumen de operaciones para control Expost

VALIDACIONES Y ESPECIFICACIONES DEL SISTEMA

Como consecuencia de la toma de requerimientos, reuniones periódicas con los usuarios y posterior presentación y análisis del prototipo, las validaciones para el Sistema de Gestión de Imágenes son:

Encargado Correspondencia

Los campos requeridos para el ingreso de una operación son:

- ❖ Folio interno.
- ❖ Código de barra de carpeta.
- ❖ Sucursal origen (Sucursal que gestiona la operación puede ser la Sucursal Centralizadora o la sucursal Satélite). Este campo debe ser del tipo combo box, con todas las oficinas del Banco, quedando seleccionada en primera instancia la oficina de conexión del usuario que digitaliza (encargado de correspondencia), la cual puede ser modificada para los casos de las sucursales satélites.
- ❖ Asistente de Negocios de la Sucursal Centralizadora o Satélite.
- ❖ Fono Asistente de negocios de la Sucursal Centralizadora o Satélite.
- ❖ Rut cliente: Este campo debe tener separado el rut del dígito verificador y con verificación automática de dígito.
- ❖ Nombre del Cliente: Este atributo deberá aparecer automáticamente a partir de la lectura del Rut.
- ❖ Tipo de producto: Este campo debe ser del tipo combo box.
- ❖ Monto bruto de la operación.

- ❖ Estado operación
- ❖ Fecha del Estado.
- ❖ Jefe de operaciones que certifica la operación en la Sucursal Centralizadora: Este dato debe aparecer automáticamente dado el usuario de Conexión (Encargado de correspondencia)

Consideraciones:

- ❖ Se debe habilitar una opción que permita eliminar imágenes de manera de anexar otras nuevas, una vez que la operación está en estado CERTIFICACIÓN RECHAZADA. Esto en el caso de que el Jefe de Operaciones al certificar debe rechazar una operación, de manera de no crearla nuevamente como operación nueva.
- ❖ En la pantalla de ingresar operaciones el campo Folio debe ubicarse en la parte superior derecha de la pantalla y llamarse folio interno.
- ❖ El campo código de barra de carpeta física debe capturar el código de barra con lector de código de barra y quedar en la misma pantalla de ingreso, además de permitir el ingreso manual de este.
- ❖ Debe quedar habilitado un botón que permita grabar una operación de manera de que cuando esté completa la pantalla con todos sus datos y con los documentos digitalizados se deben grabar los datos. El botón Grabar dejará la operación en estado EN INGRESO.
- ❖ El botón Ingresar dejará la operación en estado INGRESADA, habilitando la operación para el perfil de jefe de operaciones (JOP), mientras la operación no esté en estado INGRESADADA el JOP no puede revisar su información.
- ❖ Si se modifica el nombre de la Sucursal Centralizadora para registrar una sucursal satélite asociada, el nombre del JOP de la Centralizadora debe permanecer y el menú

pull-down de Asistentes de negocios debe modificarse para desplegar los Asistentes de Negocios de la Sucursal Satélite.

- ❖ El campo tipo de producto debe poder registrar el producto COMBO, para el cual se debe desplegar un menú pull-down que permita elegir los productos que integran el combo.

El menú principal debe quedar en el siguiente orden:

- ❖ Ingreso de Operaciones
- ❖ Resumen de Operaciones
- ❖ Consulta de Operaciones

Consideraciones al Resumen de Operaciones:

- ❖ Este menú permite desplegar todas las operaciones, no importando el estado, de manera de poder ir al detalle de cada una de ellas a consultar las imágenes y además poder eliminar una imagen de una operación y poder agregarle otra (sólo para las operaciones en estado En Ingreso). Los datos a desplegar son los mismos utilizados en las consultas.

Consideraciones a Consultar Operaciones:

- ❖ Este menú permite la consulta de operaciones Vigentes e Históricas.
- ❖ Las consultas Vigentes deben mostrar todas las operaciones que estén de acuerdo al criterio seleccionado, y en el último estado en que se encuentre.
- ❖ Las Consultas Históricas muestran las operaciones desde el último día hábil del mes anterior hacia atrás.

Los criterios de selección de consultas para operaciones vigentes e históricas son:

- ❖ Rut cliente.
- ❖ Código de barra de carpeta.
- ❖ Folio interno
- ❖ Fecha desde – fecha hasta.
- ❖ Estado operación.

Se debe desplegar lo siguiente en caso de una consulta exitosa:

- ❖ Folio interno de la aplicación.
- ❖ Código barra carpeta física.
- ❖ Sucursal origen.
- ❖ Rut cliente.
- ❖ Tipo de producto.
- ❖ Monto bruto de la operación.
- ❖ Estado operación
- ❖ Fecha del Estado.
- ❖ Jefe de operaciones que certifica la operación en la Sucursal Centralizadora.

Otras consideraciones:

- ❖ Para la operación de traspaso de operaciones a Históricas, se debe considerar un proceso automático y se debe realizar el último día hábil del mes anterior, de manera de poder visualizar las operaciones del mes anterior y de ahí hacia atrás en las operaciones históricas.

- ❖ Las operaciones en el Estado EN INGRESO e INGRESADO, deben permanecer disponibles para la Sucursal hasta un máximo de 2 días hábiles, al tercer día hábil por seguridad deben ser eliminadas.

Jefe de Operaciones

Para certificar una operación, este menú debe mostrar todas las operaciones de la Sucursal en estado INGRESADA.

Los atributos de la operación que deben aparecer para poder certificar una operación son:

- ❖ Fecha ingreso operación.
- ❖ Folio interno.
- ❖ Código de Barra de Carpeta.
- ❖ Estado operación.
- ❖ Fecha del Estado.
- ❖ Sucursal Origen.
- ❖ Asistente de Negocios.
- ❖ Nombre del Cliente
- ❖ Monto operación.
- ❖ Tipo producto.

Consideraciones:

- ❖ El vínculo a los datos de la operación deben ser por el Folio interno de la operación, a través de este vínculo se debe llegar a una pantalla donde aparecen todos los atributos

de la operación, más las opciones de CERTIFICACIÓN APROBADA Y CERTIFICACIÓN RECHAZADA.

- ❖ Los valores posibles para una certificación son: CERTIFICACIÓN APROBADA ó CERTIFICACIÓN RECHAZADA.
- ❖ La Certificación Aprobada debe inmediatamente habilitar la operación para que sea leída por el Supervisor de Control de Operaciones del CPN. Además deja la operación en estado ENVIADA SUP.
- ❖ La Certificación Rechazada deja a la operación en estado CERTIFICACIÓN RECHAZADA, por lo que el Supervisor de Control de operaciones del CPN no podrá visualizar esta operación.
- ❖ Una operación en estado certificación rechazada no puede ser enviada jamás, pero si puede ser modificada por el Encargado de Correspondencia de la Sucursal, quién la debe tomar con el estado CERTIFICACIÓN RECHAZADA, hacer los cambios y dejarla en estado INGRESADA. De esta forma el JOP puede nuevamente revisar la misma operación y dejarla ya sea en estado CERTIFICACIÓN APROBADA O CERTIFICACIÓN RECHAZADA.
- ❖ En caso de dejar la operación en Estado certificación Rechazada deberá aparecer en este un campo que permita ingresar el motivo de rechazo de la operación para el Jefe de Operaciones, en un campo abierto a narrativa.
- ❖ Se deberá habilitar una opción que permita enviar un correo electrónico al Encargado de correspondencia de la Sucursal y al Asistente de Negocios de la Sucursal, ya sea centralizadora o satélite indicando este motivo de rechazo. Este correo debe tener un mensaje definido donde se completen los siguientes datos:
 - ❖ Código de barra carpeta física.
 - ❖ Folio interno

- ❖ Tipo de producto.
 - ❖ Rut cliente.
 - ❖ Motivo de rechazo.
-
- ❖ Deberán aparecer los botones GRABAR y VER DETALLE. Donde el botón GRABAR debe dejar la operación que está en estado INGRESADA, en estado MODIFICADA POR JOP. Este evento sucede cuando el JOP modifica algunos de los atributos de la operación antes de dejarla en estado CERTIFICACIÓN APROBADA. Con lo anterior una operación puede pasar a estado de CERTIFICACIÓN APROBADA, siempre y cuando esté en los estados INGRESADA o MODIFICADA POR JOP.
 - ❖ El Botón VER DETALLE, permite mirar las imágenes digitalizadas en dos modalidades, todas juntas en una sola pantalla a través del movimiento del cursor en una barra de Scroll al lado derecho de la pantalla y una modalidad que permita mirar una por una cada operación. En este mismo menú deben aparecer todos los estados por los que ha pasado una operación junto con los usuarios que han modificado los estados y las fechas de los estados.
 - ❖ CONSULTAR OPERACIONES: Este menú permite consultar operaciones vigentes e Históricas bajo los mismos criterios definidos en las consultas del Perfil ESCANEADOR.
 - ❖ RESUMEN DIARIO: Este menú debe permitir al Jefe de Operaciones poder visualizar todas las operaciones del día que fueron enviadas al CPN a través de la habilitación de éstas para el CPN, por el estado CERTIFICACIÓN APROBADA. En esta planilla deben aparecer sólo las operaciones con CERTIFICACIÓN APROBADA.
 - ❖ Deberá aparecer en esta pantalla un encabezado fijo con los siguientes datos del cierre diario:
 - SUCURSAL : Nombre de la Sucursal

- Código Oficina: Código de la Sucursal centralizadora.
- Fecha: Fecha del día.
- Jefe de Operaciones: Nombre del Jefe de Operaciones que certifica las operaciones de la Sucursal.

Luego se deberán desplegar los siguientes atributos de las operaciones:

- Asistente de Negocios Sucursal Origen.
- Código Sucursal origen (puede ser la centralizadora o la satélite).
- Identificación de la operación.
- Folio interno operación.
- Código de barra de carpeta física.
- Estado de la operación.
- Fecha del Estado.
- Nombre completo cliente (Cliente empresa o Cliente persona natural).
- Rut cliente.
- Tipo producto (si es un combo, deberán poder ser visualizados los productos que componen el combo).
- Monto operación aprobado (Monto bruto operación)

Además deberá aparecer el siguiente detalle final del detalle de las operaciones, de la siguiente manera:

- Número de operaciones aprobadas: XX (este número corresponde a la suma de las operaciones que se despliegan en la consulta, no importando el estado en que se encuentran, y que estuvieron en estado CERTIFICACIÓN APROBADA).
- Número de operaciones rechazadas: XX (este número corresponde a la suma de operaciones que fueron rechazadas por el JOP, y que se mantienen en estado CERTIFICACIÓN RECHAZADA).

Al final de este menú se debe habilitar una opción que permita dejar disponible este cierre para el Supervisor de Control de Operaciones del CPN.

Supervisor CPN

Permite asignar las operaciones que se encuentran en estado ENVIADA SUP a los Administrativos de Control de Operaciones.

- La asignación a los Administrativos será a través de un menú pull-down donde se seleccione el nombre del Administrativo de Control de Operaciones. Esta asignación deja disponible la operación al Administrativo con el estado ASIGNADA CO (Asignada para el proceso de Control de Operaciones) y con el Nombre del Analista asociado a la operación.
- Sólo se puede asignar una operación a un Administrativo de Control de Operaciones que se encuentran en estado ENVIADA SUP.
- Las funcionalidades son: Operaciones por Asignar; Rechazos por Confirmar; Por Visar Siscred; Asignadas sin Resolver y Cierre de Ciclo.

- CONSULTAR OPERACIONES: Este menú permite revisar el estado de las operaciones registradas por los distintos roles que participan en el proceso en el CPN, a través de la elección de un criterio de consulta.

Criterios de consulta:

- Folio interno operación.
- Código de barra carpeta.
- Estado operación.
- Fecha desde – Fecha hasta.
- Sucursal origen.
- Asistente de Negocios.
- Rut Cliente.
- Tipo producto.

Datos a desplegar en la consulta:

- Nombre Sucursal origen.
- Asistente de negocios de la Sucursal.
- Identificación de la operación (folio interno, código de barra).
- Identificación del cliente (nombre y rut).
- Identificación de la operación (tipo producto, monto, estado operación, fecha del estado).

- RESUMEN DIARIO: Este menú debe permitir al Supervisor de CPN consultar por totales, las operaciones del día por Sucursal y realizar el proceso de cierre diario del

día, para lo cual deberá tener disponible una opción que le permita avisar a la Sucursal que se está realizando el cierre del día por el CPN, a través de la emisión de un correo al JOP de la Sucursal o al Escaneador de Sucursal.

- Este cierre es mandatorio sobre el cierre de la Sucursal, es decir si una sucursal no ha realizado su cierre diario, este cierre cierra el proceso diario.

Administrativo CPN

Las funcionalidades de Operaciones Visadas son:

- CONTROL DE OPERACIONES
- CURSE / VISACION
- CONTROL EX - POST
- Los posibles estados luego del proceso de Control de Operaciones son: CONTROL APROBADO, CONTROL RECHAZADO.
- El Posible estado luego del Proceso de Curse y/o Visación son: CURSE APROBADO y CURSE RECHAZADO.
- Para las operaciones en estado RECHAZADO, se debe registrar el motivo de rechazo que se registrará en un campo de texto libre.
- Se debe permitir el envío de un correo al Asistente de Negocios de la Sucursal, con copia al JOP de la Sucursal. Por lo que se deberán proveer ventanas de elección de los nombres de los asistentes de negocios y de los JOP.
- CONSULTA OPERACIONES: Este menú permite consultar operaciones por el siguiente criterio de consulta.

- Folio interno operación.
- Código de barra carpeta.
- Estado operación.
- Fecha desde – Fecha hasta.
- Sucursal origen.
- Asistente de Negocios.
- Rut Cliente.
- Tipo producto.

Datos a desplegar en la consulta:

- Nombre Sucursal origen.
- Asistente de negocios de la Sucursal.
- Identificación de la operación (folio interno, código de barra)
- Identificación del cliente (nombre y rut)
- Identificación de la operación (tipo producto, monto, estado operación, fecha del estado).

DESARROLLO DE OBJETOS COM+

BancoEstado, posee una variada y heterogénea instalación tanto en lo que respecta al Hardware como al Software. Si bien es cierto, existen definiciones que establecen políticas estrictas en cuanto al Hardware y Software que componen la plataforma informática del Banco, en ocasiones

estas políticas han de ser dejadas de lado debido a necesidades imperiosas del negocio. Un caso típico es la compra por parte de alguna Gerencia Usuaria de algún producto empaquetado que sólo existe en determinada plataforma. En esos casos ha de pasarse por alto las recomendaciones de arquitectura y debido a esta necesidad de negocio aceptar la aplicación fuera de norma. Otra situación es la de los sistemas heredados. Aun existen sistemas desarrollados en plataformas que o bien ya no son soportados por sus respectivos fabricantes o bien ya son desaconsejadas por el área de Arquitectura. En estos casos, si bien se recomienda la migración de estas aplicaciones a las plataformas recomendadas o la integración de estas funcionalidades dentro de nuevos desarrollos también es necesario aceptar la existencia de aplicaciones fuera de norma. Finalmente, existen aplicaciones heredadas que se ejecutan en el servidor central del Banco (IBM OS/390) las cuales soportan el núcleo mismo del negocio del Banco, guardando las tablas maestras del negocio. Por otro lado, se podría decir que lo que no está desarrollado para y en el Host central del Banco, está principalmente hoy en día desarrollado sobre redes de PC, ya sea como desarrollos cliente servidor o bien utilizando el estándar DNA. De esta forma es como encontramos distintos motores de datos como MS SQL Server, Oracle, Sybase, DB2 sobre PC, DB2 sobre el OS/390. También encontramos aplicaciones desarrolladas en diversos lenguajes como Cobol para OS/390, C, C++, Visual Basic, VB Script, etc.

Durante los últimos años, el Banco ha adoptado las Arquitecturas líderes en la industria en cada momento. De esta forma en lo que respecta al desarrollo sobre redes de PC, se ha pasado desde la temprana adopción de MS Windows NT 3.5.1, usando desarrollo cliente servidor con MS Visual Basic teniendo como motor de datos MS SQL Server 6.5 a desarrollos más actuales usando MS Visual Basic sobre MS Windows 2000 en el caso de la arquitectura Cliente Servidor. Por otro lado, a fines de los años 90, se adoptó también de forma bastante temprana el estándar de desarrollo Windows DNA, comenzando con desarrollos sobre MS Windows NT 4.0, IIS 4.0, COM, MTS 4.0, MS SQL Server 7.0 y VBScript para posteriormente adoptar la nueva propuesta de DNA, sobre MS Windows 2000, utilizando IIS 5.0, COM+, MS SQL Server 2000 y

VBScript. Esta es la arquitectura de desarrollo de aplicaciones distribuidas vigentes actualmente en BancoEstado.

Infraestructura de desarrollo COM+ existente en BancoEstado

En el caso de BancoEstado, actualmente se utiliza DNA 2000, vale decir, todos los servidores cuentan con Windows 2000 como Sistema Operativo, además, se utiliza IIS 5.0 en la capa de presentación, MTS 5.0 en la capa de lógica de Negocios y MS SQL 2000 en la capa de datos. Como ya se ha mencionado, la interacción de todos estos componentes se da a través de COM+.

Cabe señalar que en el caso de la capa de presentación, se utiliza una especie de shell única, llamada Plataforma Universal. Esta shell norma el uso de tipo de letras, colores, etc. a través del uso de Cascading Style Sheet (CSS). Además, esta shell provee y obliga al uso de una estructura de navegación basada en un control del tipo árbol (Treeview). Así, la navegación entre las distintas páginas que componen una aplicación está dada por urls que son inscritas en una base de datos ad-hoc la cual es leída por el control Treeview dando al usuario acceso a los links relativos a la función que ha de desempeñar dentro de cada aplicación. Además, la shell, inicialmente solo mostrará al usuario las aplicaciones disponibles según el perfil que tenga definido en dicha Base de Datos. De esta manera, al momento del desarrollo, las aplicaciones han de cumplir con el estándar definido por esta shell en cuanto a utilizar la CSS adecuada, a la inclusión de los perfiles que definen la aplicación, los usuarios que cuentan con esos perfiles, etc. Además, para la capa de presentación, se cuenta con ambientes de desarrollo en cada estación de los desarrolladores a través de una máquina virtual que cuenta con Windows 2000, IIS 5.0 y COM+, además del cliente SQL Server para Windows 2000.

Para la capa de Objetos de Negocio, se cuenta con servidores que también ejecutan Windows 2000, los cuales están asociados a los IIS de los desarrolladores durante el proceso de desarrollo de Software. Estos servidores de capa de Objetos de Negocio para el desarrollo son administrados por un área de Soporte al Desarrollo, la cual vela por la continuidad operacional de

estos, como también por el correcto uso de la infraestructura presente mediante revisiones de código fuente de los componentes, indicaciones de mejores prácticas, etc. Además, el área de Soporte al Desarrollo, administra la instalación de los componentes en estos servidores.

Finalmente, en el caso de los servidores de datos para los equipos de desarrollo, estos son administrados centralizadamente por el área de administración de Datos, la cual valida los modelos de datos, corrige los procedimientos almacenados, vela por la continuidad operacional de la plataforma, sugiere el uso de un tipo de cursor u otro, etc.

Para el desarrollo en general, se cuenta con ambientes de desarrollo, pruebas y producción. Como en el ambiente de producción se necesita que las aplicaciones estén disponibles para muchos usuarios y durante gran parte del día o en algunos casos en la modalidad de 7x24, se ha implementado una granja de servidores de presentación para el ambiente de producción, la cual cuenta en estos momentos con cerca de 70 servidores de presentación. Estos servidores, tienen asociados a su vez un número similar de servidores de la capa de Objetos de Negocio, los cuales a su vez apuntan a muchos servidores de datos con MS SQL Server 2000. Estos servidores de datos son los más potenciados en cuanto a CPU y memoria RAM.

Para el desarrollo, existen varias herramientas a ser utilizadas por los equipos de desarrollo de manera obligatoria, por ejemplo, los strings de conexión no se utilizan de forma libre, estos han de ser guardados en el registro de las maquinas de desarrollo, pruebas o producción mediante la herramienta XRegQuery (desarrollada al interior de BancoEstado) la cual encripta los strings de conexión y los almacena en el Registro de la máquina en cuestión. Para el momento de la ejecución, las aplicaciones han de utilizar la misma herramienta para descryptar los strings de conexión y de esta forma lograr la conexión efectiva con los servidores de datos.

DESARROLLO DE CAPA DE PRESENTACIÓN.

En BancoEstado, se utiliza para el desarrollo DNA 2000, la tecnología ASP 3.0, donde las páginas residen en Internet Information Server (IIS) 5.0 sobre Windows 2000 Advanced Server. Estas páginas han de estar programadas utilizando Visual Basic Scripting Edition (VBScript) para el código del lado del servidor. Además, se utiliza tanto la implementación de Microsoft (JScript) para el estándar Java Script como el mismo VBScript para el desarrollo del código que ha de ejecutarse del lado del cliente.

Por otro lado, como existen páginas que no son dinámicas, para ello se utilizan paginas HTML. Es indistinto si se utilizan páginas del tipo htm o html.

Como se mencionó anteriormente, se utilizan Cascading Style Sheet para dar un formato uniforme a las páginas, y cumplir con los estándares de la Plataforma Universal (PU).

Por otro lado, se utiliza MS XML 4.0 para el parseo de datos que se presenten en formato XML en el servidor para enviarlos posteriormente al cliente o para la transferencia de datos a otros servicios, motores de datos, aplicaciones, etc.

Cabe señalar que esta Shell, se encuentra integrada dentro del Sistema de Privilegios, que es una herramienta de carácter corporativo que permite el manejo de los permisos de los usuarios sobre las aplicaciones y sobre los distintos módulos dentro de una aplicación.

Como se mencionó anteriormente, el equipo de desarrollo ha de cumplir con una serie de normas que son verificadas y administradas por distintas Áreas del Banco, entre estas se pueden mencionar:

- El producto de administración de fuentes y versiones en la plataforma NT-2000 es CCC-Harvest, cuya administración es responsabilidad del área de Administración de Cambios. El uso de este producto por parte de los equipos de desarrollo es obligatorio.
- La implementación física del modelo de datos es responsabilidad del Área de Soporte al Desarrollo.

- La comunicación con la capa de datos se hará exclusivamente desde la capa de Lógica de Negocios invocando procedimientos almacenados de la capa de datos mediante componentes COM+.
- No se permite el uso de controles ActiveX.
- No está permitido el uso de variables de Sesión en las paginas ASP.
- Para el manejo de estados persistentes, ha de usarse la persistencia en la base de datos.

INTEGRACIÓN DE COM+ A CAPA DE PRESENTACIÓN

Para lograr la integración de las páginas de la capa de presentación a los componentes desarrollados con COM+, se utiliza el ambiente COM+ provisto en los mismos servidores de presentación.

En el caso de la conectividad hacia la capa de datos, el estándar definido es Microsoft Data Access Components (MDAC) 2.6.

El acceso a los distintos componentes se logra, inscribiendo los componentes de lógica de negocio o de acceso a datos en la capa de Lógica de Negocio dentro del entorno COM+ del servidor de Negocios. A continuación, estos componentes son inscritos de forma remota en el servidor de presentación, también utilizando los servicios de COM+. En la página ASP misma, la vinculación se logra mediante el uso de lo que se conoce como binding, utilizando la instrucción `Server.CreateObject` la cual crea una instancia del objeto que se ha referenciado desde la capa de Lógica de Negocios la cual será utilizada por la capa de presentación.

Para lograr la conectividad con el HOST IBM, se utiliza la herramienta corporativa NewMex (desarrollada dentro de Banco). Esta herramienta es a su vez un componente COM+ que permite

ejecutar servicios CICS en el HOST OS/390 a través de un intercambio de áreas comunes llamado Header Staff.

Además, para el desarrollo e integración de componentes COM+ en la capa de presentación, han de considerarse:

- Evitar pasar objetos por referencia.
- Crear los objetos lo más tarde posible y eliminarlos lo más temprano posible.
- No levantar eventos desde un componente COM+.
- No compartir variables globales.

POBLAMIENTO DE DATOS.

Para realizar el poblamiento de la base de datos del sistema se realizaron reuniones con los usuarios para definir los datos que deberían estar cargados con anterioridad, los cuales son:

1. Motivos de rechazo

<i>Código</i>	<i>Descripción</i>	<i>Tipo Rechazo</i>
1	PAGARÉ CON BASURA	1
2	FALTA HUELLA DACTILAR EN PAGARÉ	1
3	IMAGEN NO CORRESPONDE A DOCUMENTO FÍSICO	1
219	DOCUMENTO SIN FIRMA Y TIMBRE ADN	1
220	IDENTIDAD DE CLIENTE NO CORRESPONDE	1
221	CEDULA DE IDENTIDAD NO VIGENTE	1
222	CEDULA DE IDENTIDAD BLOQUEADA	1
223	SITUACION LABORAL EMPRESA NO EXISTE	1
224	Nº DE TELEFONO NO CONCUERDA	1
225	CLIENTE NO TRABAJA EN EMPRESA	1
226	CLIENTE NO EXISTE EN AFP	1
227	RUT Y NOMBRE NO EXISTE EN S.I.I	1
228	ROL PROPIEDAD NO CONCUERDA	1

229	NOMBRE PROPIETARIO NO CONCUERDA	1
230	DIRECCION NO CORRESPONDE	1
250	SE ENVÍA MAIL CON MOTIVOS DE DEVOLUCIÓN	1
251	SOLICITUD SCORING NO FORMALIZADA	1
253	OFICINA SUSPENDIDA	1
275	OFICINA SUSPENDIDA	2
276	SE ENVÍA PAUTA DE DEVOLUCIÓN POR SISDEV	2
277	SE ENVIA PAUTA DE DEVOLUCION	3

Donde los tipos de rechazo son:

- 1 → Rechazo del Jefe Operativo
- 2 → Rechazo Supervisor
- 3 → Rechazo por Curse

2. Estados de Una Operación

Código	Descripción
AC	ASIGNADA CO
AE	APROBADO CONTROL EXPOST
CA	CURSE APROBADO
CB	CONTROL APROBADO
CC	CIERRE OPERACION
CJ	CERTIFICACION RECHAZADA
CR	CURSE RECHAZADO
EC	ENVIADO A CONTROL
EJ	INGRESADA
ES	ENVIADA SUP
OI	EN INGRESO
PC	ASIGNADA A CURSE
PV	ASIGNADA A VISACION
RA	REASIGNADO CO
RC	RECHAZO CONFIRMADO

RE	RECHAZADO CONTROL EXPOST
RG	RECHAZADO CONTROL EXPOST IMAGEN
RI	IMAGEN RECHAZADA
RJ	CONTROL RECHAZADO
RM	RECHAZO CONFIRMADO POR CONTROL OPERACIONES
RO	RECHAZADO CONTROL EXPOST OPERACION
RS	RECHAZADO A SUPERVISOR
RU	RECHAZO CONFIRMADO POR CURSE
RV	RECHAZO CONFIRMADO POR VISACION
RX	RECHAZO CONFIRMADO POR CONTROL EXPOST
VA	VISADO APROBADO
VR	VISADO RECHAZADO

3. Oficinas Centralizadoras y Satélites

<i>Código Oficina</i>	<i>Tipo Oficina</i>	<i>Oficina Padre</i>	<i>Días Carpeta</i>
1	1	1	3
8	2	10	3
10	1	10	3
11	2	10	3
20	2	21	3
21	1	21	3
24	2	25	3
25	1	25	3
29	2	25	3
121	1	121	3
122	2	121	3
123	1	123	3
124	1	124	3
125	1	125	3
126	2	125	3
133	1	133	3
134	2	133	3
136	2	133	3

137	1	137	3
824	2	825	3
825	1	825	3
831	1	831	5
832	2	833	5
833	1	833	5
834	2	833	5

Donde el Tipo de Oficina corresponde a qué tipo de oficina es:

1: Centralizadora

2: Satélite

El campo código de oficina es utilizado por este sistema recurriendo a Tablas Generales (base interna del Banco) para obtener la descripción y nombre de cada oficina (sucursal del Banco) en caso de ser requerido.

Para las otras tablas del modelo los datos son ingresados según vayan generándose operaciones.

Existen en el modelo tablas o bases reconocidas como lógicas que sólo serán mencionadas y explicado su contenido, pero sus datos no pueden ser revelados, estas son:

Usuario: tabla de usuarios del Banco, desde donde se extrae información referente al usuario conectado, verificando nombre, rut y oficina.

Productos: base donde se encuentran todos los productos existentes en el Banco y del cual se extraen sólo los asociados al proceso de curse de operaciones de crédito.

Oficinas: tabla desde donde se consultan las oficinas del Banco, obteniendo su descripción.

Clientes: base desde donde se verifica la existencia de un cliente, extrayéndose su nombre, rut y su vigencia.

Monedas: tabla desde donde se verifica el tipo de moneda con el cual se está cursando la operación, entregándose la descripción de ésta.

Privilegios: base donde están almacenados todos los privilegios de acceso del Banco y desde donde se verifica que el usuario conectado tenga el perfil para acceder al GDI. Dichos perfiles son:

- Encargado Correspondencia: GDI_PRF_Escaneador
- Jefe Operaciones: GDI_PRF_JefeOperaciones
- Supervisor: GDI_PRF_SupCPN
- Administrativo: GDI_PRF_AdmCPN
- Administrador Sistema: GDI_PRF_Mantenedor
- Consultas: GDI_PRF_Consulta

Por normas de seguridad los datos expuestos son los que existen en ambiente de pruebas del usuario, no son revelados los datos correspondientes al ambiente de producción.

INTEGRACIÓN DE LA SOLUCIÓN CON SERVICIO DE DIGITALIZACIÓN DE IMÁGENES.

El servicio de Digitalización de imágenes es un servicio que se creó por el mismo equipo que lideró y desarrolló este proyecto pero fue separado de este sistema ya que puede ser utilizado desde otras aplicaciones, con esto se logra separar procesos en caso de requerir modificaciones futuras. Esto muestra una de las ventajas que posee el trabajar en un ambiente distribuido, ya

que este sistema puede ser utilizado por varios otros sistemas a la vez, sin tener relación alguna entre ellos, como es el caso de los sistemas que a continuación se indican:

- Gestión de Correspondencia – GCR
- Modelo Atención y reclamos – MARS
- Sistema Verificación Antecedentes Clientes – GDV
- Sistemas Gestión de Imágenes - GDI

Definición Proceso

Este servicio permite el ingreso, eliminación y consulta de imágenes digitalizadas desde una base de datos. Recibe como parámetro de entrada un string XML, el cual se expone a continuación:

```
<Servicio>
  <NombreServicio>value</NombreServicio>
  <TipoServicio>value</TipoServicio>
  <ParametrosdeEntrada>
    <Parametros nomparam="nombre_parametro">value</ Parametros>
    <Parametros nomparam="nombre_parametro">value</ Parametros>
  </ParametrosdeEntrada>
  <NombreTabla>value</NombreTabla>
  <CamposLLaveTabla campopag="nomcampo">
    <Campo1>value</Campo1>
    <Campo2>value</Campo2>
  </CamposLLaveTabla>
  <CamposTablas>
    <Campoact1 tipo="tipodedato">value</Campoact1>
    <Campoact2>value</Campoact2>
  </CamposTablas>
  <NombreCampImagen>value</NombreCampImagen>
  <CampoTipoDocumento>value</CampoTipoDocumento>
</Servicio>
```

Donde:

NombreServicio	Corresponde al identificador de la aplicación que esta llamando al servicio de digitalización. Con este nombre se identifica el nombre de la base de datos donde se debe grabar o consultar los datos.
----------------	--

TipoServicio	Si es consulta, Ingreso o Modificación.
ParametrosdeEntrada	Corresponde a los campos que se requieren visualizar desde la aplicación.
nomparam	Debe indicarse en cada Parámetro (<Parametros>), en este campo se indica el Nombre del Parámetro, y en el value se indica el valor del parámetro de entrada que se quiere visualizar desde la aplicación.
NombreTabla	Nombre de la tabla donde se almacenan las imágenes.
CamposLLaveTabla	Nombre y valor de los campos que son llave en la tabla, debe especificarse siempre el valor del campo llave, para realizar la consulta a la BD.
campopag	Nombre del campo que maneja la cantidad de imágenes ingresadas para un mismo documento.
CamposTablas	Nombre y valor de los campos a actualizar o consultar desde la tabla, y tipo campo (atributo tipo), debe especificarse siempre el valor de los campos.
NombreCampolImagen	Nombre del campo en la base de datos donde se guarda la imagen.
CampoTipoDocumento	Nombre del campo en el cual se almacena el tipo de archivo a adjuntar, este campo lo llena la aplicación, en el Value debe ir indicado dicho nombre; pero debe ser indicado su nombre. Puede ser nulo para aquellos casos en los cuales la tabla mencionada no posee dicho campo.

Explicación del Proceso

Servicio de Ingreso o modificación.

1. Para iniciar el proceso de Ingreso, debe especificarse una variable de nombre: **servicio**, a esta variable se asigna el xml de entrada. Esta variable es recibida por la página **Ingreso_Documentos.asp** y procesa la información solicitada.
2. Al invocar el servicio de ingreso, se pintará la pantalla de la figura 5.1, de acuerdo a los campos enviados en los parámetros de entrada.
3. Si la opción seleccionada es **Capturar Imagen**, llamar a la componente que permite escanear (**scan_jpg.dll** parte del servicio de digitalización) una vez que la componente captura la imagen se llama a la componente de negocio que permite hacer la grabación.
4. La componente de negocio, de acuerdo al servicio enviado en el parámetro de entrada se conecta a la base de datos, la conexión a la base de datos se realiza leyendo los datos necesarios desde la entrada del registro del servidor de negocio.
5. Si la opción es seleccionar una imagen desde el PC se llama a la componente de negocio **Grabar_Imagen()**.
6. Si la opción es mostrar es **Ver Imágenes**, se llama a la componente de negocio **Leer_Imágenes**, y se pinta la pantalla con las imágenes leídas desde la base de datos, ver figura 5.2.

Adjuntar Documentos

N° SS	123
Tipo	Credito Consumo
Sub Tipo	Creditazo Dieciochero
Tema	Rectificar Solicitud
Rut Cliente	1-9
Nombre y Apellidos	John Perez
Origen	Internet

IMAGEN ANEXADA

Examinar...

Seleccionar Documento en PC

Capturar en Escáner y Grabar

Grabar Documento Seleccionado

Ver Documentos Asociados

Volver

Figura 5. 8 Pantalla de ingreso de imágenes

Adjuntar Documentos

N° SS	123
Tipo	Credito Consumo
Sub Tipo	Creditazo Dieciochero
Tema	Rectificar Solicitud
Rut Cliente	1-9
Nombre y Apellidos	John Perez
Origen	Internet

Documentos Adjuntos



1 ☐ Borrar



2 ☐ Borrar



3 ☐ Borrar



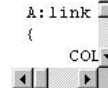
4 ☐ Borrar



5 ☐ Borrar



6 ☐ Borrar



7 ☐ Borrar

Eliminar

Ver Documentos Completos

Volver

Figura 5. 9 Despliegue de imágenes ingresadas para una operación

Servicio Consulta

1. Para iniciar el proceso de Consulta, debe especificarse una variable de nombre: servicio, a esta variable de se asigna el xml de entrada. Esta variable es recibida por la página Consulta_Documentos.asp y procesa la información solicitada.

- 2. Al invocar el servicio de Consulta, se pintará la pantalla de acuerdo a los campos enviados en los parámetros de entrada.
- 3. Luego de acuerdo al servicio invocado se conecta con la base de datos, una vez realizada la conexión forma la consulta de acuerdo a los datos enviados en el xml de entrada del servicio.
- 4. Pinta la pantalla con las imágenes leídas desde la base de datos, según Figura 5.3.



Figura 5. 10 Despliegue de imágenes consultadas

A efectos de ilustración del servicio se han utilizado sólo imágenes de fotografías, lo que en realidad se exporta en sucursales son las imágenes de:

- Fotocopia Carné Identidad.
- Fotocopia Pagaré.
- Fotocopia documento solicitud apertura productos BancoEstado.
- Fotocopia de otros documentos anexos, entre los que destacan:
 - Comprobante de domicilio.
 - Certificado renta.
 - Certificado antigüedad laboral (empleado dependiente).
 - Cotizaciones AFP
 - Copia declaración de impuestos, por mencionar algunos.

PREPARACIÓN DE AMBIENTE DE PRUEBAS

Las diferentes pruebas de software - realizadas en todas las fases del ciclo de vida - garantizan la entrega a producción de una aplicación con un número mínimo de fallas y eventualmente ninguna, aumentando la confiabilidad en las áreas de desarrollo y la satisfacción de los Usuarios.

Para este sistema dado la criticidad que representa fue elaborado un plan de pruebas e implantación que entregue un ambiente óptimo y realista a las condiciones de hardware y software que posee el BancoEstado, el cual debe comprender:

- ❖ Elaboración del Plan de Pruebas.
- ❖ Preparación de casos de prueba.
- ❖ Ambientación del lado cliente como del lado servidor.
- ❖ Documentar los resultados.

Papel importante en esta fase es el área de SQA⁷, la cual establece una metodología que contiene un conjunto de procedimientos que regulan, técnica y administrativamente, el diseño y pruebas de software a realizar en el desarrollo y mantención de software.

El propósito del SQA es garantizar la adecuada aplicación de las Metodologías definidas, aceptadas y difundidas a las diversas áreas que participan en el proceso de Desarrollo y Mantención de Software, por medio de revisiones e informes que son presentados a los responsables de cada proyecto, para lograr que el proceso sea continuamente mejorado.

Dentro de las actividades que regula SQA se define la Validación de Calidad de Software, que es el conjunto de actividades orientadas a determinar si el producto satisface las necesidades del cliente cuando es instalado en el ambiente apropiado.

Aplicación de métodos, procedimientos, técnicas y herramientas conducentes a la detección de errores y defectos del software. Esta actividad es conocida normalmente como Pruebas del

⁷ SQA: Software Quality Assurance(Seguramiento Calidad del Software)

Software. La etapa de pruebas comprende las actividades destinadas a controlar la calidad de los productos de software construidos en el sistema, dichas actividades están supervisadas por el departamento de TESTING, las cuales son (ver tabla 5.1):

Tabla 5. 1 Actividades a controlar por el área de testing

Actividades	Tareas	Descripción
Generación de la Solicitud de testing.	- Llenar formulario para la solicitud. - Enviar Solicitud.	Mediante este procedimiento se entrega a Testing el requerimiento formal para realizar el Testing de uno o más productos de software. Se debe adjuntar programas fuentes, objetos, procedimientos, archivos y condiciones de ambiente que permitan la correcta ejecución del programa en Testing. Se entrega además los antecedentes de las pruebas básicas realizadas en Desarrollo. Esto permitirá posteriormente orientar a Testing en la preparación de los ambientes necesarios.
Recepción de la Solicitud de Testing	- Entrega de módulo/aplicación. - Revisar formulario solicitud de testing.	La recepción del software traspasado al Área de Testing permite verificar si la documentación enviada es suficiente para realizar las pruebas solicitadas y en base a ello establecer las fechas de

	- Analizar los elementos entregados. - Avisar al solicitante.	en base a ello establecer las fechas de inicio y término de éstas.
Estimación de Esfuerzo	- Revisar Solicitud. - Estimación de esfuerzo.	Ingresar los datos necesarios para el cálculo del tiempo requerido para una prueba, considerando todas las actividades relacionadas con ella.
Planificar el testing	- Planificar las actividades. - Desarrollar el plan de trabajo. - Seguimiento de testing.	Establecer el detalle de las actividades a realizar y las fechas de inicio y término del Testing. Se activa al momento en que Testing recibe una solicitud para revisar un producto o probar un programa ó módulo de la aplicación.
Generar Ambiente de pruebas	- Asesoría del jefe de proyecto. Desarrollo/Usuario. - Generar ambiente de prueba. - Registrar ambiente de prueba.	El Jefe de Proyecto de Desarrollo debe proveer al Analista de Testing los elementos necesarios, para la definición del ambiente en el que deberán realizarse las actividades de test, aportando un conjunto de criterios y elementos técnicos a considerar en la generación del Ambiente de Pruebas de la aplicación.
Realizar Análisis Transaccional	- Análisis de la aplicación. - Análisis de software.	El Jefe de Proyecto deberá proveer los elementos necesarios al Analista de Testing y al usuario para que se logre

	- Identificar transacciones. - Descartar transacciones. - Priorizar transacciones. - Registrar análisis transaccional.	identificar las transacciones o funciones elementales a probar, aportando un conjunto de criterios y elementos técnicos a considerar en esta tarea.
Generar Escenarios de Prueba	- Generar escenarios de prueba. - Análisis de los escenarios de prueba. - Registrar los escenarios de prueba.	El Jefe de Proyecto debe proveer de los elementos necesarios al Analista de Testing en la determinación de los diferentes escenarios a que será sometida la aplicación como parte del proceso de pruebas. Los escenarios pueden ser ciertas fechas (Ej.: primer día del año), plataformas de software distintas (Ej. Windows 98, Windows 2000), plataformas de hardware (Ej.: PC, celular).
Generar Casos de Prueba	- Análisis de los casos de prueba. - Construir casos de prueba. - Registrar casos de prueba.	El Jefe de Proyecto debe proveer de los elementos necesarios al Analista de Testing (y al Usuario, si corresponde) en la generación de casos de prueba. Es recomendable siempre que sea posible especificar valores a las variables del sistema, quedando, de esta manera, los datos preparados para la realización del Test.

Probar	- Coordinación con el usuario. - Realizar inspección del código. - Realizar las pruebas. - Verificar los resultados. - Registrar los resultados. - Verificar el término de la prueba.	Apoyar al Analista de Testing, y al Usuario, con un conjunto de criterios y actividades a realizar en el proceso de prueba del software.
Despachar Resultados	- Registrar estado de las pruebas y fecha. - Entrega de resultados. - Rechazo de la prueba.	Poner término a las actividades de Prueba del Software y formalizar la entrega de los resultados a Desarrollo.
Respaldar y Depurar		El Jefe de Proyecto deberá proveer los elementos necesarios al Analista de Testing y al Jefe de Proyecto de Desarrollo en las labores de registro y control de la información correspondiente al test.

Se presentará un extracto de las pruebas de usuario establecidas para los roles involucrados en el sistema:

Escaneador

Tabla 5. 2 Plan de pruebas perfil Escaneador (encargado correspondencia)

Prueba Nº	Menú	Submenú	Funcionalidad / Botones	Resultado
	Gestión de Imágenes			
		Ingresar Operaciones		
1			Verificar que folio interno es asignado en forma automática por el sistema GDI.	
2			En campo Código de Barra Carpeta, ingresar código a través del lector habilitado para ello, luego verificar que quede registrado dicho código en sistema GDI.	
3			Verificar en campo de Sucursal, despliegue automático de la Sucursal del Escaneador, se debe permitir cambiar la Sucursal, para el caso de ser Sucursal Centralizadora.	
4			Verificar en campo Asistente Negocios despliegue automático del datos, debe aparecer asistente asociado a la oficina del escaneador. Este campo debe cambiar automáticamente en caso de	

			ser modificada la oficina de la operación ingresada.	
5			Ingreso Fono Asistente. Verificar validación de ingreso de sólo números a dicho campo.	
6			Ingreso del Rut del cliente. Se debe verificar el cálculo automático del dígito verificador.	
7			En campo del nombre del cliente, se despliega en forma automática al ingresar el Rut en prueba anterior.	
8			En campo Tipo de Producto, aparece como combo box, seleccionar productos según opciones indicadas.	
9			En campo Monto Bruto, ingresar en forma manual el monto que indica el Pagaré o los Pagarés cuando se trata de varios productos. Verificar validación de sólo números. Además validar formato del monto (punto "-").	
10			En campo Estado, siempre debe indicar en que parte del proceso va. Para este caso siempre debe indicar En Ingreso.	
11			En campo Fecha de Estado, siempre debe indicar el día que está ingresando	

			la operación.	
12			En campo nombre del Jefe de Operaciones debe desplegarse de acuerdo al logín del escaneador, es decir deben ser de la misma sucursal.	
13			Verificar ingreso a adjuntar imágenes mediante botón “Adjuntar Imágenes”.	
14			Verificar escaneo de imagen mediante al pinchar botón “Capturar”, que realiza la acción de la digitalización.	
15			El escáner abrirá su propia aplicación y mostrará el documento en una imagen previa, que deberá certificar que está completa. De no ser así, ir con el Mouse al documento y marcar el contorno del documento.	
16			Cambiar en la aplicación del escáner el tipo de color y dejarla en colores grises, para bajar la capacidad del disco en el envío de las imágenes.	
17			Una vez correcta la imagen del documento dar un clic en la flecha que aparece al final de las opciones.	
18			Efectuar esta operación las veces que sea necesario en función de los	

			documentos a enviar por Imagen.	
19			Verificar el almacenamiento de toda la información existente en pantalla sin cambio de estado al presionar botón Grabar.	
20			Verificar el despliegue de una pantalla que muestra estados por los cuales ha pasado la operación, usuario responsable y botón de ver imágenes digitalizadas al presionar botón Detalle.	
21			Botón Ingresar: la utilización de este botón, es para finalizar el proceso de digitalización de imágenes de una operación. Verificar envío de mensaje de que ha sido enviada a Jefe de Operaciones.	
		Resumen Operaciones	Este Submenú permite desplegar todas las operaciones, no importando el estado en que se encuentra. Para poder ir al detalle de cada una de ellas, realizar lo siguiente : Pinchar Submenú Resumen Operaciones. Se desplegarán las operaciones digitalizadas	

			El detalle de la información es igual a lo indicado en Botón Detalle, del Submenú Ingresar Operaciones. En campo Folio, está asociado con Link, que muestra el detalle de la operación.	
22			Verificar despliegue de las operaciones ingresadas y enviadas que ha realizado el escaneador. Deben aparecer los siguientes datos : - Folio Interno - Código Barra Carpeta - Sucursal de Origen - Nombre del Asistente de Negocios - Rut del Cliente - Tipo Producto - Monto Bruto - Estado de la Operación - Fecha de Estado - Nombre Jefe de Operaciones Sucursal. Centralizadora.	
23			El campo folio, está asociado con Link, que muestra el detalle de la operación	
24		Consultar	Este Submenú permite realizar	

		Operaciones	consultas de las operaciones ingresadas al sistema. Las consultas se dividen en Vigentes e Históricas: - Se definen como Vigentes, aquellas operaciones que fueron ingresadas dentro del mes en que se está realizando la consulta. - Se define como Históricas, a todas las operaciones ingresadas hasta el último día hábil del mes anterior en que se realiza la consulta. Los criterios de selección de consulta es igual para operaciones vigentes como históricas, y son: - Rut del Cliente - Código de Barra de la carpeta. - Folio Interno - Fecha desde – hasta - Estado Operación Para realizar consultas, no es necesario completar todos los criterios de selección. Una vez ingresado el o los criterios de	
--	--	--------------------	---	--

			selección, pinchar botón consultar. Sistema desplegará el detalle de los datos ingresados en Submenú Ingresar Operaciones.	
--	--	--	--	--

Jefe Operaciones

Tabla 5. 3 Plan de pruebas perfil Jefe de Operaciones

Prueba Nº	Menú	Submenú	Funcionalidad/Botones	Resultado
	Gestión de Imágenes			
1		Certificar Operaciones	Este Submenú permite certificar una operación que ha sido creada por el Escaneador y se encuentra en estado Ingresada. Al pinchar la funcionalidad Certificar Operaciones, se abre un menú que indica Listado Operaciones. La información que contiene este menú es la siguiente : - Fecha de Ingreso - Folio Interno, este dato contiene Link que permite revisar la operación al detalle, tal cual fue ingresada por el	

			escaneador. - Código Barra Carpeta - Estado de la Operación - Fecha de Estado - Sucursal de Origen - Nombre del Asistente de Negocios - Nombre del Cliente - Monto Bruto de la Operación - Tipo de Producto.	
2			Tal como fue informado en punto anterior, al pinchar Folio Interno, este dato contiene Link que permite revisar al detalle, operación ingresado por el escaneador.	
3			Verificar que nueva operación tenga folio interno, que es asignado en forma automática por el sistema GDI.	
4			En campo Código de Barra Carpeta, verificar código de la carpeta operativa	
5			En campo de Sucursal, indique la Sucursal del Escaneador, Independiente que el Asistente Negocios sea de Sucursal dependiente de la Sucursal	

			centralizadora.	
6			En campo Tipo de Producto, confirmar si corresponde el producto seleccionado.	
7			En campo Monto Bruto, verificar el monto que indica el Pagaré o los Pagarés cuando se trata de varios productos.	
8			Verificar el Rut del cliente.	
9			Verificar el nombre del cliente con los documentos de la carpeta.	
10			Verificar nombre del Asistente de Negocios, según opciones.	
11			Verificar que campo Fono Asistente, indique algún teléfono de comunicación.	
12			Verificar el nombre del Jefe de Operaciones indicado en sistema.	
13			Confirmar fecha de ingreso de operación en sistema	
14			Confirmar fecha de estado.	
15			Confirmar que el estado en que se encuentra la operación es Ingresada.	
16			Luego, debe pinchar botón Detalle, en el cual le indica las personas que han	

			intervenido en el proceso.	
17			A la vez, abajo de la información con las personas que han intervenido en el proceso, está el botón Ver Imágenes, donde muestra las imágenes que el escaneador ingresó.	
18			La función del Jefe de Operaciones, es revisar cada una de las imágenes ingresadas al sistema y dar la conformidad o rechazo de forma. Para tal efecto, Jefe de Operaciones, pincha botón volver y llega a la página que muestra el resumen de la operación.	
19			Jefe de Operaciones, tiene que pinchar en Certificación Aprobada, si la operación cumple con los controles establecidos en los puntos anteriores.	
20			En caso de no estar conforme, procede a pinchar Certificación Rechazada.	
21			Pincha el motivo de rechazo por el cual no autorizó la certificación.	
22			Pincha botón enviar, con lo cual la operación cambia de estado: de	

			Ingresada a Enviada.	
23			A partir de ese momento, puede ser vista por el Supervisor Control de Operaciones del CPN.	
24			El botón Grabar está orientado a que si por alguna circunstancia, no es posible completar el proceso en que se encuentra, se pincha este botón y guarda la operación en forma temporal.	
		Consultar Operaciones		
25			Este Submenú permite realizar consultas de las operaciones ingresadas al sistema. Las consultas se dividen en Vigentes e Históricas: Los criterios de selección de consulta es igual para operaciones vigentes como históricas, y son: - Rut del Cliente - Código de Barra de la carpeta. - Folio Interno - Fecha desde – hasta	

			- Estado Operación Para realizar consultas, no es necesario completar todos los criterios de selección. Una vez ingresado el o los criterios de selección, pinchar botón consultar. Sistema desplegará el detalle de los datos ingresados en Submenú Ingresar Operaciones.	
		Resumen Diario		
26			Este Submenú, permite consultar por totales , el estado de operaciones en que se encuentran y se detalla de la siguiente manera: - Número de Operaciones es el total de operaciones en sus distintas etapas. - Enviadas, corresponde al estado en el cual el Jefe de Operaciones, efectuó la Certificación Aprobada. - Asignadas CO, corresponde a operaciones que fueron asignadas por Supervisora de Control de	

			Operaciones a Administrativo de Control de Operaciones, para su análisis. - Control Aprobado, corresponde a operaciones que se encuentran aprobadas y listas para la siguiente etapa. - Control Rechazado, corresponde a operaciones que se rechazaron, por no cumplir con los controles establecidos. - Asignadas a Curse, corresponde a operaciones que por su forma de ingreso (Siscred), deben ser ingresadas en el CPN, a curse. - Asignadas a Visación, corresponde a operaciones que fueron cursadas en las Sucursales de origen, en donde el Supervisor de Control de Operaciones autoriza su visación en el CPN. - Curse Aprobados, corresponde a operaciones que fueron ingresadas (Cursadas) en el CPN y aprobadas por el Supervisor Control de	
--	--	--	--	--

			Operaciones. - Curses Rechazadas, corresponde a operaciones que fueron ingresadas (Cursadas) en el CPN y Rechazadas por el Supervisor Control de Operaciones. - Visado Aprobado, corresponde a todas las operaciones que cumplieron con las exigencias de Control y fueron Visadas por el Supervisor Control de Operaciones. - Visado Rechazado, corresponde a operaciones que si bien cumplieron con el Control, fueron rechazadas por otros motivos, al momento de ser Visadas, tal como error en el sistema al momento de procesar la información.	
--	--	--	--	--

Supervisor CPN

Tabla 5. 4 Plan de Pruebas perfil Supervisor CPN.

Prueba Nº	Menú	Submenú	Funcionalidad/Botones	Resultado
	Gestión de Imágenes			
		Operaciones Por Asignar	Este Submenú permite Asignar las Operaciones que se encuentran en estado Enviada a los Administrativos de Control de Operaciones.	
1			Aparecen solamente las operaciones que se encuentran con estado ENVIADA.	
2			Supervisor Pincha Link Folio Interno. El sistema muestra la operación, con todo el detalle que ingresó la Sucursal.	
3			Para asignar la Operación muestra un combo box, con el nombre de los Administrativos de Control.	
4			Puede el Supervisor pinchar botón Detalle cuya objetivo es consultar o comprobar el proceso de la operación que está viendo, más las imágenes que conforman la operación.	

5			Una vez efectuado la selección en el combo box, pinchar botón asignar y en forma automática la operación queda ingresada en el perfil del Administrativo de Control de Operaciones, donde esta acción le permite ver las imágenes para su Control.	
6		Rechazos Por Confirmar	Realizar el Proceso de Control de acuerdo a normativa vigente. Si producto del Control la operación no cumple con lo establecido. Pinchar Link Folio Interno, donde se desplegará el resumen de la operación a través de todos los datos ingresados por el escaneador. Completar opción en combo box, que indica los motivos de rechazos.	
7			El Supervisor pincha uno de los motivos indicados. A la vez, verifica que el motivo de rechazo aparezca en donde indica Observaciones Administrativo. Debe además completar en recuadro Observaciones, algún dato adicional	

			relativo al rechazo.	
8			Pinchar botón Confirmar Rechazo, en la situación que la operación no cumple con la normativa o bien algún problema de forma detectado en ese instante. En forma automática el sistema, envía mail al Asistente de Negocios, con copia al Jefe de Operaciones de la Sucursal Centralizadora, informando del rechazo ejecutado en el punto anterior.	
9			Al pinchar el botón Cambiar estado a Aprobado, se refiere a si por alguna circunstancia el Administrativo de Control, rechaza la operación, pero al llegar al Supervisor, esta tuvo solución, lo cual con esta funcionalidad permite realizar el cambio de estado y permitir seguir el proceso de curse y/o visado.	
10			Al pinchar el botón Detalle, indica las personas que han intervenido en el proceso, como también permite ver a través del botón Ver imágenes, todas las imágenes que el escaneador ingresó.	
11		Por Visar	Al ingresar en opción Por Visar Siscred,	

		Siscred	efectuar lo siguiente: Realizar el Proceso de Control de acuerdo a normativa vigente.	
12			Pinchar Link Folio Interno, donde se desplegará el resumen de la operación a través de todos los datos ingresados por el escaneador.	
13			Si operación cumple con Normativa, pinchar Visado Aprobado.	
14			Si operación no cumple con Normativa, o bien algún problema de forma detectado en ese instante pinchar Visado Rechazado.	
15			En el caso de Visado Rechazado, pinchar motivo del Rechazo indicado en combo box. Debe además completar en recuadro Observaciones, algún dato adicional relativo al rechazo.	
16			Al pinchar botón grabar, es para cuando la operación no ha podido concluir con todo su proceso, queda guardada en forma temporal en el sistema, hasta donde llegó con la revisión.	

17			Al pinchar el botón Detalle, indica las personas que han intervenido en el proceso, como también permite ver a través del botón Ver imágenes, todas las imágenes que el escaneador ingresó.	
18		Asignadas Sin resolver	Al ingresar en opción Asignadas sin Resolver, permite ver lo siguiente : Todas las operaciones que se encuentran asignadas y que no han sido resueltas, tal como Asignada a Control de Operaciones, Asignada Curse. Este informe, permite al Supervisor, tener una retroalimentación con el sistema y con las operaciones entregadas a los Administrativos que se encuentran sin resolución.	
19			Este reporte, al fin del día debe estar sin ninguna operación asignada y no resuelta. No tiene otra finalidad, mas que de informar.	
20		Cierre Ciclo	Al ingresar en opción Cierre Ciclo, realizar lo siguiente: Pinchar Link Folio Interno, donde se	

			desplegará el resumen de la operación a través de todos los datos ingresados por el escaneador.	
21			Efectuar revisión visual de todos los datos indicados por el sistema. Al estar conforme con todo lo indicado, pinchar botón Cierre Ciclo.	
22			En forma automática, el sistema informa al Jefe de Operaciones a través de la opción Resumen Diario de su perfil asignado. El resultado de la operación enviada vía Imágenes.	
		Consultar Operaciones	Este Submenú permite realizar consultas de las operaciones ingresadas al sistema. Las consultas se dividen en Vigentes e Históricas.	
23			Los criterios de selección de consulta es igual para operaciones vigentes como históricas, y son: - Rut del Cliente - Código de Barra de la carpeta. - Folio Interno - Fecha desde – hasta	

			- Estado Operación Para realizar consultas, no es necesario completar todos los criterios de selección. Una vez ingresado el o los criterios de selección, pinchar botón consultar. El sistema desplegará el detalle de los datos ingresados en Submenú Ingresar Operaciones.	
24		Resumen Diario	Este Submenú, permite consultar por totales , el estado de operaciones en que se encuentran y se detalla de la siguiente manera: Indica el nombre de la Sucursales que operan con Imágenes el día en cuestión.	
25			Número de Operaciones es el total de operaciones en sus distintas etapas.	
26			Enviadas, corresponde al estado en el cual el Jefe de Operaciones, efectuó la Certificación Aprobada.	
27			Asignadas CO, corresponde a operaciones que fueron asignadas por Supervisora de Control de Operaciones a Administrativo de Control de	

			Operaciones, para su análisis.	
28			Control Aprobado, corresponde a operaciones que fueron aprobadas. Separadas por Siscred y Plataforma Universal (PU)	
29			Control Rechazado, corresponde a operaciones que fueron rechazadas, por no cumplir con los controles establecidos. Separadas por Siscred y Plataforma Universal (PU). En esta opción, permite efectuar modificar el rechazo, dado que estos se encuentran con Link, y desde ahí otro Link con el folio Interno , donde permite al Supervisor de Control de Operaciones cambiar de estado de Visado Rechazado a Visado Aprobado.	
30			Curse Aprobado, corresponde a operaciones, que por su forma de ingreso fueron ingresadas en el CPN	
31			Curse Rechazado, corresponde a operaciones, que fueron Rechazadas por el CPN.	
32			Visado Aprobado, corresponde a operaciones que fueron cursadas en las	

			Sucursales de origen, en donde el Supervisor de Control de Operaciones autoriza su visación en el CPN. Separadas por Siscred y Plataforma Universal (PU).	
33			Visado Rechazado, corresponde a operaciones que si bien cumplieron con el Control, fueron rechazadas por otros motivos, al momento de ser Visadas, tal como error en el sistema al momento de procesar la información. Separadas por Siscred y Plataforma Universal (PU).	

Administrativo CPN

Tabla 5. 5 Plan de pruebas del perfil Administrativo CPN

Prueba Nº	Menú	Submenú	Funcionalidad/Botones	Resultado
	Gestión de Imágenes			
		Operaciones Asignadas	Este Submenú permite al Administrativo, registrar el resultado del proceso de Control de Operaciones, a todas aquellas que se encuentran en estado Asignado CO.	
1			Aparecen solamente las operaciones	

			que se encuentran con estado Asignado CO.	
2			Administrativo Pincha Link Folio Interno. El sistema muestra la operación, con todo el detalle que ingresó la Sucursal.	
			Pinchar en Tipo, por que vía fue ingresada la Operación.	
3			Si es por Plataforma Universal, viene Cursada. El CPN, realiza solamente la visación. Si la operación está conforme con Normativa , pinchar Control Aprobado.	
4			Si la operación no está conforme con Normativa, pinchar Control Rechazado. Completar opción en combo box, que indica los motivos de rechazos. Administrativo pincha uno de los motivos indicados. Debe además completar en recuadro Observaciones, algún dato adicional relativo al rechazo	
5			Al pinchar botón grabar, es para cuando la operación no ha podido concluir con todo su proceso, queda guardada en	

			forma temporal en el sistema, hasta donde llegó con la revisión.	
6			Puede el Administrativo pinchar botón Detalle cuya objetivo es consultar o comprobar el proceso de la operación que está viendo, más las imágenes que conforman la operación.	
7		Curse/Visación	Aparecen solamente las operaciones que se encuentran con estado Asignada Curse y Asignada Visación.	
8			Pinchar Link Folio Interno, donde se desplegará el resumen de la operación a través de todos los datos ingresados por el escaneador.	
9			Administrativo pincha Link Folio Interno. El sistema muestra la operación, con todo el detalle que ingresó la Sucursal. Si la operación pasó conforme el proceso de Curse o Visación, pinchar Curse Aprobado.	
10			Si la operación, no pasa conforme el proceso de Curse o Visación, pinchar Curse Rechazado. Completar opción en combo box, el	

			motivo de rechazo. Administrativo pincha uno de los motivos de rechazo. Debe además completar en recuadro Observaciones, algún dato adicional relativo al rechazo.	
11			Al pinchar botón grabar, es para cuando la operación no ha podido concluir con todo su proceso, queda guardada en forma temporal en el sistema hasta donde llegó con el ingreso de datos.	
12			Puede el Administrativo pinchar botón Detalle, cuyo objetivo es consultar o comprobar el proceso de la operación que está viendo, más las imágenes que conforman la operación.	
13		Control Expost	Al ingresar en opción Control Expost , debe efectuar lo siguiente: Realizar el Proceso de Control Expost de acuerdo a normativa vigente.	
14			Aparecen solamente las operaciones que se encuentran con estado Visado Aprobado.	
15			Administrativo pincha Pinchar Link Folio	

			Interno. El sistema muestra la operación con todo el detalle que ingresó la Sucursal.	
16			El Control Expost se realiza en dos etapas: La primera etapa, tiene relación en confirmar los documentos originales con las imágenes enviadas sean iguales. Si los documentos están conformes, en Control Imagen pinchar Aprobado. Si los documentos no están conformes, en Control Imagen pinchar Rechazado. Completar opción en combo box, que indica los motivos de rechazos. El administrativo pincha uno de los motivos indicados.	
17			La Segunda etapa , tiene relación en realizar Control de Operaciones a los demás documentos no enviados por Imágenes. Si la revisión está conforme, ir a Control Operación y pinchar Aprobado. Si la revisión no está conforme ir a Control Operación y pinchar Rechazado.	

			Completar opción en combo box, que indica los motivos de rechazos. Administrativo pincha uno de los motivos indicados.	
18			Para ambas etapas, además debe completar en recuadro Observaciones, algún dato adicional relativo a los rechazos.	
19			Al pinchar botón grabar, es para cuando la operación no ha podido concluir con todo su proceso, queda guardada en forma temporal en el sistema hasta donde llegó con el ingreso de datos.	
20			Puede el Administrativo pinchar botón Detalle, cuyo objetivo es consultar o comprobar el proceso de la operación que está viendo, mas las imágenes que conforman la operación	
		Consultar Operaciones	Este Submenú permite realizar consultas de las operaciones ingresadas al sistema. Las consultas se dividen en Vigentes, Históricas y del Día.	
21			Los criterios de selección de consulta	

			para ambos casos son: - Rut del Cliente - Código de Barra de la carpeta. - Folio Interno - Fecha desde – hasta - Estado Operación	
22			Para realizar consultas, no es necesario completar todos los criterios de selección. Una vez ingresado el o los criterios de selección, pinchar botón consultar.	
23			Sistema desplegará el detalle de los datos ingresados en Submenú Ingresar Operaciones.	
		Del día	- Enviadas, corresponde al estado en el cual el Jefe de Operaciones, efectuó la Certificación Aprobada. - Asignadas CO, corresponde a operaciones que fueron asignadas por Supervisora de Control de Operaciones a Administrativo de Control de Operaciones, para su análisis. - Control Aprobado, corresponde a	

			operaciones que fueron aprobadas. Separadas por Siscred y Plataforma Universal (PU) - Control Rechazado, corresponde a operaciones que fueron rechazadas, por no cumplir con los controles establecidos. Separadas por Siscred y Plataforma Universal (PU). En esta opción, permite efectuar modificar el rechazo, dado que estos se encuentran con Link, y desde ahí otro Link con el folio Interno , donde permite al Supervisor de Control de Operaciones cambiar de estado de Visado Rechazado a Visado Aprobado. - Curse Aprobado, corresponde a operaciones, que por su forma de ingreso fueron ingresadas en el CPN - Curse Rechazado, corresponde a operaciones, que fueron Rechazadas por el CPN - Visado Aprobado, corresponde a operaciones que fueron cursadas en las Sucursales de origen, en donde el	
--	--	--	--	--

			Supervisor de Control de Operaciones autoriza su visación en el CPN. Separadas por Siscred y Plataforma Universal (PU). - Visado Rechazado, corresponde a operaciones que si bien cumplieron con el Control, fueron rechazadas por otros motivos, al momento de ser Visadas, tal como error en el sistema al momento de procesar la información. Separadas por Siscred y Plataforma Universal (PU).	
--	--	--	---	--

CAPITULO 6: IMPLANTACIÓN DE LA SOLUCIÓN INFORMÁTICA EN

PRODUCCIÓN

PUESTA EN PRODUCCIÓN DE LA SOLUCIÓN

Para una aplicación DNA 2000, el proceso de paso a producción en Banco Estado comienza con la aprobación por parte del área usuaria de la aplicación en cuanto a las funcionalidades que fueron comprometidas, al cumplimiento de los objetivos deseados por parte de la aplicación, etc. Normalmente esto se logra luego de un proceso de pruebas exhaustivas de la aplicación por parte del área usuaria. Posteriormente, esta aprobación es formalizada mediante el envío de un formulario tipo de aprobación, el formulario ha de seguir el siguiente formato:

	CORREO DE REQUERIMIENTO Y CERTIFICACION			Fecha 01/01/2006
INFORMACION DE LA APLICACIÓN				
NOMBRE DE APLICACION				
SUB-APLICACIÓN / MODULO / FUNCIONALIDAD (si aplica)				
PLATAFORMA (Indicar con una X)	CLIENTE/SERVIDOR	HOST	DNA X	
A UNTO	Paso a producción			
UUARIO CERTIFICADOR	Nombre del Usuario Certificador			
DESCRIPCION DEL REQUERIMIENTO INICIAL				
En esta sección se explica someramente en que consiste la aplicación.				
SATISFACCION DEL REQUERIMIENTO				
(Indicar con una X)				

SI	NO	OBSERVACIONES (Si aplica) :	
X			
APRUEBA PASO A PRODUCCION (Indicar con una X)		SI X	NO

Tanto en el proceso de desarrollo de la aplicación, como en el posterior paso a producción de esta, los equipos de desarrollo son apoyados por el área de Soporte al Desarrollo (SDM), la cual, mediante la implementación de un sitio Web ad-hoc interactúa con los equipos de desarrollo permitiendo acceder al manejador de versiones CCC Harvest, solicitar pruebas de estrés, modificaciones en los componentes COM+, etc. En el paso a producción, el área de SDM provee a la aplicación de una herramienta llamada Matriz de Componentes Electrónica (MCE) la cual identifica tanto los componentes COM+, paginas ASP, paginas HTML, etc. involucrados en una aplicación en particular y sus archivos de código fuente en CCC Harvest.

A continuación, la aplicación es presentada al área de Testing y Aseguramiento de la Calidad, la cual validará si es necesario el someterla a pruebas de estrés. Estas pruebas de estrés tienen como objetivo el asegurar que la aplicación una vez instalada en producción, podrá soportar la cantidad de usuarios y el número de transacciones proyectados. Para estas pruebas de estrés, se utiliza la herramienta WebStress. En esta herramienta, ha de definirse primeramente una estructura de navegación por la aplicación, la cual normalmente define los pasos necesarios para el desarrollo de una funcionalidad tipo. Esto da origen a un guión (script) el cual será utilizado posteriormente por la herramienta para estresar la aplicación y sus objetos. Además, es necesario definir el o los usuarios que serán personificados por la herramienta de estrés para el desarrollo del guión de estrés. Como previamente se ha definido el nivel de carga que ha de soportar la aplicación, el estresamiento consiste en ejecutar el guión de estrés, con el o los

usuarios definidos, tantas veces como se ha definido sobre la aplicación instalada en el ambiente de pruebas de producción. A continuación, y una vez terminan las pruebas de estrés sobre la aplicación, el área de Soporte al Desarrollo (SDM) elabora un informe de resultados de las pruebas de estrés, el cual sigue el siguiente formato.

Informe de Pruebas de Stress

Detalle de Solicitud de Stress

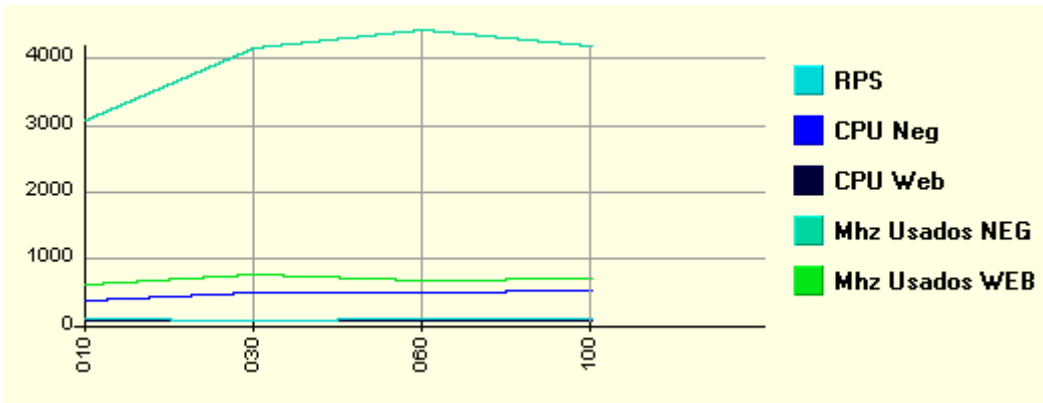
- » **Fecha** 01/01/2006
- » **aplicación** Nombre de la Aplicación
- » **M.C.E** 5969
- » **Certificador** Nombre del Usuario Certificador
- » **Observaciones**

Detalle de Asignación de Stress

- » **Fecha** 01/01/2006 - 10:00
- » **Encargado** Nombre del Encargado
- » **Resultado** APROBADO
- » **Observaciones** *
- » **Conclusiones** *

Detalle Cuadro Estadístico

	Items	10 Usr	30 Usr	60 Usr	100 Usr
1	RPS	10,11	9,47	9,69	9,54
2	CPU Neg	38,97	49,09	50,00	53,84
3	CPU Web	7,82	9,17	8,30	8,40
4	Mhz Usados NEG	308,37	414,66	444,50	419,29
5	Mhz Usados WEB	61,84	77,42	68,52	70,44



Posteriormente, como normalmente las aplicaciones son de las llamadas “de la Intranet Aplicativa”, han de seguir su camino hacia la shell integrada (Plataforma Universal) donde conviven todas las aplicaciones DNA del Banco. De otro modo, si la aplicación no representa en si mismo una aplicación relativa estrictamente al negocio del Banco, residirá en un área de servidores llamados de difusión, los cuales si bien es cierto están sometidos a control para asegurar su continuidad operacional, este monitoreo no es tan estricto como con los servidores orientados a las aplicaciones relacionadas con el negocio del Banco pues son menos críticas para el negocio.

Como se mencionó anteriormente, las aplicaciones conviven dentro de la Plataforma Universal, y la Plataforma Universal a su vez basa su interacción en el llamado Sistema de Privilegios, el cual mediante entradas en una Base de Datos define las aplicaciones o partes de una aplicación que estarán disponibles para los usuarios. Para definir como una aplicación en particular ha de interactuar con el Sistema de Privilegios, se define lo que se conoce como Matriz de Privilegios, la cual es concordada tanto por el equipo de desarrollo como por el área Usuaría y el área de Administración de Privilegios.

Antes de dar el vamos al paso a producción, es necesario realizar una presentación a las áreas involucradas en la explotación del sistema, es decir Administración de Redes, la Mesa de Atención de Usuarios (MAU), Administración de la Producción (explotación) y Administración de Cambios. En esta presentación ha de explicarse brevemente la funcionalidad de la aplicación, la o las aplicaciones relacionadas, un diagrama de la arquitectura de la aplicación, los perfiles que se han definido para la aplicación, las áreas usuarias involucradas y una breve navegación por el sistema.

A continuación, se coordina el paso a producción propiamente tal a través del área de Administración de Cambios, la cual tomara la Matriz de Componentes Electrónicos, los archivos fuentes de la aplicación y los componentes compilados previamente para realizar el paso a producción.

IMPLANTAR LA SOLUCIÓN EN SUCURSALES PILOTO Y CPN

Se eligieron como sucursales piloto, las siguientes oficinas:

- Arica
- Iquique
- Punta Arenas

Esta elección se basó en la cantidad de operaciones que estas sucursales debían atender mensualmente y a que corresponden a oficinas extremas, lo que dificultaba el despacho de la correspondencia vía valija.

Para implementar la solución en Sucursales y en CPN se siguieron los siguientes pasos:

1. Se crearon manuales de usuario para cada perfil involucrado en el sistema, dichos manuales debe contener:

En Sucursales:

- Cómo es el ingreso al sistema (mediante plataforma Universal, escaneador y jefe de operaciones).
- Cómo se ingresa una operación (escaneador).
- Parámetros que debe contener la pantalla de Ingreso de una operación (escaneador).
- Cómo digitalizar una imagen (escaneador).
- Cómo se finaliza el ingreso y posterior envío al Jefe de Operaciones (escaneador).
- Visualización de Operaciones ingresadas (jefe de operaciones).
- Certificación o rechazo de Operaciones (jefe de operaciones).

- Envío de operaciones a CPN.
- Realizar Consultas (escaneador y jefe de operaciones)
- Realizar el resumen Diario (jefe de operaciones)

En CPN:

- Cómo se Ingresa al sistema (Supervisor y Administrativo).
- Visualización de menú Operaciones por asignar (supervisor).
- Visualización de menú Operaciones para confirmar los rechazos (supervisor).
- Visualización de Operaciones por Visar (supervisor).
- Visualización de menú de operaciones que fueron asignadas y no han sido resueltas (supervisor).
- Visualización de menú Cierre Ciclo (supervisor).
- Visualización de menú consultar operaciones (administrativo y supervisor).
- Visualización de menú Resumen Diario (supervisor).
- Visualización de menú Controlar operaciones (administrativo).
- Visualización de menú Curse/Visación (administrativo).
- Visualización de menú Control expost (Administrativo).

2. Se preparó un procedimiento de implantación del proceso en sucursales y CPN, el cual se detalla a continuación:

Actividades del implantador.

Tabla 6. 1 Actividades a realizar por el implantador en sucursales

DIA	Nº	ACCION
1	1	Presentación al Agente y jefe de operaciones.
	2	Confirmar que escáner este operativo y que cuente con los parámetros de memoria RAM y GB.
	3	Revisar perfil de plataforma universal- imágenes, opción gestión.
	4	Revisar perfil de plataforma universal – créditos.
	5	Realizar reunión de capacitación a plataforma comercial, detallando las funciones que tiene cada perfil.
	6	Revisión a los demás sistemas que la subgerencia ha implantado, como son: sistema único de normas, control flujo documental, sistema de normalización, mesa de ayuda y plataforma universal. En los productos líneas de créditos y tarjetas de créditos.
	7	Hacer un resumen diario de lo realizado y enviar a jefe de proyecto.
2	8	Participar con cada asistente de negocios en el curso de las operaciones por plataforma universal - créditos, como también las operaciones que se desarrollan en los otros sistemas (siscred, PU, pantalla administrativa)
	9	Desarrollar con cada asistente de negocios el proceso de envío de operaciones vía imágenes, enviando mail y registrando carpeta en cfd.
	10	Confirmar que la digitalización de los documentos (títulos ejecutivos), esté correcta en forma y fondo.
	11	Participar con el jefe de operaciones, de la recepción de los mail de los asistentes de negocios y revisión de operaciones en GDI.
	12	Estar en forma permanente con el funcionario encargado de correspondencia y de envío de imágenes, para apoyarlo en el proceso.

	13	Presentar contingencia en el caso de caída del sistema de PU - créditos e imágenes.
3	14	Verificar abonos de visación de imágenes del CPN al día siguiente.
	15	Emitir informe final con conclusión de la implantación.
	16	Pedir al agente realice evaluación de implantación final.

Actividades del Técnico IBM

Se recurre a un técnico en terreno, para coordinar las actividades que no es posible realizar en forma remota. El Banco cuenta con una red de técnicos provistos por la empresa IBM para resolver este tipo de necesidades en su red de sucursales. Una vez instalado el escáner, se procede a realizar la siguiente prueba:

Tabla 6. 2 Actividades a realizar por el técnico IBM

Nº	Acción
1	Ingresar por menú Inicio.
2	Seleccionar Programas.
3	Seleccionar Internet Explorer.
4	En la Dirección del portal Intranet, escribir srv_plataforma
5	Pinchar vínculo (pestaña) superior Gestión.
6	Pinchar GESTIÓN DE IMÁGENES.
7	Seleccionar NUEVA OPERACIÓN.
8	Anotar en Nº. de folio, para usarlo al envío de la imagen.
9	Indicar en campo observaciones que se trata de una prueba.

10	Pinchar adjuntar imágenes.
11	Colocar documento o papel en escáner.
12	Seleccionar capturar.
13	Aparecerá una pantalla propia del software del escáner donde se pre-visualiza la imagen, en esa pantalla hay que seleccionar la imagen, y luego poner en la opción 2 del menú del escáner ESCALA DE GRISES, luego seleccionar opción 4 del menú que aparece en el lado izquierdo.
14	Seleccionar volver.
15	Para revisar si la imagen está digitalizada se debe hacer lo siguiente: con la opción volver seleccionada en el punto 14, se retorna a la pantalla donde se crea la “carpeta electrónica”, en esa pantalla seleccionar botón DETALLE, y luego en esa pantalla seleccionar botón VER IMÁGENES, en esta pantalla se podrán visualizar las imágenes digitalizadas.
16	Luego de finalizada la prueba se debe informar del resultado al jefe del departamento de instalaciones.

El proceso de implantación se puede entender más claramente a través de la siguiente bitácora:

Día 1: Sucursal

- ❖ Revisión con Técnico de IBM de instalación correcta de escáner y PC.
- ❖ Capacitación en la Sucursal Modelo de Curse con Imágenes, Procedimiento detallado y herramienta para digitalización, envío y certificación de imágenes:
 - o Definición Operaciones a enviar vía imágenes.
 - o VºBº documentos a digitalizar por Asistente de Negocios.
 - o Entrega documentos a Administrativo de Correspondencia para Digitalización.

- o Digitalización de Documentos por Administrativo de Correspondencia.
- o Envío de Imágenes vía Herramienta de Apoyo por Administrativo de Correspondencia.
- o Certificación de imágenes vía herramienta por Jefe de Operaciones.
- o Respuesta del Centro de Procesos con Aprobaciones y Rechazos.
- o Reenvíos si son requeridos.

Participantes: Asistente de Negocios (1), Jefe de Operaciones, Administrativo de Correspondencia

Día 1: Centro de Procesos – CPN

- ❖ Seguimiento del Proceso.
- ❖ Soporte presencial a dotación designada para curse y visación con imágenes.
- ❖ Revisión de Controles.
- ❖ Apoyo en el Proceso de prueba para acciones de control, curse y visación
- ❖ Cierre de proceso diario y comunicación a Sucursal.

Día 2: Sucursal.

- ❖ Continuación capacitación y apoyo, Envío de imágenes, verificación de Curse de la Operación en Centro de Procesos Masivo, de acuerdo a estándar definido (medición de tiempos).
- ❖ Envío documentación física para controles ex post.
- ❖ Seguimiento proceso de envío de imágenes ambiente controlado.

Participantes: Asistente de Negocios (2), Jefe de Operaciones

Día 2: Centro de Procesos – CPN

- ❖ Seguimiento del Proceso.
- ❖ Soporte presencial a dotación designada para curse y visación con imágenes.
- ❖ Revisión de Controles.
- ❖ Apoyo en el Proceso de prueba para acciones de control, curse y visación.
- ❖ Cierre de proceso diario y comunicación a Sucursal.

Día 3: Sucursal

- ❖ Continuación capacitación y apoyo a la Sucursal, Envío de imágenes, verificación de Curse de la Operación en CPN, de acuerdo a estándar definido (medición de tiempos).
- ❖ Envío documentación física para controles ex post.
- ❖ Seguimiento proceso de envío de imágenes ambiente controlado.

Participantes: Asistente de Negocios (3), Jefe de Operaciones, Administrativo de Correspondencia

Día 3: Centro de Procesos – CPN

- ❖ Revisión llegada carpeta física día 1 y Control del Flujo de Procesos posterior
- ❖ Seguimiento del Proceso.
- ❖ Revisión de Controles.
- ❖ Envío de Nómina de Recepción de Imágenes.

- ❖ Apoyo en el Proceso de prueba para acciones de control, curse y visación.
- ❖ Cierre de proceso diario y comunicación a Sucursal.

Día 4 y 5: Sucursal

- ❖ Capacitación en Modelo de Curse con Imágenes, Procedimiento detallado y herramienta de Apoyo al total de los Asistentes de Negocios de la Sucursal.
- ❖ Ejecución Proceso completo desde el envío de la operación hasta la verificación del curse y visación de la operación en el CPN.

Participantes: Grupo completo de Asistentes de Negocios de la Sucursal, Jefe de Operaciones, Administrativo de Correspondencia.

Día 4 y 5: Centro de Procesos – CPN

- ❖ Revisión llegada carpeta física día 1 y Control del Flujo de Procesos posterior.
- ❖ Seguimiento del Proceso.
- ❖ Revisión de Controles.
- ❖ Apoyo en el Proceso de prueba para acciones de control, curse y visación
- ❖ Cierre de proceso diario y comunicación a Sucursal.

El material de apoyo con el cual se realizó la implantación y capacitación fue:

- ❖ Manuales de GDI.
- ❖ Descriptivo Sucursal (documento explicativo del proceso a realizarse en sucursales).
- ❖ Presentación power point de apoyo.

- Se realizan las pruebas de funcionalidad del sistema, mediante el ingreso de operaciones y seguimiento de éstas hasta CPN.

Estas pruebas están claramente explicadas en la bitácora anteriormente expuestas.

SEGUIMIENTO DEL PROCESO DE CURSE IMPLANTADO.

Para el seguimiento de proceso de curse de operaciones – sistema informático GDI – se solicitó la ayuda de la Mesa de Atención de Usuarios (MAU), la cual interactuando con los usuarios definidos y establecidos para el sistema preparó una lista de preguntas frecuentes para satisfacer las necesidades de sucursales y CPN. Estas se exponen en la siguiente tabla:

Tabla 6. 3 Preguntas frecuentes para GDI - MAU

Nombre	Pregunta Frecuente	Solución
Créditos - Cursación Modalidad Vía Imágenes.	¿Qué productos operan bajo la Modalidad Vía Imágenes?	Banca de Personas: Créditos de Consumo, Créditos Universitarios, Apertura de Líneas de Crédito, Activación de TC para montos Superiores a \$1.000.000.-, Combos, Créditos Comerciales, Créditos Comerciales con Garantía
		FOGAPE

Créditos - Curse y Visación Oficinas en Modalidad Vía Imágenes	¿Cuáles son las Oficinas autorizadas para lograr curse y visación en Modalidad Vía Imágenes?	Modalidad 1: Son aquellas sucursales donde la recepción de la documentación física en CPN demore más de 24 horas y que califiquen en categoría de Sucursal Crítica (Arica, Iquique, Punta Arenas).
		Modalidad 2: Sucursal Centralizadora actúa como centralizadora de Sucursales Satélites.
		Sucursales Satélites envían Carpetas con Operaciones a Digitalizar a su Sucursal Centralizadora y ésta envía operaciones vía imágenes al CPN (Punta Arenas 18 de Septiembre, Puerto Natales y Puerto Porvenir centralizan en Punta Arenas)
Créditos - Documentación exigida para cursar operaciones en Modalidad Vía Imágenes	¿Cuál es la documentación exigida para cursar operaciones con imágenes?	Toda la documentación exigida para los diferentes productos se puede visualizar en el SUN - PROCESOS CENTRALIZADOS-Documentación exigida para Cursar Operación con Imágenes.
Créditos - Envío de Carpetas con desfase Modalidad Vía Imágenes	¿Qué pasa si la Sucursal No envió la carpeta al día hábil siguiente de enviada la Imagen y la operación ya	La Sucursal tiene un plazo máximo compuesto de 3 días hábiles para enviar la carpeta y si al 5º día hábil CPN no recibe la Carpeta, no se cursa operación con imágenes para esa sucursal, hasta que llegue la carpeta física.

	está cursada?	
Créditos - Envío de Carpetas en Modalidad Vía Imágenes	¿Cuándo deben enviarse las Carpetas al CPN?	Las Carpetas deben enviarse al día hábil siguiente al envío de la Operación por Imágenes, previa verificación de Operación Cursada y Visada por el Asistente de Negocios de la Sucursal
Créditos - Estándar de Servicio CPN en Modalidad Vía Imágenes	¿Cuál es el estándar de servicio del CPN en esta Modalidad?	Para Operaciones simuladas en SISCREDA. Las operaciones son cursadas y visadas antes de las 14 horas del día hábil siguiente al envío de la operación vía imágenes.
		Para Operaciones de créditos ingresadas por Plataforma Universal. Las Operaciones ingresadas en la Sucursal por PU, se visan en el CPN el mismo día de envío de las imágenes.
Créditos - Participantes en Sucursal en el proceso Modalidad Vía Imágenes	¿Quiénes participan en la Sucursal este proceso?	Los Asistentes de Negocios, Jefe de Operaciones y el Encargado de Correspondencia.
Créditos - Proceso de Carpeta Física en CPN Modalidad	Qué pasa con la Carpeta Física cuando llega al CPN?	Se hace un control ex - post compuesto por dos controles: Que los documentos originales de la operación sean exactamente iguales a las imágenes enviadas.

Vía Imágenes		Se hace el proceso de control de operaciones y documental a todos los documentos. Si ambos controles están OK, la carpeta va a Control de Salida para luego ser ingresados en la Custodia Central
Créditos - Recepción de Rechazos Operación en Sucursal Modalidad Vía Imágenes	¿Cómo se reciben los rechazos en la Sucursal?	La Sucursal recibe los rechazos de la operación a través de Correo Electrónico directo al Asistente de Negocios con copia al Jefe de Operaciones. Los rechazos son "virtuales" ya que se tiene las imágenes y con éstas se cursará y visará la operación.
Créditos - Rechazos de Operación Modalidad Vía Imágenes	¿Puede haber rechazos en este proceso?	Sí. Los rechazos de la operación pueden ocurrir en algunas de estas fases del proceso: Control de Operaciones de las imágenes Curse y/o Visación de las imágenes Control ex - post a la Carpeta Física
Créditos - Aplicación que soporta al Proceso Modalidad Vía Imágenes	¿Qué Aplicación soporta al Proceso de Envío y Recepción de Imágenes	La Aplicación se llama GDI, Gestión de Imágenes. Está Aplicación apoya al proceso desde el ingreso de la operación en la sucursal hasta que es recepcionada en el CPN, incluyendo el registro y seguimiento completo de la operación.

Paralelo a este set de preguntas frecuentes se presentaron informes de cómo fue el proceso de implantación en sucursales y en CPN. Uno de estos informes puede ser revisado en el anexo 3.

CAPITULO 7: ESTABLECIMIENTO DE NUEVOS PARÁMETROS DE MEDICIÓN

Una vez terminado el proceso de implantación en las sucursales piloto se procedió a implantar el sistema en otras sucursales, tanto centralizadoras como satélites.

Las sucursales que se implantaron posteriormente fueron:

Tabla 7. 1 Sucursales implementadas con GDI

Sucursal centralizadora	Sucursal Satélite
ARICA	Arica Chichorro
	Putre
IQUIQUE	Iquique Zofrí
	Iquique Cavancha
	Iquique Alto Hospicio
CHAITEN	Futalelfú
	Alto Palena
CASTRO	Dalcahue
	Achao
	Chonchi
	Quellón
ANCUD	Opera sin satélites
COYHAIQUE	Coyhaique Río Simpson
	Puerto Aysen
	Chile Chico
	Cochrane
PUNTA ARENAS	Punta Arenas 18 de Septiembre
	Puerto Porvenir
	Puerto Natales
OVALLE	Montepatria

	Ovalle Benavente
LA SERENA	La Serena Huanhualí
	La Serena La Recoba
	La Serena Las Compañías
COQUIMBO	Coquimbo Tierras Blancas
COPIAPO	Copiapó Los Carrera
	Copiapó El Palomar
PUERTO MONTT	Puerto Montt presidente Ibáñez
ANTOFAGASTA	Costanera Sur
	Angamos
	Latorre
	La Portada
CALAMA	Chuquicamata
	Calama Diego Portales
PUERTO VARAS	Frutillar
OSORNO	Osorno Patricio Lynch
LA UNIÓN	
RIO BUENO	
PAILLACO	
VALDIVIA	Valdivia Aguirre Cerda
	Valdivia Los Torreones
TEMUCO	Temuco Av. Alemania
	Nueva Imperial
	Padre las Casas

El procedimiento de implantación fue el mismo descrito en el capítulo anterior, no se realizaron cambios y la bitácora fue la misma en todas las sucursales.

RECOPIACIÓN DE INFORMACIÓN SOBRE EL FUNCIONAMIENTO DEL SISTEMA

El área usuaria de este sistema – Ingeniería de Procesos y Gerencia de Sucursales – preparó una encuesta de satisfacción sobre el material entregado y la capacitación dada en sucursales. Con esta iniciativa se pretende detectar posibles falencias que pudieran presentar los usuarios finales del sistema sobre el comportamiento y funcionamiento de GDI.

A continuación se presenta un extracto de dicha encuesta:

Tabla 7. 2 Encuesta a usuarios de GDI

Pregunta	Tabulación de Respuesta de Asistentes.
1. El material entregado en la capacitación le pareció: A: Muy completo B: Adecuado C: Insuficiente	A: 50% B: 50% C: 0
2. La exposición del relator(a) le pareció: A: Muy clara B: Clara C: Poco Clara D: Deficiente	A: 80% B: 20% C: 0 D: 0
3. La capacitación a su juicio le proporcionó los elementos para satisfacer consultas de usuarios en un grado: A: Alto. B: Adecuado. C: Escaso	A: 30% B: 70% C: 0
4. A su juicio requiere profundizar mas en las materias expuestas: SI (precisar temas) NO	SI: 20% (no se precisaron temas) NO: 80% Precisiones: 1: Muy clara exposición de la relatora 2: Relatora explicó muy bien 3: Exposición clara y completa

Adicionalmente se recopiló información de los tiempos de respuesta involucrados en el proceso e rse de operaciones que se registraron una vez implementado el sistema en sucursales.

Tabla 7. 3 Tiempos involucrados en el proceso de curse de operaciones antes y después del GDI

Nombre Actividad en Sucursales	Antes	Tiempo estimado GDI	Tiempo real GDI	Antes	Tiempo estimado GDI	Tiempo real GDI	Tiempo Optimizado estimado	Tiempo Optimizado real
DIGITALIZACIÓN Y CERTIFICACIÓN FORMA				7	5,5		1,5	
REGISTRO CARPETA EN CFD				7	7			
REGISTRO OPERACIÓN EN PLANILLA	2	0					1	
CERTIFICACIÓN DE OPERACIÓN CON IMAGEN	3	3					0	
CUADRE CIERRE DIA ENVIO Y RECEP. CIERRE DIARIO	1	0					1	
Total Actividades	6	3		14	12,5		3,5	
Tiempo Total Actual	20							
Nuevo Tiempo Total estimado	15,5							
% de Optimización	23%							

Todos los datos expuestos son expresados en minutos.

Tabla 7. 4 Tiempos de respuesta en CPN antes y después de implementarse el GDI

Nombre Actividad en CPN	Tiempo Antes	Tiempo estimado GDI	Tiempo real GDI	Tiempo Antes	Tiempo estimado GDI	Tiempo Real GDI	Dif. Tiempo Optimizado	Tiempo optimizado real
RECEP. OP. Y ASIGNAC. DE ADM. CONTROL OP. Y REG OP. PLANILLA CONTROL	1,22	0,5					0,72	
RECEPCIÓN DE OPERACIÓN E IMPRESIÓN				2	1,3		0,7	
PROCESO CONTROL OPERAC. SOBRE IMÁGENES IMPRESAS				3	3		0	
REGISTRO ESTADO OP. A TRAVÉS DE PLANILLA				1	0,1		0,9	
RECEPCIÓN PLANILLA SUC. Y CIERRE	0,15	0					0,15	
DEVOLUCIÓN PLANILLA ACEPT. A SUC. VÍA MAIL CON EST. OPERAC.	0,45	0,1					0,35	
CURSE DE LA OPERACIÓN				1,3	1,03		0,27	
REGISTRO ESTADO OP. POR EL PROCESO CURSE				1	0,1		0,9	
VISACIÓN EN SISTEMA PRODUCTO (SISCRED)	1,3	1,3					0	
VISACIÓN EN SISTEMA PRODUCTO (PU)				1,3	1,3		0	
REGISTRO ESTADO VISAC. EN PLANILLA	1	0,1					0,9	
Total Actividades Minutos	4,12	2		9,6	6,83		4,89	
Tiempo Total Actual	13,72							
Nuevo Tiempo Total estimado	8,83							
% de Optimización	36%							

Todos los datos expuestos son expresados en minutos.

Tabla 7. 5 Volúmenes de operaciones cursadas con GDI

								Volumen Base Datos (anual)		
Piloto	Sucursal	Créditos	Líneas de Crédito	Prom. diario	Usuario de Apc.	Tráfico diario (byte)	Horario	Imagen (Kb)	Data (Kb)	Indice (kb)
Si	Punta Arenas	206	20	11	3	32.385.024	14:00 - 18:00	2.087.316	894.564	298.188
Si	Arica	318	16	17	3	47.810.560	14:00 - 18:00	3.081.540	1.320.660	440.220
Si	Iquique	260	24	14	3	40.678.400	14:00 - 18:00	2.621.850	1.123.650	374.550
No	María Elena	25	8	2	2	3.505.152	14:00 - 18:00	301.224	129.096	43.032
No	Tal-Tal	43	6	2	2	5.193.216	14:00 - 18:00	446.292	191.268	63.756
No	Puerto Natales.	46	9	3	2	5.897.472	14:00 - 18:00	506.814	217.206	72.402
No	Achao	41	12	3	2	5.687.808	14:00 - 18:00	488.796	209.484	69.828
No	Puerto Aysen	42	9	3	2	5.505.024	14:00 - 18:00	473.088	202.752	67.584
No	Coyhaique	171	24	10	3	27.983.872	14:00 - 18:00	1.803.648	772.992	257.664
No	Coyhaique Río Simpson	9	21	2	2	3.225.600	14:00 - 18:00	277.200	118.800	39.600
No	Puerto Porvenir	30	17	2	2	5.048.064	14:00 - 18:00	433.818	185.922	61.974
No	Quellón.	48	5	3	2	5.628.672	14:00 - 18:00	483.714	207.306	69.102
No	Chaiten	30	6	2	2	3.876.096	14:00 - 18:00	333.102	142.758	47.586
No	Punta Arenas 18 de Sept.	10	9	1	2	1.999.872	14:00 - 18:00	171.864	73.656	24.552
No	Chile Chico	23	14	2	2	3.945.984	14:00 - 18:00	339.108	145.332	48.444
No	Cochrane	16	4	1	2	2.198.784	14:00 - 18:00	188.958	80.982	26.994

No	Isla de Pascua	57	7	3	2	6.929.664	14:00 - 18:00	595.518	255.222	85.074
No	Arica Chinchorro	60	6	3	2	7.042.560	14:00 - 18:00	605.220	259.380	86.460
No	Iquique Cavancha	33	23	3	2	6.015.744	14:00 - 18:00	516.978	221.562	73.854
No	Zofri Iquique	17	6	1	2	2.494.464	14:00 - 18:00	214.368	91.872	30.624
No	Iquique Alto Hospicio	18	8	1	2	2.747.136	14:00 - 18:00	236.082	101.178	33.726
Totales		1500	254	88	46			16.206 Mb	6.946 Mb	2.315 Mb
								25 Gigas		

ESTUDIO DE MEJORAS AL SISTEMA INFORMÁTICO.

Si bien es cierto que el sistema cumple con los requerimientos planteados originalmente, a poco andar y como siempre sucede con los Sistemas de Información realizados a pedido, los usuarios mismos y la operación del sistema sugiere mejoras para una próxima versión. Esencialmente, estas mejoras se pueden resumir en:

- Integración con la PU:
 - o Aun cuando el GDI permite ingresar operaciones de crédito como la apertura de un Crédito de consumo, esto ha de hacerse una vez que ya ha sido ingresados los datos en la aplicación correspondiente en la Plataforma Universal (PU). Una mejora sugerida consiste en integrar a los Sistemas de Producto con el GDI, de tal forma que al ser generado un nuevo producto para un cliente (por ejemplo una línea de crédito) el Sistema de Créditos alimente directamente al GDI con los datos del cliente y del producto, de tal forma que el encargado de correspondencia ya cuenta con estos datos para ingresar la operación en el GDI. Además, se sugiere integrar la PU con el GDI a través de la implementación de: Consulta de Oficina inscrita en GDI, Actualización de Estado en PU, Consulta de Estado en PU.
- Integración con Sistema de Tracking:
 - o El Sistema de Tracking es un Sistema de Ámbito Corporativo que centraliza información relativa al estado de los procesos internos del Banco. Estos datos son utilizados posteriormente para Gestión. Se plantea que el GDI interactúe con el Sistema de Tracking para que de esta forma reporte sobre los avances (o

cambios de estado) en el estado de tramitación del curso de las Operaciones de crédito, además de la estimación de los tiempos involucrados en estos procesos. Algunos de los estados de operación que se sugiere podrían reportarse al Sistema de Tracking son:

- Ingreso de Operaciones.
 - Control de Aprobado.
 - Control de Rechazado.
 - Visado Aprobado.
 - Visado Rechazado.
 - Aprobado Control Expost.
 - Rechazado Control Expost.
 - Rechazado Control Expost Imagen.
 - Rechazado Control Expost Operación.
- Integración con Sistema de Control de Devoluciones.
 - o El Sistema de Control de Devoluciones (DGO), también de carácter corporativo cuya función es llevar un control pormenorizado de las devoluciones o rechazos a las distintas solicitudes que se tramitan. De esta forma, es necesario que el GDI, al registrar una operación como "Control Rechazado", informe al Sistema DGO.
 - Incorporación de Informes al GDI.
 - o Se plantea la generación (dentro del GDI) de Informes Diarios tanto por producto como por gestor, correos electrónicos al Agente para notificación en caso de que la sucursal presente muchos rechazos, Informes sobre los Supervisores asignados a un producto en particular, etc.

- Mejoramiento de Procesos en el CPN.
 - o Se plantean modificaciones al CGI para soportar optimizaciones de proceso en el CPN, entre las propuestas se pueden mencionar:
 - Modificar flujo de asignación de imágenes de manera que los administrativos puedan auto asignarse las operaciones.
 - Incorporar el actual formulario impreso del "check list" en el GDI.
 - Adicionar una nueva modalidad de visualización de las imágenes enviadas por la Sucursal.
 - Implementar funcionalidad en el GDI que en forma aleatoria y automática determine carpetas a revisar por el supervisor del CPN.
 - Incorporar funcionalidad que permita controlar y administrar el acceso al GDI por incumplimiento del estándar en envío de carpetas para Control Expost desde sucursales.
 - Automatización y consolidación de cuadraturas manuales.

- Ampliación del Ámbito del GDI.
 - o Originalmente, el GDI fue diseñado para ser utilizado solo con los productos asociados a la llamada Banca de Personas. A raíz del éxito que ha supuesto la implantación del GDI tanto en la disminución de costos asociados como en la agilización de los tiempos requeridos para el Curse de las Operaciones de Crédito, otras áreas usuarias, como Banca Comercial (Orientada a ofrecer servicios Financieros a Empresas) y MYPE (orientada a atender a las Micro y Pequeñas Empresas) han mostrado su interés en un proyecto de ampliación del ámbito de acción del GDI con el objeto de cubrir también a sus productos. Esta ampliación de su ámbito de acción ampliaría radicalmente la gama de productos operados mediante el GDI pues si bien es cierto en muchos casos comprenden

los mismos tipos de producto (Tarjeta de Crédito, Crédito de Consumo, Línea de Crédito, etc.) ha de hacerse el distingo en cuanto al aplicarse a clientes diferentes estos productos también tienen una distinta caracterización.

CAPITULO 8: CONCLUSIONES

Actualmente todos los clientes del Banco esperan respuestas en tiempos cortos, más aún con los crecientes niveles de bancarización que ha tenido el Banco en el último tiempo. Sin embargo, al implementarse el procesamiento centralizado de las operaciones de crédito se presenta un problema de logística y distribución con las sucursales geográficamente más alejadas. Es así, como surge la necesidad de proveer una solución para agilizar los tiempos de respuesta hacia los clientes de estas sucursales.

El análisis de este sistema se realizó en conjunto con el área usuaria, realizando visitas a las sucursales para recopilar información sobre el flujo del proceso de curse. Se recibieron sugerencias y se indicaron las principales falencias del modelo actual. De esta forma se llegó a un modelo consensuado, entre el equipo de desarrollo y el área usuaria para ser implementado.

Para el diseño de la solución se colaboró con las áreas relacionadas con el inicio de un proceso de desarrollo de software al interior del Banco. Estas áreas entregaron las directrices a seguir con el fin de cumplir toda la normativa exigida y vigente para dichos desarrollos. La normativa establecida obliga a seguir pautas en cuanto a arquitectura, modelo de objetos, servicio de datos, manejo de código fuente, manejo de versiones, inclusión de objetos de carácter corporativo, uso de los ambientes de desarrollo, prueba y producción siguiendo para ello todo el ciclo establecido.

Dentro del diseño aprobado por el área usuaria y las áreas de soporte al proceso de desarrollo, se incluye el uso de prototipos, definiéndose claramente con esto las interfaces a crear y el flujo de datos a seguir en el proceso de curse de operaciones.

En la metodología basada en prototipos su etapa de análisis puede ser un poco larga, dependiendo de las veces que se deba modificar los prototipos para conseguir afinidad con el usuario, es muy efectivo desde el punto de vista del producto final, ya que fue el mismo usuario el

que fue definiendo las características específicas del software al ir evaluando este progresivamente en su etapa de análisis.

Al utilizar el servicio de digitalización de imágenes vemos claramente las ventajas de utilizar el estándar Windows DNA, puesto que este servicio permite integrar sistemas y aplicaciones heterogéneos en hardware y software, diferenciando claramente todos los componentes involucrados.

Al utilizar el Sistema de Privilegios del Banco para la generación de los perfiles involucrados en el sistema, logramos que GDI aproveche las sinergias presentes en un entorno integrado como la Shell de Servicios Financieros, concentrando en una interfaz única los accesos a este.

Se establecieron perfiles para las sucursales (escaneador y jefe de operaciones) y para CPN (supervisor y administrativo), en definitiva fueron 4 perfiles utilizados en el modelo, los cuales mediante la inclusión de la tabla de Sucursales en el sistema diseñado podrán determinar si el usuario conectado de la sucursal X tiene acceso al sistema, ya que puede tener el perfil, pero además debe existir dicha sucursal asociada al proceso de curse.

Fueron seleccionadas 3 sucursales críticas para probar la solución informática, estas son Arica, Iquique y Punta Arenas; para esto fue desarrollada una bitácora de implantación, que involucraba el proceso de instalación del escáner, la creación y envío de operaciones desde sucursales al CPN a través de la solución disponible en el ambiente de test (pre-producción). Estas sucursales tuvieron un tiempo de prueba de 1 mes, este tiempo permitió contar con un escenario realista del comportamiento del sistema, donde pudieron detectarse errores en el procedimiento, ya que se ingresaron operaciones reales, además se detectó que ante un gran volumen de operaciones era necesario balancear la carga (previo resultado de pruebas de estrés) del servidor de presentación, lo que originó que el sistema estuviera destinado a la granja de servidores aplicativos con que cuenta el Banco.

Además estas sucursales vieron una significativa disminución en los tiempos de respuesta asociados a sus operaciones de curse de crédito (ver tabla 7.3).

Con la solución implementada se logró un aumento de operaciones de crédito, llegando a volúmenes de operaciones de entre 8 mil a 10 mil mensuales (información proporcionada por CPN). Lo anteriormente descrito, llevó a que otras sucursales fueran incorporadas al sistema GDI; hoy en día existe un 80% de las sucursales totales del BancoEstado usando este sistema.

Además, el éxito de este sistema ha generado el interés de otras áreas usuarias, como son banca PYME y banca Comercial, por utilizar el sistema GDI, esto origina la necesidad de levantar un nuevo proyecto que implemente el estudio de si el actual sistema satisfacería las necesidades que estas nuevas áreas usuarias requieren. Para ello ya fueron presentados los requerimientos e iniciado el proceso de análisis.

BIBLIOGRAFÍA

PRESSMAN, R. 1993. Ingeniería de Software, un Enfoque Práctico. 3ra Edición, España McGraw-Hill.

Francis B., Et. Al. 1998, "Professional Active Server Pages 2.0", Wrox Press.

Homer A., Et. Al., 1999, "Professional Active Server Pages 3.0", Wrox Press.

Blexrud C., Et. Al., 2001, "Professional Windows DNA: Building Distributed Web Applications with VB, COM+, MSMQ, SOAP, and ASP", Wrox Press.

Microsoft Corporation, Microsoft Official Curriculum, "Building Distributed Applications for Microsoft Windows 2000 with Visual Basic".

Microsoft Corporation, Microsoft Official Curriculum, "Mastering Distributed Application Design and Development Using Microsoft Visual Studio 6".

DOCUMENTOS NORMATIVA BANCOESTADO

Algunos de los documentos de la normativa BancoEstado utilizados fueron:

Para Desarrollo

- Normas BECH GUI.doc
- Herramientas a utilizar en nuevos desarrollos.pdf
- Normas de definicion de stored procedure.doc
- Normas para la definicion de componentes de una base de datos.doc
- Políticas a la implementación de bases de datos.doc
- Norma para el tamaño de páginas web y flujo de datos a través de la wan.doc
- Manejando errores y excepciones.doc
- Normas de programacion dna.doc
- Arquitectura de desarrollo de aplicaciones dna.doc
- Normas para nombres de librerias.doc
- Metodologia de desarrollo de aplicaciones informáticas.doc
- Procedimiento para la instalación de componentes mts.doc
- Normas xml para el bech.pdf

Para Testing

- Estándares SQA: QA-NMN01 18/08/2005.doc
- Probar el rendimiento de su aplicacion web.pdf
- Plan de testing: TG-PLN02 24/04/2006.doc

- Plan de Calidad : [TG-PLN01 24/04/2006.doc](#)
- Formulario de registro pruebas funcionales : [TG-FMU06 24/04/2006.xls](#)
- Metodología de Testing: [TG-PDM01 31/08/2005 .doc](#)
- Informe de resultados de pruebas de usabilidad: [TG-INF04 16/03/2006.xls](#)
- Informe de Testing: [TG-INF01 16/03/2006.doc](#)
- Certificado de Aprobación: [TG-FMU08 16/03/2006.doc](#)
- Formulario de aprobación de Testing: [TG-FMU07 16/03/2006.doc](#)
- Checklist de inspección de código: [TG LDV01 30/11/2005.xls](#)

Para Producción

- [Sistema privilegios.doc](#)
- [Tipos de pasos a producción en dna y criterios para su ejecución.doc](#)

REFERENCIAS INTERNET.

Las referencias consultas de internet fueron:

UML

- <http://www.dcc.uchile.cl/~psalinas/uml/introduccion.html>
- <http://www.clikear.com/manuales/uml/>

Metodologías de Desarrollo

- <http://www2.netexplora.com/tombrad/arteyciencia.html>
- http://gbtcr.chileforge.cl/info_web/node131.html
- http://www.inter-media.cl/desarrollo/metodologia_trabajo.php

APENDICE 1: GLOSARIO

- **Sucursal Centralizadora:** sucursal que digitaliza y envía operaciones al CPN a través de la aplicación.
- **Sucursal Satélite:** sucursal donde se reciben operaciones, pero son enviadas a las centralizadoras.
- **Componente:** archivo del tipo librería, es decir es código ya compilado, que usualmente se utiliza para conexiones a bases de datos y para aplicar reglas de negocio.
- **DLL:** Es una componente creada en lenguaje Visual Basic, formada por una o varias clases las cuales contienen un conjunto de métodos que son los encargados de realizar las distintas funcionalidades que posee la DLL.
- **Jscript:** implementación de Microsoft del lenguaje JavaScript.
- **Plataforma:** hardware o software que definen un estándar sobre el cual se puede desarrollar un sistema.
- **Servicio de datos:** programa que se encuentra dentro del ambiente de una base de datos y que accesa distintas tablas para entregar una respuesta a quien lo invoque.
- **Scripts:** lista de comandos que pueden ser ejecutados sin la intervención de un usuario.
- **VBScript:** lenguaje script desarrollado por Microsoft, usado en el navegador Internet Explorer.
- **BECH:** abreviatura de BancoEstado de Chile.
- **WebSphere:** Implementación de múltiples capas de IBM
- **Mozilla:** Navegador heredero de Netscape Navigator, desarrollado por Mozilla.org
- **Opera:** Navegador independiente de origen comercial, desarrollado por Opera Software.

APENDICE 2: ABREVIATURAS UTILIZADAS.

Las abreviaturas utilizadas en este proyecto se describen y explican en la siguiente tabla:

Sigla	Significado	Traducción
ADO	ActiveX Data Object	Objeto de datos ActiveX
API	Application Program Interface	Interfaz de Programas de Aplicación
ASP	Active Server Page	Página Activa del Servidor
CICS	Customer Information Control System	Sistema De Control De la Información Del Cliente
CLASS	Class	Clase
COM	Component Object Model	Modelo de Componentes de Objeto
CORBA	Common Object Request Broker Architecture	Arquitectura de componentes para ambientes distribuidos.
DLL	Dynamic Link Library	Librería de Enlace Dinámico
DNA	Distributed InterNet Architecture	Arquitectura Distribuidas en Internet.
EJB	Enterprise JavaBeans	
GUID	Global Unique Identifier	Identificador Unico Global
HTML	HyperText Markup Language	Lenguaje de Descripción de HiperTexto
HTTP	HyperText Transport Protocol	Protocolo para Transferencia de HiperTexto
IBM	International Business machines	Maquinas de Negocio Internacional

ID	ID	Identificador
IIS	Internet Information Server	Servidor de Información Internet
J2EE	Java 2 Enterprise Edition	Implementación de Java para uso corporativo.
MTS	Microsoft Transaction Server	Servidor Transaccional
SP	Store Procedure	Procedimiento Almacenado
SQL	Structured Query Language	Lenguaje de consulta estructurado
URL	Uniform Resource Locator	Localizador Uniforme de Recursos
VB	Visual Basic	Visual Basic
XML	Extensible Markup Language	Lenguaje de Marcas Extensible
XSL	Extensible Stylesheet Language	Lenguaje de hoja de estilos extensible

ANEXO 1: EXTRACTO CODIGO FUENTE

En el extracto del código fuente se indicará parte de lo realizado en la COM de DATOS y en NEGOCIOS.

DATOS

Clase c_CUR_DATA

```
Public Enum DataBaseConn
    bgdi_img = 1
    bsap_cen = 2
    btab_gen = 3
    bcen_cli = 5
End Enum

Enum HKEYClavesRaices
    HKEY_CLASSES_ROOT = 4
    HKEY_CURRENT_USER = 3
    XXHKEY_LOCAL_MACHINEX = 2
    HKEY_USERS = 1
End Enum

Function RunSP(ByVal strSP As String, _
    ByVal db As DataBaseConn, _
    ParamArray params() As Variant)
    On Error GoTo errorHandler

    data_base = db

    Dim cmd As ADODB.Command
    Set cmd = CreateObject("ADODB.Command")

    cmd.ActiveConnection = ConnectionString
    cmd.CommandText = strSP
    cmd.CommandType = adCmdStoredProc
    collectParams cmd, params

    cmd.Execute , , adExecuteNoRecords

    Set cmd.ActiveConnection = Nothing
    Set cmd = Nothing

    Exit Function

errorHandler:
    Set cmd = Nothing
    RaiseError m_modName, "RunSP(" & strSP & ", ...)"
End Function

Function RunSPReturnRS( ByVal strSP As String, _
    ByVal db As DataBaseConn, _
    ParamArray params() As Variant) As
    ADODB.Recordset
    On Error GoTo errorHandler

    data_base = db

    Dim rs As ADODB.Recordset, cmd As ADODB.Command
```

```

Set rs = CreateObject("ADODB.RecordSet")
Set cmd = CreateObject("ADODB.Command")

cmd.ActiveConnection = ConnectionString
cmd.CommandText = strSP
cmd.CommandType = adCmdStoredProc

collectParams cmd, params

rs.CursorLocation = adUseClient
rs.Open cmd, , adOpenForwardOnly, adLockReadOnly

Set RunSPReturnRS = rs

Set cmd.ActiveConnection = Nothing
Set cmd = Nothing
Set rs.ActiveConnection = Nothing

Exit Function

errorHandler:
Set rs = Nothing
Set cmd = Nothing
RaiseError m_modName, "RunSPReturnRS(" & strSP & ", ...)"
End Function

Function RunSPReturnRS_RW(ByVal strSP As String, _
                          ByVal db As DataBaseConn, _
                          ParamArray params() As Variant) As
ADODB.Recordset
On Error GoTo errorHandler

data_base = db

Dim rs As ADODB.Recordset, cmd As ADODB.Command
Set rs = CreateObject("ADODB.RecordSet")
Set cmd = CreateObject("ADODB.Command")

cmd.ActiveConnection = ConnectionString
cmd.CommandText = strSP
cmd.CommandType = adCmdStoredProc
collectParams cmd, params
rs.CursorLocation = adUseClient

rs.Open cmd, , adOpenDynamic, adLockBatchOptimistic
Set cmd = Nothing

Set RunSPReturnRS_RW = rs
Exit Function

errorHandler:
Set rs = Nothing
Set cmd = Nothing
RaiseError m_modName, "RunSPReturnRS_RW(" & strSP & ", ...)"
End Function

Function RunSQLReturnRS(ByVal strSP As String, _
                       ByVal db As DataBaseConn, _
                       ParamArray params() As Variant) As
ADODB.Recordset
On Error GoTo errorHandler

data_base = db

Dim rs As ADODB.Recordset, cmd As ADODB.Command

```

```

Set rs = CreateObject("ADODB.RecordSet")
Set cmd = CreateObject("ADODB.Command")

cmd.ActiveConnection = ConnectionString
cmd.CommandText = strSP
cmd.CommandType = adCmdText

collectParams cmd, params

rs.CursorLocation = adUseClient
rs.Open cmd, , adOpenForwardOnly, adLockReadOnly

Set cmd.ActiveConnection = Nothing
Set cmd = Nothing
Set rs.ActiveConnection = Nothing

Set RunSQLReturnRS = rs
Exit Function

errorHandler:
Set rs = Nothing
Set cmd = Nothing
RaiseError m_modName, "RunSQLReturnRS(" & strSP & ", ...)"
End Function

Function RunSQLReturnRS_RW(ByVal strSP As String, _
                          ByVal db As DataBaseConn, _
                          ParamArray params() As Variant) As
ADODB.Recordset
On Error GoTo errorHandler

data_base = db

Dim rs As ADODB.Recordset, cmd As ADODB.Command
Set rs = CreateObject("ADODB.RecordSet")
Set cmd = CreateObject("ADODB.Command")

cmd.ActiveConnection = ConnectionString
cmd.CommandText = strSP
cmd.CommandType = adCmdText
collectParams cmd, params
rs.CursorLocation = adUseClient

rs.Open cmd, , adOpenDynamic, adLockBatchOptimistic
Set cmd = Nothing

Set RunSQLReturnRS_RW = rs
Exit Function

errorHandler:
Set rs = Nothing
Set cmd = Nothing
RaiseError m_modName, "RunSQLReturnRS_RW(" & strSP & ", ...)"
End Function

Function RunSQL(ByVal strSP As String, _
               ByVal db As DataBaseConn, _
               ParamArray params() As Variant)
On Error GoTo errorHandler

data_base = db

Dim cmd As ADODB.Command
Set cmd = CreateObject("ADODB.Command")

```

```

cmd.ActiveConnection = ConnectionString
cmd.CommandText = strSP
cmd.CommandType = adCmdText
collectParams cmd, params

cmd.Execute , , adExecuteNoRecords

Set cmd.ActiveConnection = Nothing
Set cmd = Nothing

Exit Function

errorHandler:
Set cmd = Nothing
RaiseError m_modName, "RunSQL(" & strSP & ", ...)"
End Function

Function RunSPReturnInteger(ByVal strSP As String, _
                           ByVal db As DataBaseConn, _
                           ParamArray params() As Variant) As Long
On Error GoTo errorHandler

data_base = db

Dim cmd As ADODB.Command
Set cmd = CreateObject("ADODB.Command")

cmd.ActiveConnection = ConnectionString
cmd.CommandText = strSP
cmd.CommandType = adCmdStoredProc
collectParams cmd, params

cmd.Parameters.Append cmd.CreateParameter("@retval", adInteger,
adParamOutput, 4)

cmd.Execute , , adExecuteNoRecords
RunSPReturnInteger = cmd.Parameters("@retval").Value

Set cmd.ActiveConnection = Nothing
Set cmd = Nothing

Exit Function

errorHandler:
Set cmd = Nothing
RaiseError m_modName, "RunSPReturnInteger(" & strSP & ", ...)"
End Function

Private Sub collectParams(ByRef cmd As Object, ParamArray argparams()
As Variant)
Dim params As Variant, V As Variant
Dim i As Integer, L As Integer, U As Integer

params = argparams(0)
For i = LBound(params) To UBound(params)
L = LBound(params(i))
U = UBound(params(i))
If U - L = 3 Then
If VarType(params(i)(3)) = vbString Then
V = IIf(params(i)(3) = vbNullString, Null,
params(i)(3))
Else
V = params(i)(3)
End If

```



```

        cmd.Parameters.Append cmd.CreateParameter(params(i)(0),
params(i)(1), adParamInput, params(i)(2), V)

    Else
        CtxRaiseError m_modName, "collectParams(...): numero de
parametros incorrector"
    End If
Next i
End Sub

```

Clase c_CUR_DATA.cls

```

Public Function mSVA_OPE_SUP(ByVal usr_spv_cod As String, _
                             ByVal pop_fol_ope As Long _
                             ) As Boolean

    Dim Obj As c_GDI_DAT_COM.C_CUR_DBD
    Set Obj = CreateObject("c_GDI_DAT_COM.C_CUR_DBD")

    On Error GoTo errorHandler

    Obj.RunSP "<nombre store procedure>", bgdi_img, _
        mp("@usr_spv_cod", adVarChar, 10, usr_spv_cod), _
        mp("@pop_fol_ope", adInteger, 4, pop_fol_ope)

    mSVA_OPE_SUP = True

    Set Obj = Nothing
    Exit Function

errorHandler:
    App.LogEvent m_modName & "." & "mSVA_OPE_SUP" & Err.description

    ErrLogAplicacion "c_GDI_DAT_COM.C_CUR_DATA.mSVA_OPE_SUP", _
        Array(usr_spv_cod, pop_fol_ope), EA_DFTRBKCLSCR
    Err.Raise mErrNumber, mErrSource, mErrDescr

    Set Obj = Nothing
End Function

Public Function mSVA_OPE_RCH(ByVal pop_fol_ope As Long, _
                             ByVal mrc_mtv_cod As Integer, _
                             ByVal usr_cod As String, _
                             ByVal orc_obs_rch As String _
                             ) As Boolean

    Dim Obj As c_GDI_DAT_COM.C_CUR_DBD
    On Error GoTo errorHandler

    Set Obj = CreateObject("c_GDI_DAT_COM.C_CUR_DBD")

    Obj.RunSP "<nombre store procedure>", bgdi_img, _
        mp("@pop_fol_ope", adInteger, 4, pop_fol_ope), _
        mp("@mrc_mtv_cod", adInteger, 4, mrc_mtv_cod), _
        mp("@usr_cod", adChar, 10, usr_cod), _
        mp("@orc_obs_rch", adChar, 300, orc_obs_rch)

    mSVA_OPE_RCH = True

    Set Obj = Nothing
    Exit Function

```

```

errorHandler:
    App.LogEvent m_modName & "." & "mSVA_OPE_RCH" & Err.description

    ErrLogAplicacion "c_GDI_DAT_COM.C_CUR_DATA.mSVA_OPE_RCH", _
        Array(pop_fol_ope, mrc_mtv_cod, usr_cod, _
            orc_obs_rch), EA_DFTRBKCLSCR
    Err.Raise mErrNumber, mErrSource, mErrDescr

    Set Obj = Nothing
End Function

Public Function mSVC_MTV_RCH_OPE(ByVal pop_fol_ope As Long _
    ) As ADODB.Recordset

    Dim Obj As c_GDI_DAT_COM.C_CUR_DBD ' Object

    On Error GoTo errorHandler

    Set Obj = CreateObject("c_GDI_DAT_COM.C_CUR_DBD")

    Set mSVC_MTV_RCH_OPE = Obj.RunSPReturnRS("<nombre store
procedure>", bgdi_img, _
        mp("@pop_fol_ope", adInteger, 4, pop_fol_ope))

    Set Obj = Nothing
    Exit Function

errorHandler:
    App.LogEvent m_modName & "." & "mSVC_MTV_RCH_OPE" & Err.description

    ErrLogAplicacion "c_GDI_DAT_COM.C_CUR_DATA.mSVC_MTV_RCH_OPE", _
        Array(pop_fol_ope), EA_DFTRBKCLSCR
    Err.Raise mErrNumber, mErrSource, mErrDescr

    Set Obj = Nothing
End Function

```

NEGOCIOS:

Clase C_CUR_NEG.cls

```

Public Function mSVA_OPE_SUP(ByVal usr_spv_cod As String, _
    ByVal pop_fol_ope As Long _
    ) As Boolean

    Dim Obj As c_GDI_DAT_COM.c_CUR_DATA

    On Error GoTo errorHandler

    Set Obj = CreateObject("c_GDI_DAT_COM.c_CUR_DATA")

    mSVA_OPE_SUP = Obj.mSVA_OPE_SUP(usr_spv_cod, _
        pop_fol_ope)

    Set Obj = Nothing

    Exit Function

errorHandler:
    App.LogEvent m_modName & "." & "mSVA_OPE_SUP" & Err.Description

    ErrLogAplicacion "c_GDI_NEG.c_CUR_NEG.mSVA_OPE_SUP", _
        Array(usr_spv_cod, pop_fol_ope), EA_DFTRBKCLSCR
    Err.Raise mErrNumber, mErrSource, mErrDescr

```

```

        Set Obj = Nothing
End Function

Public Function mSVA_OPE_RCH(ByVal pop_fol_ope As Long, _
                             ByVal mrc_mtv_cod As Integer, _
                             ByVal usr_cod As String, _
                             ByVal orc_obs_rch As String _
                             ) As Boolean

    Dim Obj As c_GDI_DAT_COM.c_CUR_DATA

    On Error GoTo errorHandler

    Set Obj = CreateObject("c_GDI_DAT_COM.c_CUR_DATA")

    mSVA_OPE_RCH = Obj.mSVA_OPE_RCH(pop_fol_ope, _
                                     mrc_mtv_cod, _
                                     usr_cod, _
                                     orc_obs_rch)

    Set Obj = Nothing
    Exit Function

errorHandler:
    App.LogEvent m_modName & "." & "mSVA_OPE_RCH" & Err.Description

    ErrLogAplicacion "c_GDI_NEG.c_CUR_NEG.mSVA_OPE_RCH", _
                    Array(pop_fol_ope, mrc_mtv_cod, _
                        usr_cod, orc_obs_rch), EA_DFTRBKCLSCR

    Err.Raise mErrNumber, mErrSource, mErrDescr

    Set Obj = Nothing
End Function

Public Function mSVA_OPE_ADM(ByVal eop_est_cod As String, _
                             ByVal pop_fec_est As Date, _
                             ByVal pop_hra_act As Date, _
                             ByVal usr_cod As String, _
                             ByVal pop_ide_sis As Integer, _
                             ByVal pop_fol_ope As Long _
                             ) As Boolean

    Dim Obj As c_GDI_DAT_COM.c_CUR_DATA

    On Error GoTo errorHandler
    Set Obj = CreateObject("c_GDI_DAT_COM.c_CUR_DATA")

    mSVA_OPE_ADM = Obj.mSVA_OPE_ADM(eop_est_cod, _
                                     pop_fec_est, _
                                     pop_hra_act, _
                                     usr_cod, _
                                     pop_ide_sis, _
                                     pop_fol_ope)

    Set Obj = Nothing

    Exit Function

errorHandler:
    App.LogEvent m_modName & "." & "mSVA_OPE_RCH" & Err.Description

    ErrLogAplicacion "c_GDI_NEG.c_CUR_NEG.mSVA_OPE_RCH", _
                    Array(pop_fol_ope, mrc_mtv_cod, _
                        usr_cod, orc_obs_rch), EA_DFTRBKCLSCR

```

```

Err.Raise mErrNumber, mErrSource, mErrDescr

Set Obj = Nothing
End Function

Public Function mSVA_ING_OPE_ENC_CRR(ByVal pop_fol_ope_ing As Long, _
ByVal ofi_cod_ing As Integer, _
ByVal mon_cod_ing As String, _
ByVal cli_rut_ing As Double, _
ByVal eop_est_cod_ing As String, _
ByVal usr_cod_ing As String, _
ByVal cli_rut_dv_ing As String, _
ByVal pop_cpa_cod_ing As String, _
ByVal usr_jef_cod_ing As String, _
ByVal pop_fec_ing_ing As Date, _
ByVal usr_adm_cod_ing As String, _
ByVal pop_mnt_brt_ing As Double, _
ByVal usr_ast_cod_ing As String, _
ByVal usr_spv_cod_ing As String, _
ByVal pop_nom_no_cli_ing As
String,_
ByVal pop_fec_est_ing As Date, _
ByVal pop_hra_act_ing As Date, _
ByVal pop_ide_sis As Integer, _
Optional ByVal segmento As
Integer,_
Optional ByVal clase As Integer, _
Optional ByVal numeroope As
String,_
Optional ByVal idrcredito As
String_
) As Boolean 'String

Dim Obj As c_GDI_DAT_COM.c_CUR_DATA
On Error GoTo errorHandler

Set Obj = CreateObject("c_GDI_DAT_COM.c_CUR_DATA")

mSVA_ING_OPE_ENC_CRR = Obj.mSVA_ING_OPE_ENC_CRR(pop_fol_ope_ing, _
ofi_cod_ing, _
mon_cod_ing, _
cli_rut_ing, _
eop_est_cod_ing, _
usr_cod_ing, _
cli_rut_dv_ing, _
pop_cpa_cod_ing, _
usr_jef_cod_ing, _
pop_fec_ing_ing, _
usr_adm_cod_ing, _
pop_mnt_brt_ing, _
usr_ast_cod_ing, _
usr_spv_cod_ing, _
pop_nom_no_cli_ing,
_
pop_fec_est_ing, _
pop_hra_act_ing, _
pop_ide_sis, _
segmento, _
clase, _
numeroope, _
idrcredito)

Set Obj = Nothing

```



```

        Err.Raise mErrNumber, mErrSource, mErrDescr

        Set Obj = Nothing
    End Function

    Public Function mSVC_OPE_VGT_FOL(ByVal pop_fol_ope As Long) As String

        Dim Obj As c_GDI_DAT_COM.c_CUR_DATA
        On Error GoTo errorHandler
        Set Obj = CreateObject("c_GDI_DAT_COM.c_CUR_DATA")

        mSVC_OPE_VGT_FOL = mGeneraXML(Obj.mSVC_OPE_VGT_FOL(pop_fol_ope),
        "Consulta")

        Set Obj = Nothing
        Exit Function

    errorHandler:
        App.LogEvent m_modName & "." & "mSVC_OPE_VGT_FOL" & Err.Description

        ErrLogAplicacion "c_GDI_NEG.c_CUR_NEG.mSVC_OPE_VGT_FOL", _
            Array(pop_fol_ope), EA_DFTRBKCLSCR
        Err.Raise mErrNumber, mErrSource, mErrDescr

        Set Obj = Nothing
    End Function

    Public Function mSVC_EST_OPE(ByVal eop_est_cod As String) As String
        Dim Obj As c_GDI_DAT_COM.c_CUR_DATA

        On Error GoTo errorHandler

        Set Obj = CreateObject("c_GDI_DAT_COM.c_CUR_DATA")

        mSVC_EST_OPE = mGeneraXML(Obj.mSVC_EST_OPE(eop_est_cod),
        "Consulta")

        Set Obj = Nothing
        Exit Function

    errorHandler:
        App.LogEvent m_modName & "." & "mSVC_EST_OPE" & Err.Description

        ErrLogAplicacion "c_GDI_NEG.c_CUR_NEG.mSVC_EST_OPE", _
            Array(eop_est_cod), EA_DFTRBKCLSCR
        Err.Raise mErrNumber, mErrSource, mErrDescr

        Set Obj = Nothing
    End Function

    Public Function mSVC_BCCBUS_DAT001(ByVal rutCli As String, _
        ByVal header_out As String _
        ) As String

        Dim Obj As c_GDI_DAT_COM.c_CUR_DATA

        On Error GoTo errorHandler

        Set Obj = CreateObject("c_GDI_DAT_COM.c_CUR_DATA")

        mSVC_BCCBUS_DAT001 = mGeneraXML(Obj.mSVC_BCCBUS_DAT001(rutCli, _
            header_out), _
            "Consulta")

        Set Obj = Nothing
    End Function

```

```

Exit Function

errorHandler:
    App.LogEvent m_modName & "." & "mSVC_BCCBUS_DAT001" &
Err.Description

    ErrLogAplicacion "c_GDI_NEG.c_CUR_NEG.mSVC_BCCBUS_DAT001", _
        Array(rutCli, header_out), EA_DFTRBKCLSCR
    Err.Raise mErrNumber, mErrSource, mErrDescr

    Set Obj = Nothing

End Function

Public Function mSVC_JEF_VGT_PRU(ByVal ofi_pde_cod As Integer, _
                                ByVal tot As Integer) As String

    Dim Obj As c_GDI_DAT_COM.c_CUR_DATA

    On Error GoTo errorHandler

    Set Obj = CreateObject("c_GDI_DAT_COM.c_CUR_DATA")

    mSVC_JEF_VGT_PRU = mGeneraXML(Obj.mSVC_JEF_VGT_PRU(ofi_pde_cod, _
                                                         tot), _
                                   "Consulta")

    Set Obj = Nothing

Exit Function

errorHandler:
    App.LogEvent m_modName & "." & "mSVC_JEF_VGT_PRU" & Err.Description

    ErrLogAplicacion "c_GDI_NEG.c_CUR_NEG.mSVC_JEF_VGT_PRU", _
        Array(ofi_pde_cod, tot), EA_DFTRBKCLSCR
    Err.Raise mErrNumber, mErrSource, mErrDescr

    Set Obj = Nothing
End Function

Public Function mSVC_MTV_RCH(ByVal mrc_mtv_cod As Integer, _
                             ByVal mrc_tip_rch As Integer _
                             ) As String

    Dim Obj As c_GDI_DAT_COM.c_CUR_DATA

    On Error GoTo errorHandler

    Set Obj = CreateObject("c_GDI_DAT_COM.c_CUR_DATA")

    mSVC_MTV_RCH = mGeneraXML(Obj.mSVC_MTV_RCH(mrc_mtv_cod, _
                                                mrc_tip_rch), _
                              "Consulta")

    Set Obj = Nothing

Exit Function

errorHandler:
    App.LogEvent m_modName & "." & "mSVC_MTV_RCH" & Err.Description

    ErrLogAplicacion "c_GDI_NEG.c_CUR_NEG.mSVC_MTV_RCH", _
        Array(mrc_mtv_cod, mrc_tip_rch), EA_DFTRBKCLSCR
    Err.Raise mErrNumber, mErrSource, mErrDescr

```

```

        Set Obj = Nothing
End Function

Public Function mSVC_OPE_RCH(ByVal pop_fol_ope_ing As Long _
                             ) As String

    Dim Obj As c_GDI_DAT_COM.c_CUR_DATA

    On Error GoTo errorHandler

    Set Obj = CreateObject("c_GDI_DAT_COM.c_CUR_DATA")

    mSVC_OPE_RCH = mGeneraXML(Obj.mSVC_OPE_RCH(pop_fol_ope_ing),
    "Consulta")

    Set Obj = Nothing

    Exit Function

errorHandler:
    App.LogEvent m_modName & "." & "mSVC_OPE_RCH" & Err.Description

    ErrLogAplicacion "c_GDI_NEG.c_CUR_NEG.mSVC_OPE_RCH", _
        Array(pop_fol_ope_ing), EA_DFTRBKCLSCR
    Err.Raise mErrNumber, mErrSource, mErrDescr

    Set Obj = Nothing
End Function

Public Function mSVC_PRD_CBO_OPE(ByVal pop_fol_ope As Long _
                                  ) As String

    Dim Obj As c_GDI_DAT_COM.c_CUR_DATA

    On Error GoTo errorHandler

    Set Obj = CreateObject("c_GDI_DAT_COM.c_CUR_DATA")

    mSVC_PRD_CBO_OPE = mGeneraXML(Obj.mSVC_PRD_CBO_OPE(pop_fol_ope),
    "Consulta")

    Set Obj = Nothing

    Exit Function

errorHandler:
    App.LogEvent m_modName & "." & "mSVC_PRD_CBO_OPE" & Err.Description

    ErrLogAplicacion "c_GDI_NEG.c_CUR_NEG.mSVC_PRD_CBO_OPE", _
        Array(pop_fol_ope), EA_DFTRBKCLSCR
    Err.Raise mErrNumber, mErrSource, mErrDescr

    Set Obj = Nothing

End Function

Public Function mSVC_MTV_RCH_OPE(ByVal pop_fol_ope As Long _
                                  ) As String

    Dim Obj As c_GDI_DAT_COM.c_CUR_DATA

    On Error GoTo errorHandler

    Set Obj = CreateObject("c_GDI_DAT_COM.c_CUR_DATA")

```



```

mSVC_MTV_RCH_OPE = mGeneraXML(Obj.mSVC_MTV_RCH_OPE(pop_fol_ope),
"Consulta")

Set Obj = Nothing

Exit Function

errorHandler:
App.LogEvent m_modName & "." & "mSVC_MTV_RCH_OPE" & Err.Description

ErrLogAplicacion "c_GDI_NEG.c_CUR_NEG.mSVC_MTV_RCH_OPE", _
Array(pop_fol_ope), EA_DFTRBKCLSCR
Err.Raise mErrNumber, mErrSource, mErrDescr

Set Obj = Nothing
End Function

Public Function mSVC_OPE_VGT_ENC_CRR_PRU(ByVal ofi_pde_cod As Integer,
_
ByVal tot As Integer) As
String

Dim Obj As c_GDI_DAT_COM.c_CUR_DATA

On Error GoTo errorHandler

Set Obj = CreateObject("c_GDI_DAT_COM.c_CUR_DATA")

mSVC_OPE_VGT_ENC_CRR_PRU =
mGeneraXML(Obj.mSVC_OPE_VGT_ENC_CRR_PRU(ofi_pde_cod, _
tot), _
"Consulta")

Set Obj = Nothing

Exit Function

errorHandler:
App.LogEvent m_modName & "." & "mSVC_OPE_VGT_ENC_CRR_PRU" &
Err.Description

ErrLogAplicacion "c_GDI_NEG.c_CUR_NEG.mSVC_OPE_VGT_ENC_CRR_PRU", _
Array(ofi_pde_cod, tot), EA_DFTRBKCLSCR
Err.Raise mErrNumber, mErrSource, mErrDescr

Set Obj = Nothing
End Function

Public Function mSVC_OPE_HIS_ENC_CRR_DAT(ByVal tip_consulta As Integer,
_
ByVal eop_est_cod As String, _
ByVal ofi_pde As Integer, _
ByVal ohi_fol_ope As Long, _
ByVal ohi_cpa_cod As String, _
ByVal pop_fec_desde As Date, _
ByVal pop_fec_hasta As Date, _
ByVal cli_rut As Double, _
ByVal tot As Integer) As String

Dim Obj As c_GDI_DAT_COM.c_CUR_DATA

On Error GoTo errorHandler

```

```

Set Obj = CreateObject("c_GDI_DAT_COM.c_CUR_DATA")

mSVC_OPE_HIS_ENC_CRR_DAT =
mGeneraXML(Obj.mSVC_OPE_HIS_ENC_CRR_DAT(tip_consulta, _
                                         eop_est_cod, _
                                         ofi_pde, _
                                         ohi_fol_ope, _
                                         ohi_cpa_cod, _
                                         pop_fec_desde, _
                                         pop_fec_hasta, _
                                         cli_rut, _
                                         tot),"Consulta")

Set Obj = Nothing

Exit Function

errorHandler:
    App.LogEvent m_modName & "." & "mSVC_OPE_HIS_ENC_CRR_DAT" &
Err.Description

    ErrLogAplicacion "c_GDI_NEG.c_CUR_NEG.mSVC_OPE_HIS_ENC_CRR_DAT", _
        Array(tip_consulta, eop_est_cod, ofi_pde, _
              ohi_fol_ope, ohi_cpa_cod, pop_fec_desde, _
              pop_fec_hasta, cli_rut, tot),
EA_DFTRBKCLSCR
    Err.Raise mErrNumber, mErrSource, mErrDescr
    Set Obj = Nothing
End Function

Public Function mSVC_OPE_VGT_SUP_CPN_DAT(ByVal tip_consulta As Integer,
_
                                         ByVal eop_est_cod As String, _
                                         ByVal pop_fol_ope As Long, _
                                         ByVal pop_cpa_cod As String, _
                                         ByVal pop_fec_desde As Date, _
                                         ByVal pop_fec_hasta As Date, _
                                         ByVal cli_rut As Double, _
                                         ByVal ofi_cod As Integer, _
                                         ByVal usr_ast_cod As String, _
                                         ByVal epr_prd_cod As Integer, _
                                         ByVal tot As Integer) As String

    Dim Obj As c_GDI_DAT_COM.c_CUR_DATA

    On Error GoTo errorHandler

    Set Obj = CreateObject("c_GDI_DAT_COM.c_CUR_DATA")

    mSVC_OPE_VGT_SUP_CPN_DAT =
mGeneraXML(Obj.mSVC_OPE_VGT_SUP_CPN_DAT(tip_consulta, _
                                         eop_est_cod, _
                                         pop_fol_ope, _
                                         pop_cpa_cod, _
                                         pop_fec_desde, _
                                         pop_fec_hasta, _
                                         cli_rut, _
                                         ofi_cod, _
                                         usr_ast_cod, _
                                         epr_prd_cod, _
                                         tot), "Consulta")

    Set Obj = Nothing

Exit Function

```

```

errorHandler:
    App.LogEvent m_modName & "." & "mSVC_OPE_VGT_SUP_CPN_DAT" &
Err.Description

    ErrLogAplicacion "c_GDI_NEG.c_CUR_NEG.mSVC_OPE_VGT_SUP_CPN_DAT", _
        Array(tip_consulta, eop_est_cod, pop_fol_ope, _
            pop_cpa_cod, pop_fec_desde, pop_fec_hasta,
        _
            cli_rut, ofi_cod, usr_ast_cod, epr_prd_cod,
tot), EA_DFTRBKCLSCR
    Err.Raise mErrNumber, mErrSource, mErrDescr
    Set Obj = Nothing
End Function

Public Function mSVC_OPE_HIS_SUP_CPN_DAT(ByVal tip_consulta As Integer,
    _
        ByVal eop_est_cod As String, _
        ByVal ohi_fol_ope As Long, _
        ByVal ohi_cpa_cod As String, _
        ByVal pop_fec_desde As Date, _
        ByVal pop_fec_hasta As Date, _
        ByVal cli_rut As Double, _
        ByVal ofi_cod As Integer, _
        _
        ByVal ohi_usr_cod_ast As
String,
        byVal epr_prd_cod As Integer,
    _
        ByVal tot As Integer) As String

    Dim Obj As c_GDI_DAT_COM.c_CUR_DATA

    On Error GoTo errorHandler

    Set Obj = CreateObject("c_GDI_DAT_COM.c_CUR_DATA")

    mSVC_OPE_HIS_SUP_CPN_DAT =
mGeneraXML(Obj.mSVC_OPE_HIS_SUP_CPN_DAT(tip_consulta, _
        eop_est_cod, _
        ohi_fol_ope, _
        ohi_cpa_cod, _
        pop_fec_desde, _
        pop_fec_hasta, _
        cli_rut, _
        ofi_cod, _
        ohi_usr_cod_ast, _
        epr_prd_cod, _
        tot), "Consulta")

    Set Obj = Nothing

    Exit Function

errorHandler:
    App.LogEvent m_modName & "." & "mSVC_OPE_HIS_SUP_CPN_DAT" &
Err.Description

    ErrLogAplicacion "c_GDI_NEG.c_CUR_NEG.mSVC_OPE_HIS_SUP_CPN_DAT", _
        Array(tip_consulta, eop_est_cod, ohi_fol_ope, _
            ohi_cpa_cod, pop_fec_desde, pop_fec_hasta,
        _
            cli_rut, ofi_cod, ohi_usr_cod_ast,
epr_prd_cod, tot), EA_DFTRBKCLSCR
    Err.Raise mErrNumber, mErrSource, mErrDescr
    Set Obj = Nothing

```

```

End Function

Public Function mSVC_OPE_VGT_SUP_CPN_PRU(estado As String, _
                                           sistema As Integer, _
                                           tot As Integer) As String

    Dim Obj As c_GDI_DAT_COM.c_CUR_DATA

    On Error GoTo errorHandler

    Set Obj = CreateObject("c_GDI_DAT_COM.c_CUR_DATA")

    mSVC_OPE_VGT_SUP_CPN_PRU =
mGeneraXML(Obj.mSVC_OPE_VGT_SUP_CPN_PRU(estado, _

sistema, _

tot), "Consulta")

    Set Obj = Nothing
    Exit Function

errorHandler:
    App.LogEvent m_modName & "." & "mSVC_OPE_VGT_SUP_CPN_PRU" &
Err.Description

    ErrLogAplicacion "c_GDI_NEG.c_CUR_NEG.mSVC_OPE_VGT_SUP_CPN_PRU", _
        Array(estado, sistema, tot), EA_DFTRBKCLSCR
    Err.Raise mErrNumber, mErrSource, mErrDescr
    Set Obj = Nothing
End Function

Public Function mSVC_OPE_VGT_ADM_CPN_PRU(eop_est_cod As String, _
                                           usr_adm_cod As String, _
                                           tot As Integer) As String

    Dim Obj As c_GDI_DAT_COM.c_CUR_DATA

    On Error GoTo errorHandler

    Set Obj = CreateObject("c_GDI_DAT_COM.c_CUR_DATA")

    mSVC_OPE_VGT_ADM_CPN_PRU =
mGeneraXML(Obj.mSVC_OPE_VGT_ADM_CPN_PRU(eop_est_cod, _

usr_adm_cod, _

tot), "Consulta")

    Set Obj = Nothing

    Exit Function

errorHandler:
    App.LogEvent m_modName & "." & "mSVC_OPE_VGT_ADM_CPN_PRU" &
Err.Description

    ErrLogAplicacion "c_GDI_NEG.c_CUR_NEG.mSVC_OPE_VGT_ADM_CPN_PRU", _
        Array(eop_est_cod, usr_adm_cod, tot),
EA_DFTRBKCLSCR
    Err.Raise mErrNumber, mErrSource, mErrDescr
    Set Obj = Nothing
End Function

```

```

Public Function mSVC_PRD_CFD_MAN(Optional ByVal epr_prd_cod As Integer,
-
                                Optional ByVal epr_prd_cbo As Integer
-
                                ) As String

    Dim Obj As c_GDI_DAT_COM.c_CUR_DATA

    On Error GoTo errorHandler

    Set Obj = CreateObject("c_GDI_DAT_COM.c_CUR_DATA")

    mSVC_PRD_CFD_MAN = mGeneraXML(Obj.mSVC_PRD_CFD_MAN(epr_prd_cod,
epr_prd_cbo), "Consulta")

    Set Obj = Nothing

    Exit Function

errorHandler:
    App.LogEvent m_modName & "." & "mSVC_PRD_CFD_MAN" & Err.Description

    ErrLogAplicacion "c_GDI_NEG.c_CUR_NEG.mSVC_PRD_CFD_MAN", _
                    Array(epr_prd_cod, epr_prd_cbo), EA_DFTRBKCLSCR
    Err.Raise mErrNumber, mErrSource, mErrDescr
    Set Obj = Nothing
End Function

Public Function mSVC_PRD_CFD(Optional ByVal epr_prd_cod As Integer _
                                ) As String

    Dim Obj As c_GDI_DAT_COM.c_CUR_DATA

    On Error GoTo errorHandler

    Set Obj = CreateObject("c_GDI_DAT_COM.c_CUR_DATA")

    mSVC_PRD_CFD = mGeneraXML(Obj.mSVC_PRD_CFD(epr_prd_cod),
"Consulta")

    Set Obj = Nothing

    Exit Function

errorHandler:
    App.LogEvent m_modName & "." & "mSVC_PRD_CFD" & Err.Description

    ErrLogAplicacion "c_GDI_NEG.c_CUR_NEG.mSVC_PRD_CFD", _
                    Array(epr_prd_cod), EA_DFTRBKCLSCR
    Err.Raise mErrNumber, mErrSource, mErrDescr
    Set Obj = Nothing
End Function

Public Function mSVC_PRD_CFD_CBO(Optional ByVal epr_prd_cod As Integer
-
                                ) As String

    Dim Obj As c_GDI_DAT_COM.c_CUR_DATA

    On Error GoTo errorHandler

    Set Obj = CreateObject("c_GDI_DAT_COM.c_CUR_DATA")

    mSVC_PRD_CFD_CBO = mGeneraXML(Obj.mSVC_PRD_CFD_CBO(epr_prd_cod),
"Consulta")

```

```

        Set Obj = Nothing

Exit Function

errorHandler:
    App.LogEvent m_modName & "." & "mSVC_PRD_CFD_CB0" & Err.Description

    ErrLogAplicacion "c_GDI_NEG.c_CUR_NEG.mSVC_PRD_CFD_CB0", _
        Array(epr_prd_cod), EA_DFTRBKCLSCR
    Err.Raise mErrNumber, mErrSource, mErrDescr
    Set Obj = Nothing
End Function

Public Function mSVC_NUM_FOL(ByVal ofi_cod_ing As Integer, _
    ByVal usr_cod_ing As String, _
    ByVal jefe As String, _
    ByVal asistente As String, _
    ByVal fecha As Date, _
    ByVal pop_hra_act_ing As Date) As String

    Dim Obj As c_GDI_DAT_COM.c_CUR_DATA

    On Error GoTo errorHandler

    Set Obj = CreateObject("c_GDI_DAT_COM.c_CUR_DATA")

    mSVC_NUM_FOL = mGeneraXML(Obj.mSVC_NUM_FOL(ofi_cod_ing, _
        usr_cod_ing, _
        jefe, _
        asistente, _
        fecha, _
        pop_hra_act_ing), _
        "Consulta")

    Set Obj = Nothing

Exit Function

errorHandler:
    App.LogEvent m_modName & "." & "mSVC_NUM_FOL" & Err.Description

    ErrLogAplicacion "c_GDI_NEG.c_CUR_NEG.mSVC_NUM_FOL", _
        Array(ofi_cod_ing, usr_cod_ing, jefe, _
            asistente, fecha, pop_hra_act_ing),
EA_DFTRBKCLSCR
    Err.Raise mErrNumber, mErrSource, mErrDescr
    Set Obj = Nothing
End Function

Public Function mSVC_TTAB_USR_BUS_LGN(ByVal lgn As String _
    ) As String
    Dim Obj As c_GDI_DAT_COM.c_CUR_DATA

    On Error GoTo errorHandler

    Set Obj = CreateObject("c_GDI_DAT_COM.c_CUR_DATA")

    mSVC_TTAB_USR_BUS_LGN = mGeneraXML(Obj.mSVC_TTAB_USR_BUS_LGN(lgn),
"Consulta")

    Set Obj = Nothing

Exit Function

```

```

errorHandler:
    App.LogEvent m_modName & "." & "mSVC_TTAB_USR_BUS_LGN" &
Err.Description

    ErrLogAplicacion "c_GDI_NEG.c_CUR_NEG.mSVC_TTAB_USR_BUS_LGN", _
        Array(lgn), EA_DFTRBKCLSCR
    Err.Raise mErrNumber, mErrSource, mErrDescr
    Set Obj = Nothing
End Function

Public Function mSVC_TTAB_MON_BUS_EQA(ByVal mon_cod As Integer _
    ) As String
    Dim Obj As c_GDI_DAT_COM.c_CUR_DATA

    On Error GoTo errorHandler

    Set Obj = CreateObject("c_GDI_DAT_COM.c_CUR_DATA")

    mSVC_TTAB_MON_BUS_EQA =
mGeneraXML(Obj.mSVC_TTAB_MON_BUS_EQA(mon_cod), "Consulta")

    Set Obj = Nothing

Exit Function

errorHandler:
    App.LogEvent m_modName & "." & "mSVC_TTAB_MON_BUS_EQA" &
Err.Description

    ErrLogAplicacion "c_GDI_NEG.c_CUR_NEG.mSVC_TTAB_MON_BUS_EQA", _
        Array(mon_cod), EA_DFTRBKCLSCR
    Err.Raise mErrNumber, mErrSource, mErrDescr
    Set Obj = Nothing
End Function

Public Function mSVC_TTAB_MON_BUS_TOD() As String

    Dim Obj As c_GDI_DAT_COM.c_CUR_DATA

    On Error GoTo errorHandler

    Set Obj = CreateObject("c_GDI_DAT_COM.c_CUR_DATA")

    mSVC_TTAB_MON_BUS_TOD = mGeneraXML(Obj.mSVC_TTAB_MON_BUS_TOD(),
"Consulta")

    Set Obj = Nothing

Exit Function

errorHandler:
    App.LogEvent m_modName & "." & "mSVC_TTAB_MON_BUS_TOD" &
Err.Description

    ErrLogAplicacion "c_GDI_NEG.c_CUR_NEG.mSVC_TTAB_MON_BUS_TOD", _
        Array(), EA_DFTRBKCLSCR
    Err.Raise mErrNumber, mErrSource, mErrDescr
    Set Obj = Nothing
End Function

Public Function mSVC_TTAB_OFI_BUS_EQA(ByVal ofi_cod As Integer _
    ) As String
    Dim Obj As c_GDI_DAT_COM.c_CUR_DATA

    On Error GoTo errorHandler

```

```

        Set Obj = CreateObject("c_GDI_DAT_COM.c_CUR_DATA")

        mSVC_TTAB_OFI_BUS_EQA =
mGeneraXML(Obj.mSVC_TTAB_OFI_BUS_EQA(ofi_cod), "Consulta")

        Set Obj = Nothing

Exit Function

errorHandler:
    App.LogEvent m_modName & "." & "mSVC_TTAB_OFI_BUS_EQA" &
Err.Description

        ErrLogAplicacion "c_GDI_NEG.c_CUR_NEG.mSVC_TTAB_OFI_BUS_EQA", _
                        Array(ofi_cod), EA_DFTRBKCLSCR
        Err.Raise mErrNumber, mErrSource, mErrDescr
        Set Obj = Nothing
End Function

Public Function mSVC_DAT_USR(ByVal usr_cod As String _
                        ) As String
    Dim Obj As c_GDI_DAT_COM.c_CUR_DATA

    On Error GoTo errorHandler

    Set Obj = CreateObject("c_GDI_DAT_COM.c_CUR_DATA")

    mSVC_DAT_USR = mGeneraXML(Obj.mSVC_DAT_USR(usr_cod), "Consulta")

    Set Obj = Nothing

Exit Function

errorHandler:
    App.LogEvent m_modName & "." & "mSVC_DAT_USR" & Err.Description

        ErrLogAplicacion "c_GDI_NEG.c_CUR_NEG.mSVC_DAT_USR", _
                        Array(usr_cod), EA_DFTRBKCLSCR

        Err.Raise mErrNumber, mErrSource, mErrDescr
        Set Obj = Nothing
End Function

Public Function msvc_cse_ast_neg(ByVal cod_ofi As Integer) As String

    Dim Obj As c_GDI_DAT_COM.c_CUR_DATA

    On Error GoTo errorHandler

    Set Obj = CreateObject("c_GDI_DAT_COM.c_CUR_DATA")

        msvc_cse_ast_neg = mGeneraXML(Obj.msvc_cse_ast_neg(cod_ofi),
"Consulta")

        Set Obj = Nothing

Exit Function

errorHandler:
    App.LogEvent m_modName & "." & "msvc_cse_ast_neg" & Err.Description

        ErrLogAplicacion "c_GDI_NEG.c_CUR_NEG.msvc_cse_ast_neg", _
                        Array(cod_ofi), EA_DFTRBKCLSCR
        Err.Raise mErrNumber, mErrSource, mErrDescr

```



```

    Set Obj = Nothing
End Function

Public Function msvc_cse_jef_ope(ByVal cod_ofi As Integer) As String

    Dim Obj As c_GDI_DAT_COM.c_CUR_DATA

    On Error GoTo errorHandler

    Set Obj = CreateObject("c_GDI_DAT_COM.c_CUR_DATA")

    msvc_cse_jef_ope = mGeneraXML(Obj.msvc_cse_jef_ope(cod_ofi),
"Consulta")

    Set Obj = Nothing

Exit Function

errorHandler:
    App.LogEvent m_modName & "." & "msvc_cse_jef_ope" & Err.Description

    ErrLogAplicacion "c_GDI_NEG.c_CUR_NEG.msvc_cse_jef_ope", _
        Array(cod_ofi), EA_DFTRBKCLSCR
    Err.Raise mErrNumber, mErrSource, mErrDescr
    Set Obj = Nothing
End Function

Public Function msvc_cse_ast_tdo() As String

    Dim Obj As c_GDI_DAT_COM.c_CUR_DATA

    On Error GoTo errorHandler

    Set Obj = CreateObject("c_GDI_DAT_COM.c_CUR_DATA")

    msvc_cse_ast_tdo = mGeneraXML(Obj.msvc_cse_ast_tdo(), "Consulta")

    Set Obj = Nothing

Exit Function

errorHandler:
    App.LogEvent m_modName & "." & "msvc_cse_ast_tdo" & Err.Description

    ErrLogAplicacion "c_GDI_NEG.c_CUR_NEG.msvc_cse_ast_tdo", _
        Array(), EA_DFTRBKCLSCR
    Err.Raise mErrNumber, mErrSource, mErrDescr
    Set Obj = Nothing
End Function

Public Function mSVC_CGR_USR_PFL(ByVal NOM_PFL As String _
) As String
    Dim Obj As c_GDI_DAT_COM.c_CUR_DATA

    On Error GoTo errorHandler

    Set Obj = CreateObject("c_GDI_DAT_COM.c_CUR_DATA")

    mSVC_CGR_USR_PFL = mGeneraXML(Obj.mSVC_CGR_USR_PFL(NOM_PFL),
"Consulta")

    Set Obj = Nothing

Exit Function

```

```

errorHandler:
    App.LogEvent m_modName & "." & "mSVC_CGR_USR_PFL" & Err.Description

    ErrLogAplicacion "c_GDI_NEG.c_CUR_NEG.mSVC_CGR_USR_PFL", _
        Array(NOM_PFL), EA_DFTRBKCLSCR
    Err.Raise mErrNumber, mErrSource, mErrDescr
    Set Obj = Nothing
End Function

Public Function mSVC_OPE_SCA(ByVal ope_fol As Long _
    ) As String

    Dim Obj As c_GDI_DAT_COM.c_CUR_DATA

    On Error GoTo errorHandler

    Set Obj = CreateObject("c_GDI_DAT_COM.c_CUR_DATA")

    mSVC_OPE_SCA = mGeneraXML(Obj.mSVC_OPE_SCA(ope_fol), "Consulta")

    Set Obj = Nothing

Exit Function

errorHandler:
    App.LogEvent m_modName & "." & "mSVC_OPE_SCA" & Err.Description

    ErrLogAplicacion "c_GDI_NEG.c_CUR_NEG.mSVC_OPE_SCA", _
        Array(ope_fol), EA_DFTRBKCLSCR
    Err.Raise mErrNumber, mErrSource, mErrDescr
    Set Obj = Nothing

End Function

Public Function mSVC_VGT_EST_OPE(ByVal tip_consulta As Integer, _
    ByVal pop_fol_pos As Long, _
    ByVal ofi_pde As Integer, _
    ByVal estado1 As String, _
    ByVal estado2 As String, _
    ByVal estado3 As String, _
    ByVal pop_ide_sis As Integer _
    ) As String

    Dim Obj As c_GDI_DAT_COM.c_CUR_DATA

    On Error GoTo errorHandler

    Set Obj = CreateObject("c_GDI_DAT_COM.C_CUR_DATA")

    mSVC_VGT_EST_OPE = mGeneraXML(Obj.mSVC_VGT_EST_OPE(tip_consulta, _
        pop_fol_pos, _
        ofi_pde, _
        estado1, _
        estado2, _
        estado3, _
        pop_ide_sis), _
        "Consulta")

    Set Obj = Nothing

Exit Function

errorHandler:
    App.LogEvent m_modName & "." & "mSVC_VGT_EST_OPE" & Err.Description

```

```

        ErrLogAplicacion "c_GDI_NEG.c_CUR_NEG.mSVC_VGT_EST_OPE", _
                        Array(tip_consulta, pop_fol_pos, ofi_pde, _
                            estado1, estado2, estado3, pop_ide_sis),
EA_DFTRBKCLSCR
    Err.Raise mErrNumber, mErrSource, mErrDescr
    Set Obj = Nothing
End Function

Public Function mSVC_JEF_VGT_CRR(ByVal tip_consulta As Integer, _
                                ByVal pop_fec_ing As Date, _
                                ByVal pop_fec_est_hoy As Date, _
                                ByVal eop_est_cod As String, _
                                ByVal ofi_pde_cod As Integer _
                                ) As String
Dim Obj As c_GDI_DAT_COM.c_CUR_DATA

On Error GoTo errorHandler

    Set Obj = CreateObject("c_GDI_DAT_COM.C_CUR_DATA")

    mSVC_JEF_VGT_CRR = mGeneraXML(Obj.mSVC_JEF_VGT_CRR(tip_consulta, _
                                                        pop_fec_ing, _
                                                        pop_fec_est_hoy,
-
                                                        eop_est_cod, _
                                                        ofi_pde_cod), _
                                "Consulta")

    Set Obj = Nothing

Exit Function

errorHandler:
    App.LogEvent m_modName & "." & "mSVC_JEF_VGT_CRR" & Err.Description

    ErrLogAplicacion "c_GDI_NEG.c_CUR_NEG.mSVC_JEF_VGT_CRR", _
                    Array(tip_consulta, pop_fec_ing, pop_fec_est_hoy,
-
                        eop_est_cod, ofi_pde_cod), EA_DFTRBKCLSCR
    Err.Raise mErrNumber, mErrSource, mErrDescr
    Set Obj = Nothing
End Function

Public Function mSVC_CRR_OPE_CPN(ByVal tpcon As Integer, _
                                ByVal fecha_hoy As Date, _
                                ByVal eop_est_cod As String, _
                                ByVal ofi_cod_pde As Integer _
                                ) As String

Dim Obj As c_GDI_DAT_COM.c_CUR_DATA

On Error GoTo errorHandler

    Set Obj = CreateObject("c_GDI_DAT_COM.C_CUR_DATA")

    mSVC_CRR_OPE_CPN = mGeneraXML(Obj.mSVC_CRR_OPE_CPN(tpcon, _
                                                        fecha_hoy, _
                                                        eop_est_cod, _
                                                        ofi_cod_pde), _
                                "Consulta")

    Set Obj = Nothing

Exit Function

```

```

errorHandler:
    App.LogEvent m_modName & "." & "mSVC_CRR_OPE_CPN" & Err.Description

    ErrLogAplicacion "c_GDI_NEG.c_CUR_NEG.mSVC_CRR_OPE_CPN", _
        Array(tpcon, fecha_hoy, eop_est_cod, _
            ofi_pde_cod), EA_DFTRBKCLSCR
    Err.Raise mErrNumber, mErrSource, mErrDescr
    Set Obj = Nothing
End Function
Public Function mSVC_CRR_OPE_CPN_OFI(ByVal tpcon As Integer, _
    ByVal fecha_hoy As Date, _
    ByVal eop_est_cod As String, _
    ByVal ofi_cod As Integer _
    ) As String

Dim Obj As c_GDI_DAT_COM.c_CUR_DATA

On Error GoTo errorHandler

    Set Obj = CreateObject("c_GDI_DAT_COM.C_CUR_DATA")

    mSVC_CRR_OPE_CPN_OFI = mGeneraXML(Obj.mSVC_CRR_OPE_CPN_OFI(tpcon, _
        fecha_hoy, _
        eop_est_cod, _
        ofi_cod), _
        "Consulta")

    Set Obj = Nothing

Exit Function

errorHandler:
    App.LogEvent m_modName & "." & "mSVC_CRR_OPE_CPN_OFI" &
Err.Description

    ErrLogAplicacion "c_GDI_NEG.c_CUR_NEG.mSVC_CRR_OPE_CPN_OFI", _
        Array(tpcon, fecha_hoy, eop_est_cod, _
            ofi_cod), EA_DFTRBKCLSCR
    Err.Raise mErrNumber, mErrSource, mErrDescr
    Set Obj = Nothing
End Function

Public Function mSVC_CON_OFI_PDE() As String

Dim Obj As c_GDI_DAT_COM.c_CUR_DATA

On Error GoTo errorHandler

    Set Obj = CreateObject("c_GDI_DAT_COM.c_CUR_DATA")

    mSVC_CON_OFI_PDE = mGeneraXML(Obj.mSVC_CON_OFI_PDE(), "Consulta")

    Set Obj = Nothing

Exit Function

errorHandler:
    App.LogEvent m_modName & "." & "mSVC_CON_OFI_PDE" & Err.Description

    ErrLogAplicacion "c_GDI_NEG.c_CUR_NEG.mSVC_CON_OFI_PDE", _
        Array(), EA_DFTRBKCLSCR
    Err.Raise mErrNumber, mErrSource, mErrDescr
    Set Obj = Nothing
End Function

```

```

Public Function msvc_det_usr_lgn(ByVal login As String _
                                ) As String

    Dim Obj As c_GDI_DAT_COM.c_CUR_DATA

    On Error GoTo errorHandler

    Set Obj = CreateObject("c_GDI_DAT_COM.c_CUR_DATA")

    msvc_det_usr_lgn = mGeneraXML(Obj.msvc_det_usr_lgn(login), _
                                "Consulta")

    Set Obj = Nothing

Exit Function

errorHandler:
    App.LogEvent m_modName & "." & "msvc_det_usr_lgn" & Err.Description

    ErrLogAplicacion "c_GDI_NEG.c_CUR_NEG.msvc_det_usr_lgn", _
                    Array(login), EA_DFTRBKCLSCR
    Err.Raise mErrNumber, mErrSource, mErrDescr
    Set Obj = Nothing

End Function

Public Function mSVC_OPE_OFI_PDE(ByVal oficina As Integer _
                                ) As String

    Dim Obj As c_GDI_DAT_COM.c_CUR_DATA

    On Error GoTo errorHandler

    Set Obj = CreateObject("c_GDI_DAT_COM.c_CUR_DATA")

    mSVC_OPE_OFI_PDE = mGeneraXML(Obj.mSVC_OPE_OFI_PDE(oficina), _
                                "Consulta")

    Set Obj = Nothing

Exit Function

errorHandler:
    App.LogEvent m_modName & "." & "mSVC_OPE_OFI_PDE" & Err.Description

    ErrLogAplicacion "c_GDI_NEG.c_CUR_NEG.mSVC_OPE_OFI_PDE", _
                    Array(oficina), EA_DFTRBKCLSCR
    Err.Raise mErrNumber, mErrSource, mErrDescr
    Set Obj = Nothing

End Function

Public Function mSVA_OFI_SAT(ByVal opcion As Integer, _
                             ByVal ofi_cod As Integer, _
                             ByVal sat_tip_ofi As Integer, _
                             ByVal ofi_pde_cod As Integer, _
                             ByVal sat_dia_est As Integer _
                             ) As Boolean

    Dim Obj As c_GDI_DAT_COM.c_CUR_DATA

    On Error GoTo errorHandler

    Set Obj = CreateObject("c_GDI_DAT_COM.c_CUR_DATA")

    mSVA_OFI_SAT = Obj.mSVA_OFI_SAT(opcion, _
                                    ofi_cod, _

```

```

                                sat_tip_ofi, _
                                ofi_pde_cod, _
                                sat_dia_est)

    Set Obj = Nothing

Exit Function

errorHandler:
    App.LogEvent m_modName & "." & "mSVA_OFI_SAT" & Err.Description

    ErrLogAplicacion "c_GDI_NEG.c_CUR_NEG.mSVA_OFI_SAT", _
                    Array(opcion, ofi_cod, sat_tip_ofi, _
                        ofi_pde_cod, sat_dia_est), EA_DFTRBKCLSCR
    Err.Raise mErrNumber, mErrSource, mErrDescr
    Set Obj = Nothing
End Function

Public Function mSVA_MTV_RCH(ByVal opcion As Integer, _
                            ByVal mrc_mtv_cod As Integer, _
                            ByVal mrc_gls As String, _
                            ByVal mrc_tip_rch As Integer _
                            ) As Boolean
    Dim Obj As c_GDI_DAT_COM.c_CUR_DATA

    On Error GoTo errorHandler

    Set Obj = CreateObject("c_GDI_DAT_COM.c_CUR_DATA")

    mSVA_MTV_RCH = Obj.mSVA_MTV_RCH(opcion, _
                                    mrc_mtv_cod, _
                                    mrc_gls, _
                                    mrc_tip_rch)

    Set Obj = Nothing

Exit Function

errorHandler:
    App.LogEvent m_modName & "." & "mSVA_MTV_RCH" & Err.Description

    ErrLogAplicacion "c_GDI_NEG.c_CUR_NEG.mSVA_MTV_RCH", _
                    Array(opcion, mrc_mtv_cod, mrc_gls, _
                        mrc_tip_rch), EA_DFTRBKCLSCR
    Err.Raise mErrNumber, mErrSource, mErrDescr
    Set Obj = Nothing

End Function

Public Function mSVA_EST_COD(ByVal opcion As Integer, _
                            ByVal eop_est_cod As String, _
                            ByVal eop_est_gls As String _
                            ) As Boolean
    Dim Obj As c_GDI_DAT_COM.c_CUR_DATA

    On Error GoTo errorHandler

    Set Obj = CreateObject("c_GDI_DAT_COM.c_CUR_DATA")

    mSVA_EST_COD = Obj.mSVA_EST_COD(opcion, _
                                    eop_est_cod, _
                                    eop_est_gls)

    Set Obj = Nothing

```

```

Exit Function

errorHandler:
    App.LogEvent m_modName & "." & "mSVA_EST_COD" & Err.Description

    ErrLogAplicacion "c_GDI_NEG.c_CUR_NEG.mSVA_EST_COD", _
        Array(opcion, eop_est_cod, eop_est_gls),
EA_DFTRBKCLSCR
    Err.Raise mErrNumber, mErrSource, mErrDescr
    Set Obj = Nothing

End Function

Public Function mSVA_DEL_PRD_OPE(ByVal epr_prd_cod As Integer _
    ) As Boolean
    Dim Obj As c_GDI_DAT_COM.c_CUR_DATA

    On Error GoTo errorHandler

    Set Obj = CreateObject("c_GDI_DAT_COM.c_CUR_DATA")

    mSVA_DEL_PRD_OPE = Obj.mSVA_DEL_PRD_OPE(epr_prd_cod)

    Set Obj = Nothing

Exit Function

errorHandler:
    App.LogEvent m_modName & "." & "mSVA_DEL_PRD_OPE" & Err.Description

    ErrLogAplicacion "c_GDI_NEG.c_CUR_NEG.mSVA_DEL_PRD_OPE", _
        Array(epr_prd_cod), EA_DFTRBKCLSCR
    Err.Raise mErrNumber, mErrSource, mErrDescr
    Set Obj = Nothing

End Function

Public Function mSVA_DEL_OFI_SAT(ByVal ofi_cod As Integer _
    ) As Boolean
    Dim Obj As c_GDI_DAT_COM.c_CUR_DATA

    On Error GoTo errorHandler

    Set Obj = CreateObject("c_GDI_DAT_COM.c_CUR_DATA")

    mSVA_DEL_OFI_SAT = Obj.mSVA_DEL_OFI_SAT(ofi_cod)

    Set Obj = Nothing

Exit Function

errorHandler:
    App.LogEvent m_modName & "." & "mSVA_DEL_OFI_SAT" & Err.Description

    ErrLogAplicacion "c_GDI_NEG.c_CUR_NEG.mSVA_DEL_OFI_SAT", _
        Array(ofi_cod), EA_DFTRBKCLSCR
    Err.Raise mErrNumber, mErrSource, mErrDescr
    Set Obj = Nothing
End Function

Public Function mSVA_DEL_MTV_RCH(ByVal mrc_mtv_cod As Integer _
    ) As Boolean
    Dim Obj As c_GDI_DAT_COM.c_CUR_DATA

    On Error GoTo errorHandler

```

```

        Set Obj = CreateObject("c_GDI_DAT_COM.c_CUR_DATA")

        mSVA_DEL_MTV_RCH = Obj.mSVA_DEL_MTV_RCH(mrc_mtv_cod)

        Set Obj = Nothing

Exit Function

errorHandler:
    App.LogEvent m_modName & "." & "mSVA_DEL_MTV_RCH" & Err.Description

    ErrLogAplicacion "c_GDI_NEG.c_CUR_NEG.mSVA_DEL_MTV_RCH", _
        Array(mrc_mtv_cod), EA_DFTRBKCLSCR
    Err.Raise mErrNumber, mErrSource, mErrDescr
    Set Obj = Nothing
End Function

Public Function mSVA_DEL_EST_COD(ByVal eop_est_cod As String _
    ) As Boolean
    Dim Obj As c_GDI_DAT_COM.c_CUR_DATA

    On Error GoTo errorHandler

    Set Obj = CreateObject("c_GDI_DAT_COM.c_CUR_DATA")

    mSVA_DEL_EST_COD = Obj.mSVA_DEL_EST_COD(eop_est_cod)

    Set Obj = Nothing

Exit Function

errorHandler:
    App.LogEvent m_modName & "." & "mSVA_DEL_EST_COD" & Err.Description

    ErrLogAplicacion "c_GDI_NEG.c_CUR_NEG.mSVA_DEL_EST_COD", _
        Array(eop_est_cod), EA_DFTRBKCLSCR
    Err.Raise mErrNumber, mErrSource, mErrDescr
    Set Obj = Nothing
End Function

Public Function mSVC_RCH_MAN(ByVal mrc_mtv_cod As Integer _
    ) As String

    Dim Obj As c_GDI_DAT_COM.c_CUR_DATA

    On Error GoTo errorHandler

    Set Obj = CreateObject("c_GDI_DAT_COM.c_CUR_DATA")

    mSVC_RCH_MAN = mGeneraXML(Obj.mSVC_RCH_MAN(mrc_mtv_cod), _
        "Consulta")

    Set Obj = Nothing

Exit Function

errorHandler:
    App.LogEvent m_modName & "." & "mSVC_RCH_MAN" & Err.Description

    ErrLogAplicacion "c_GDI_NEG.c_CUR_NEG.mSVC_RCH_MAN", _
        Array(mrc_mtv_cod), EA_DFTRBKCLSCR
    Err.Raise mErrNumber, mErrSource, mErrDescr
    Set Obj = Nothing
End Function

```



```

Public Function mSVC_EXI_PRD_OPE(ByVal epr_prd_cod As Integer _
                                ) As String

    Dim Obj As c_GDI_DAT_COM.c_CUR_DATA

    On Error GoTo errorHandler

    Set Obj = CreateObject("c_GDI_DAT_COM.c_CUR_DATA")

    mSVC_EXI_PRD_OPE = mGeneraXML(Obj.mSVC_EXI_PRD_OPE(epr_prd_cod), _
                                "Consulta")

    Set Obj = Nothing

Exit Function

errorHandler:
    App.LogEvent m_modName & "." & "mSVC_EXI_PRD_OPE" & Err.Description

    ErrLogAplicacion "c_GDI_NEG.c_CUR_NEG.mSVC_EXI_PRD_OPE", _
                    Array(epr_prd_cod), EA_DFTRBKCLSCR
    Err.Raise mErrNumber, mErrSource, mErrDescr
    Set Obj = Nothing
End Function

Public Function mSVC_RCH_EXI_OPE(ByVal mrc_mtv_cod As Integer _
                                ) As String

    Dim Obj As c_GDI_DAT_COM.c_CUR_DATA

    On Error GoTo errorHandler

    Set Obj = CreateObject("c_GDI_DAT_COM.c_CUR_DATA")

    mSVC_RCH_EXI_OPE = mGeneraXML(Obj.mSVC_RCH_EXI_OPE(mrc_mtv_cod), _
                                "Consulta")

    Set Obj = Nothing

Exit Function

errorHandler:
    App.LogEvent m_modName & "." & "mSVC_RCH_EXI_OPE" & Err.Description

    ErrLogAplicacion "c_GDI_NEG.c_CUR_NEG.mSVC_RCH_EXI_OPE", _
                    Array(mrc_mtv_cod), EA_DFTRBKCLSCR
    Err.Raise mErrNumber, mErrSource, mErrDescr
    Set Obj = Nothing
End Function

Public Function mSVC_OPE_HIS_EST(ByVal eop_est_cod As String _
                                ) As String

    Dim Obj As c_GDI_DAT_COM.c_CUR_DATA

    On Error GoTo errorHandler

    Set Obj = CreateObject("c_GDI_DAT_COM.c_CUR_DATA")

    mSVC_OPE_HIS_EST = mGeneraXML(Obj.mSVC_OPE_HIS_EST(eop_est_cod), _
                                "Consulta")

    Set Obj = Nothing

Exit Function

```

```

errorHandler:
    App.LogEvent m_modName & "." & "mSVC_OPE_HIS_EST" & Err.Description

    ErrLogAplicacion "c_GDI_NEG.c_CUR_NEG.mSVC_OPE_HIS_EST", _
        Array(eop_est_cod), EA_DFTRBKCLSCR
    Err.Raise mErrNumber, mErrSource, mErrDescr
    Set Obj = Nothing
End Function

Public Function mSVC_PRD_VGT_HIS(ByVal epr_prd_cod As Integer _
                                ) As String

    Dim Obj As c_GDI_DAT_COM.c_CUR_DATA

    On Error GoTo errorHandler

    Set Obj = CreateObject("c_GDI_DAT_COM.c_CUR_DATA")

    mSVC_PRD_VGT_HIS = mGeneraXML(Obj.mSVC_PRD_VGT_HIS(epr_prd_cod), _
                                   "Consulta")

    Set Obj = Nothing

Exit Function

errorHandler:
    App.LogEvent m_modName & "." & "mSVC_PRD_VGT_HIS" & Err.Description

    ErrLogAplicacion "c_GDI_NEG.c_CUR_NEG.mSVC_PRD_VGT_HIS", _
        Array(epr_prd_cod), EA_DFTRBKCLSCR
    Err.Raise mErrNumber, mErrSource, mErrDescr
    Set Obj = Nothing
End Function

Public Function mSVC_OPE_OFI_BTG(ByVal opcion As Integer, _
                                ByVal ofi_cod As Integer _
                                ) As String

    Dim Obj As c_GDI_DAT_COM.c_CUR_DATA

    On Error GoTo errorHandler

    Set Obj = CreateObject("c_GDI_DAT_COM.c_CUR_DATA")

    mSVC_OPE_OFI_BTG = mGeneraXML(Obj.mSVC_OPE_OFI_BTG(opcion,
ofi_cod), _
                                   "Consulta")

    Set Obj = Nothing

Exit Function

errorHandler:
    App.LogEvent m_modName & "." & "mSVC_OPE_OFI_BTG" & Err.Description

    ErrLogAplicacion "c_GDI_NEG.c_CUR_NEG.mSVC_OPE_OFI_BTG", _
        Array(opcion, ofi_cod), EA_DFTRBKCLSCR
    Err.Raise mErrNumber, mErrSource, mErrDescr
    Set Obj = Nothing
End Function

Public Function mSVA_CIE_DIA(ByVal pop_fol_ope As String, _
                             ByVal fecha_hoy As Date, _
                             ByVal hora_hoy As Date _

```

```

        ) As Boolean

Dim Obj As c_GDI_DAT_COM.c_CUR_DATA

On Error GoTo errorHandler

Set Obj = CreateObject("c_GDI_DAT_COM.c_CUR_DATA")

mSVA_CIE_DIA = Obj.mSVA_CIE_DIA(pop_fol_ope, _
                                fecha_hoy, _
                                hora_hoy)

Set Obj = Nothing

Exit Function

errorHandler:
App.LogEvent m_modName & "." & "mSVA_CIE_DIA" & Err.Description

ErrLogAplicacion "c_GDI_NEG.c_CUR_NEG.mSVA_CIE_DIA", _
                Array(pop_fol_ope, fecha_hoy, hora_hoy),
EA_DFTRBKCLSCR
Err.Raise mErrNumber, mErrSource, mErrDescr
Set Obj = Nothing
End Function

Public Function mSVC_CSU_CIE_DIA(ByVal pop_fol_ope As String, _
                                ByVal oficina As Integer, _
                                ByVal fecha_hoy As Date _
                                ) As String

Dim Obj As c_GDI_DAT_COM.c_CUR_DATA

On Error GoTo errorHandler

Set Obj = CreateObject("c_GDI_DAT_COM.c_CUR_DATA")

mSVC_CSU_CIE_DIA = mGeneraXML(Obj.mSVC_CSU_CIE_DIA(pop_fol_ope, _
                                                    oficina, _
                                                    fecha_hoy), "Consulta")

Set Obj = Nothing

Exit Function

errorHandler:
App.LogEvent m_modName & "." & "mSVC_CSU_CIE_DIA" & Err.Description

ErrLogAplicacion "c_GDI_NEG.c_CUR_NEG.mSVC_CSU_CIE_DIA", _
                Array(pop_fol_ope, oficina, fecha_hoy),
EA_DFTRBKCLSCR
Err.Raise mErrNumber, mErrSource, mErrDescr
Set Obj = Nothing
End Function

Public Function mSVC_OPE_ADM(ByVal usr_cod As String) As String

Dim Obj As c_GDI_DAT_COM.c_CUR_DATA

On Error GoTo errorHandler

Set Obj = CreateObject("c_GDI_DAT_COM.c_CUR_DATA")

mSVC_OPE_ADM = mGeneraXML(Obj.mSVC_OPE_ADM(usr_cod), "Consulta")

```

```

        Set Obj = Nothing

Exit Function

errorHandler:
    App.LogEvent m_modName & "." & "mSVC_OPE_ADM" & Err.Description

    ErrLogAplicacion "c_GDI_NEG.c_CUR_NEG.mSVC_OPE_ADM", _
        Array(usr_cod), EA_DFTRBKCLSCR
    Err.Raise mErrNumber, mErrSource, mErrDescr
    Set Obj = Nothing
End Function

Public Function mSVC_OPE_SUP_ADM_CPN(eop_est_cod As String, _
                                     tot As Integer) As String

    Dim Obj As c_GDI_DAT_COM.c_CUR_DATA

    On Error GoTo errorHandler

    Set Obj = CreateObject("c_GDI_DAT_COM.c_CUR_DATA")

    mSVC_OPE_SUP_ADM_CPN =
mGeneraXML(Obj.mSVC_OPE_SUP_ADM_CPN(eop_est_cod, _
                                     tot),
"Consulta")

    Set Obj = Nothing

Exit Function

errorHandler:
    App.LogEvent m_modName & "." & "mSVC_OPE_SUP_ADM_CPN" &
Err.Description

    ErrLogAplicacion "c_GDI_NEG.c_CUR_NEG.mSVC_OPE_SUP_ADM_CPN", _
        Array(eop_est_cod, tot), EA_DFTRBKCLSCR
    Err.Raise mErrNumber, mErrSource, mErrDescr
    Set Obj = Nothing
End Function

Public Function mSVC_SGM_CLS_PRD_CFD(ByVal segmento As Integer, _
                                     ByVal clase As Integer) As String
Dim Obj As c_GDI_DAT_COM.c_CUR_DATA

    On Error GoTo errorHandler

    Set Obj = CreateObject("c_GDI_DAT_COM.c_CUR_DATA")

    mSVC_SGM_CLS_PRD_CFD = mGeneraXML(Obj.mSVC_SGM_CLS_PRD_CFD(segmento,
_
                                     clase),
"Consulta")

    Set Obj = Nothing

Exit Function

errorHandler:
    App.LogEvent m_modName & "." & "mSVC_SGM_CLS_PRD_CFD" &
Err.Description

    ErrLogAplicacion "c_GDI_NEG.c_CUR_NEG.mSVC_SGM_CLS_PRD_CFD", _
        Array(segmento, clase), EA_DFTRBKCLSCR
    Err.Raise mErrNumber, mErrSource, mErrDescr

```

```

    Set Obj = Nothing
End Function

Public Function mSVC_REL_SGM_CLS_PRD_XML(ByVal segmento As Integer, _
                                         ByVal clase As Integer) As
String

    Dim Obj As c_GDI_DAT_COM.c_CUR_DATA
    Dim objXmlSalida As STRCATLib.Catter
    Dim rsrec As ADODB.Recordset

    On Error GoTo errorHandler

    Set objXmlSalida = CreateObject("Strcat.Catter.1")
    objXmlSalida.Append "<SALIDA><ERROR><CODERROR>"

    Set Obj = CreateObject("c_GDI_DAT_COM.c_CUR_DATA")
    Set rsrec = Obj.mSVC_SGM_CLS_PRD_CFD(segmento, clase)

    If Not rsrec.EOF Then
        objXmlSalida.Append "00</CODERROR></ERROR><DATOS>"
        Do While Not rsrec.EOF
            objXmlSalida.Append "<REGISTRO>"
            objXmlSalida.Append "<CODPRD>"
            objXmlSalida.Append rsrec("epr_prd_cod")
            objXmlSalida.Append "</CODPRD>"
            objXmlSalida.Append "<GLSPRD>"
            objXmlSalida.Append rsrec("epr_gls_prd")
            objXmlSalida.Append "</GLSPRD>"
            objXmlSalida.Append "<COMBO>"
            objXmlSalida.Append rsrec("epr_prd_cbo")
            objXmlSalida.Append "</COMBO>"
            objXmlSalida.Append "<IDRCOMBO>"
            objXmlSalida.Append rsrec("pcf_idr_cbo")
            objXmlSalida.Append "</IDRCOMBO>"
            objXmlSalida.Append "</REGISTRO>"
            rsrec.MoveNext
        Loop
        objXmlSalida.Append "</DATOS></SALIDA>"
    Else
        objXmlSalida.Append
"11</CODERROR><TIPERROR/><GLSEERROR>VACIO</GLSEERROR></ERROR></SALIDA>"
    End If
    mSVC_REL_SGM_CLS_PRD_XML = objXmlSalida.Resolve

    Set objXmlSalida = Nothing
    Set Obj = Nothing
    rsrec.Close
    Set rsrec = Nothing

Exit Function

errorHandler:
    App.LogEvent m_modName & "." & "mSVC_REL_SGM_CLS_PRD_XML" &
Err.Description

    ErrLogAplicacion
"c_GDI_DAT_COM.c_CUR_DATA.C_CUR_NEG.mSVC_REL_SGM_CLS_PRD_XML", _
        Array(segmento, clase), EA_DFTRBKCLSCR
    Err.Raise mErrNumber, mErrSource, mErrDescr
    Set objXmlSalida = Nothing
    Set Obj = Nothing
    rsrec.Close
    Set rsrec = Nothing
End Function

```

ANEXO 2: DICCIONARIO DATOS

En este anexo se explicará el diccionario de datos utilizado para cada tabla de la Base de Datos BGDIMG, modelado para SQL server 7.0.

Tabla: Estados		
Contiene la descripción de los estados por los cuales podría pasar una operación.		
Campo	Tipo de dato	Descripción
eop_est_cod	Char(2)	Código del tipo de estado
eop_est_gls	Char(60)	Descripción del estado

Tabla: Estados Históricos de la operación		
Contiene todos los estados por los cuales pasa una operación. Esta tabla es histórica de la tabla Estados		
Campo	Tipo de dato	Descripción
His_num_crv	Integer	Correlativo de la tabla
Pop_fol_ope	Integer	Folio interno de la operación
Usr_cod	Char(10)	Usuario que realiza el cambio de estado
Eop_est_cod	Char(2)	Codigo estado operación
His_fec_est	Datetime	Fecha en que se realizó el cambio de estado
His_hra_est	Datetime	Hora en que se realizó el cambio de estado.

Tabla: Motivos de Rechazo		
Tipo de rechazos de una operación		
Campo	Tipo de dato	Descripción
mrc_mtv_cod	Integer	Códigos motivos de rechazos
mrc_tip_rch	Smallint	Tipo de rechazos.

Mrc_gls	Char(200)	Descripción de los motivos de rechazo
---------	-----------	---------------------------------------

Tabla: Oficinas Satélites Contiene la información de las oficinas que pueden participar en el GDI.		
Campo	Tipo de dato	Descripción
ofi_cod	Integer	Código oficina
sat_tip_ofi	Integer	Tipo de oficina
ofi_pde_cod	Integer	Código Oficina Padre(Centralizadora)
sat_dia_esr	Integer	Número de días que debe esperar el CPN la llegada de la carpeta física.

Tabla: Operaciones Rechazadas Contiene la información de las operaciones que han sido rechazadas.		
Campo	Tipo de dato	Descripción
pop_fol_ope	Integer	Folio Interno de la operación
mrc_mtv_cod	Integer	Código del rechazo
usr_cod	Char(10)	Usuario que rechazó la operación
orc_fec_rch	Datetime	Fecha en que fue rechazada la operación
orc_obs_rch	Varchar(300)	Observación al rechazo

Tabla: Documentos Digitalizados Contiene la las imágenes en formato binario de los documentos digitalizados.		
Campo	Tipo de dato	Descripción
ope_fol	Integer	Folio interno
sca_num_crv	Integer	Correlativo de la imagen
sca_img_ope	Image	Imagen asociada a la operación

Tabla: Productos Combo(más de un producto asociado a la operación) Contiene la información de los productos combos asociados a la operación		
--	--	--

Campo	Tipo de dato	Descripción
epr_prd_cbo	Integer	Código del producto combo
epr_prd_cod	Integer	Código del producto
pop_fol_ope	Integer	Folio interno de la operación

Tabla: Productos

Contiene la información de los productos vigentes a utilizar.

Campo	Tipo de dato	Descripción
epr_prd_cod	Integer	Código del producto
epr_prd_cbo	Integer	Código producto combo
epr_gls_prd	Nvarchar(500)	Nombre de productos
pcf_idr_cbo	Char(1)	Identifica si es o no producto combo

Tabla: Operaciones

Contiene la información en detalle de una operación vigente

Campo	Tipo de dato	Descripción
pop_fol_ope	Integer	Identificador de la operación(Folio interno)
mon_cod	Char(3)	Código de la moneda
ofi_cod	Integer	Código de la oficina de curse
eop_est_cod	Char(2)	Código del estado de la operación
usr_cod	Char(10)	Identificador del último usuario que interactúo con la operación
cli_rut	Numeric(9)	Rut del cliente
cli_rut_dv	Char(1)	Dígito verificador del rut del cliente
pop_nom_no_cli	Varchar(30)	Nombre del cliente
pop_cpa_cod	Char(20)	Código de barra de la carpeta física que contiene la operación
usr_jef_cod	Char(10)	Código del jefe de operaciones
usr_adm_cod	Char(10)	Código del administrativo CPN
usr_ast_cod	Char(10)	Código del asistente de negocios en sucursales

usr_spv_cod	Char(10)	Código del supervisor de CPN
pop_fec_ing	Datetime	Fecha de ingreso de la operación
pop_mnt_brt	Numeric(13,2)	Monto bruto de la operación
pop_fec_est	Datetime	Fecha de estado de la operación
pop_hra_act	Datetime	Hora de actualización del estado
pop_ide_sis	Integer	Identificador del sistema por el cual se está cursando la operación.
pop_fec_cie	Datetime	Fecha de corte del proceso diario en sucursales.
pop_hra_cie	datetime	Hora de cierre de operaciones en sucursales.

Tabla: Histórico de estados de la operación
 Contiene la información histórica de todos los estados posibles por los cuales pasa la operación.

Campo	Tipo de dato	Descripción
her_crv	Integer	Correlativo interno
ohi_fol_ope	Integer	Folio de la operación
eop_est_cod	Char(2)	Código del estado
her_usr_cod	Char(10)	Usuario actualizador
hes_fec	Datetime	Fecha en que se realizó el respaldo histórico
hes_hra	Datetime	Hora en que se realizó el respaldo.

Tabla: Histórico Productos Combo(más de un producto asociado a la operación)
 Contiene la información de los productos combos históricos asociados a la operación

Campo	Tipo de dato	Descripción
epr_prd_cbo	Integer	Código del producto combo
epr_prd_cod	Integer	Código del producto
ohi_fol_ope	Integer	Folio interno de la operación

Tabla: Operaciones Históricas
 Contiene la información en detalle de una operación histórica.

Campo	Tipo de dato	Descripción
ohi_fol_ope	Integer	Identificador de la operación(Folio interno)
mon_cod	Char(3)	Código de la moneda
ofi_cod	Integer	Código de la oficina de curse
eop_est_cod	Char(2)	Código del estado de la operación
usr_cod	Char(10)	Identificador del último usuario que interactúo con la operación
cli_rut	Numeric(9)	Rut del cliente
cli_rut_dv	Char(1)	Dígito verificador del rut del cliente
ohi_nom_no_cli	Varchar(30)	Nombre del cliente
ohi_cpa_cod	Char(20)	Código de barra de la carpeta física que contiene la operación
ohi_usr_cod_jef	Char(10)	Código del jefe de operaciones
ohi_usr_cod_adm	Char(10)	Código del administrativo CPN
ohi_usr_cod_ast	Char(10)	Código del asistente de negocios en sucursales
ohi_usr_cod_sup	Char(10)	Código del supervisor de CPN
ohi_fec_ing	Datetime	Fecha de ingreso de la operación
ohi_mnt_brt	Numeric(13,2)	Monto bruto de la operación
ohi_fec_his	Datetime	Fecha que se traspasó la operación a histórica.
ohi_fec_est	Datetime	Fecha de estado de la operación histórica
ohi_hra_act	Datetime	Hora de actualización del estado
ohi_ide_sis	Integer	Identificador del sistema por el cual se está cursando la operación.

Tabla: Operaciones Históricas Rechazadas
 Contiene la información de las operaciones históricas que han sido rechazadas.

Campo	Tipo de dato	Descripción
ohi_fol_ope	Integer	Folio Interno de la operación

Mrc_mtv_cod	Integer	Código del rechazo
Usr_cod	Char(10)	Usuario que rechazó la operación
Orh_fec_rch	Datetime	Fecha en que fue rechazada la operación
Orh_obs_rch	Varchar(300)	Observación al rechazo

ANEXO 3: INFORME PRUEBA CONTROLADA PUNTA ARENAS -

CPN

INFORME DE EVALUACIÓN

El Proyecto Piloto Curse con Imágenes realizado en Punta Arenas, se dividió en dos Fases:

Fase 1: Prueba Controlada:

Período: 2-4 de Diciembre.

Objetivo: Prueba del Modelo de Curse con Imágenes definido con operaciones controladas.

Iniciar un proceso controlado enviando operaciones vía imágenes al CPN de acuerdo al Procedimiento establecido, realizando un seguimiento completo al proceso en la Sucursal y Centro de Procesos Masivos.

Sucursal: Capacitación a uno o dos Asistentes de Negocios de la banca de Personas, Encargado de Correspondencia para la digitalización y envío de imágenes al CPN, y el Jefe de Operaciones para la certificación de imágenes, control y cierre el Proceso diario con el CPN.

Dotación CPN: 1 supervisor de curse con imágenes, 2 analistas de control de Operaciones, 1 supervisor de curse, 1 cursador, 1 visador capacitados para operar con nuevo modelo.

Dotación presencial Subgerencia de Ingeniería de Procesos:

Sucursal:

2, 3 de Diciembre: 1 persona.

4 de Diciembre: 3 personas.

CPN:

2, 3 de Diciembre: 2 personas, más supervisión designada para controlar proceso.

4 de Diciembre: 1 persona, más supervisión designada para controlar proceso.

Tipo de Operaciones cursadas: Segmento Personas.

Hito de Cierre Fase 1: Primera Evaluación, acciones posibles:

Modelo probado con en Sucursal y CPN necesita ajustes en procedimientos, controles y herramientas y con resultados obtenidos se decide cómo continuar el proceso.

Resultado Satisfactorio del Modelo, Modelo opera normalmente, se da inicio a Fase 2, PILOTO.

Fase 2: PILOTO:

Período: 5-6 de Diciembre.

Objetivo: Realizar el primer Stress Testing del Modelo probado en Fase 1, incorporando todos los asistentes de negocios de la Banca de personas y aumentando el volumen de Operaciones enviadas vía imágenes al CPN.

Sucursal: Capacitación a la totalidad de Asistentes de Negocios de la Banca de Personas, Encargado de Correspondencia para reemplazo, y Jefe de Atención a clientes para reemplazo de rol asignada a Jefe de Operaciones para el proceso.

Dotación CPN: Misma dotación definida para prueba controlada.

Dotación presencial Subgerencia de Ingeniería de Procesos:

Sucursal:

5 y 6 de Diciembre: 1 persona.

CPN:

5 de Diciembre: 1 persona.

6 de Diciembre: 2 personas

Tipo de Operaciones cursadas: Segmento Personas.

Hito de Cierre Fase 2: 2da. Evaluación, acciones posibles:

Modelo necesita ajustes para procedimiento en Sucursal y CPN, con resultados obtenidos se decide cómo continuar el proceso.

Resultado Satisfactorio del Modelo, Modelo opera normalmente, sólo son necesarios ciertos ajustes, Se decide continuar con Fase 2 de Piloto, incorporando ejecutivos segmentos MYPE y Empresas, hasta la entrada en Régimen de la Sucursal para el nuevo Modelo de operación.

RESULTADOS 1RA. EVALUCIÓN FASE 1: PRUEBA CONTROLADA

El informe contempla las actividades realizadas en la Sucursal y en CPN, además de sugerencias y mejoras propuestas en los ámbitos de Sucursales, CPN y Apoyo Tecnológico.

ACTIVIDAD EN LA SUCURSAL:

Las actividades de la Sucursal se analizarán según los actores participantes definidos en el Modelo: Asistentes de Negocios - Plataforma Comercial, Encargado de envío de imágenes, Jefe de Operaciones.

Ejecutivos de Plataforma Comercial:

Se capacitó y entregó formalmente los procedimientos a la totalidad de los ejecutivos de Banca de Personas, a 1 Ejecutivo de Empresas y 1 Ejecutivo de Microempresa, advirtiéndoles desde el inicio de la semana que estos dos últimos segmentos no enviarían operaciones en esta primera semana del Piloto, trasladando este segmento para la semana del 9 al 13 de Diciembre (2da. Semana proyecto piloto).

La principal observación de los ejecutivos tiene que ver con el envío del correo al Jefe de Operaciones, informando del envío de operaciones con imágenes al CPN, el cual en su opinión les consume tiempo. Se les explicó la necesidad de contar con ese control, y el mínimo tiempo consumido para informar sólo 5 datos, el resto de las actividades del Asistente es similar a su labor antes de este proceso.

Es relevante para un cumplimiento de los horarios de cierre de envío a CPN, optimizar los flujos internos de envío a firmas de comité, de tal forma que idealmente las carpetas a enviar al CPN lleguen ya preparadas a las firmas de Comité y que no se confeccionen después. El alto flujo de atención de público en horario bancario le impide a cada ejecutivo entregarle temprano al encargado de envío de imágenes las carpetas, concentrando en consecuencia en poco tiempo el flujo de envío de imágenes, (Conversado con el Sr. Agente quien implementará una solución al respecto).

Producto del envío exitoso de los dos primeros días de Prueba Controlada, se decidió enviar las operaciones del día miércoles para toda la Plataforma Personas, alcanzando un volumen de 9 operaciones, algunas de las cuales fueron rechazadas en el CPN por corresponder el pagaré al día siguiente.

Encargado de envío de imágenes:

Capacitado en procedimientos, el uso de la Aplicación para la digitalización y envío de imágenes y utilización de scanner. Sin dificultades técnicas, bastante hábil para su utilización y con fortalezas para el control de las operaciones. El flujo de actividades con el Jefe de Operaciones es fluido y coordinado.

Se capacitó adicionalmente un reemplazante para casos de contingencia.

Partió el primer día demorando hasta 10 minutos en una operación de 4 imágenes, disminuyendo en el transcurso de los días a 5 minutos para la operación con mayor número de papeles (4), incluidas actividades manuales. Puede mejorar algo más con práctica, pero no será nunca menor a 3 minutos producto del viaje de imágenes.

Se recomienda por el volumen de operaciones de la Sucursal que el cierre del proceso de digitalización y envío de imágenes para la recepción de carpetas para envío, cierre una hora antes del cierre establecido para envío de planilla de cierre diario al CPN, y que idealmente tenga un flujo continuo durante el día, evitando así una concentración de operaciones en horario de cierre.

Jefe de Operaciones:

Capacitado en procedimientos, herramienta para consulta y certificación de imágenes y en el flujo completo de la sucursal. El énfasis de su tarea está en la certificación de imágenes, en el control del flujo y llenado de planillas para cierre y control con CPN. Se le generaron las carpetas

para archivo electrónico, y se trabajó en aspectos metodológicos para cuidar el orden y control de las tareas.

También fue capacitado su reemplazante para situaciones de contingencia.

Se coordinó con él y el Sr. Agente para que en los horarios de cierre de proceso diario, este funcionario no sea incluido en reuniones que distraigan su atención al cumplimiento de esta responsabilidad crítica para la Sucursal, o en su defecto opere el reemplazante.

ACTIVIDADES CENTRO DE PROCESOS NACIONAL:

Se elaborará un informe detallado por unidad este viernes, incluidas las mediciones de tiempos.

Las generales en esta fase son:

De las 3 personas asignadas en control de operaciones (incluido el Supervisor), sólo se cuenta con 2 de ellas hasta las 18:00 hrs. la tercera se retira a las 16:00, el primer día el Supervisor no efectuó el cierre de control con Sucursal y quedó para el día siguiente, problema a la fecha ya corregido.

En líneas generales los tiempos de procesos y productividad deben ser analizados con detalle para definir con precisión dotaciones, tiempos de servicio, y por ende instruir a sucursales en consecuencia para las fechas de simulación de las operaciones.

El día miércoles hubo 3 devoluciones de operaciones a la Sucursal, lo que incrementa los tiempos de procesos en el CPN. La actitud del CPN a este respecto es ser lo más estricto posible, por lo cual la Sucursal debe ser reforzada permanentemente en relación a que deben realizar todos los controles necesarios antes de enviar las imágenes al CPN y que de esta manera se optimiza el tiempo en la Sucursal y en CPN al no reprocesar operaciones.

Aún se pueden optimizar algunos tiempos dentro del CPN ya que hay muchos tiempos destinados a revisión que crean esperas innecesarias para detectar errores de la información recibida, lo que puede llevar a realizar doble labor en algunos casos.

La utilización del Sistema Access como mecanismo de control debe ser estudiada, ya que los procedimientos establecen planillas de control que permiten tener información rápida sobre el estado de la operación, y aquí hay tiempos invertidos no considerados inicialmente.

El supervisor del CPN debe contar con teléfono con salida a regiones para facilitar su tarea de control y retroalimentación permanente con Sucursales.

El supervisor del CPN debe anular diariamente las operaciones que no consten de las planillas de cierre de Sucursales, evitando así posibles errores, informando del hecho a la sucursal de origen. (A incluir en procedimientos actuales).

APOYO TECNOLÓGICO

Se formularán observaciones de orden general, y se elaborará en conjunto con Informática un informe detallado sobre tiempos de respuesta y probables desarrollos para optimizar el uso de la aplicación.

Los tiempos de respuesta para esta Sucursal se consideran buenos, dado el enlace satelital que posee. Los usuarios no perciben una lentitud en la Aplicación para envío de imágenes, ya que están habituados a estos tiempos de respuesta para cualquier aplicación. A modo de ejemplo el Jefe de Operaciones cuando requiere revisar una operación ejecuta 2 acciones; llamar al documento, (demora 40 segundos en traer la respuesta y la pantalla de detalle), y ver imágenes (demora 20 segundos para visualizar 4 imágenes). Como parámetro de comparación, el correo electrónico tarda tiempos similares en las acciones de levantar la aplicación y solicitar lista de usuarios, alrededor de 40 segundos.

Un análisis más detallado con la información de mediciones se obtendrá de la 2da. Evaluación, la cual nos permitirá extraer conclusiones para otras Sucursales en términos de tiempos y calidad.

Se estima relevante trabajar sobre aspectos como el control de estados en la medida que se avanza en el proceso para un mejor control, por ejemplo contar con un aviso, por ejemplo un

mensaje que alerte al usuario cuando llega una nueva operación, (facilitando por tanto la revisión y optimización del tiempo).

Una observación relevante tiene que ver con el tamaño de la documentación a digitalizar, ya que el scanner acepta papeles tamaño oficio, pero la documentación del Banco tiene papeles que exceden en 2 centímetros a ese tamaño, lo que obliga al digitalizador a encuadrar el papel en el Scanner. Caso concreto es la Solicitud de Créditos emitida por Plataforma Comercial y el formulario Resolución de Comité generado por la Plataforma Comercial. (Requerimiento a gestionar con Iván Cortés (Informática), y Banca de Personas).

Conclusión:

La conclusión general es que el Modelo para Curse con Imágenes opera bajo las condiciones establecidas según lo previsto, los procedimientos definidos están operativos y con mínimas observaciones, los tiempos de respuesta de las aplicaciones y el servicio del CPN son buenos, los volúmenes de operaciones son manejables por todos los actores del proceso, y por tanto se recomienda continuar con la Fase 2 de Piloto, según lo acordado.