



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE ESTUDIOS SUPERIORES ARAGÓN

**“ADMINISTRADOR DE DEPOSITADORES
AUTOMÁTICOS”**

**TRABAJO ESCRITO
EN LA MODALIDAD DE DESARROLLO DE UN CASO
PRACTICO
QUE PARA OBTENER EL TÍTULO DE:
INGENIERO EN COMPUTACIÓN**

P R E S E N T A N :

**PABLO BEJARANO ESPINOZA
Y
ORLANDO SANTAELLA VARGAS**

ASESOR: ING. RAFAEL CANTO GALLO



MÉXICO, 2010



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Índice

Introducción 3

Capítulo 1 5

1. Situación Actual 5

1.1. Problemática.....	5
1.2. Solución.....	6
1.3. Planeación.....	7
1.3.1. Alcances.....	8
1.3.2. Entregables.....	8
1.3.3. Recursos Humanos.....	8
1.4. Costos de Software.....	8
1.4.1. Selección del lenguaje de programación.....	8
1.4.2. Selección de herramientas de desarrollo.....	10
1.4.3. Selección del Servidor de Aplicaciones.....	11
1.4.4. Software de productividad y administración de proyectos.....	16
1.5. Costos de Hardware.....	17
1.6. Costos de recursos humanos.....	18
1.7. Factibilidad económica.....	22
1.8. Factibilidad técnica.....	29
1.9. Factibilidad operativa.....	30

Capítulo 2 31

2. Depositadores Automáticos 31

2.1. Descripción.....	31
2.2. Ventajas para el banco y sus clientes.....	32
2.3. Especificaciones de Hardware.....	33
2.4. Especificaciones de Software.....	34
2.4.1. ProClassic/Enterprise.....	34
2.4.2. Protopas/Suite.....	36

Capítulo 3 38

3. Diseño y Desarrollo del Problema 38

3.1. Descripción del problema.....	38
3.2. Diseño de la interfaz de usuario.....	40
3.2.1. Generación de archivos manual.....	40
3.2.2. Generación de reportes en Excel.....	41

3.2.3.	Administrador de días festivos.....	42
3.2.4.	Administrador de usuarios.....	43
3.3.	Diseño y Construcción.....	44
3.3.1.	Diagramas de Casos de uso.....	44
3.3.2.	Referencia de los casos de uso.....	45
3.3.4.	Diccionario de datos.....	46
3.3.5.	Diagrama de secuencia.....	49
3.3.6.	Diagrama de estados.....	53
3.3.7.	Diagrama de Clases.....	57
Capítulo 4 58		
4.	Pruebas y Resultados 58	
4.1.	Pruebas de generación de reportes en TXT.....	58
4.2.	Resultados de la generación y envío de reportes en TXT.....	58
4.3.	Pruebas de generación de reportes en Excel.....	58
4.4.	Resultados de generación de reportes en Excel.....	58
4.5.	Pruebas del módulo Administración de usuarios.....	59
4.6.	Resultados del módulo Administrador de usuarios.....	59
4.7.	Pruebas del módulo de Administrador de días festivos.....	59
4.8.	Resultados del modulo de Administrador de días festivos.....	59
4.9.	Conclusiones.....	59
Conclusiones 60		
Bibliografía 61		
Mesohemerografía 62		

Introducción

La calidad del software es una lucha a la que se le dedican múltiples esfuerzos, sin embargo, el software casi nunca es perfecto. Todo proyecto tiene el objetivo de producir aplicaciones de calidad, que cumplan con el funcionamiento para el que fueron creados y si es posible, que supere las expectativas del usuario, ya que finalmente, este es quien mide el grado de satisfacción según lo que tenga o no.

La calidad del software depende de quien la juzgue y el hecho de que una empresa tenga certificación de calidad no garantiza que su software sea de calidad.

En la actualidad, la competencia por ofrecer diferentes servicios ha ido creciendo exponencialmente en múltiples rubros de la industria y uno de ellos es la industria bancaria.

La industria bancaria intenta atraer la atención de los clientes con diferentes productos y ofreciendo servicios que otros no pueden o no tienen y es por esto que se observó la opción de ofrecer depósitos de efectivo a cuentas, a través de un Depositador de Efectivo (CDM – por sus siglas en inglés Cash Deposit Machine) el cual pretende brindar mayor rapidez en los tiempos de espera dentro de una sucursal y con esto que el cliente salga mas satisfecho.

El presente proyecto tiene como objetivo realizar una aplicación Web la cual va a generar y enviar los archivos de cortes y transacciones que se realizan en los depositadores automáticos al área de gestión y soporte, dicha área nos van a garantizar la conciliación de las operaciones que diariamente se realizan, dicha aplicación también nos va permitir una mejor administración de servicios como pagos a empresas telefónicas, tarjetas de crédito, etc. y que se pueden pagar en los mismos.

En el capítulo uno contiene la descripción del problema, la propuesta para resolver la situación mencionada al principio del capítulo, que actualmente se encuentra productiva , las tecnologías de desarrollo y los costos de recursos tanto de software y hardware como los recursos humanos que se tomaron en cuenta para realizar la aplicación Web.

En el segundo capítulo dos se presenta una descripción de lo que son los depositadores automáticos, que ventajas traen al banco y a los usuarios del mismo, especificaciones de hardware y de software.

El análisis y diseño de la aplicación son manejados en el capítulo número tres, que contiene desde la descripción general, hasta las herramientas utilizadas para el diseño y desarrollo del proyecto tales como: UML el cual incluye diagramas de

casos de uso, de secuencia, de clases, de estados, Modelo entidad relación de la base de datos y el diccionario de datos.

Por último en el capítulo número cuatro se muestran las pruebas que se hicieron de la aplicación en conjunto con el usuario, los resultados que se obtuvieron de estas y las correcciones que se tuvieron que hacer para que dichos resultados fueran totalmente satisfactorios, antes de poder dar un visto bueno y así la aplicación fuera puesta en producción.

Capítulo 1

1. Situación Actual

1.1. Problemática

Una parte importante del proceso de Depositadores Automáticos es la conciliación de las transacciones realizadas diariamente, la cual consiste en revisar que los depósitos que se tienen registrados por el depositador sean los mismos que tiene registrado el sistema de depósitos bancario, si se presenta una situación en la que estas dos cifras no coincidan se tiene que realizar un análisis por parte del área de desarrollo y es necesario dar un dictamen explicando cual es la razón por la cual estas cifras no están coincidiendo

Existe una aplicación que trabaja diariamente para generar 4 archivos, los cuales contienen la información de las transacciones y cortes que se realizaron en los Depositadores Automáticos y que cuentan con el siguiente nombre:

1. NPMXMMXMPOP6D001_smtrc_aaaammdd.txt
2. NPMXMMXMPOP6D002_461_aaaammdd.txt
3. NPMXMMXMPOP6D003_483_aaaammdd.txt
4. NPMXMMXMPOP6D004_daily_aaaammdd.txt

Donde:

- aaa es el año en 4 dígitos
- mm es el mes
- dd es el día

Actualmente el área de gestión y soporte es la encargada de realizar la conciliación de las transacciones en depositadores, para esto recibe diariamente los 4 archivos ya mencionados, los cuales son transmitidos con una herramienta llamada FEP¹. Si esta herramienta tiene muchos archivos en espera de ser transmitidos comienza a tener problemas para enviar información, por lo tanto los archivos no son enviados correctamente y afecta directamente a la correcta conciliación de los movimientos realizados en los Depositadores Automáticos que puede ocasionar que se tengan diferencias en las cuentas utilizadas para llevar la contabilidad de los CDMs. Esto puede impactar con pérdidas en la operación de los CDMs.

¹ FEP File Exchange Process (Proceso de Intercambio de Archivos)

En algunos casos puede haber consecuencias directas con nuestros clientes directos que son los usuarios de los Depositadores Automáticos, ya que si existiera alguna reclamación por parte del cliente no habría forma de darle un dictamen en un corto tiempo, provocando desconfianza en el servicio que se está ofreciendo y un bajo nivel en las transacciones.

1.2. Solución

El objetivo de este proyecto es asegurar que los archivos de corte ya antes mencionados se generen correctamente y se transmitan de lunes a viernes por medio de otra herramienta llamada GTS² sin excepción, los archivos son transformados por dicha herramienta en archivos propios de un equipo mainframe³ los cuales son llamados etiquetas, estas etiquetas son tomadas en automático por los procesos de Gestión & Soporte y de esta manera, se logrará la conciliación de las transacciones realizadas en CDMs⁴ diariamente.

Las etiquetas tienen el siguiente nombre:

5. PHOG.NFSS.CDMSMTRC.P22DS
6. PHOG.NFSS.CDM461.P22DS
7. PHOG.NFSS.CDM483.P22DS
8. PHOG.NFSS.CDMDFILE.P22DS

y están relacionadas con cada uno de los archivos que se generan.

La solución que se propone, en conjunto debe también incluir una opción que cuente con re-procesos, los cuales consisten en, realizar el proceso de generación y envío de los archivos de manera manual, en caso de que el proceso automático falle por cualquier situación y que las etiquetas mencionadas anteriormente sean actualizadas con la información correcta casi de manera inmediata, así el área encargada de la conciliación pueda continuar con su operación normal.

Debe contar con la opción de mostrar las transacciones de los depositadores en formato Excel y con la opción de ser descargado de forma local, tomando en cuenta un rango de fechas que el usuario tendrá

2 Group Trade Services (Comercio Exterior del Grupo)

3 Computadora grande, potente y costosa usada principalmente por una gran compañía para el procesamiento de una gran cantidad de datos

4 Por sus siglas en inglés Cash Deposit Machine

que seleccionar

Un módulo que administre los usuarios que tendrán acceso al sistema también debe estar presente, en el cual se podrán dar de alta, baja y editar usuarios que según su perfil tendrán acceso a las opciones del menú.

El proceso automático que genera los archivos esta programado para que funcione todos los días pero en fines de semana y días festivos estos archivos no deben generarse y para administrar esos días también se debe de contar con un módulo en el que se puedan dar de alta y eliminar dichos días.

1.3. Planeación

Una vez entregada la propuesta al usuario, lo siguiente fue realizar el análisis para desarrollar la aplicación que dará solución al problema planteado y el cual consiste en mejorar con lo que actualmente se cuenta.

1.3.1. Alcances

- 1) El sistema deberá contar con:
 - Una base de datos en la cual contendrá información de cortes,
 - depósitos, usuarios, fines de semana y días festivos
- 2) La interfaz de usuario contará con:
 - Una interfaz gráfica para el reproceso
 - Una interfaz gráfica para obtener el listado de transacciones por fecha en formato Excel
 - Una interfaz gráfica para la administración de usuarios que pueden acceder al sistema (altas, bajas y cambios)
 - Una interfaz gráfica administrar días festivos (altas, bajas y cambios)

1.3.2. Entregables

La documentación y archivos con la que contará el usuario es la siguiente:

- 1) Manal de usuario
- 2) Plantilla con el formato de los archivos de cortes y transacciones
- 3) Archivos de cortes y transacciones

1.3.3. Recursos Humanos

Los recursos humanos necesarios para el desarrollo del proyecto son:

- 4) Líder de proyecto
- 5) Analista
- 6) Usuarios

En el siguiente punto se analizarán las tecnologías propuestas tanto de hardware, software como de recursos humanos y el costo que representan.

1.4. Costos de Software

1.4.1. Selección del lenguaje de programación

El lenguaje de programación seleccionado deberá cubrir totalmente o la mayoría de las siguientes características tecnológicas deseadas:

- **Ligero.** No debe requerir grandes inversiones en hardware para poder instalarlo.
- **Sistema Web.** No debe requerir de un software cliente.
- **Multiplataforma.** Debe funcionar sobre sistemas operativos Windows, Unix y Mac.
- **Multi-idioma.** Debe permitir configuración de idioma y formato de moneda pensando en internacionalización del producto.
- **Amigable.** Debe proporcionar una interfaz que facilite su uso.
- **Independiente del proveedor de servidor de aplicaciones.** Debe tener la facilidad de instalarse en diferentes servidores de aplicaciones
- **Amplia oferta de personal capacitado.** Debe contar con una amplia oferta de personal capacitado en el mercado nacional.
- **Costo competitivo.** No debe requerir hardware robusto, ni servidores para reducir el costo de implementación.
- **De fácil instalación y configuración.** Debe usar componentes configurables para disminuir el número de adecuaciones al instalar.

Actualmente los lenguajes más utilizados en el desarrollo de aplicaciones empresariales y que ofrecen mayor oferta de recursos humanos en el mercado mexicano son Java y .Net.

Ambos son lenguajes orientados a objetos, que permiten desarrollo de aplicaciones Web, con mucha documentación disponible, oferta de

cursos de capacitación y foros de discusión entre muchas otras similitudes.

En la tabla 1.1 se presenta una comparación de las características técnicas deseadas que puedan realizarse con Java y .Net.

SE PUEDE CONSTRUIR UN SISTEMA:	JAVA (JAVA, 2009)	.NET(.NET, 2009)
Ligero.	✓	✓
Sistema Web.	✓	✓
Multiplataforma.	✓	✗
Multi-idioma.	✓	✓
Amigable.	✓	✓
Independiente del servidor de aplicaciones.	✓	✗
Amplia oferta de personal capacitado.	✓	✓
Costo competitivo.	✓	✓
De fácil instalación y configuración.	✓	✓
Total aciertos:	9	7

Tabla 1.1. Comparación de características proporcionadas.

Como se aprecia en la tabla 1.1 Java es ligeramente superior a .Net, esta última herramienta a pesar de ser un producto robusto, estable y muy completo para crear e integrar rápidamente aplicaciones de Microsoft Windows y soluciones Web, funciona exclusivamente sobre la plataforma Windows.

1.4.2. Selección de herramientas de desarrollo.

Un entorno de desarrollo integrado o IDE (acrónimo en inglés de Integrated Development Environment), es un programa informático compuesto por un conjunto de herramientas de programación. Puede dedicarse en exclusiva a un sólo lenguaje de programación o bien, poder utilizarse para varios.

CARACTERÍSTICA	JDEVELOPER 11g (JDEVELOPER, 2009)	eCLIPSE Galileo (eclipse, 2009)	net beans (NETBEANS, 2009)
Empresa	Oracle	Fundación Eclipse	Sun Microsystems
Ambiente de ejecución	Java Runtime Environment	Java Runtime Environment	Java Runtime Environment
Lenguajes de programación	Java	JRE	Java
Generación de páginas dinámicas	JSF, JSP, Servlets, PHP, UIX	JSF, JSP, Servlets, PHP	JSF, JSP, Servlets
Acceso a datos	JDBC	JDBC	ODBC
Diseño UML	Modelado UML y generación integrado	Requiere componentes de modelado	Incluido
Control de versiones	Requiere Software SVN componente	Requiere componente SVN	SI
Plataforma	Multiplataforma	Multiplataforma	Multiplataforma
Costo	Gratis	Gratis	Gratis

Tabla 1.2. Comparativa diferentes ambientes de desarrollo

El análisis, diseño y desarrollo de software es una actividad continua para la mayoría de las aplicaciones, por lo que es importante fortalecer estas actividades con herramientas que faciliten el trabajo. A continuación algunos IDEs disponibles que utilizan el lenguaje Java seleccionado. En la tabla 1.2 se muestra la comparación de diversos entornos de desarrollo disponibles.

Al consultar el sitio de Internet de los IDEs comparados en la tabla 1.2, destaca Eclipse por ser el IDE que cuenta con mayor número de componentes gratuitos desarrollado por terceros y que sirven de apoyo para el desarrollo de software empresarial.

1.4.3. Selección del Servidor de Aplicaciones

Una de las decisiones más importantes que deben tomarse antes de empezar a escribir la primera aplicación en Java es la de qué servidor de aplicaciones web utilizar. A IBM le gustaría que creyera que la respuesta obvia es WebSphere Application Server (WAS), pero, también apoya Apache Tomcat para las aplicaciones desarrolladas en Java. Básicamente, WebSphere se recomienda para aplicaciones grandes y complejas en sitios web con mucho tráfico y Tomcat para aplicaciones básicas y sitios web con poco tráfico.

1.4.3.1. IBM WebSphere Application Server Ventajas

IBM WebSphere Application Server agiliza los negocios, ya que proporciona a millones de desarrolladores y de arquitectos de TI una base innovadora y con un gran rendimiento para crear, reutilizar, ejecutar, integrar y gestionar las aplicaciones y servicios SOA (Arquitectura Orientada a Servicios).

Ejecuta los servicios y las aplicaciones en un entorno fiable, escalable y de alta disponibilidad para asegurarse de que no se pierden oportunidades empresariales a causa del tiempo de inactividad de las aplicaciones.

Reduce los requisitos de memoria y utiliza procesadores que consumen menos energía gracias a la optimización del tiempo de ejecución.

Proteje las aplicaciones y los datos con una infraestructura de seguridad flexible basada en estándares, que elimina las amenazas y maximiza la productividad de los desarrolladores.

Protege las inversiones reutilizando e integrando activos de software con estándares de Web 2.0 y servicios Web mejorados, así como nuevos dispositivos de gestión de aplicaciones.

Se anticipa y ajusta a la evolución de los requisitos empresariales gracias a la eficacia en el tiempo de ejecución, una infraestructura simplificada y un control de las aplicaciones eficaz y flexible.

Aumenta la productividad de los desarrolladores mediante un soporte para estándares mejorado, una tecnología emergente y una gran variedad de infraestructuras de desarrollo que simplifican los modelos de programación.

WebSphere Application Server continuará presentando Feature Packs sobre V7, lo que facilitará la adopción de nuevos estándares como la Arquitectura de Componentes de Servicio (SCA).

Sistemas operativos soportados: z/OS², AIX, HP Unix, Linux, Sun Solaris⁶, Windows®,⁷

Con un alto rendimiento Mantiene tu negocio listo y se adapta fácilmente ante un ambiente de cambios. WebSphere Application Server proporciona una manera rápida, fiable, disponible, segura y escalable que permite el desarrollo y la eficiencia de aplicación para que las empresas pueden hacer más con menos recursos.

Optimiza el rendimiento del tiempo de ejecución a través los servicios web que provee y de las mejoras los EJB's⁸, las cuales pueden dar como resultado un menor consumo en el rendimiento del procesador con las mismas cargas de trabajo de versiones anteriores.

Seguridad muy confiable Sientase seguro, las aplicaciones y los datos están seguros de los ataques fuera de las configuraciones de seguridad y registros de usuario, cuenta con el cumplimiento de los estándares gubernamentales y una estricta seguridad de los Web Services. Añade nuevas capas de seguridad a niveles mas profundos de la administración, gestión de usuarios y de auditoria para reducir las vulnerabilidades del sistema y maximizar la productividad del desarrollador.

2 Es el sistema operativo actual de los mainframes de IBM

6 AIX, HP Unix, , Linux, Sun Solaris Sistemas operativos y versiones de UNIX

7 Sistema operativo desarrollado por Microsoft

8 Enterprise JavaBeans son una de las API que forman parte del estándar de construcción de aplicaciones empresariales J2EE

Los dominios de seguridad, que ofrecen una mayor flexibilidad y control sobre los usuarios y la infraestructura para incrementar la eficacia de la administración de estos.

WebSphere es un galardonado servidor de aplicaciones web conforme con J2EE⁹ que lidera el mercado. WebSphere es un producto comercial con un completo servicio técnico (a quienes puede llamar para desahogarse si algo va mal). Puede utilizarse en entornos de clusters, permite la conmutación por anomalía y tiene un alto nivel de escalabilidad. Asimismo, WebSphere es totalmente compatible con EJB.

En realidad, WebSphere no sólo le ofrece todo lo que pueda desear de un servidor de aplicaciones web, sino que también dispone de asistentes que le ayudarán en la configuración y el mantenimiento de las aplicaciones. WebSphere incluye agrupaciones de conexiones JDBC¹⁰. Integra un juego de herramientas XML¹¹. Tiene el entorno de servicios web más integrado del sector y ofrece una mirada de potentes opciones de seguridad. WebSphere permite utilizar perfiles de aplicaciones (que, por cierto, son imprescindibles para ajustar el rendimiento de las aplicaciones desarrolladas en Java). También incluye un mecanismo de registro cronológico integrado.

Aunque puede que lo mejor de WebSphere sea lo bien que está integrado con WSAD¹². Ningún otro IDE para Java del mercado se integra de tal manera con un servidor de aplicaciones web.

1.4.3.2. Problemas con WebSphere

Los tres problemas principales de WebSphere son el coste, la complejidad y los requisitos de hardware. WAS¹³ cuesta unos 8.000 euros por procesador (dependiendo de descuentos, la selección de productos y otras opciones). Si tiene una máquina multiprocesador, el precio que tendrá que pagar por WebSphere puede salirse del presupuesto de la empresa.

9 Java Platform, Enterprise Edition es una plataforma de programación para desarrollar y ejecutar software de aplicaciones en Lenguaje de programación Java

10 Java Database Connectivity es una API que permite la ejecución de operaciones sobre bases de datos desde el lenguaje de programación Java

11 Extensible Markup Language s un metalenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium (W3C)

12 BM Websphere Development Studio es un Integrated Development Environment (IDE) basado en eclipse, creado por IBM el cual se puede integrar con WebSphere

13 WebSphere Application Server, servidor de aplicaciones web de IBM

Por otro lado, WebSphere es un producto complejo que incluye varios entornos integrados. WAS 4.0 dispone de dos versiones de una GUI¹⁴ de administración: el producto Single Server incluye una aplicación de administración basada en web, mientras que Advanced Edition se entrega con una aplicación cliente-servidor que necesita para su funcionamiento un potente PC. El desarrollo de aplicaciones exige utilizar tanto la Herramienta de ensamblado de aplicaciones de WebSphere como su Consola administrativa. Si habitualmente se despliegan aplicaciones con estas dos herramientas pronto se convierte en un trabajo monótono, por lo que WAS también proporciona una docena de sofisticadas herramientas para agilizar las operaciones de despliegue. Entre estas herramientas de despliegue se incluye XMLConfig, el programa de control de WebSphere (WSCP) y plug-ins para Ant (otro de los recursos de código libre para la creación de aplicaciones Java).

Por último, no nos podemos olvidar del tema de sus requisitos de hardware. A diferencia de Tomcat, WebSphere no puede ejecutarse en máquinas iSeries de gama baja. WebSphere es muy escalable, pero sólo en modelos 270 de gama alta o superiores con 1 GB o más de memoria principal.

1.4.3.3. Apache Tomcat

De Apache Tomcat es una aplicación de software de código abierto para las tecnologías Java Servlet y JavaServer Pages

Tomcat es gratis, es fácil de instalar, se ejecuta en máquinas más pequeñas y es compatible con las API¹⁵ más recientes de Java. Puede descargarse, instalarse y probarse en el iSeries¹⁶ en menos de una hora. Tomcat ocupa muy poco espacio, teniendo su código binario (todo clases de Java) un tamaño total de apenas un megabyte, de modo que no es raro que se ejecute tan deprisa.

Otra ventaja de Tomcat es que es muy fiable, lo que no quiere decir que WebSphere no lo sea. Innumerables empresas utilizan Tomcat (aunque es imposible saber cuántas debido a la falta de licencia comercial). Y como dijo Linus Torvalds acerca del código libre,

14 Graphical Userinterface s un programa informático que actúa de interfaz de usuario, utilizando un conjunto de imágenes y objetos gráficos

15 Interfaz de programación de aplicaciones es el conjunto de funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software

16 Línea de Servidores fabricados por IBM que originalmente fueron conocidos como AS/400

"Dado un número suficiente de ojos, todos los errores son irrelevantes". Lo que quiere significar es que si el número de usuarios es lo bastante grande, siempre habrá alguien capaz de arreglar lo que los demás pueden pensar que es un error muy complejo. Dicho de otra forma, la solidez de Tomcat se basa en que miles de desarrolladores contribuyen con código.

Tomcat pone a disposición de todo el mundo las últimas actualizaciones de Java. Por ejemplo, Tomcat 4.0 es compatible con JSP 1.2 y Servlet 2.3 (como WAS 5.0), mientras que WAS 4.0 y Tomcat 3.x son compatibles con JSP 1.1 y Servlet. Además, como Tomcat puede ejecutarse utilizando la JVM que se quiera, puede utilizarse JDK 1.4 (por cierto que Java 1.4 ofrece aserciones, expresiones regulares, mejor rendimiento que Java 1.3 y una infraestructura de registro cronológico estándar). Tomcat ahora tiene seguridad de nivel de aplicación, una aplicación de administración intuitiva basada en web, expresiones regulares compatibles con JDK mas actual y, como se mencionaba antes, mejor escalabilidad y rendimiento.

Tomcat es compatible de forma integrada con el último IDE para Java de IBM, WebSphere Studio Application Development (WSAD) y su versión para el iSeries, WebSphere Development Studio Client (WDSC). Incluso aunque vaya a desplegar sus aplicaciones para WebSphere sería una buena idea hacerlo también con Tomcat. ¿Por qué? Por dos razones: la primera es que desplegar aplicaciones para Tomcat se hace en un momento. Y la segunda es que si prueba la aplicación web con Tomcat se asegurará de que no sólo puede desplegarse en WebSphere, sino también en otros servidores de aplicaciones web, como los WebLogic de BEA o iPlanet de Sun.

1.4.3.4. Problemas con Tomcat

El mayor problema de Tomcat es que no hay nadie a quien echarle las culpas si algo va mal. No es posible telefonar al servicio técnico para que le resuelva sus problemas. Además, Tomcat sólo puede trabajar con un único servidor; no puede utilizarse en clusters. WebSphere puede utilizarse en entornos de clusters y permite la conmutación por anomalía (failover). Y, por supuesto, tiene servicio técnico.

Además, Tomcat no tiene tanta escalabilidad como WebSphere. Tomcat no es compatible con EJB... de hecho, no es compatible con muchas de las otras API de J2EE. Tomcat tampoco dispone de un

entorno integrado con una sofisticada GUI, asistentes y ayuda en línea.

Compatibilidad con J2EE. Tomcat no es directamente compatible con todas las API de J2EE que soporta WebSphere (como JDBC, JNDI, JavaMail, RMI, JMS).

Como se puede ver en la tabla 1.3 tomcat es de licencia libre y esto puede ahorrarle a la empresa licencias costosas, son embargo los lineamientos actuales es que todas las aplicaciones web deben de trabajar con un alta disponibilidad y por lo tanto se tiene que trabajar con clusters, esto es algo que tomcat no nos ofrece y la razón por que se trabajara con WebSphere.

Características	WebSphere	Tomcat
Compatibilidad jsp' y java	✓	✓
Costo	8.000 euros por procesador	Licencia Libre
Compatibilidad con J2EE	✓	No soporta ciertas API's de java
Escalabilidad	✓	No cuenta con la suficiente escalabilidad
Cluster	✓	No maneja Cluster

Tabla 1.3 Comparativa entre diferentes servidores de aplicaciones

1.4.4. Software de productividad y administración de proyectos.

Para la elaboración de documentación y administración del proyecto se propone utilizar herramientas de la empresa Microsoft, por ser un software muy utilizado en México, y con ello tener la certeza de conseguir personal capacitado en dichas herramientas. Microsoft Office Professional y Microsoft Project sirven para la elaboración de documentación, presentaciones y administración de la planeación y seguimiento proyectos.

Existe software de productividad gratuito como OpenOffice, este producto en su página de Internet describe características similares al Microsoft Office Professional. La solución OpenOffice no es contemplada en el proyecto por el riesgo que genera la incertidumbre de conseguir personal capacitado o el tiempo que se invertiría en lograr una auto capacitación. Además, un problema que se puede generar en el proyecto sería la compatibilidad en el intercambio de información con el personal externo al proyecto, como el personal que se tiene planeado contratar para la elaboración del diseño del sistema y los problemas de compatibilidad que podrían existir al enviar documentación técnica y de usuario al área en cuestión.

Tipo de Software	Software	Costo
Base de Datos	DB2	Sin Costo
IDE	Eclipse Galileo	Sin Costo
Control de versiones	SVN	Sin Costo
Admon de proyectos	Microsoft Project 2007	\$5,693 pesos
Software de productividad	MS Office Professional 2007	\$4,728 pesos

En la tabla 1.4 se concentran los costos de software.

1.5. Costos de Hardware.

En la tabla 1.5 se concentra información sobre costos de hardware obtenidos del sitio de Internet de los proveedores nacionales PC OnLine S.A. de C.V¹⁷ y Sumicom Telemarketing S.A. de C.V¹⁸. Las características del hardware propuesto fueron estimadas con base en los requisitos de sistema que exige el software que se instalará de acuerdo al perfil del usuario.

¹⁷ <http://www.pcenlinea.com/>

¹⁸ <http://www.sumitel.com/>

EQUIPO	Características	Costo
Desarrollador	PC Procesador Core2Duo 2 GB ram	\$8,000 pesos
Herramienta de desarrollo seleccionada	80 GB disco duro Monitor LCD 19"	
Analista.	PC Procesador Dual Core. 2 GB RAM	\$7,000 pesos
Herramientas para documentación y presentaciones.	MS Windows 80 GB disco duro Monitor LCD 15"	
Servidor.	Procesador Intel 7i 4 GB RAM	\$18,000 pesos
Manejador de BD, servidor de aplicaciones y repositorio de código fuente	1 Terabyte de disco duro Monitor LCD 22"	

En la tabla 1.5 se concentran los costos de hardware

1.6. Costos de recursos humanos.

Se pretende reclutar personal que resida en el Estado de México o el Distrito Federal con experiencia mínima de 5 años en desarrollo de software y que cuente con alguna certificación de acuerdo a su perfil tecnológico.

La información que se muestra aquí es obtenida resultado de una encuesta abierta contestada por 2,162 personas a través del sitio de Internet de Software Gurú¹⁹ durante el mes de octubre del 2008, publicada por Pedro Galván el 3 de diciembre de 2008. Se resaltaré la información que esté lo más acorde al perfil requerido para el proyecto aquí propuesto.

Sueldo de acuerdo a la región. El DF tiene su liderazgo, gracias a ofrecer los mejores sueldos para profesionistas de software, dejando a Nuevo León en segundo lugar y al Estado de México en tercero. Estos estados son también los que emplean a un mayor porcentaje de los profesionistas de software en México. En la tabla 1.6 se presenta la información por región de acuerdo a la encuesta realizada (Galván, 2008).

¹⁹ <http://www.sg.com.mx/> [Consulta: 2 de junio de 2009]

Estado	Salario	%
DF	\$28,899	35.2%
NL	\$25,885	8.3%
Edomex	\$22,232	9.7%
Queretaro	\$21,031	4.5%
Jalisco	\$19,842	8.7%

Tabla 1.6 Tabla con salarios más altos de acuerdo a la región.

Sueldo de acuerdo al género. En la tabla 1.7 se puede ver claramente que la participación del sexo femenino en desarrollo de software es muy baja, y con un salario significativamente menor. La diferencia de salario es del 33%. (Galván, 2008).

GENERO	Salario	%
Femenino	\$17,513	14.4%
Masculino	\$23,423	85.6%

Tabla 1.7 Tabla con salarios por género.

Sueldo de acuerdo a la edad. Como se aprecia en la tabla 1.8 el salario de un profesionalista de TI sube paulatinamente de acuerdo a su edad, y llega a un máximo entre las personas que tienen 50 a 59 años. Posteriormente baja drásticamente después de los 60 (Galván, 2008).

EDAD	Salario	%
18-24	\$11,410	14.6%
25-29	\$17,378	31.9%

30-39	\$25,563	40.5%
40-49	\$37,166	10.4%
50-59	\$46,276	2.3%
60 o más	\$27,143	0.3%

Tabla 1.8 Tabla con salarios por edad.

Sueldo de acuerdo a la experiencia. El salario sube paulatinamente respecto a los años de experiencia. Analizando la tabla 1.9, se determina que le hacen falta más escalas después de los 10 años de experiencia, para poder analizar a este segmento en mayor detalle (Galván, 2008).

EDAD	Salario	%
Menos de un año	\$10,328	6.4%
1 a 3	\$12,746	20.8%
3 a 5	\$17,706	17.1%
5 a 10	\$23,304	25%
Más de 10	33,871	25%

Tabla 1.9 Tabla con salarios por experiencia

Sueldo por perfil tecnológico. En la tabla 2.0 se muestran los resultados de la encuesta mostrando solamente los lenguajes de programación que interesan para el presente estudio, en la tabla 2.1 las bases de datos y en la tabla 2.2 los conocimientos de documentación y pruebas deseables en los desarrolladores y que serían obligatorios para los analistas (Galván, 2008).

LENGUAJE	Salario	%
.Net	\$27,827	20.7%
JEE	\$26,577	26.7%

Tabla 2.0 Lenguaje de programación

BASE DE DATOS	Salario	%
DB2	\$26,217	39.3%
SQL Server	\$23,140	59.8%

Tabla 2.1 Base de datos.

HABILIDADES	Salario	%
Test	\$26,294	24.6%
UML	\$22,970	44.9%

Tabla 2.2 Otras habilidades.

Certificaciones. La mejor forma de demostrar conocimientos sobre alguna práctica o tecnología específica es una certificación. En la tabla 4.4.h se presentan los resultados de la encuesta por certificaciones. Considerando solamente las certificaciones de interés para el presente documento (Galván, 2008).

CERTIFICACIÓN	Salario	%
Project Managment Professional	\$43,215	5.5%
OMG-UML	\$35,929	1.3%
DB2	\$35,577	3.1%
Microsoft Solution Developer	\$28,039	3.1%
Sun Java	\$27,981	7.7%

Tabla 2.3 Salario por certificaciones.

1.7. Factibilidad económica.

El objetivo de esta sección es hacer una estimación de los recursos necesarios para desarrollar el sistema basados en la información que se ha ido incorporando al presente estudio.

Sommerville (2005:358) afirma que existen tres parámetros involucrados en el cálculo del costo total de un proyecto de desarrollo de software:

Los costos hardware y software.

Los costos de viajes y capacitación.

Los costos de esfuerzo (Recursos humanos).

En la presente investigación no se consideran costos de capacitación porque el personal seleccionado debe cumplir con las habilidades técnicas que se requieren para el proyecto. De igual manera tampoco se pretende que el personal viaje, con la excepción que durante el desarrollo se deba realizar alguna presentación en sitio con algún posible cliente, además estos costos serían una pequeña parte comparados con los costos de esfuerzo.

Los costos de esfuerzo no sólo son los salarios del personal que interviene en el proyecto. Las organizaciones calculan los costos de esfuerzo en función de los costos totales, donde se tiene en cuenta el costo total para hacer funcionar la organización (Sommerville, 2005:562). Por lo tanto, los siguientes costos son parte de los costos totales:

Costos de proveer, aclimatar e iluminar las oficinas.

Los costos del personal de apoyo como administrativos, secretarias, limpiadores y técnicos.

Los costos de redes y las comunicaciones.

Los costos de los recursos centralizados como las bibliotecas, los recursos recreativos, etc.

Los costos de seguridad social. pensiones. seguros privados, etc.

Con el actual avance de las comunicaciones, el uso de correo electrónico, sitios Web compartidos y videoconferencias ya no es necesario disponer de un espacio físico y la domiciliación de sociedades, una oficina virtual es una buena elección para un negocio de reciente implantación en un territorio concreto. En la tabla 2.4 se listan algunas ventajas y desventajas del uso de oficinas virtuales.

VENTAJAS	DESVENTAJAS
✓ Reducción de costos.	× Falta de entendimiento.
✓ Comodidad.	× Dependencia absoluta de Internet.
✓ Aprovechamiento del tiempo	× Falta de un espacio para reuniones.

Tabla 2.4 Ventas y desventajas de oficinas virtuales.

Como explica Sommerville (2005:60) aunque existen muchos procesos diferentes para el desarrollo de software, algunas actividades fundamentales son comunes para todos ellos:

1. Especificación del software. Se debe definir la funcionalidad del software y las restricciones en su operación.
2. Diseño e implementación del software. Se debe producir software que cumpla su especificación.
3. Validación del software. Se debe validar el software para asegurar que hace lo que el cliente desea.
4. Evolución del software. El software debe evolucionar para cubrir las necesidades cambiantes del cliente.

Utilizar un enfoque formal para desarrollo de software implica una planificación cuidadosa del proyecto, la utilización de métodos de análisis, diseño y procesos de desarrollo de software controlados y rigurosos, el esfuerzo invertido puede ser tan grande que algunas veces se puede pasar más tiempo pensando en cómo se debe desarrollar el software que en escribir el código y las pruebas. Si cambian los requerimientos, es esencial

rehacer el trabajo, y al menos en principio, la especificación y el diseño tienen que cambiar con el programa. Esto implica un aumento en los tiempos de entrega y por consecuencia un costo mayor.

Los procesos de desarrollo rápido de software están diseñados para producir software útil de forma rápida. Generalmente, son procesos iterativos en los que se entrelazan la especificación, el diseño, el desarrollo y las pruebas. El software no se desarrolla y utiliza en su totalidad, sino en una serie de incrementos, donde en cada incremento se incluyen nuevas funcionalidades al sistema (Sommerville, 2005:358).

Es fundamental que el software nuevo se desarrolle rápidamente para aprovechar las oportunidades de negocio. Por esto como única propuesta en el presente estudio se propone un único escenario con los requerimientos mínimos y utilice para la actividad de implementación y validación del software el proceso de desarrollo de Programación Extrema²⁰ (XP) para la, con la finalidad de concluir de manera más rápida el producto y reducir costos. En la figura 2.5 se muestra el ciclo de entrega de la Programación Extrema.

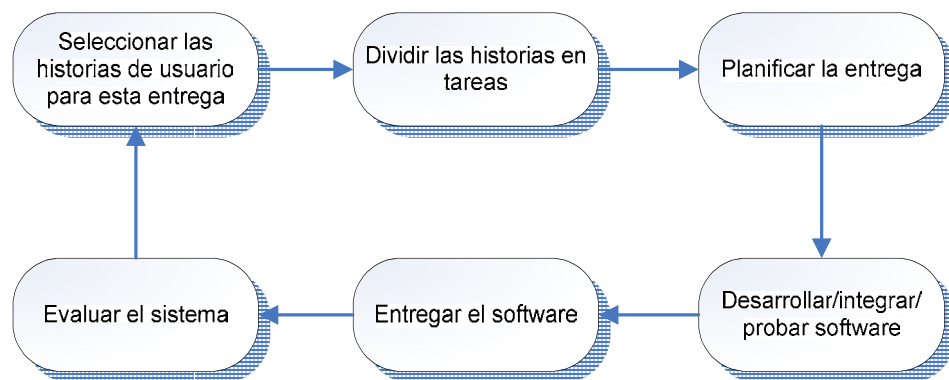


Figura 2.5 Ciclo de entrega de la Programación Extrema (Sommerville, 2005:364).

²⁰ La Programación Extrema (XP por sus siglas en inglés Extreme Programming) es el método ágil más conocido y utilizado. El nombre acuñado por Beck debido a que el enfoque fue desarrollado utilizando buenas prácticas reconocidas, como el desarrollo iterativo. y con la participación del cliente en niveles extremos (Sommerville, 2005:364).

Las prácticas que se utilizarán de la XP para la construcción del nuevo sistema son:

Práctica	Descripción
Entregas pequeñas.	El mínimo conjunto de funcionalidad que proporcione valor de negocio se desarrolla primero. Las entregas del sistema son frecuentes e incrementalmente añaden funcionalidad a la primera entrega.
Desarrollo previamente probado.	Se utiliza un sistema de pruebas de unidad automatizado para escribir pruebas para nuevas funcionalidades antes de que éstas se implementen.
Refactorización.	Se espera que todos los desarrolladores refactoricen el código continuamente tan pronto como encuentren posibles mejoras en el código. Esto conserva el código sencillo y mantenible.
Integración continua.	En cuanto acaba el trabajo en una tarea, se integra en el sistema entero. Después de la integración, se deben pasar al sistema todas las pruebas de unidad.
Ritmo sostenible.	No se consideran aceptables grandes cantidades de horas extras, ya que a menudo el efecto que tienen es que se reduce la calidad del código y la productividad a medio plazo.
Cliente presente.	Debe estar disponible al equipo un representante de los usuarios finales del sistema. En un proceso de la programación extrema, el cliente es miembro del equipo de desarrollo y es responsable de formular al equipo los requerimientos del sistema para su implementación.

Tabla 2.6 Prácticas consideradas de la XP para el nuevo software
(Sommerville:364).

Especificación del software. El diseño será elaborado por personal con experiencia en la operación y mantenimiento de sistemas financieros para conciliaciones que se hayan implementado exitosamente, este personal será contratado como asesor externo. El analista se encargará de ampliar la documentación agregando notas y comentarios a los documentos entregados por los asesores externos. También tendrá la tarea de verificar que la documentación cumpla con una simbología estándar del Lenguaje

Unificado de Modelado (UML, por sus siglas en inglés, Unified Modeling Language).

Implementación y validación. El desarrollo, integración y pruebas se realizarán acorde al proceso de programación extrema. El conjunto de casos de prueba serán proporcionados por los asesores externos que participen en la especificación del software, una vez que los desarrolladores hayan concluido alguna funcionalidad nueva, será integrada al sistema, la integración será verificada por el líder de proyecto asegurándose de que el código generado cumpla con los estándares de programación, una vez verificada y aceptada la integración se aplicarán las pruebas a todo el sistema, verificando que los resultados sean los esperados y no haya afectado a otros procesos, se enviarán evidencia de pruebas a los diseñadores del sistema para que estos realicen sus observaciones (si las hay) al líder de proyecto para planificará el proceso de cambio o en su defecto dar el visto bueno al proceso. En la figura 2.7 se resume el proceso propuesto para la etapa de implementación y validación.

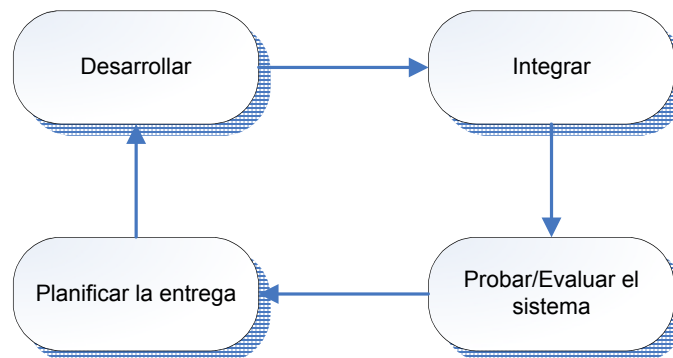


Figura 2.7 Proceso propuesto de desarrollo/integración/pruebas/aceptación del proyecto.

En la tabla 2.8 se muestra el concentrado de los tiempos estimados en 110 días hombre que fueron calculados basando en la experiencia en otros desarrollos similares.

Departamento	Días hombre.
Generación Reportes en Excel	32
Generación de Archivos de Corte	40
Administración de Días Festivos	11
Administración de Usuarios	27
Total días hombre:	110

2.8 Estimación de tiempo de desarrollo.

Al ser un sistema modular se pueden construir los diferentes procesos en paralelo, el proceso más grande es el módulo de generación de archivos de corte, con un tiempo de 40 días hombre (consultar la tabla 2.8). Para la elaboración de la parte correspondiente al front-ent, durante el mismo tiempo se debe contar con personal programador PL/SQL para apoyar a los programadores Java.

Al tomar en cuenta el riesgo de los días y ausentismo en que llegarán a incurrir los desarrolladores, los 40 días hombre que se necesitan para el proceso contable y considerando 20 días hábiles por mes se puede traducir en un mínimo de 3 meses (60 días hombre).

En la tabla 2.9 se concentran los costos de recursos humanos con un tiempo deseado de desarrollo de 3 meses.

Cantidad	Concepto	Costo	Duración	Total
1	Elaboración de análisis y diseño	\$100,000 único	1 mes	\$100,000
1	Analista certificado UML	\$35,000 mensual	2 Meses	\$70,000
3	Programador Certificado Java	\$28,000 mensual	3 Meses	\$252,000
1	Programador Store Procedure.	\$35,000 mensual	3 Meses	\$105,000
1	Líder de proyecto certificado	\$43,000 mensual	3 Meses	\$129,000
Total:				\$656,000

Tabla 2.9 Costos recursos humanos.

Al conocer el número de recursos humanos que se incorporarán al proyecto, en la tabla 3.0 se presentan los costos de hardware requeridos para el personal contemplado en la tabla 2.9.

Cantidad	Concepto	Costo	Total
1	PC analista	\$7,000	\$7,000
3	PC programador	\$8,000	\$24,000
1	PC servidor	\$18,000	\$18,000
1	Laptop	\$12,000	\$12,000
		Total:	\$62,000

Tabla 3.0 Costos hardware.

En la tabla 3.1 se concentran los costos de las partes requeridas para la construcción del Administrador de 200 depositadores automáticos utilizando el lenguaje de programación Java y DB2 en un tiempo estimado de 3 meses.

Rubro	Costo
Software	\$1,0421
Hardware	\$62,000
Recursos humanos	\$656,000
Total días hombre:	\$728,421

Tabla 3.1 Concentrado de costos

1.8. Factibilidad técnica.

La factibilidad técnica consiste en realizar una evaluación de la tecnología existente. En el apartado 1.4 se refiere al software propuesto y el punto 1.5 se concentra el hardware requerido para soportar el software.

Software: El desarrollo del proyecto lleva consigo la utilización de software libre y comercial que se concentra en la tabla 3.2.

TIPO	SOFTWARE
Sistema operativo	Windows XP Profesional SP3 o superior
Software de Desarrollo	ECLIPSE JEE Galileo SR1 32 bits
	Java Enterprise Edition 6 SDK
Base de Datos	DB2
Software para control de versiones.	Subversión 1.6.1
Software para documentación	Microsoft Office Professional 2007
Software para administración del proyecto	Microsoft Project 2007

Tabla 3.2. Software propuesto para el desarrollo, documentación y administración

Hardware: Las características del hardware requerido se concentran en la tabla 1.5.

Al existir la tecnología requerida de hardware y software para el desarrollo del proyecto se cumple con la factibilidad técnica.

1.9. Factibilidad operativa.

La factibilidad operativa permite predecir, si se pondrá en marcha el sistema propuesto, investigar si el personal existente es capaz de operar el sistema o se requiere de capacitación.

Para evitar el rechazo de los usuarios finales en el uso del sistema y facilitar su capacitación, el sistema debe tener una funcionalidad sencilla, intuitiva, amigable, que cubra todos los requerimientos y proporcione la información en forma oportuna y confiable.

Para aumentar la aceptación del equipo técnico de las IES, los estándares de desarrollo a utilizar son los propuestos por Sun Microsystems que son reconocidos a nivel mundial, y son conocimientos que requieren tener las personas que obtienen la certificación en programador Java. De la misma manera los estándares de desarrollo propuestos para la base de datos DB2 son los propuestos por la empresa IBM, al utilizar estos estándares se puede garantizar un mejor entendimiento con el área técnica responsable de dar soporte y mantenimiento al sistema.

Para la elaboración de la documentación se utilizará el Lenguaje Unificado de Modelado (UML, por sus siglas en inglés, Unified Modeling Language) que es el lenguaje de modelado de sistemas de software utilizado en la actualidad; está respaldado por el OMG (Object Management Group). Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un plano del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables.

En el siguiente capítulo se hará una breve descripción de los depositadores automáticos, así como sus características de software y de hardware.

Capítulo 2

2. Depositadores Automáticos.

2.1. Descripción.

La aceptación y el proceso de llenado de documentos para realizar un depósito ocupan la mayor parte del tiempo del personal bancario y del cliente, por lo tanto los depositadores automáticos ofrecen el mayor potencial de ahorro del mismo. La automatización de este proceso, en otras palabras, sustituir el papel bancario por actividades rápidas y sencillas, permite integrar al cliente en el proceso bancario mediante la utilización de una máquina capaces de proporcionar múltiples servicios de forma automática y totalmente desatendida a través de un interfaz interactivo, diseñado para que cualquier persona pueda utilizarlo, las cuales son llamadas terminales de autoservicio.

La institución bancaria HSBC introduce al mercado los depositadores de efectivo, los cuales son terminales de autoservicio (CDM – por sus siglas en inglés Cash Deposit Machine), en los cuales se pueden realizar depósitos a cuentas HSBC o pago de servicios y se muestra en la figura 2.1

Para obtener la aceptación del cliente es necesario que las operaciones en los CDM's sean familiares y de apariencia sencilla. El producto que HSBC ofrece a través del proveedor Wincor-Nixdorf y el cual traslada el procesamiento de efectivo a los cajeros automáticos, ofreciendo el Check/Cash Deposit Module (CCDM) es el equipo ProCash 3100xe. Este dispositivo permite al banco añadir un sistema de depósito de efectivo y cheques.



Figura 2.1

Con la utilización de los sistemas de autoservicio el largo proceso manual de entrada de efectivo queda enormemente simplificado y el trabajo administrativo se reduce significativamente. Al mismo tiempo, este es el primer paso hacia la automatización del proceso, el procedimiento de comprobación completo de efectivo hasta la disposición en la cuenta del cliente. Se acelera el proceso de operaciones en las cajas, no es necesario rellenar formularios al realizar depósitos y el personal bancario no perderá tiempo en el proceso de realizar los mismos

2.2. Ventajas para el banco y sus clientes.

- Menor coste del proceso de gestión de efectivo (hasta un 75% en la cadena de valor completa)
- Menor coste de transporte de papel
- El uso frecuente convierte los CDM's en una solución financiera más viable; las transacciones de autoservicio suponen una reducción del coste
- Menos efectivo circulando en la ventanilla
- El sistema inteligente de depósito calcula el total de las cantidades
- Los errores topológicos son identificados y corregidos inmediatamente
- Menos horas de trabajo para los cajeros, más horas disponibles para los clientes
- El efectivo es depositado inmediatamente en la cuenta del cliente
- Independencia de los horarios de apertura
- Los clientes reciben un listado de sus billetes/documentos y una imagen de los documentos en la pantalla que pueden confirmar o cancelar
- Por motivos de seguridad los documentos pueden ser impresos y puestos a la disposición del cliente

2.3. Especificaciones de Hardware.

Opciones para el manejo de efectivo:

- Dispensador de efectivo
- Sistema de depósito de efectivo
- Sistema de aceptación y devolución de efectivo
- Sistema de reciclaje de efectivo
- Carga frontal y trasera

Características sobresalientes:

- Dispensa hasta 8 denominaciones
- ECB approval as deposit + recycling system
- Sistema ideal de autoservicio
- Aceptación de faja de hasta 400 billetes
- Devolución de hasta 100 billetes
- Opciones de colores personalizados y etiquetas de braille

Descripción:

- Tecnología PC estandar
- Variantes de instalación con (marcos) prefabricados
- Certificado por EMV 2000
- Puede recibir hasta 400 billetes en un solo depósito
- Puede dispensar hasta 100 billetes
- Hasta 5 módulos de batería
- Hasta 4 cartucheras para dispensar efectivo
- Hasta 7 cartucheras para recibir depósitos
- 8 teclas de acero inoxidable
- Teclado numérico incluye 4 teclas funcionales
- Lectora motorizada

- Ranura a prueba de alteraciones
- Impresora de recibos y journal
- El panel de logo alumbrado\iluminado

2.4. Especificaciones de Software.

En este punto se realizará la descripción del software con el que trabajan los depositadores automáticos y que es necesario para su buen funcionamiento.

2.4.1. ProClassic/Enterprise.

2.4.1.1. Descripción.

Es una solución de software de última generación que responde a los requisitos y necesidades del canal de autoservicio. Su arquitectura abierta se basa en el estándar independiente e internacionalmente reconocido J2EE. Se implementa en la infraestructura de servidores de aplicaciones compatibles con J2EE, cuya escalabilidad lo convierten en una herramienta ideal para soportar soluciones centradas en la red.

Constituye una plataforma para el canal de autoservicio, pero además ofrece todas las funciones necesarias para diseñar una estrategia multicanal plenamente integrada. Esto facilita a las instituciones financieras la optimización de todos sus canales y el desarrollo paralelo de nuevas estrategias de negocio.

Está basado en componentes y se nutre de diversos módulos de software, cada uno capaz de aportar uno de los servicios concretos que se requieren en una solución de autoservicio. Estos servicios comprenden desde Smart Client, que corre en el mismo sistema de autoservicio, hasta servicios de negocio genéricos y específicos corriendo sobre el servidor, pasando por conectores host encargados de facilitar una integración transparente del entorno de transacciones y otros sistemas back-end.

2.4.1.2. Ventajas.

Reducidos costes de inversión y mantenimiento, dado que la infraestructura necesaria para adoptar ProClassic/Enterprise normalmente ya está disponible en los centros de datos de las entidades bancarias.

Simplificación de los procesos de negocio, en la medida que nuevas funciones, productos y servicios requieren una única implementación, pero pueden utilizarse a través de todos los canales.

Nuevos productos y servicios pueden lanzarse al mercado más rápidamente, debido a que no es necesario esperar a que se produzcan cambios del software en los sistemas host.

Mayor satisfacción del cliente a través de direcciones personalizadas, un look universal y un portafolio de servicios para todos los canales.

Menores costes operativos al poder delegar la lógica de procesos y decisiones en el servidor de aplicaciones central.

Los datos están disponibles en un punto central: optimización de los procesos de monitorización, gestión y evaluación del sistema de autoservicio. Se ahorran costes en la distribución de software.

Incremento de productividad mediante el uso de estándares, la reutilización de componentes, limitación de dependencias y el foco en integración y optimización.

Libre elección a la hora de seleccionar el proveedor de terminales, servidores y sistemas operativos.

Alta disponibilidad y escalabilidad gracias a su robustez y distribución de cargas.

2.4.1.3. Detalles.

ProClassic/Enterprise es una colección de componentes de software diseñada para una arquitectura centrada en la red. Dado que la mayoría de entidades bancarias cuentan ya con una moderna infraestructura TI equipada con servidores web y servidores de aplicaciones con el fin de gestionar canales individuales como

Internet Banking, el canal de autoservicio está normalmente separado de estas soluciones orientadas a la multicanalidad.

Así, los sistemas de autoservicio operan por lo general como una red independiente en un entorno vertical de canales específicos. Gracias a su arquitectura de n-capas, ProClassic/Enterprise puede ser integrado en canales específicos (cliente-host) así como en los entornos Internet y call center, soportando además los procesos de reestructuración de las modernas sucursales.

ProClassic/Enterprise cubre los requisitos del canal de autoservicio, y consiste en:

ProClassic/Enterprise Smart Client

ProClassic/Enterprise Server cuenta con

- 1) Conectores Smart Client
- 2) Servicios de Negocio
- 3) Conectores Host

Diseñado como suite de soluciones modulares, ProClassic/Enterprise cuenta con módulos listos para utilizarse en las funciones provistas por los sistemas de autoservicio, para el control del servidor de autoservicio y para la administración del servidor, así como para integrarse en los sistemas de gestión existentes, como ProView, IBM Tivoli, HP OpenView, etc.

Los departamentos TI del cliente, los partners de software y las instituciones financieras pueden utilizar el kit de desarrollo ProClassic/Enterprise para diseñar, integrar, implementar y poner en marcha sus propias soluciones.

2.4.2. Protopas/Suite.

2.4.2.1. Descripción.

ProTopas/Suite combina las ventajas de una aproximación de producto estándar con las de una plataforma de programación flexible. Los módulos de aplicaciones pre-diseñados facilitan un rápido despliegue de funciones estándares de terminal de autoservicio, así como una integración flexible de los requisitos individuales de cada cliente.

Gracias al soporte de servicios web, ProTopas/Suite constituye una plataforma de integración basada en el paradigma SOA (Arquitecturas Orientadas a Servicios).

2.4.2.2. Ventajas.

ProTopas/Suite proporciona módulos de aplicaciones basadas en la web para el despliegue de terminales de información, cajeros automáticos y sistemas de depósito. Gracias al uso de los estándares de la industria CEN/XFS y J/XFS, ProTopas/Suite soporta capacidades multivendedor.

Basado en la sólida arquitectura ProClassic, ProTopas/Suite proporciona una amplia gama de funciones de autoservicio, incluyendo el depósito de efectivo y el reciclado de efectivo. La posibilidad de acceso web on line combinada con el producto ProSales abre el camino para oportunidades de venta cruzada a través de ventas personalizadas.

2.4.2.3. Detalles.

ProTopas/Base
(provisión de funciones básicas para soportar, por ejemplo, terminales kiosco²¹)
ProTopas/Cash
(funciones que soportan máquinas dispensadoras de efectivo)

ProTopas/CashIn
(funciones que soportan el depósito de efectivo y máquinas recicladoras)
ProTopas/TopUp
(recarga de tarjetas prepago para teléfonos móviles)

ProTopas/Webservice
(interfaz abierto para comunicar con sistemas de terceros, como el servidor de proceso de cheques)

ProTopas/CashBox
(solución de depósito off line que permite depositar efectivo a bancos, empresas de transporte de seguridad y otros)

ProTopas/Development Suite
(herramientas y librerías del sistema para personalizar proyectos)

²¹ Cubículo de autoservicio, o Terminal de computación de datos libre situado en un entorno de venta al por menor

Capítulo 3

3. Diseño y Desarrollo del Problema.

En este capítulo, tomando en cuenta lo que se analizó en el capítulo número uno se hará el análisis y el diseño para el desarrollo de la aplicación la cual actualmente se encuentra productiva, haciendo uso de las tecnologías que mas se adaptaron a las necesidades del banco.

3.1. Descripción del problema.

Para resolver el problema que se planteo en el capítulo uno se propone colocar la aplicación Web en un Servidor Solaris con Websphere Application Server interactuando con una base de datos DB2 llamada BDPROCLA y que es de donde se obtendrá la información de todos los módulos.

Para poder ingresar en el sistema, la autenticación del usuario se realizará a través de un servicio que actualmente opera en el banco llamado (LDAP)²² y que se encarga de validar que los usuarios estén registrados en la institución.

Todo el desarrollo deberá realizarse sobre el framework²³ de Execution Services²⁴, como lenguaje de programación se utilizara java ya que en el capítulo uno destaco sobre .NET y bajo los estándares de diseño que marca el banco los cuales incluyen colores y tipos de letra institucionales.

La información de la que se alimenta el sistema se obtendrá de la Base de Datos llamada BDProcla la cual se encuentra en DB2, es de esta base de datos de donde se obtendrá la información necesaria para que se generen los archivos de transacciones y cortes que serán enviados por la aplicación de lunes a viernes de forma automática utilizando la herramienta llamada GTS²⁵. Esta descripción se puede ver representada en la figura 3.1, en la cual se pueden ver tanto servidores de base de datos como Web.

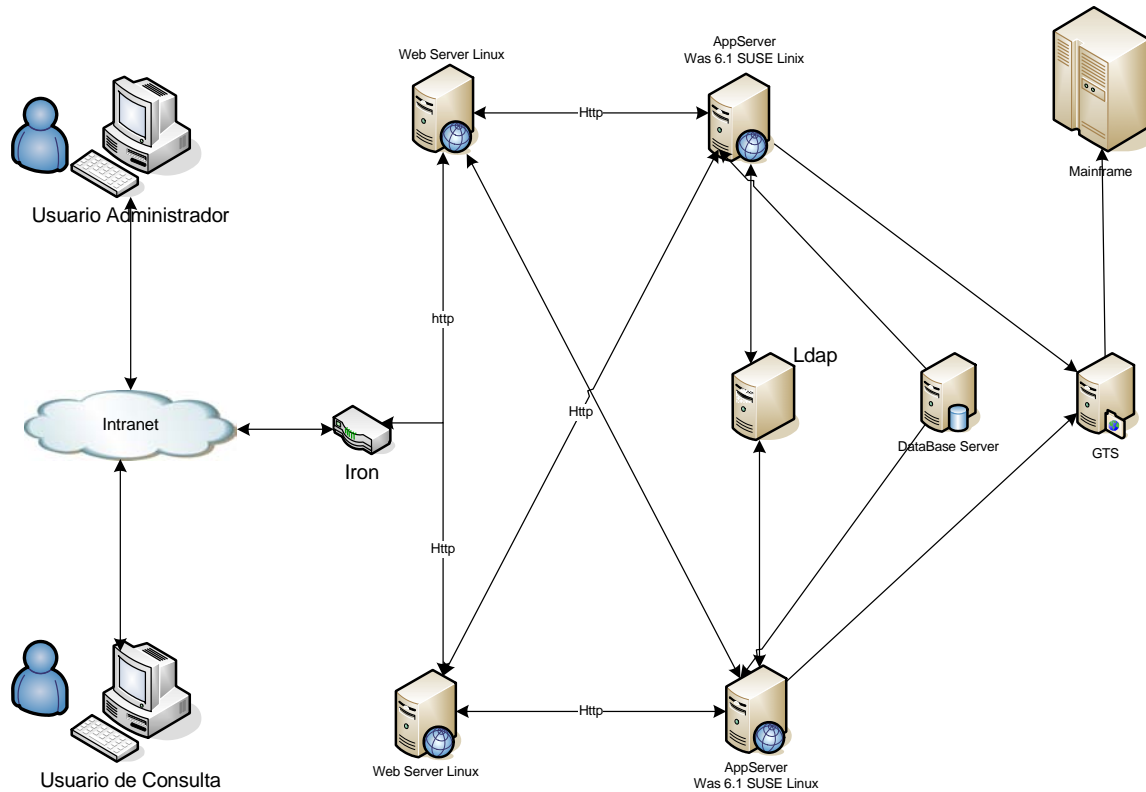
La aplicación se encuentra en un esquema de alta disponibilidad, esto quiere decir que si en por algún motivo alguno de los servidores ya sea de base de datos o Web llega a tener algún contratiempo, existe un respaldo, lo cual permite que el servicio que brinda el aplicativo siempre este disponible.

22 Lightweight Directory Access Protocol (Protocolo Ligero de Acceso a Directorios)

23 Implementaciones de patrones de diseño que facilitan la reutilización de diseño y código

24 Implementación del modelo 2/patrón de diseño MVC que facilita la creación de aplicaciones web en Java

25 Group Trade Services (Comercio Exterior del Grupo)



(Figura 3.1)

En la aplicación y tomando en cuenta el alcance, se podrán firmar 2 tipos de usuarios:

- Administrador y Usuario

Características del administrador y del usuario:

1) Administrador:

- Podrá agregar, borrar y modificar usuarios del sistema.
- Podrá generar los archivos de conciliación diaria y transmitirla.
- Podrá consultar las transacciones y exportarlas a Excel
- Podrá actualizar días festivos
- Solo deberá de existir un usuario con estas características y estará a cargo del líder de proyecto del área de Negocio.

2) Usuario:

- Podrá generar los archivos de conciliación diaria y transmitirla.
- Los usuarios de la información es principalmente Gestión y Soporte, Aclaraciones y logística monetaria.

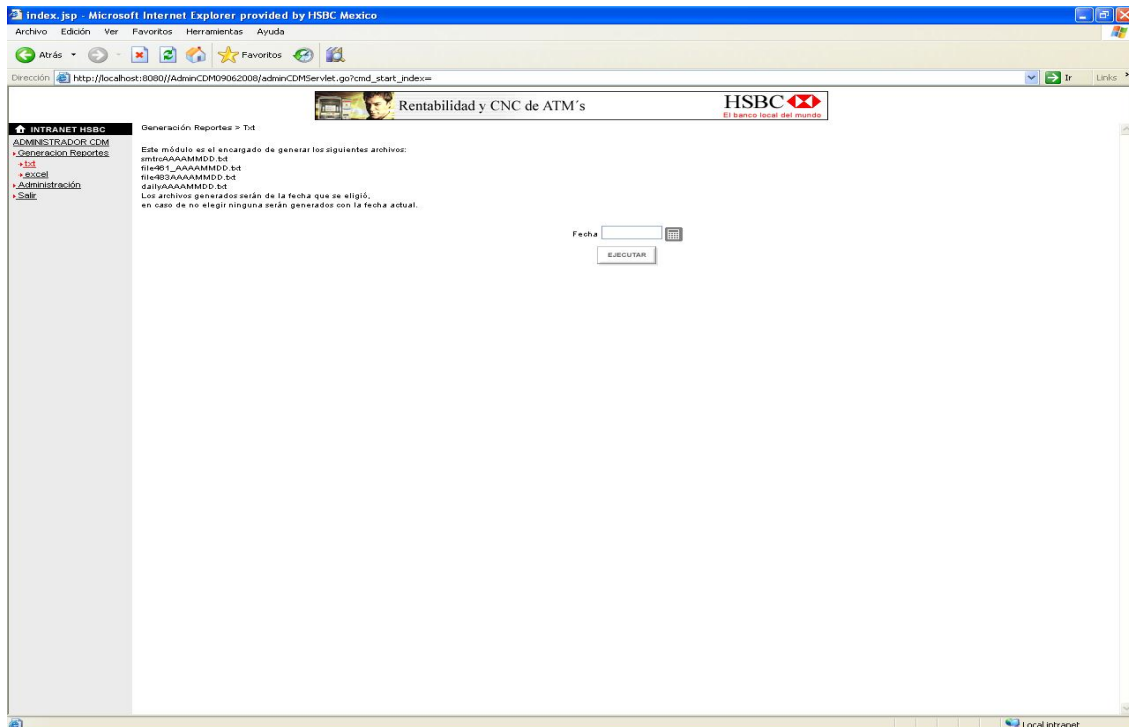
3.2. Diseño de la interfaz de usuario.

En este punto se mostrará un detalle de la parte visual de la interfaz del usuario y la cual debe de cumplir con las normas y estándares de la institución bancaria HSBC.

La aplicación Web que resolverá nuestro problema consta de 4 módulos y a continuación se muestran pantallas con su detalle.

3.2.1. Generación de archivos manual.

Este es el módulo en el cual se realizan los re-procesos en los casos en que ocurran contingencias y sea necesario el reenvío de los archivos de cortes y transacciones de forma manual.



3.2.2. Generación de reportes en Excel.

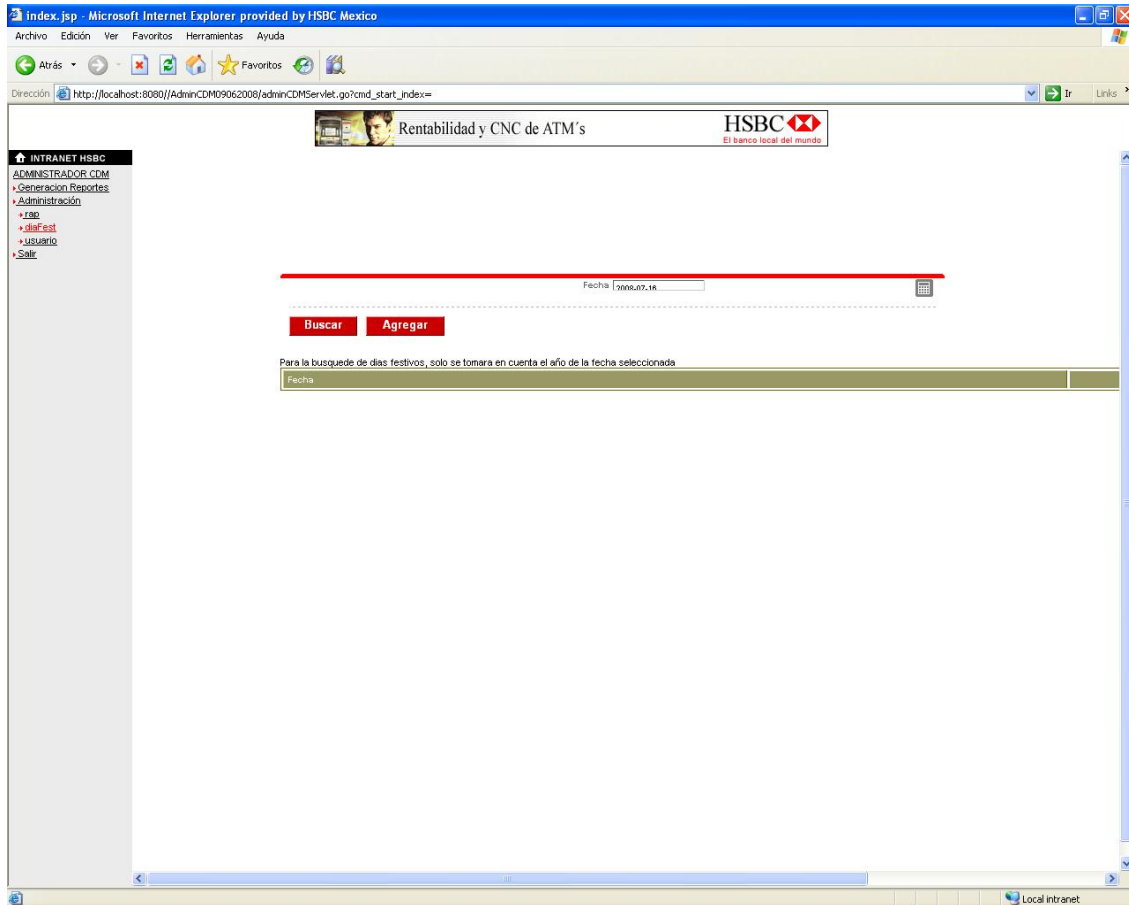
Este modulo es el encargado de generar un reporte de los depósitos hechos en los CDM's en formato Excel.

The screenshot shows a web browser window with the URL `http://localhost:8080/AdminCDM9062008/adminCDMServlet.do?cmd_start_index=`. The page title is "Rentabilidad y CNC de ATM's" and features the HSBC logo. A search interface includes fields for "Fecha Inicial" (2007.11.01) and "Fecha Final" (2007.11.30), with a "Buscar" button. Below the search area is a table with the following columns: Plástico, Fecha, Clase FX, Nemo, Nombre, Monto, Comercio, Origen, Fecha Operación, Hora, Folio, Cheques, Banco Auth, and a final column with values 0. The table contains 20 rows of transaction data.

Plástico	Fecha	Clase FX	Nemo	Nombre	Monto	Comercio	Origen	Fecha Operación	Hora	Folio	Cheques	Banco Auth	
00000000000000000000	20080718	1	D00003	D00003	00000001000	000000000	H	20071101	220444	0000000000023	000000004005641105	00000000000	0
00000000000000000000	20080718	1	D00003	D00003	00000000500	000000000	H	20071105	160749	0000000000031	000000004005641105	00000000000	0
00000000000000000000	20080718	1	D00004	D00004	00000001500	000000000	H	20071105	204802	0000000000000	000000004005641105	00000000000	0
00000000000000000000	20080718	1	S1A12345	S1A123456	00000001950	000000000	H	20071106	223143	0000000000015	00000000488888888	00000000000	0
00000000000000000000	20080718	1	D00004	D00004	00000050000	000000000	H	20071106	223225	0000000000032	00000000488888888	00000000000	0
00000000000000000000	20080718	1	D00004	D00004	00000025000	000000000	H	20071106	223945	0000000000034	00000000488888888	00000000000	0
00000000000000000000	20080718	1	D00004	D00004	00000095000	000000000	H	20071106	224135	0000000000035	00000000488888888	00000000000	0
00000000000000000000	20080718	2	S1A12345	S1A123456	00000001950	000000000	H	20071107	180843	0000000000031		00000000000	0
00000000000000000000	20080718	1	S1A12345	S1A123456	00000195000	000000000	H	20071107	184463	0000000000038	00000000488888888	00000000000	0
00000000000000000000	20080718	1	D00004	D00004	00000000500	000000000	H	20071107	210841	0000000000052	000000004005641105	00000000000	0
00000000000000000000	20080718	1	D00003	D00003	00000000800	000000000	H	20071107	212549	0000000000001	00000000488888888	00000000000	0
00000000000000000000	20080718	2	D00003	D00003	00000000100	000000000	H	20071107	212837	0000000000002		00000000000	0
00000000000000000000	20080718	1	D00004	D00004	00000000020	000000000	H	20071107	215025	0000000000053	000000004005641105	00000000000	0
00000000000000000000	20080718	1	D00004	D00004	00000002000	000000000	H	20071107	224638	0000000000056	000000004005641105	00000000000	0
00000000000000000000	20080718	1	D00004	D00004	00000350000	000000000	H	20071108	181855	0000000000063	00000000488888888	00000000000	0
00000000000000000000	20080718	1	D00004	D00004	00000077900	000000000	H	20071108	183417	0000000000067	00000000488888888	00000000000	0
00000000000000000000	20080718	1	D00004	D00004	00000002500	000000000	H	20071108	181504	0000000000002	00000000488888888	00000000000	0
00000000000000000000	20080718	1	D00003	D00003	000000001000	000000000	H	20071108	200301	0000000000005	000000004005641105	00000000000	0
00000000000000000000	20080718	1	S1A12345	S1A123456	00000272000	000000000	H	20071108	205417	0000000000003	00000000488888888	00000000000	0
00000000000000000000	20080718	1	D00003	D00003	000000001000	000000000	H	20071108	224421	0000000000012	000000004005641105	00000000000	0
00000000000000000000	20080718	1	D00003	D00003	00000010100	000000000	H	20071109	182557	0000000000016	00000000488888888	00000000000	0

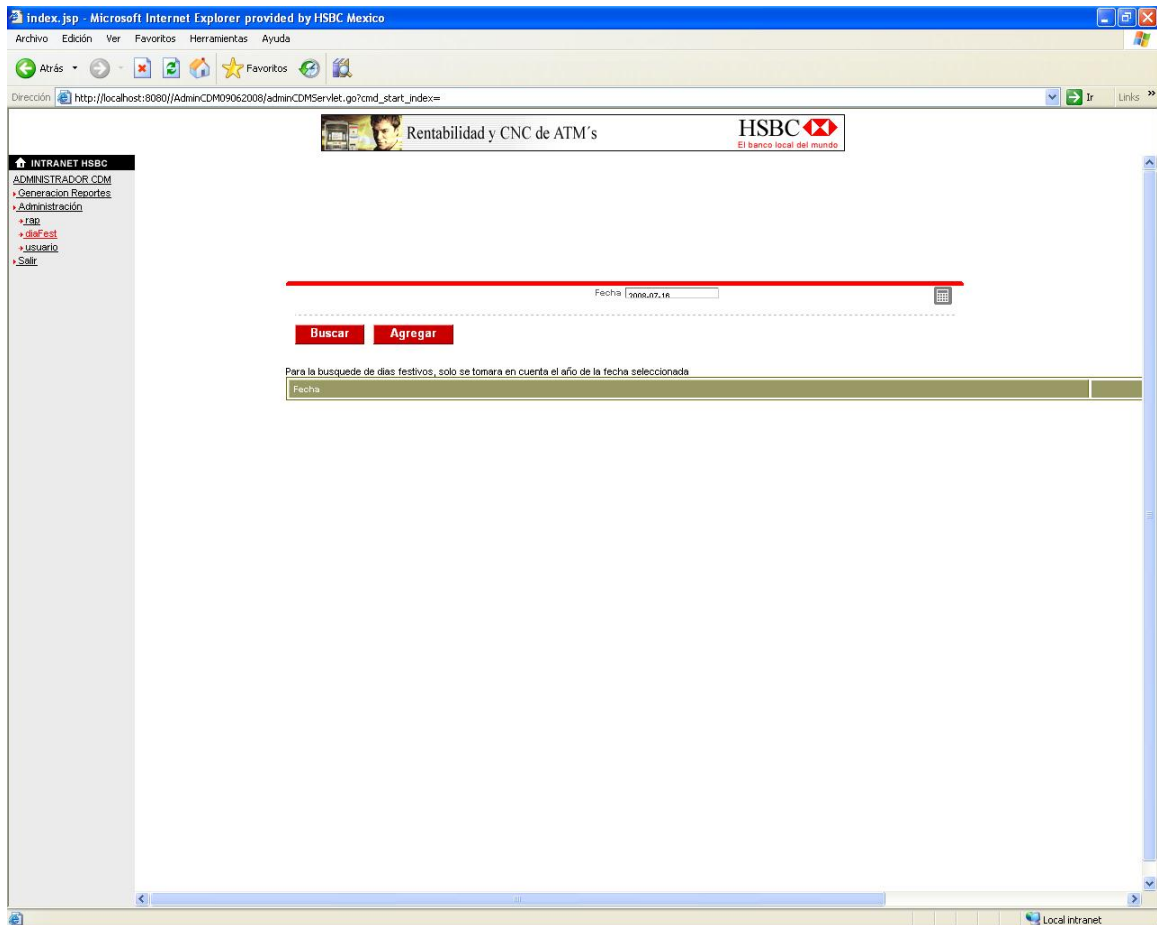
3.2.3. Administrador de días festivos.

En el módulo que se muestra, se podrán realizar altas, bajas y búsquedas de días festivos, los cuales nos indican si se deben o no generar los archivos de transacciones.



3.2.4. Administrador de usuarios.

Es el módulo al cual, solo el administrador puede entrar y donde se podrán dar de baja, alta y hacer actualizaciones a los usuarios del sistema.



3.3. Diseño y Construcción.

En este punto se utilizara el lenguaje de modelado de sistemas de software llamado UML el cual es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema.

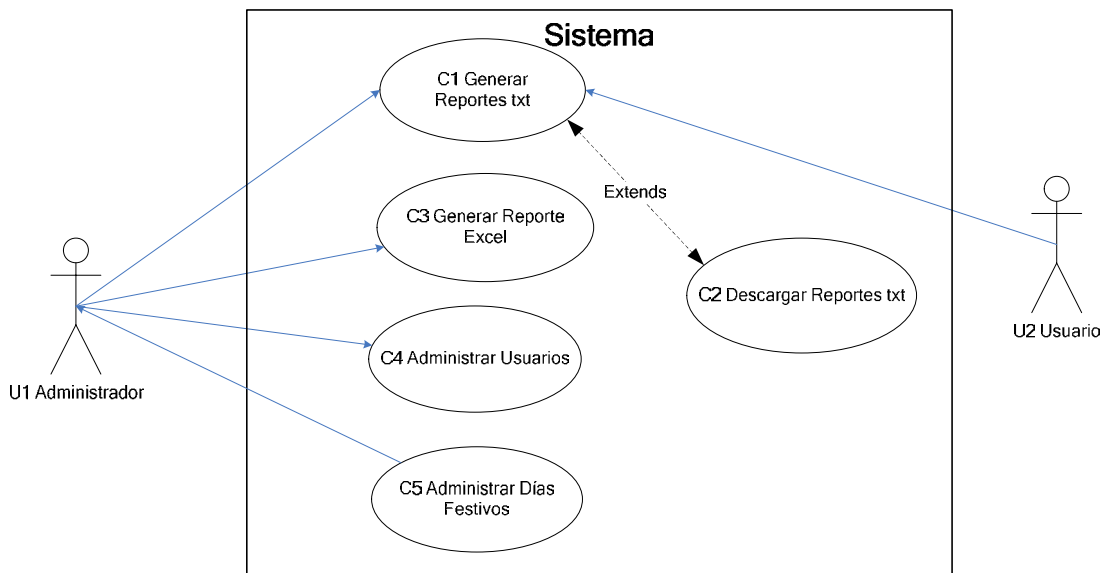
Un modelo UML muestra lo que supuestamente hará el sistema, más no como lo hará y esto lo hace mediante diagramas como son: diagramas de casos de uso, diagramas de clases, diagramas de estados y de secuencia y los cuales mostramos a continuación.

3.3.1. Diagramas de Casos de uso.

Un caso de uso es una descripción de la secuencia de interacciones que se producen entre actor y el sistema, cuando el actor usa el sistema para llevar a cabo una tarea específica.

Un actor es una entidad externa al sistema que realiza algún tipo de interacción con el mismo.

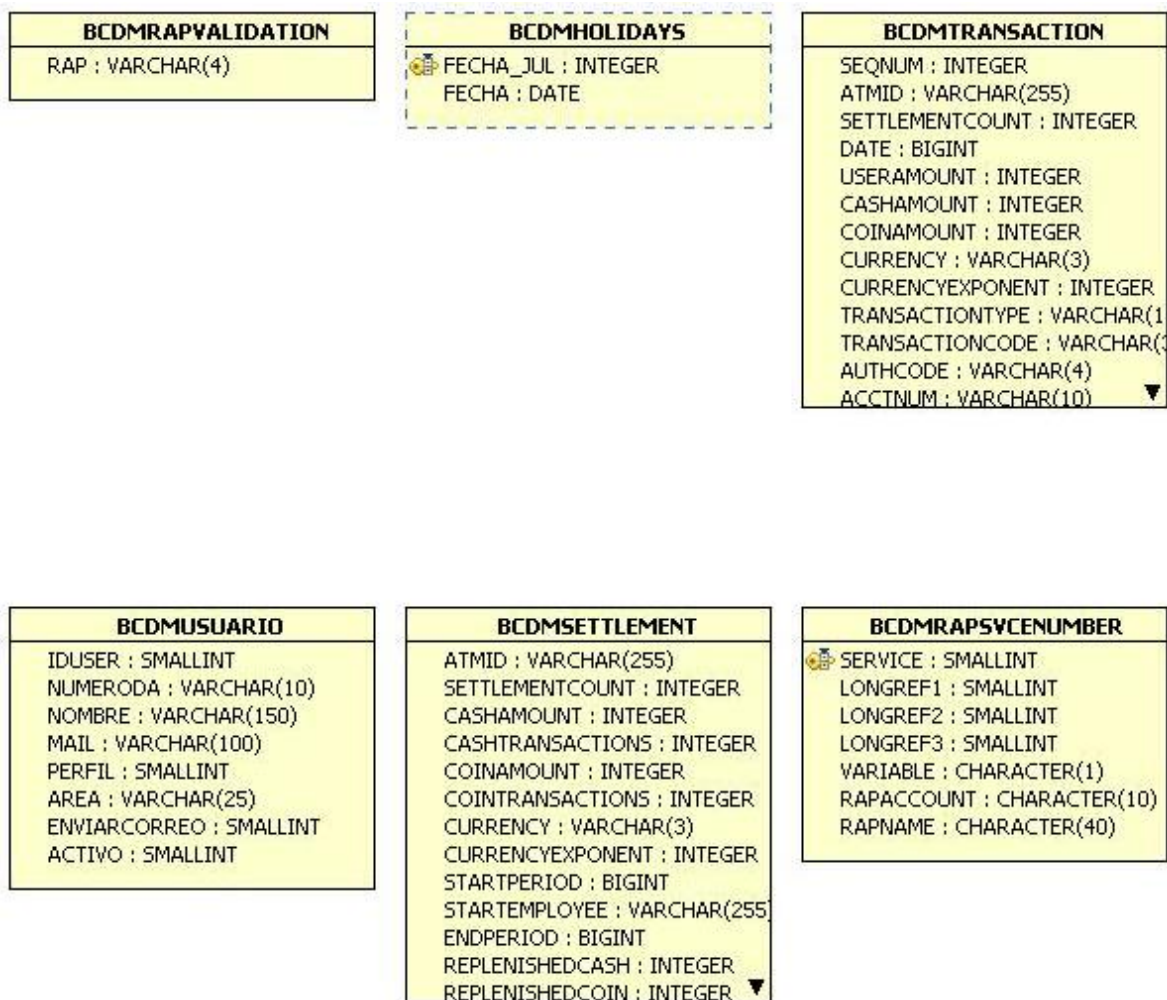
Un diagrama de casos de uso representa la funcionalidad que ofrece el sistema en lo que se refiere a su interacción externa y la relación entre actores y las funcionalidades del sistema.



3.3.2. Referencia de los casos de uso.

Código del Actor	Nombre del Actor	Descripción
U1	Administrador	Usuario con los suficientes privilegios para generar reportes en formato de texto y Excel, descargar dichos reportes y administrar usuario y días festivos.
U2	Usuario	Usuario con privilegio solo de generar reportes en Excel, texto y descargarlos

3.3.3. Diagrama Entidad Relación.



3.3.4. Diccionario de datos.**BCDMRAPVALIDATION**

Nombre de Columna	Tipo de Dato	Longitud	Nulo
RAP	VARCHAR	4	FALSE

BCDMHOLIDAYS

Nombre de Columna	Tipo de Dato	Longitud	Nulo
FECHA_JUL	INTEGER		FALSE
FECHA	DATE		FALSE

BCDMTRANSACTION

Nombre de Columna	Tipo de Dato	Longitud	Nulo
SEQNUM	INTEGER		FALSE
ATMID	VARCHAR	255	FALSE
SETTLEMENTCOUNT	INTEGER		FALSE
DATE	BIGINT		FALSE
USERAMOUNT	INTEGER		FALSE
CASHAMOUNT	INTEGER		FALSE
COINAMOUNT	INTEGER		FALSE
CURRENCY	VARCHAR	3	FALSE
CURRENCYEXPONENT	INTEGER		FALSE
TRANSACTIONTYPE	VARCHAR	1	FALSE
TRANSACTIONCODE	VARCHAR	3	FALSE
AUTHCODE	VARCHAR	4	FALSE
ACCTNUM	VARCHAR	10	FALSE

RAPREFNUM	VARCHAR	40	FALSE
CLISEQNUM	INTEGER		TRUE

BCDMUSUARIO

Nombre de Columna	Tipo de Dato	Longitud	Nulo
IDUSER	SMALLINT		false
NUMERODA	VARCHAR	10	true
NOMBRE	VARCHAR	150	true
MAIL	VARCHAR	100	true
PERFIL	SMALLINT		true
AREA	VARCHAR	25	true
ENVIARCORREO	SMALLINT		true
ACTIVO	SMALLINT		true

BCDMSETTLEMENT

Nombre de Columna	Tipo de Dato	Longitud	Nulo
ATMID	VARCHAR	255	false
SETTLEMENTCOUNT	INTEGER		false
CASHAMOUNT	INTEGER		false
CASHTRANSACTIONS	INTEGER		false
COINAMOUNT	INTEGER		false
COINTRANSACTIONS	INTEGER		false
CURRENCY	VARCHAR	3	false
CURRENCYEXPONENT	INTEGER		false
STARTPERIOD	BIGINT		false
STARTEMPLOYEE	VARCHAR	255	true
ENDPERIOD	BIGINT		true

REPLENISHED CASH	INTEGE R		true
REPLENISHED COIN	INTEGE R		true
ENDEMPLOYE E	VARCHA R	255	true

BCDMRAPSVCENUMBER

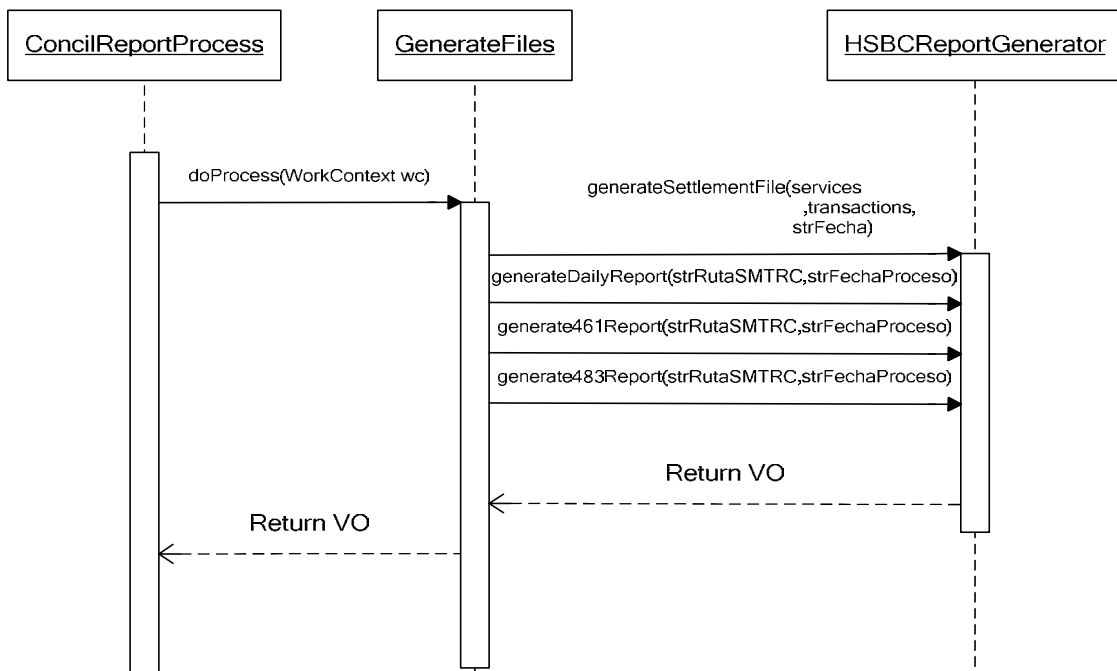
Nombre de Columna	Tipo de Dato	Longitud	Nulo
SERVICE	SMALLIN T		false
LONGREF1	SMALLIN T		true
LONGREF2	SMALLIN T		true
LONGREF3	SMALLIN T		true
VARIABLE	CHARAC TER	1	true
RAPACCOUNT	CHARAC TER	10	true
RAPNAME	CHARAC TER	40	true

3.3.5. Diagrama de secuencia.

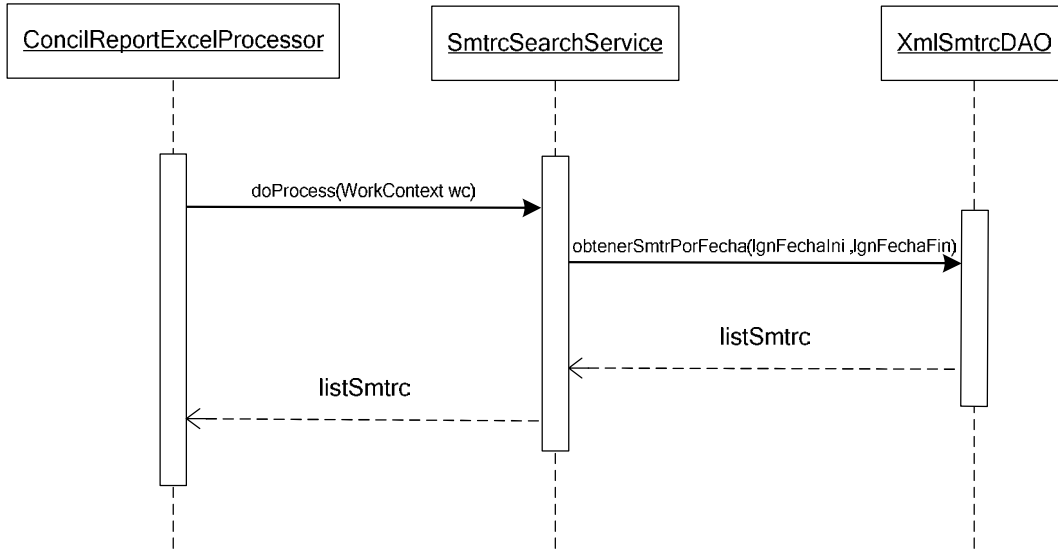
Un diagrama de secuencia muestra las interacciones entre objetos, ordenadas mediante una secuencia temporal. Muestra los objetos que se encuentran en un escenario y la secuencia de mensajes intercambiados entre los objetos para llevar a cabo la funcionalidad.

A continuación se muestran los diagramas de secuencia de los diferentes módulos.

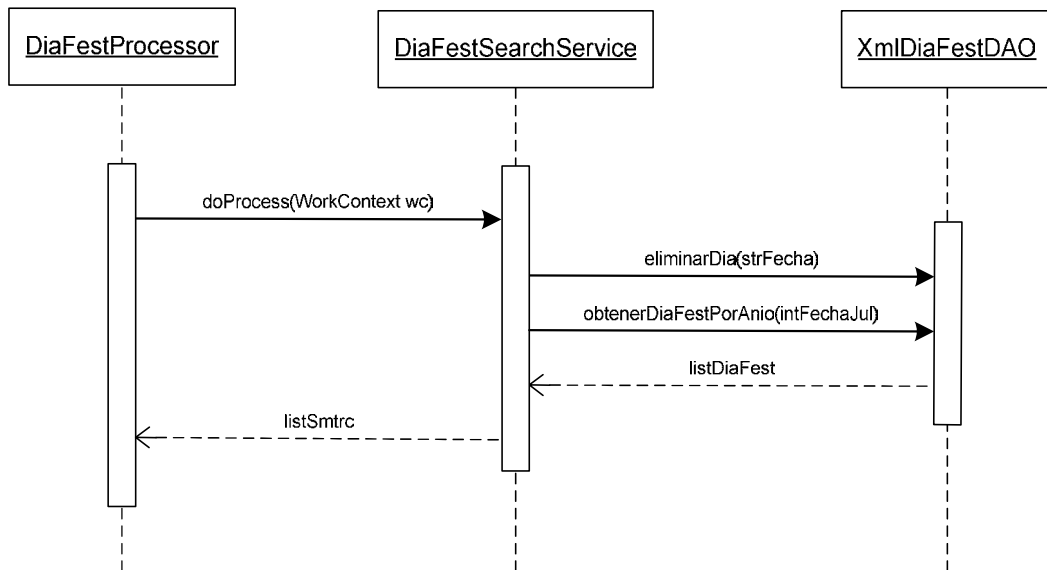
3.3.5.1. Modulo de Generación Reportes txt.



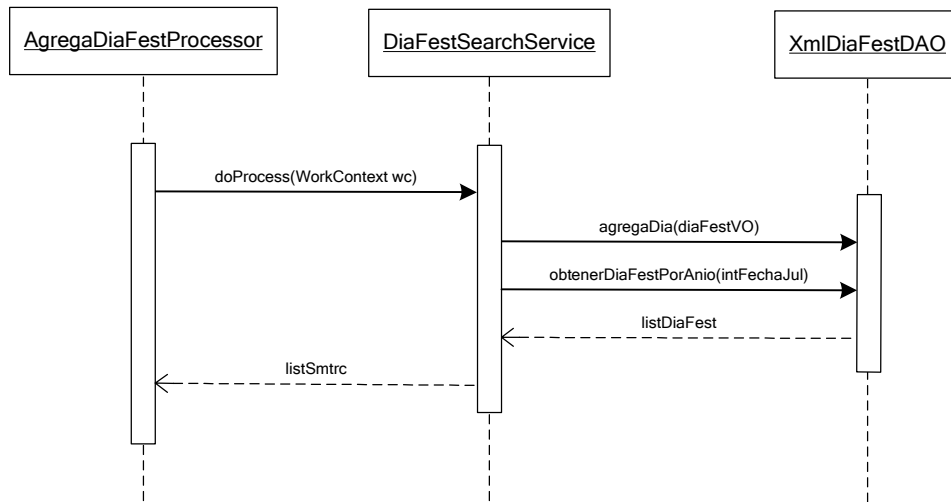
3.3.5.2. Generar Reportes en Excel.



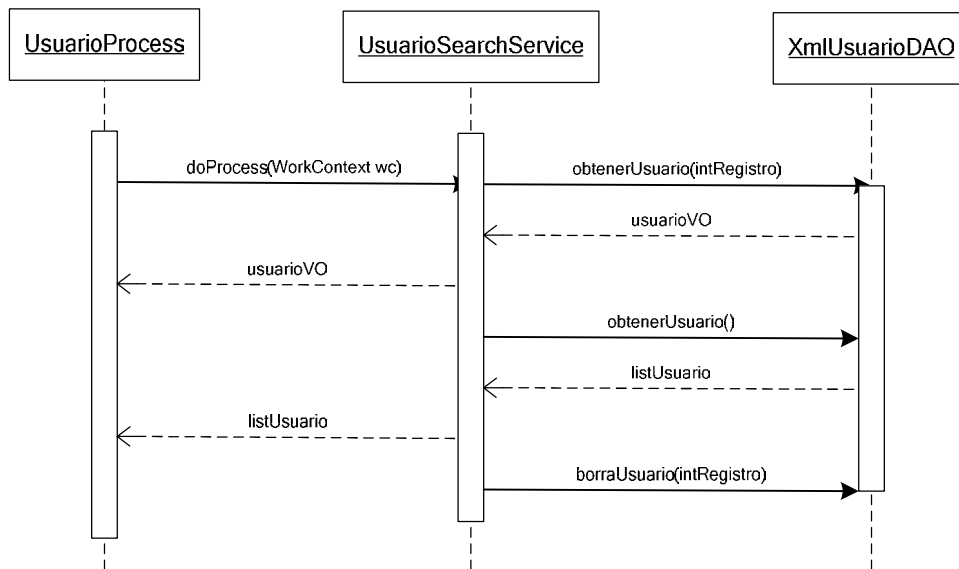
3.3.5.3. Consultar y Eliminar días festivos.



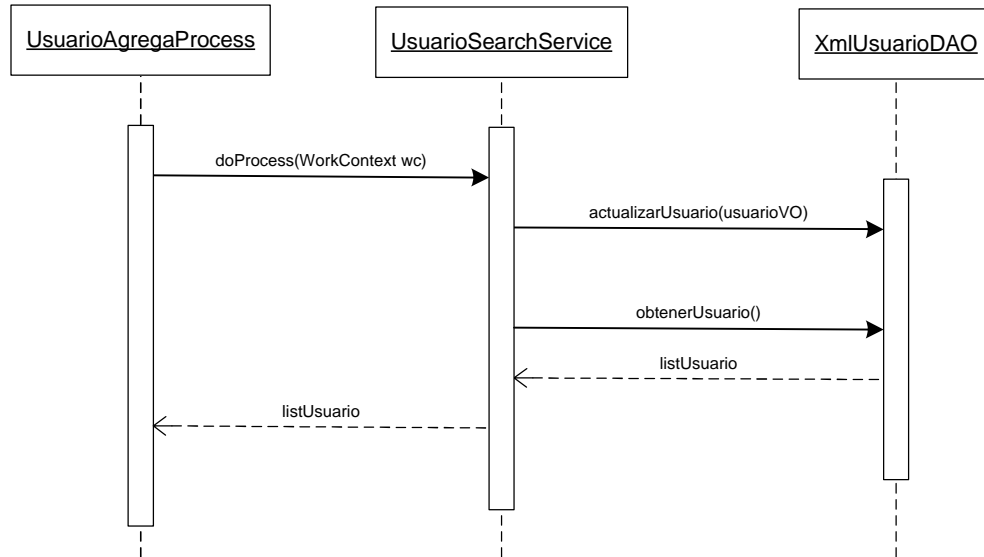
3.3.5.4. Insertar y actualizar Días Festivos.



3.3.5.5. Consultar y elimina usuarios.



3.3.5.6. Inserta y actualiza usuarios.



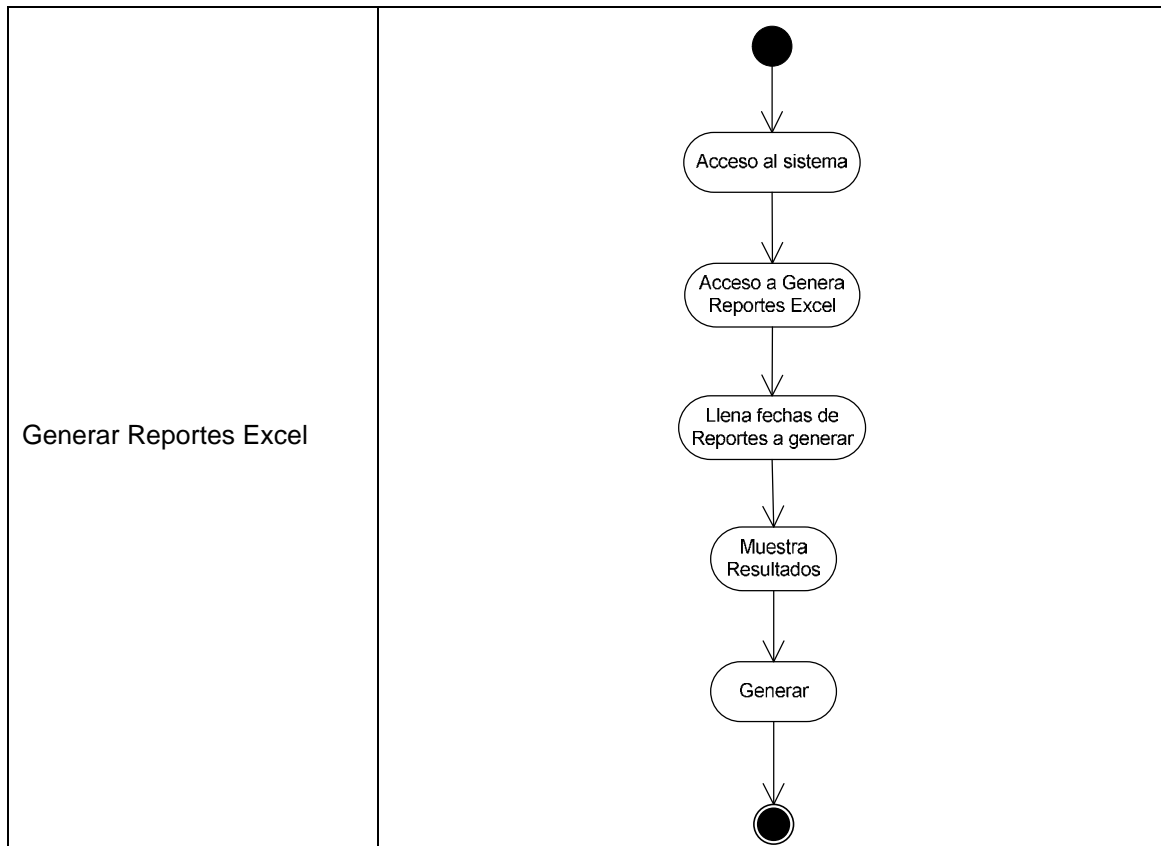
3.3.6. Diagrama de estados.

Es un diagrama utilizado para identificar cada una de las rutas o caminos que puede tomar un flujo de información luego de ejecutarse cada proceso.

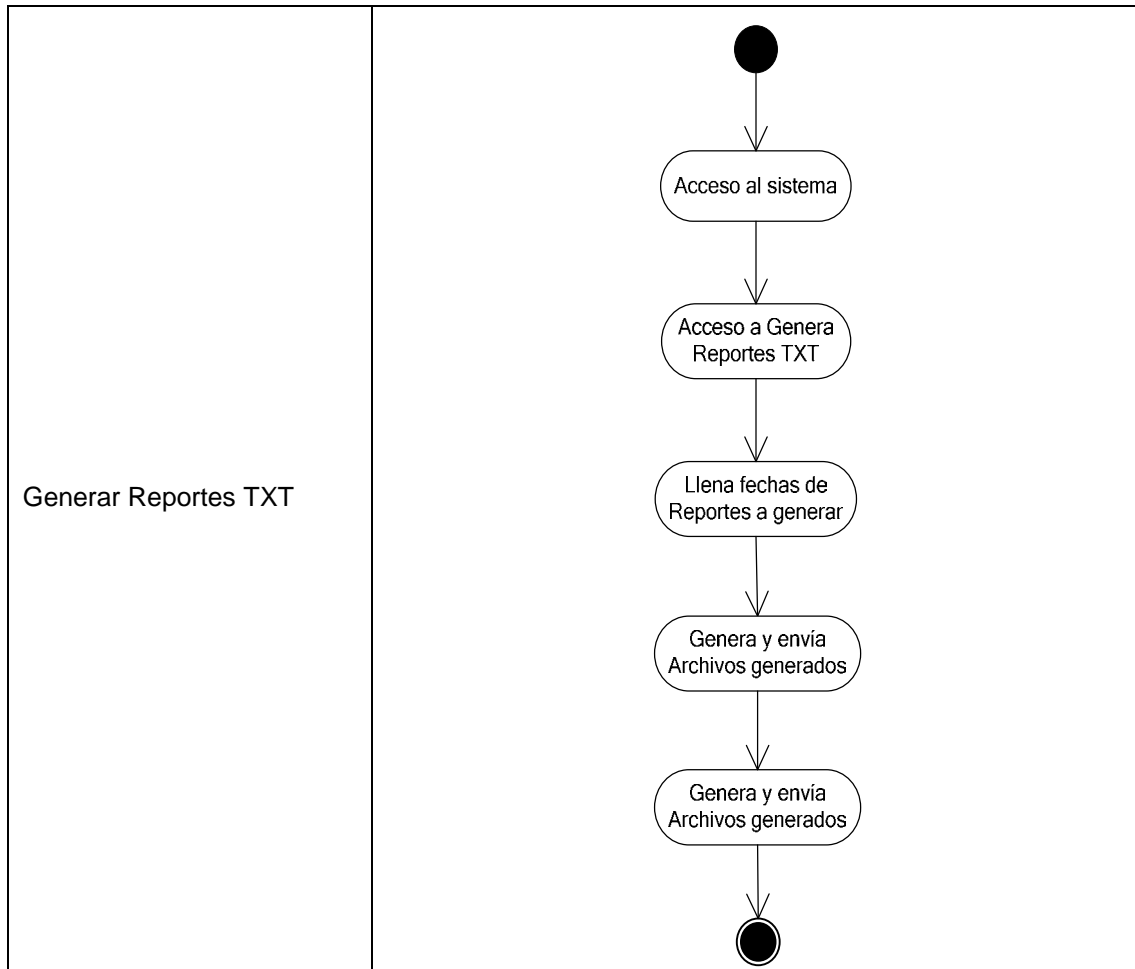
Permite identificar bajo qué argumentos se ejecuta cada uno de los procesos y en qué momento podrían tener una variación.

El diagrama de estados permite visualizar de una forma secuencial la ejecución de cada uno de los procesos.

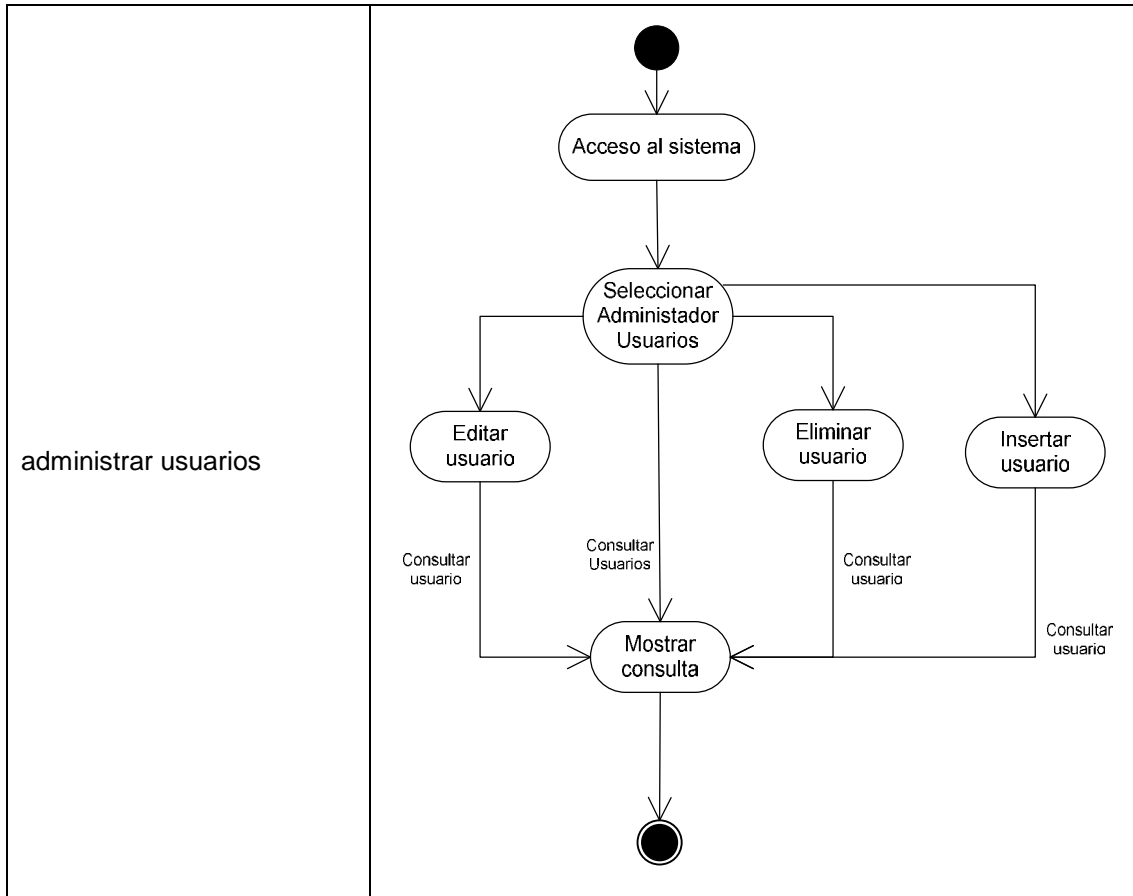
3.3.6.1. Diagrama de estados para la generación de reportes en Excel.



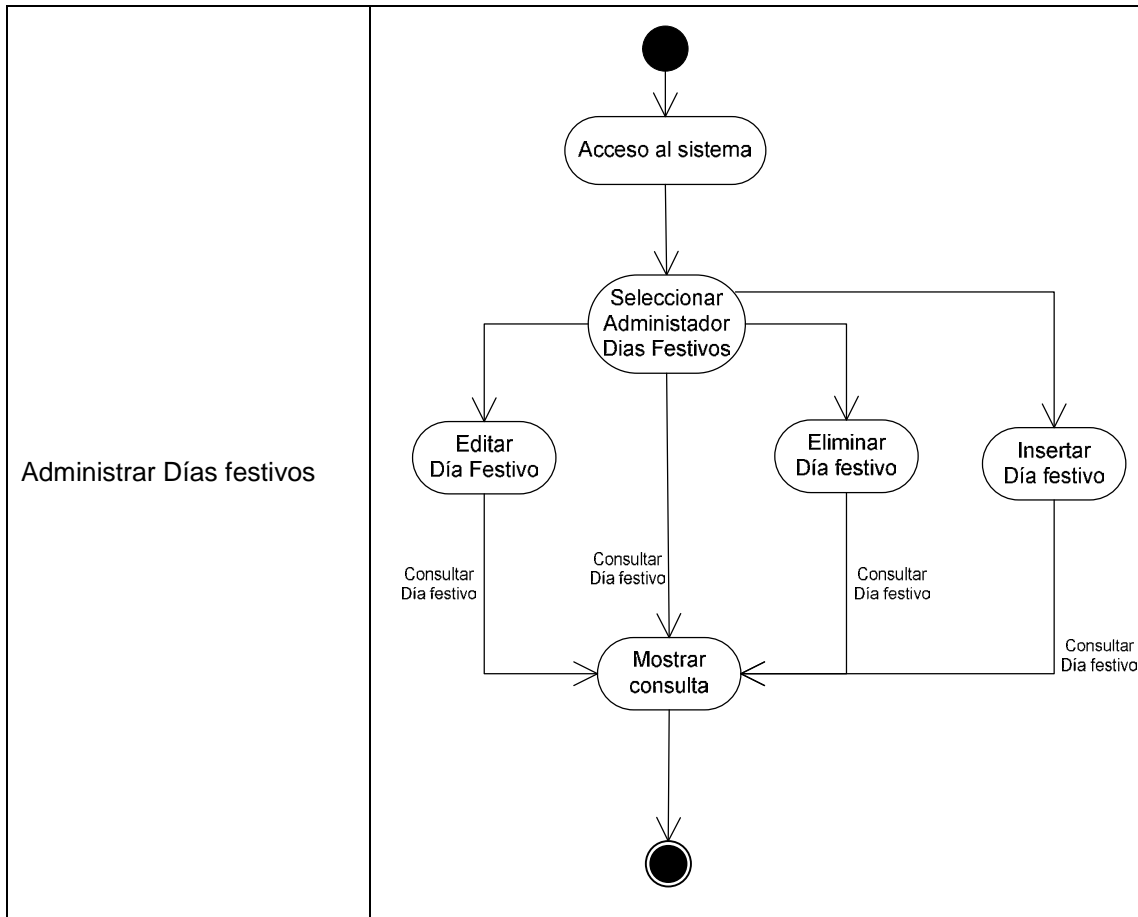
3.3.6.2. Diagrama de estados para la generación archivos de corte.



3.3.6.3. Diagrama de estados para la administración de usuarios.



3.3.6.4. Diagrama de estado para la administración de días festivos.



3.3.7. Diagrama de Clases.

Se encargan de mostrar las diferentes clases de las que se compone un sistema y la relación que existe entre ellas. Este tipo de diagramas son los estáticos porque muestran las clases junto con sus métodos y atributos, así como las relaciones estáticas entre ellas.



Capítulo 4

4. Pruebas y Resultados.

4.1. Pruebas de generación de reportes en TXT.

En estas pruebas se verificaron 2 situaciones.

- Que los archivos se generaran correctamente y fueran enviados a las etiquetas correspondientes en un lapso no mayor a 5 minutos.
- Que adicionalmente el proceso automático generara y enviara los archivos de manera desatendida.

4.2. Resultados de la generación y envío de reportes en TXT.

Se monitoreó durante una semana la generación y envío de archivos, para esto se involucró al usuario de la aplicación -área de Gestión & Soporte- y al área encargada de la herramienta GTS. Los resultados fueron satisfactorios en cuanto a la generación y envío de archivos y en cuanto a la parte de reproceso manual, el proceso tardaba entre 1 y 3 minutos desde que se generan hasta que se reciben en las etiquetas de mainframe y el usuario puede correr su proceso de conciliación.

4.3. Pruebas de generación de reportes en Excel.

En estas pruebas se verificó que el archivo en Excel se generara de manera correcta y con los datos solicitados en los campos de búsqueda

4.4. Resultados de generación de reportes en Excel.

Los datos se muestran correctamente y en el formato requerido, tanto en el explorador como en el archivo en Excel. Se recomienda mejorar el tiempo de respuesta del despliegue de datos en el explorador ya que cuando los datos son en grandes cantidades la aplicación se muestra lenta.

4.5. Pruebas del módulo Administración de usuarios.

En este módulo se realizaron 3 pruebas

- Dar de alta usuarios y verificar los permisos sobre el aplicativo.
- Editar datos de usuarios.
- Eliminar usuarios.

4.6. Resultados del módulo Administrador de usuarios.

De las tres pruebas que se corrieron, dos fueron satisfactorias y una no, al dar de alta un usuario el rol no se insertaba correctamente y no mostraba correctamente el menú. Se corrigió el problema y se volvieron a realizar pruebas con el usuario. Esta vez las 3 pruebas fueron satisfactorias.

4.7. Pruebas del módulo de Administrador de días festivos.

En este módulo se realizaron 3 pruebas.

- Dar de alta días festivos.
- Eliminar días festivos.
- Verificar que el depositador tome el día festivo y no permita depósitos.

4.8. Resultados del modulo de Administrador de días festivos.

De las tres pruebas que se hicieron una fue exitosa y dos salieron mal, los días no se estaban insertando correctamente y por tal motivo el depositador no tomaba el día festivo y permitía depósitos. Se corrigió este error y se volvieron a realizar pruebas con el usuario, esta vez las pruebas fueron satisfactorias.

4.9. Conclusiones.

Las presentes pruebas tuvieron resultados satisfactorios, después de realizar algunas correcciones, estas pruebas fueron necesarias para que se diera visto bueno por parte del usuario y finalmente el sistema fuera puesto en producción.

Conclusiones.

En el presente proyecto se dan a conocer algunos productos existentes en el mercado mexicano con sus ventajas y desventajas. Por ejemplo, algunas desventajas técnicas que se identificaron son: el uso de un lenguaje de programación, no ser multiplataforma, uso servidor de aplicaciones de un único proveedor entre otras.

El administrador de depositadores automáticos es una aplicación que actualmente genera información de operaciones bancarias como lo son depósitos y pago de servicios. Ha sido utilizado con éxito para casos de emergencia y se espera pueda terminar completamente con los problemas que se tenían

El proyecto se trató con cierta confidencialidad por lo delicado que es el manejo del dinero de las empresas bancarias, que es utilizado por los depositadores automáticos y que es abordado a través del documento

Actualmente los depositadores automáticos es una tecnología nueva y con algunas fallas, pero que actualmente con la ayuda de mejoras en el software y el hardware se han ido corrigiendo. En el futuro cercano se espera tener más de 200 equipos en el mercado a nivel nacional, los cuales se pretende que brinden servicio al público que desee realizar depósitos de una manera más rápida y fácil

Para poder realizar este desarrollo fueron necesarios conocimientos de programación, análisis y diseño de sistemas, así como de software y hardware. Fue necesario adquirir conocimientos de contabilidad y conciliación de transacciones.

Bibliografía.

E. Kendall, Kenneth y E. Kendall, Julie (2005). Análisis y Diseño de Sistemas. 6a. ed. México, Pearson

Flower Martin, (2003), UML Distilled. 3a. ed. (s.l.), Adison-Wesley.

Schmuller, Joseph (2001), Aprendiendo Uml En 24 Horas, (s.l.), Prentice Hall

Pérez, Agustín (2009), Modelo de Tecnología de Información para la adquisición y reemplazo de hardware y software, caso: Universidad Autónoma del Estado de México, Plan Rector de Desarrollo Institucional 2001-2005. Tesis profesional de Licenciatura, Universidad Iberoamericana, México D.F.

Penker, Magnus y Eriksson, Hans-Erik. (2000), Business Modeling With UML: Business Patterns at Work, 1 ed., USA, John Wiley & Sons.

Mesohemerografía

Depositadores de efectivo[En línea] Disponible en <http://www.wincor-nixdorf.com>

Proclassic Enterprise[En línea]Disponible en

http://www.wincor-nixdorf.com/internet/site_VE/VE/Products/Software/Banking/MultivendorMultichannel/ProClassicEnterprise/ProClassicEnterprise_node.html

IBM WebSphere[En línea] Disponible en

<http://www-142.ibm.com/software/products/mx/es/appserv-was/>

Apache Tomcat

<http://es.wikipedia.org/wiki/Tomcat>

<http://tomcat.apache.org>