

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA DE SISTEMAS

**MIGRACIÓN DE DATOS EN FOXBASE A PLATAFORMAS DE BASES
DE DATOS RELACIONALES**

**PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO
EN SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN**

DANNY FERNANDO PUENTE CARRERA

dannyfepu@gmail.com

SYLVIA ESTEFANÍA RAMÍREZ TAPIA

zelmaylula@gmail.com

DIRECTOR: ING. FRANCISCO VILLAVICENCIO

Quito, octubre 2008

DECLARACIÓN

Nosotros, Danny Fernando Puente Carrera y Sylvia Estefanía Ramírez Tapia, declaramos bajo juramento que el trabajo aquí descrito es de nuestra autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y que hemos consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedemos nuestros derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su reglamento y por la normatividad institucional vigente.

Danny Fernando Puente Carrera

Sylvia Estefanía Ramírez Tapia

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Sylvia Estefanía Ramírez Tapia y Danny Fernando Puente Carrera, bajo mi supervisión.

Ing. Francisco Villavicencio
DIRECTOR DE PROYECTO

AGRADECIMIENTOS

Principalmente quisiera agradecer a mis padres, quienes con su amor y comprensión me supieron educar en valores y siempre me dieron las pautas para hacer lo correcto, ya que gracias a ellos soy la mujer que soy.

A mi hermano ya que aunque no tengo la costumbre de pedir ayuda siempre estuvo a mi lado para darme ánimo y fuerzas para seguir adelante.

A mis hermanas que con sus peleas siempre me hicieron ver que había un mundo diferente fuera de las paredes de la universidad.

A mi sobrinita que con su dulzura y su afecto siempre tuvo un abrazo y un beso para darme cuando lo pedí.

A mi mejor amigo aquel que siempre estuvo allí cuando lo necesité sin importar la hora, el día, el lugar o la situación, para apoyarme darme ánimo cuando hacía falta y sobre todo por filosofar en aquellos momentos en los que lo único que quería era poder hablar. Gracias por ser mi confidente, escucharme y ayudarme a tomar algunas de las decisiones importantes en mi vida. Gracias por tu cariño y tu comprensión durante todo el tiempo que somos amigos, quisiera agradecerte tantas cosas pero creo que todo se resume en Gracias por ser Tú.

A mis amigos aquellos que sin importar lo doloroso que tuvieran que decirme siempre encontraron la forma de decirlo y a lo largo de la carrera fueron un apoyo incondicional para que el día de hoy pueda concluirla. Obviamente también les doy las gracias por los momentos de sano esparcimiento que compartimos por las charlas y los momentos felices, los cuales están almacenados en mi mente y en mi corazón. Prefiero no mencionar nombres ya que tengo miedo de olvidar alguno que

sea importante, pero estoy segura que cada una de las personas a quienes yo considero amigos sabrá que este párrafo es para ella, ya sea porque me conocen o porque saben lo importante que son en mi vida.

A aquella persona que fue parte importante de mi vida y me acompañó durante casi todo el transcurso de la carrera por enseñarme tantas cosas entre ellas a amar y ser amada, por demostrarme que el que persevera alcanza, por hacerme conocer y sentir la felicidad, por ser mi compañero de tesis y por guardar respaldos que aunque sé que es difícil para él, lo empezó a hacer. Gracias por haberte cruzado en mi camino ya que gracias a ti he madurado mucho y me diste el impulso que necesitaba para cumplir mis sueños.

A los Mafios, mi grupo de amigos que junto a ellos, a la par que empezamos la carrera hicimos crecer una amistad, la cual nos ayudo a conocernos más y gracias a la cual empecé a conocer más la ciudad ya sea por las reuniones o salidas. Gracias a cada uno de los que forman parte del grupo de amigos, por considerarme una amiga más y por demostrarme que cuando todos buscan un mismo objetivo es mejor estar juntos para alcanzarlo.

A nuestro tutor el Ing. Francisco Villavicencio por su paciencia y buena voluntad ya que en muchas ocasiones fue la clave para el poder continuar y finalmente terminar con éxito esta tesis, que sabe que nos ha costado mucho esfuerzo. Gracias también por compartirnos su conocimiento el cual nos será muy valioso en un futuro profesional.

Y finalmente al más importante al ser supremo, Dios, por escucharme, guiarme y enseñarme el camino a seguir cuando todas las luces estaban apagadas. También por darme tantas alegrías que son las que llenan mi corazón de júbilo y de esperanza.

Stefy

AGRADECIMIENTOS

Esta tesis de grado, si bien ha requerido de esfuerzo y mucha dedicación por parte de los autores y su director de tesis, no hubiese sido posible su finalización sin la cooperación desinteresada de todas y cada una de las personas que a continuación citaré y muchas de las cuales han sido un soporte muy fuerte en momentos de angustia y desesperación.

Primero y antes que nada, dar gracias a Dios, por estar conmigo en cada paso que doy, por fortalecer mi corazón e iluminar mi mente y por haber puesto en mi camino a aquellas personas que han sido mi soporte y compañía durante todo el periodo de estudio.

Agradecer hoy y siempre a mi familia, y está claro que si no fuese por el esfuerzo realizado por ellos, mis estudios universitarios no hubiesen sido posible. A mis padres Telmo y Aida, mis hermanos Junior y Cristian, porque a pesar de todos los malos ratos que les he hecho pasar, el ánimo, apoyo y alegría que me brindan me dan la fortaleza necesaria para seguir adelante.

De igual manera mi más sincero agradecimiento al Ing. Francisco Villavicencio mi director de tesis, y todos los profesores que de una u otra manera supieron inculcarme todos sus conocimientos y enfocarme a una vida profesional digna y respetable.

Un agradecimiento especial a mis grandes amigos; Xime, por su paciencia, apoyo brindados desde siempre y sobre todo por esa gran amistad que me brindó y me brinda, por escucharme, cuidarme y aconsejarme siempre, A Stefy, por hacer de mi carrera universitaria un hermoso recuerdo, a tu apoyo incondicional, comprensión, cariño y compañía que me supiste brindar en todo este tiempo, ya

que sin ti, mucho de lo que soy y he logrado no lo hubiese conseguido. Y lo más importante es que 5 años me has demostrado lo hermoso que es amar. Gracias porque a pesar de todo el daño que te he hecho has seguido ahí apoyándome y dándome ánimos sin importar lo que duela, se que vas a salir adelante y vas a lograr todo lo que te propongas, suerte, y gracias por ser más fuerte de lo que se puede ser, a Omar, por brindarme su ayuda cuando más la necesitaba, por ser una persona con la que puedo contar siempre, por los ánimos que me diste y por confiar siempre en mi, a Héctor por su apoyo, ánimo, por compartir conmigo muchos momentos tanto alegres como tristes, por tener siempre tendida su mano amiga, por escucharme, en fin, por darme su amistad desde el día en que nos conocimos, a Evita, porque en ti encontré al principio de mi carrera a la mejor amiga que alguien puede encontrar, no tienes idea lo mucho que te quiero y que ese sentimiento siempre estará ahí aunque la vida se encargó de mandarnos por distintos caminos, se que eres feliz y a mí también me hace feliz, a Fabián porque me has demostrado ser una gran persona y sobre todo un gran amigo, con el que sé que podré contar siempre, y lo más importante porque siempre estuviste ahí desde el comienzo compartiendo tus conocimientos conmigo sin esperar nada a cambio, de verdad te agradezco mucho por eso, a mis demás amigos que de una u otra manera fueron muy importantes en mi vida universitaria, Edu, Jaime, Pao, Carlitos C., Andy, Víctor, José, los Mafios, el grupo Doteró, a los Papayos, a mis compañeros de Carrera, a los que hoy ya son Ingenieros. En general a todas y cada una de las personas que han vivido conmigo la realización de esta tesis, con sus altos y bajos, desde los más profundo de mi corazón les agradezco el haberme brindado todo el apoyo, colaboración, ánimo y sobre todo cariño y amistad.

A Viki mi amor, tal vez al final, pero nunca la menos importante, por llegar a mi vida y hacerme el hombre más feliz del mundo, por tu paciencia, cariño, amor y comprensión durante todo este tiempo que me tomo terminar la tesis.

Danny

DEDICATORIA

A ti... que me escuchaste y me ayudaste cuando tenía problemas.

A ti... que estuviste a mi lado y fuiste mi compañía.

A ti... que confiaste y creíste en mi ciegamente aunque todo el mundo dudara.

A ti... que me animaste y sacaste una sonrisa cuando lo único que quería era llorar.

A ti... que me escuchaste y estuviste a mi lado aun sin saber cuanto te necesitaba.

A ti... que estuviste siempre presente aun cuando me asomara sin avisarte.

A ti... que esperaste el mejor momento para hablar conmigo y hacerme reaccionar.

Y por sobre todo a mi familia que siempre estuvo a mi lado durante todo el transcurso de la carrera.

Stefy

DEDICATORIA

Esta tesis de grado está dedicada a todas aquellas personas que supieron confiar en mí y que me apoyaron en todo el transcurso de mi carrera, a aquellos que de una u otra forma, en las buenas y en las malas, siempre han sabido tenderme su mano amiga.

Danny

CONTENIDO

INTRODUCCIÓN.....	1
CAPÍTULO 1. SELECCIÓN DE MODELO DE DESARROLLO	4
1.1 SELECCIONAR EL MODELO DE DESARROLLO	4
1.1.1 <i>METODOLOGÍA PARA EL DESARROLLO DE SOFTWARE</i>	4
1.1.2 <i>Modelo de CASCADA</i>	5
1.1.2.1 Ingeniería y Análisis del Sistema	6
1.1.2.1.1 Especificaciones de Requerimientos del Sistema.....	6
1.1.2.1.2 Especificaciones del Desarrollo del Sistema	6
1.1.2.2 Análisis de los Requerimientos.....	7
1.1.2.3 Diseño.....	7
1.1.2.4 Codificación.....	8
1.1.2.5 Pruebas	8
1.1.2.6 Mantenimiento	8
1.1.3 <i>Justificación del Modelo de Desarrollo</i>	9
1.2 SELECCIONAR LENGUAJE DE PROGRAMACIÓN.....	10
1.2.1 <i>Factibilidad de desarrollo con .NET</i>	10
1.2.1.1 Características Principales.....	11
1.2.2 <i>C#.NET: características PRINCIPALES</i>	12
1.2.2.1 Sencillez	12
1.2.2.2 Modernidad	12
1.2.2.3 Orientación a objetos	13
1.2.2.4 Orientación a componentes.....	13
1.2.2.5 Seguridad de tipos.....	13
1.2.2.6 Instrucciones seguras.....	13
1.2.2.7 Sistema de tipos unificado	14
1.2.2.8 Extensibilidad de tipos básicos	14
1.2.2.9 Extensibilidad de operadores	14
1.2.2.10 Extensibilidad de modificadores	15
1.2.2.11 Versionable.....	15
1.2.2.12 Eficiente:.....	16
1.2.2.13 Compatible.....	17
CAPÍTULO 2. DESARROLLO DE LA APLICACIÓN	18
2.1 DISEÑO	18
2.1.1 <i>Ingeniería y Análisis del Sistema</i>	18
2.1.1.1 Especificaciones de Requerimientos del Sistema.....	18
2.1.1.1.1 Retos	18
2.1.1.1.2 La solución de Informática	19
2.1.1.2 Especificaciones del Desarrollo del Sistema	19
2.1.2 <i>Análisis de los Requerimientos</i>	20
2.1.2.1 Lo que debe hacer la herramienta	20
2.1.2.2 Lo que no debe hacer la herramienta.....	21
2.1.3 <i>Diseño</i>	22
2.1.3.1 Interfaz de Usuario	22
2.1.3.2 Diseño General de la Interfaz de Usuario	22
2.1.3.3 Estructura de Foxbase.....	23
2.1.3.3.1 Definición del foxbase.....	23
2.1.3.3.2 Archivos en Foxbase	24
2.1.3.3.2.1 La cabecera del .DBF.....	24

2.1.3.3.2	Registros de Datos	29
2.1.3.3.3	Definición de Base de Datos.....	31
2.1.3.3.4	Tipos de Datos A Utilizar	33
2.1.3.3.5	Modelo de base de datos relacional.....	36
2.1.3.3.6	Cardinalidad de Relaciones.....	36
2.2	PLANIFICACIÓN.....	37
2.2.1	<i>Responsables del Proceso de Migración</i>	38
2.3	IMPLEMENTACIÓN.....	41
2.3.1	<i>Diagrama de Flujo</i>	41
2.3.2	<i>Estructura de la Herramienta dentro de la Programación</i>	42
2.3.3	<i>Codificación</i>	43
2.3.4	<i>Funcionamiento de la Herramienta</i>	48
2.3.4.1	Selección de la Opción a Realizar en la Herramienta	48
2.3.4.1.1	Crear Tabla.....	48
2.3.4.1.2	Generar Script	49
2.3.4.2	Selección de la Base de Datos Destino	54
2.3.4.3	Parámetros de Conectividad.....	54
2.3.4.4	Selección de la Base de Datos	56
2.3.4.5	Selección del Archivo a Migrar	57
CAPÍTULO 3. PRUEBAS		62
3.1	ELABORACIÓN DE APLICACIONES EN MODO GRAFICO	62
3.1.1	<i>Descripción del Caso de Prueba</i>	62
3.1.1.1	Misión.....	62
3.1.1.2	Objetivos	62
3.1.1.2.1	La Administración Financiera en la Cooperativa Politécnica.....	63
3.1.1.3	Situación Tecnológica	63
3.1.2	<i>Definición del Problema</i>	64
3.1.3	<i>Descripción del nuevo Sistema</i>	64
3.1.4	<i>Implementación con Software LIBRE</i>	65
3.1.4.1	Base de Datos PostgreSQL	65
3.1.4.1.1	Características	66
3.1.4.1.2	Otras características.....	66
3.1.4.1.3	Funciones	67
3.1.4.2	Servidor HTTP Apache.....	68
3.1.4.2.1	Ventajas.....	68
3.1.4.2.2	Módulos.....	69
3.1.4.3	PHP	70
3.1.4.3.1	Usos de PHP	70
3.1.4.3.2	Ventajas de PHP	71
3.1.4.4	JavaScript	71
3.1.4.5	Navegador: Mozilla Firefox.....	72
3.1.5	<i>Modelos de Bases de Datos</i>	73
3.1.5.1	Modelo de Bases Datos basado en XBase	73
3.1.5.2	Modelo de Base de Datos Relacional del Sistema de Caso de Prueba.....	76
3.1.5.2.1	Esquema DOLA2008	77
3.1.5.2.2	Esquema Prestaciones	78
3.1.6	<i>Interfaz del Sistema de Caso de Prueba</i>	79
3.1.6.1	Acceso al Sistema	80
3.1.6.2	Pantalla Principal	81
3.2	PRUEBAS DE ACOPLAMIENTO	82
3.2.1	<i>Tipos de Pruebas</i>	82
3.2.1.1	Pruebas de Migración de Datos	83
3.2.1.2	Pruebas de Integración	85
3.2.1.3	Pruebas del Sistema	91
3.2.1.3.1	Pruebas de Recuperación	91

3.2.1.3.2	Pruebas de Seguridad	93
3.2.1.3.3	Pruebas de Resistencia	94
3.2.1.3.4	Pruebas de Rendimiento.....	94
CAPÍTULO 4. CONCLUSIONES Y RECOMENDACIONES		96
4.1	CONCLUSIONES	96
4.2	RECOMENDACIONES	97
BIBLIOGRAFÍA		98
ANEXOS		102

ÍNDICE DE TABLAS Y FIGURAS

INTRODUCCIÓN.....	1
SELECCIÓN DE MODELO DE DESARROLLO.....	4
FIGURA 1. PROCESO DE LA METODOLOGÍA EN CASCADA	6
CAPÍTULO 2. DESARROLLO DE LA APLICACIÓN	18
FIGURA 2. INTERFAZ DE MIGRACIÓN	23
TABLA 1.A. ESTRUCTURA DE LA CABECERA DE LOS ARCHIVOS .DBF	25
TABLA 1.B. ESTRUCTURA DE LA CABECERA DE LOS ARCHIVOS .DBF	27
TABLA 2. VALORES DE TIPOS DE CAMPOS	28
TABLA 3. SISTEMA DE ARCHIVOS VS. SISTEMA DE BASE DE DATOS.....	32
TABLA 4. EQUIVALENCIAS DE TIPOS DE DATOS PARA LOS MOTORES DE BASES DE DATOS	36
FIGURA 3. DIAGRAMA DE FLUJO	42
FIGURA 4. MIGRACIÓN POR SELECCIÓN DE LA OPCIÓN TABLA	49
FIGURA 5. SELECCIÓN DEL ARCHIVO DBF PARA GENERAR SCRIPT EN SQL	50
FIGURA 6. MIGRACIÓN POR SELECCIÓN DE LA OPCIÓN GENERAR SCRIPT	51
FIGURA 7. PROGRESO DE GENERACIÓN DEL SCRIPT EN SQL.....	52
FIGURA 8. GENERACIÓN DEL SCRIPT EXITOSA	52
FIGURA 9. SELECCIÓN DEL NOMBRE Y UBICACIÓN DEL ARCHIVO DEL SCRIPT GENERADO.....	53
FIGURA 10. SCRIPT GENERADO	53
FIGURA 11. SELECCIÓN DEL MOTOR DE BASE DE DATOS PARA EL PROCESO DE MIGRACIÓN.....	54
FIGURA 12. PARÁMETROS DE CONECTIVIDAD A LA BASE DE DATOS.....	55
FIGURA 13. CONEXIÓN EXITOSA A LA BASE DE DATOS.....	56
FIGURA 14. CONEXIÓN FALLIDA A LA BASE DE DATOS	56
FIGURA 15. SELECCIÓN DE LA BASE DE DATOS PARA EL PROCESO DE MIGRACIÓN.....	57
FIGURA 16. SELECCIÓN DEL ARCHIVO DBF A MIGRAR.....	58
FIGURA 17. PREVIO A LA MIGRACIÓN DE LOS DATOS.....	59
FIGURA 18. PROCESO DE MIGRACIÓN DE DATOS.....	60
FIGURA 19. MENSAJE DE MIGRACIÓN EXITOSA.....	61
FIGURA 20. MENSAJE DE MIGRACIÓN FALLIDA.....	61
CAPÍTULO 3. PRUEBAS	62
FIGURA 21. LOGO POSTGRESQL	65
FIGURA 22. LOGO SERVIDOR HTTP APACHE	68
FIGURA 23. LOGO PHP	70
FIGURA 24. LOGO MOZILLA FIREFOX.....	72
FIGURA 25. MODELO DE BASES DATOS BASADO EN XBASE.....	76
FIGURA 26. ESQUEMA DOLA2008.....	78
FIGURA 27. ESQUEMA PRESTACIONES	79
FIGURA 28. PANTALLA DE ACCESO AL SISTEMA (SISTEMA ANTIGUO)	80
FIGURA 29. PANTALLA DE ACCESO AL SISTEMA (SISTEMA NUEVO)	80
FIGURA 30. PANTALLA PRINCIPAL (SISTEMA ANTIGUO)	81
FIGURA 31. PANTALLA PRINCIPAL (SISTEMA NUEVO).....	82
FIGURA 32. DATOS DEL ARCHIVO ASIENTOS.DBF.....	83
FIGURA 33. DATOS DE LA TABLA ASIENTOS.....	84

FIGURA 34.	DATOS DEL ARCHIVO COMPROBA.DBF	84
FIGURA 35.	DATOS DE LA TABLA COMPROBA	84
FIGURA 36.	DATOS DEL ARCHIVO CUENTAS.DBF	84
FIGURA 37.	DATOS DE LA TABLA CUENTAS	84
FIGURA 38.	DATOS DEL ARCHIVO ELIASI.DBF	85
FIGURA 39.	DATOS DE LA TABLA ELIASI	85
FIGURA 40.	CONSULTA DE COMPROBANTES (VENTANA #1) SISTEMA ANTIGUO	87
FIGURA 41.	CONSULTA DE COMPROBANTES (VENTANA #2) SISTEMA ANTIGUO	87
FIGURA 42.	CONSULTA DE COMPROBANTES. SISTEMA NUEVO	88
FIGURA 43.	REGISTRO DE USUARIOS. SISTEMA ANTIGUO	89
FIGURA 44.	REGISTRO DE USUARIOS. SISTEMA NUEVO	89
FIGURA 45.	AHORROS/APORTES/NAVIDAD. SISTEMA ANTIGUO	90
FIGURA 46.	AHORROS/APORTES/NAVIDAD. SISTEMA NUEVO	91
TABLA 5.	PRUEBAS DE RECUPERACIÓN (PRUEBA NO. 1)	92
TABLA 6.	PRUEBAS DE RECUPERACIÓN (PRUEBA NO. 2)	92
TABLA 7.	PRUEBAS DE RECUPERACIÓN (PRUEBA NO. 3)	93
CAPÍTULO 4.	CONCLUSIONES Y RECOMENDACIONES	96
BIBLIOGRAFÍA	98
ANEXOS	102

INTRODUCCIÓN

Con el rápido devenir tecnológico los sistemas informáticos en modo texto tanto en hardware como en software van quedando huérfanos de soporte pues ambos componentes de poco en poco han sido casi totalmente abandonados.

Estos sistemas han demostrado ser muy eficientes en soportes de hardware sencillo y sistema operativo modo texto tipo DOS por lo que varias instituciones han invertido considerables recursos en la confección de aplicaciones basadas en soportes Xbase.

El presente trabajo permite brindar una solución para continuar con el procesamiento de los datos en aplicaciones basadas en plataformas Xbase. Por lo que se desarrollo una plataforma intermedia entre Foxbase y un sistema de base de datos relacional tal que permita la migración de los datos de una manera ágil y confiable. Con estos datos migrados al nuevo modelo de base de datos relacional, nos brinda la posibilidad de construir nuevas aplicaciones con tecnologías actuales para así probar que los datos y el procesamiento de los mismos siguen siendo correctos y consistentes; lo más importante en este punto es mostrar que el nuevo sistema a desarrollar, arroja los mismo resultados que el sistema antiguo, el cual se basa en tecnología Visual Fox 2.5.

Para el desarrollo de la Herramienta de Migración de Datos se tuvo como idea principal el implementar una solución informática que ayude a muchas organizaciones a llevar a cabo con éxito proyectos de migración de datos, sin salirse del presupuesto y con una gran precisión, proporcionando así una plataforma de integración de datos empresariales segura y escalable, dotada de todas las funciones necesarias para realizar la migración de datos desde foxbase a plataformas de bases de datos relacionales utilizando los motores de bases de

datos más comerciales del mercado, siendo estos: Oracle, MySQL, SQL Server y Postgres. También se tomó la posibilidad de generar un script en SQL para el caso de que la herramienta no posea un motor al cual se desee dirigir el proceso de migración como puede ser el caso de DB2, Informix, etc.

La herramienta se encuentra desarrollada en C# de la tecnología Visual .NET. Por lo que no se uso software libre sino software propietario de Microsoft ©. Restringiendo a la herramienta al solo ser utilizada en plataformas de sistema operativo Windows tanto XP como Vista de 32 bits.

Para la construcción de la nueva aplicación, la cual probará los datos y el procesamiento de los mismos; se tomo como caso de prueba al sistema contable que posee la Cooperativa Politécnica de Ahorro y Crédito Ltda., el cual se basa en tecnología casi obsoleta y con muy poco soporte. Por este motivo surge la necesidad de actualizar el manejo de esta información utilizando herramientas modernas que permitan a los empleados de la Cooperativa acceder a la información de sus obligaciones con la Institución desde cualquier lugar en el que se encuentren, a través del web.

Los costos elevados de las licencias de software propietario, han ido ocasionando que las empresas no puedan contar con herramientas modernas que permitan mejorar sus servicios. Por lo que para el proceso de creación e implementación del sistema de caso de prueba se utilizará software libre para aplicaciones Web, debido a que el precio de las aplicaciones es mucho menor y la mayoría de las veces son gratuitas. Por otro lado el usuario que adquiere software propietario depende al 100% de la empresa distribuidora.

El software libre utilizado para el nuevo sistema de caso de prueba es: Motor de Bases de Datos: *PostgreSQL*; Servidor Web: *Apache*; Lenguajes de Programación: *PHP*, *JavaScript*; Navegador Web: *Mozilla Firefox*. Dando com

o resultado que el nuevo sistema no restringe su uso a una plataforma de Sistema Operativo, sino que puede ser usado tanto en Windows XP o Vista de 32 bits, como también en cualquiera de las distribuciones de Linux.

Al comprobar que el sistema de caso de prueba arroja los mismo resultados que el sistema antiguo, se pudo concluir que el proceso de migración de FoxBase a un modelo de Base de Datos Relacional ha concluido con éxito. Con esto se logró brindar una nueva solución para que las empresas continuen con el procesamiento de sus datos junto a tecnologías actuales.

CAPÍTULO 1. SELECCIÓN DE MODELO DE DESARROLLO

1.1 SELECCIONAR EL MODELO DE DESARROLLO

Ningún modelo de desarrollo de software es bueno o malo por sí solo. En la aplicación y evaluación para cada situación específica se debe tener en cuenta los parámetros específicos y un buen criterio de evaluación, logrando así determinar cuál es el más apropiado para cada ocasión. ^[6]

Para seleccionar un modelo se debe tener en cuenta que este cumpla con las cualidades más importantes para el desarrollo de software y se acople a las necesidades de nuestro sistema, teniendo en cuenta que es una herramienta para migrar datos de foxbase a modelos de bases de datos relacionales.

1.1.1 METODOLOGÍA PARA EL DESARROLLO DE SOFTWARE

Los requerimientos de un sistema constituyen la definición del comportamiento del software, inclusive si se trata de un utilitario para migrar datos en Foxbase a Plataformas de Bases de Datos Relacionales. Una aplicación de calidad no necesariamente está asociada a la sofisticación interna del software. Si los requerimientos definidos para este utilitario son simples, el software necesariamente va a ser simple y calidad puede ser perfecta.

Después de una perfecta definición de los requerimientos, es necesario que la producción del software esté organizada para asegurar la calidad requerida. Hay que contar con una metodología, técnicas y herramientas para producir software de manera organizada. Aplicar una metodología de ingeniería de software significa que en lugar de llegar inmediatamente a escribir el código, previamente se produce una serie de pasos intermedios, cada uno facilitando el desarrollo del próximo, desde el

^[6] DACCACH, José Camilo. Seleccionando el Modelo de Desarrollo.

análisis del sistema, la definición de requerimientos hasta el diseño, programación, implementación y pruebas.

Existen varias metodologías para el desarrollo de software, de las cuales se ha planteado utilizar la del Ciclo de Vida del Desarrollo de Software conocido como *Cascada* (Waterfall Software Development Life Cycle), puesto que este modelo está basado en la lógica de desarrollo de software, que en definitiva propone:

- Ingeniería y Análisis del Sistema
- Análisis de los Requerimientos
- Diseño
- Codificación
- Pruebas
- Mantenimiento

1.1.2 MODELO DE CASCADA

El método de Cascada involucra una serie de pasos como se puede observar en la *Figura 1* que continuamente interactúan entre sí, comenzando con el análisis del sistema, requerimientos, diseño, codificación, pruebas y mantenimiento.

Este modelo provee estabilidad en los requerimientos, se determinan también una gran cantidad de retroalimentaciones entre etapas. El éxito de la finalización del sistema se fundamenta en el análisis profundo de los requerimientos del usuario y, más particularmente, de las especificaciones del software.

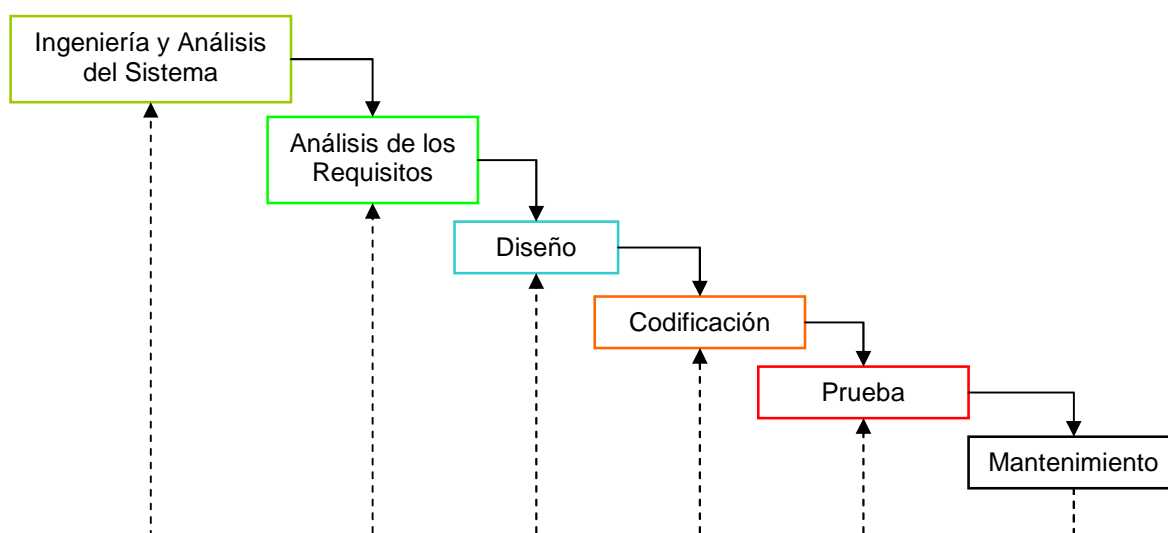


Figura 1. Proceso de la Metodología en Cascada^[15]

1.1.2.1 Ingeniería y Análisis del Sistema

Debido a que el software es siempre parte de un sistema mayor el trabajo comienza estableciendo los requisitos de todos los elementos del sistema y luego asignando algún subconjunto de estos requisitos al software.

1.1.2.1.1 Especificaciones de Requerimientos del Sistema

Constituyen los objetivos a cumplirse desde el punto de vista de los administradores del proyecto. Las necesidades del usuario identifican los requerimientos para el sistema, que deben estar de acuerdo a los objetivos funcionales (actividades, distribución, concurrencia, límites, seguridad, etc.), de rendimiento (velocidad, memoria, capacidad en disco), y estratégicos (portabilidad, sistemas abiertos, extensibilidad, versatilidad).

1.1.2.1.2 Especificaciones del Desarrollo del Sistema

Estas especificaciones se detallan a continuación:

- Análisis de requisitos funcionales y de desempeño.

^[15] ROJAS, Richard; BOUCHECHTER, Israel. Ciclos de Vida de Ingeniería del Software

- Selección de la tecnología, software de apoyo a la ejecución del sistema
- Portabilidad entre varias plataformas de desarrollo
- Conectividad a distintos motores de Bases de Datos
- Requerimientos de Hardware: Procesadores, hardware de comunicación (redes), etc.

1.1.2.2 Análisis de los Requerimientos

El objetivo principal consiste en obtener un sistema de calidad partiendo de las especificaciones del software. Se reconocen las transformaciones de los datos (funciones), transformaciones de controles (eventos externos), flujo de datos (discreto o continuo), flujo de control (discreto o continuo) y almacenamiento de datos (estructura de datos en RAM, archivos y Bases de Datos). Se realiza la especificación del módulo (diseño y selección de algoritmos) y pseudocódigo, la especificación de datos (estructura de datos y archivos junto con su comportamiento en pseudocódigo), para finalmente realizar una verificación y validación.

1.1.2.3 Diseño

El diseño del software se enfoca en cuatro atributos distintos del programa: la estructura de los datos, la arquitectura del software, el detalle procedimental y la caracterización de la interfaz. El proceso de diseño traduce los requisitos en una representación del software con la calidad requerida antes de que comience la codificación.

El sistema implica el siguiente diseño:

Interfaz de Usuario: Consiste en determinar la organización, disposición y uso de gráficos en la pantalla. Basándose en las especificaciones del software se determinan interfaces, las cuales están orientadas a las actividades del usuario que involucra la migración de datos de foxbase a plataformas de bases de datos relacionales.

1.1.2.4 Codificación

El diseño debe traducirse en una forma legible para la maquina. El paso de codificación realiza esta tarea. Si el diseño se realiza de una manera detallada la codificación puede realizarse mecánicamente.

En base al análisis y diseño del proyecto se escoge una herramienta de programación, lo cual debe estar acorde a las especificaciones obtenidas anteriormente. Si en el desarrollo se necesita programación multitarea o programación orientada a objetos, el lenguaje adecuado será aquel que soporte estas características. Los resultados de diseño, permiten generar definiciones de subrutinas y trasladar el pseudocódigo al código final. La codificación involucra el desarrollo de la interfaz de usuario. Se requiere la implementación de las transacciones asociadas en la migración de datos a los motores de bases de datos más conocidos y utilizados en el mercado (SQL Server, Oracle, PostgreSQL, MySQL).

1.1.2.5 Pruebas

Una vez que se ha generado el código comienza la prueba del programa. La prueba se centra en la lógica interna del software, y en las funciones externas, realizando pruebas que aseguren que la entrada definida produce los resultados que realmente se requieren.

De manera general, se requiere describir cada uno de los casos de prueba que se van a efectuar sobre el software y los resultados que se espera recibir. Para cada uno de los casos de prueba se realiza la revisión de requerimientos y resultados obtenidos.

1.1.2.6 Mantenimiento

El software sufrirá cambios después de que se realice la entrega final de la herramienta. Los cambios ocurrirán debido a que hayan encontrado errores, a que

el software deba adaptarse a cambios del entorno externo (Ejemplo: sistema operativo), o debido a que el cliente requiera ampliaciones funcionales o del rendimiento.

Tanto la etapa de diseño como la de implementación deben estar orientadas al mantenimiento del software. Las tareas efectuadas en esta etapa incluyen la transformación de código no estructurado a estructurado, estandarización de nombres, eliminación de datos y código no utilizado, etc.

1.1.3 JUSTIFICACIÓN DEL MODELO DE DESARROLLO

Se ha decidido utilizar la metodología de cascada para el desarrollo de una herramienta para migración de datos en foxbase a plataformas de bases de datos relacionales puesto que abarca el desarrollo de software de manera más general, tiene una secuencia ordenada y permite que en sus definiciones se involucren los aspectos particulares orientados a migración de datos.

Se tomaron en cuenta dos aspectos principales para utilizar el proceso de metodología en cascada para el desarrollo de la herramienta, los cuales son:

- El área de investigación para nuestro proyecto de titulación está orientada a bases de datos y no a desarrollo de software, por lo que se hará más énfasis en la parte de migración que corresponde al área de bases de datos. Con esto no se quiere decir que el modelo cascada no sea adecuado para el desarrollo de nuestra herramienta, en contrario, este método es excelente para trabajar con proyectos pequeños debido a que posee una metodología muy estructurada.
- La herramienta que se va a desarrollar es un utilitario y no un aplicativo por lo que no posee los requisitos necesarios para implementar un modelo de desarrollo que abarque una metodología más completa (Casos de uso,

Diagramas de Clase, etc.). Viendo así que nuestra mejor opción es optar por el proceso de la metodología en cascada.

1.2 SELECCIONAR LENGUAJE DE PROGRAMACIÓN

Se plantea usar como plataforma de desarrollo para la herramienta de migración de datos en Foxbase a plataformas de bases de datos relacionales el lenguaje C#.NET.

1.2.1 FACTIBILIDAD DE DESARROLLO CON .NET

.NET es una plataforma de desarrollo de software con énfasis en transparencia de redes, con independencia de plataforma de hardware y que permita un rápido desarrollo de aplicaciones.

Su propuesta es ofrecer una manera rápida y económica, a la vez que segura y robusta, de desarrollar aplicaciones permitiendo una integración más rápida y ágil entre empresas y un acceso más simple y universal a todo tipo de información desde cualquier tipo de dispositivo.^[18]

Los principales componentes del marco de trabajo son:

- El conjunto de lenguajes de programación
- La Biblioteca de Clases Base o BCL
- El Entorno Común de Ejecución para Lenguajes o CLR por sus siglas en inglés.

^[18] WIKIPEDIA, La Enciclopedia Libre; .NET

1.2.1.1 Características Principales ^[18]

Es el encargado de proveer lo que se llama código administrado, es decir, un entorno que provee servicios automáticos al código que se ejecuta. Los servicios son variados:

- *Cargador de clases*: Permite cargar en memoria las clases
- *Compilador MSIL a nativo*: Transforma código intermedio de alto nivel independiente del hardware que lo ejecuta a código de máquina propio del dispositivo que lo ejecuta.
- *Administrador de código*: Coordina toda la operación de los distintos subsistemas del Common Language Runtime.
- *Recolector de basura*: Elimina de memoria objetos no utilizados.
- *Motor de seguridad*: Administra la seguridad del código que se ejecuta.
- *Motor de depuración*: Permite hacer un seguimiento de la ejecución del código aún cuando se utilicen lenguajes distintos.
- *Verificador de tipos*: Controla que las variables de la aplicación usen el área de memoria que tienen asignado.
- *Administrador de excepciones*: Maneja los errores que se producen durante la ejecución del código.
- *Soporte de multiproceso (threads)*: Permite ejecutar código en forma paralela.
- *Empaquetador de COM*: Coordina la comunicación con los componentes COM para que puedan ser usados por el .NET Framework.
- *Soporte de la Biblioteca de Clases Base*: Interfaz con las clases base del .NET Framework. Esto quiere decir que existen tipos de estructuras como es la de java y la .NET

^[18] WIKIPEDIA, La Enciclopedia Libre; .NET

1.2.2 C#.NET: CARACTERÍSTICAS PRINCIPALES ^[4]

A continuación se mencionan algunas características de C#.NET:

1.2.2.1 Sencillez

C# elimina muchos elementos que otros lenguajes incluyen y que son innecesarios en .NET. Por ejemplo:

- El código escrito en C# es auto contenido, lo que significa que no necesita de ficheros adicionales al propio fuente tales como ficheros de cabecera o ficheros IDL.
- El tamaño de los tipos de datos básicos es fijo e independiente del compilador, sistema operativo o máquina para quienes se compile (no como en C++), lo que facilita la portabilidad del código.
- No se incluyen elementos poco útiles de lenguajes como C++ tales como macros, herencia múltiple o la necesidad de un operador diferente del punto (.) acceder a miembros de espacios de nombres (::)

1.2.2.2 Modernidad

C# incorpora en el propio lenguaje elementos que a lo largo de los años ha ido demostrándose son muy útiles para el desarrollo de aplicaciones y que en otros lenguajes como Java o C++ hay que simular, como un tipo básico decimal que permita realizar operaciones de alta precisión con reales de 128 bits, la inclusión de una instrucción foreach que permita recorrer colecciones con facilidad y es ampliable a tipos definidos por el usuario, la inclusión de un tipo básico string para representar cadenas o la distinción de un tipo bool específico para representar valores lógicos.

^[4] BARRETEROS.NET; Características Introdutorias a C#.NET

1.2.2.3 Orientación a objetos

Como todo lenguaje de programación de propósito general actual, C# es un lenguaje orientado a objetos, aunque eso es más bien una característica del CTS que de C#. Una diferencia de este enfoque orientado a objetos respecto al de otros lenguajes como C++ es que el de C# es más puro en tanto que no admiten ni funciones ni variables globales sino que todo el código y datos han de definirse dentro de definiciones de tipos de datos, lo que reduce problemas por conflictos de nombres y facilita la legibilidad del código. C# soporta todas las características propias del paradigma de programación orientada a objetos: encapsulación, herencia y polimorfismo.

1.2.2.4 Orientación a componentes

La propia sintaxis de C# incluye elementos propios del diseño de componentes que otros lenguajes tienen que simular mediante construcciones más o menos complejas. Es decir, la sintaxis de C# permite definir cómodamente propiedades (similares a campos de acceso controlado), eventos (asociación controlada de funciones de respuesta a notificaciones) o atributos (información sobre un tipo o sus miembros).

1.2.2.5 Seguridad de tipos

C# incluye mecanismos que permiten asegurar que los accesos a tipos de datos siempre se realicen correctamente, lo que permite evitar que se produzcan errores difíciles de detectar por acceso a memoria no perteneciente a ningún objeto y es especialmente necesario en un entorno gestionado por un recolector de basura.

1.2.2.6 Instrucciones seguras

Para evitar errores muy comunes, en C# se han impuesto una serie de restricciones en el uso de las instrucciones de control más comunes. Por ejemplo, la guarda de toda condición ha de ser una expresión condicional y no aritmética, con lo que se evitan errores por confusión del operador de igualdad (==) con el de

asignación (=); y todo caso de un switch ha de terminar en un break o goto que indique cuál es la siguiente acción a realizar, lo que evita la ejecución accidental de casos y facilita su reordenación.

1.2.2.7 Sistema de tipos unificado

A diferencia de C++, en C# todos los tipos de datos que se definan siempre derivarán, aunque sea de manera implícita, de una clase base común llamada System.Object, por lo que dispondrán de todos los miembros definidos en ésta clase (es decir, serán “objetos”) A diferencia de Java, en C# esto también es aplicable a los tipos de datos básicos Además, para conseguir que ello no tenga una repercusión negativa en su nivel de rendimiento, se ha incluido un mecanismo transparente de boxing y unboxing con el que se consigue que sólo sean tratados como objetos cuando la situación lo requiera, y mientras tanto puede aplicárseles optimizaciones específicas.

El hecho de que todos los tipos del lenguaje deriven de una clase común facilita enormemente el diseño de colecciones genéricas que puedan almacenar objetos de cualquier tipo.

1.2.2.8 Extensibilidad de tipos básicos

C# permite definir, a través de estructuras, tipos de datos para los que se apliquen las mismas optimizaciones que para los tipos de datos básicos. Es decir, que se puedan almacenar directamente en pila (luego su creación, destrucción y acceso serán más rápidos) y se asignen por valor y no por referencia. Para conseguir que lo último no tenga efectos negativos al pasar estructuras como parámetros de métodos, se da la posibilidad de pasar referencias a pila a través del modificador de parámetro ref.

1.2.2.9 Extensibilidad de operadores

Para facilitar la legibilidad del código y conseguir que los nuevos tipos de datos básicos que se definan a través de las estructuras estén al mismo nivel que

los básicos predefinidos en el lenguaje, al igual que C++ y a diferencia de Java, C# permite redefinir el significado de la mayoría de los operadores -incluidos los de conversión, tanto para conversiones implícitas como explícitas- cuando se apliquen a diferentes tipos de objetos. Las redefiniciones de operadores se hacen de manera inteligente, de modo que a partir de una única definición de los operadores ++ y—el compilador puede deducir automáticamente como ejecutarlos de manera prefija y postfija; y definiendo operadores simples (como +), el compilador deduce cómo aplicar su versión de asignación compuesta (+=) Además, para asegurar la consistencia, el compilador vigila que los operadores con opuesto siempre se redefinan por parejas (por ejemplo, si se redefine ==, también hay que redefinir !=) También se da la posibilidad, a través del concepto de indizador, de redefinir el significado del operador [] para los tipos de dato definidos por el usuario, con lo que se consigue que se pueda acceder al mismo como si fuese una tabla. Esto es muy útil para trabajar con tipos que actúen como colecciones de objetos.

1.2.2.10 Extensibilidad de modificadores

C# ofrece, a través del concepto de atributos, la posibilidad de añadir a los metadatos del módulo resultante de la compilación de cualquier fuente información adicional a la generada por el compilador que luego podrá ser consultada en tiempo ejecución a través de la librería de reflexión de .NET. Esto, que más bien es una característica propia de la plataforma .NET y no de C#, puede usarse como un mecanismo para definir nuevos modificadores.

1.2.2.11 Versionable

C# incluye una política de versionado que permite crear nuevas versiones de tipos sin temor a que la introducción de nuevos miembros provoquen errores difíciles de detectar en tipos hijos previamente desarrollados y ya extendidos con miembros de igual nombre a los recién introducidos.

Si una clase introduce un nuevo método cuyas redefiniciones deban seguir la regla de llamar a la versión de su padre en algún punto de su código, difícilmente seguirían esta regla miembros de su misma signatura definidos en clases hijas previamente a la definición del mismo en la clase padre; o si introduce un nuevo campo con el mismo nombre que algún método de una clase hija, la clase hija dejará de funcionar. Para evitar que esto ocurra, en C# se toman dos medidas:

1. Se obliga a que toda redefinición deba incluir el modificador `override`, con lo que la versión de la clase hija nunca sería considerada como una redefinición de la versión de miembro en la clase padre ya que no incluiría `override`. Para evitar que por accidente un programador incluya este modificador, sólo se permite incluirlo en miembros que tengan la misma signatura que miembros marcados como re definibles mediante el modificador `virtual`. Así además se evita el error tan frecuente en Java de creerse haber redefinido un miembro, pues si el miembro con `override` no existe en la clase padre se producirá un error de compilación.
2. Si no se considera redefinición, entonces se considera que lo que se desea es ocultar el método de la clase padre, de modo que para la clase hija sea como si nunca hubiese existido. El compilador avisará de esta decisión a través de un mensaje de aviso que puede suprimirse incluyendo el modificador `new` en la definición del miembro en la clase hija para así indicarle explícitamente la intención de ocultación.

1.2.2.12 Eficiente:

En principio, en C# todo el código incluye numerosas restricciones para asegurar su seguridad y no permite el uso de punteros. Sin embargo, y a diferencia de Java, en C# es posible saltarse dichas restricciones manipulando objetos a través de punteros. Para ello basta marcar regiones de código como inseguras (modificador `unsafe`) y podrán usarse en ellas punteros de forma similar a cómo se

hace en C++, lo que puede resultar vital para situaciones donde se necesite una eficiencia y velocidad procesamiento muy grandes.

1.2.2.13 Compatible

Para facilitar la migración de programadores, C# no sólo mantiene una sintaxis muy similar a C, C++ o Java que permite incluir directamente en código escrito en C# fragmentos de código escrito en estos lenguajes, sino que el CLR también ofrece, a través de los llamados Platform Invocation Services (PInvoke), la posibilidad de acceder a código nativo escrito como funciones sueltas no orientadas a objetos tales como las DLLs de la API Win32. Nótese que la capacidad de usar punteros en código inseguro permite que se pueda acceder con facilidad a este tipo de funciones, ya que éstas muchas veces esperan recibir o devuelven punteros.

CAPÍTULO 2. DESARROLLO DE LA APLICACIÓN

2.1 DISEÑO

A continuación se describirá detalladamente cada una de las fases del proceso de la metodología en cascada para la implementación de una herramienta para migración de datos en foxbase a plataformas de bases de datos relacionales.

2.1.1 INGENIERÍA Y ANÁLISIS DEL SISTEMA

2.1.1.1 Especificaciones de Requerimientos del Sistema^[13]

La creciente necesidad de dotar al negocio con nuevas funciones (a menudo, implementando aplicaciones nuevas o actualizando las existentes) es ahora más constante y acentuada que nunca. Para responder a esta progresiva demanda, muchas organizaciones deben migrar los datos almacenados en las aplicaciones existentes a aplicaciones nuevas. Sin embargo, la migración puede ser compleja, lenta y cara, generando costes (a menudo inesperados) que multiplican por 10 el presupuesto original o que requieren unos esfuerzos que contrastan bruscamente con el alcance de los recursos disponibles. Muchos directores de TI aplazan este cambio hasta que la presión empresarial es tan fuerte que no tienen más opción que llevarlo a cabo.

2.1.1.1.1 Retos

La migración de datos suele resultar cara y arriesgada por los siguientes motivos:

- Para acceder y entender sistemas más antiguos y heredados se necesita un conocimiento especializado;
- Los sistemas carecen de documentación actualizada;

^[13] POWER DATA; Migración de Datos

- La migración de muchos sistemas pasa por la resolución de redundancias e incoherencias;
- Los procesos de diseño iterativos para “intentar aprender” provocan dificultades de control y demoras relacionadas con la codificación manual.

2.1.1.1.2 La solución de Informática

Lo que se requiere es implementar es una solución informática que ayude a muchas organizaciones a llevar a cabo con éxito proyectos de migración de datos, sin salirse del presupuesto y con una gran precisión, proporcionando una plataforma de integración de datos empresariales segura y escalable, dotada de todas las funciones necesarias para realizar la migración de datos desde foxbase a plataformas de bases de datos relacionales utilizando los motores de bases de datos más comerciales del mercado, siendo estos: Oracle, MySQL, SQL Server y Postgres.

Las empresas que utilizan la herramienta para llevar a cabo sus proyectos de migración de datos, se benefician de:

- Una **reducción del tiempo** de entrega de los proyectos de migración de datos;
- Una **disminución del riesgo** de reducir los esfuerzos para el proyecto, rebasar los costes o retrasar el proyecto;
- Unos **datos más coherentes** en todos los sistemas, procesos y organizaciones;
- Una **mayor capacidad de adaptación** al negocios

2.1.1.2 Especificaciones del Desarrollo del Sistema

Todo el proceso de migración tiene que tener en cuenta muy especialmente, las maneras que tenga cada base de datos de guardar la información, es decir, del

formato de sus tipos de datos. Tenemos que contar siempre con la posible necesidad de transformar algunos datos como pueden ser los campos booleanos, fechas, campos memo (texto con longitud indeterminada), etc., que pueden almacenarse de maneras distintas en cada uno de los sistemas gestores, origen y destino.

El proceso de la migración puede ser bastante complejo y, como hay cuatro motores de bases de datos distintos, difícil de dar una receta que funcione en todos los casos. Posiblemente tengamos que realizar algún script que realice los cambios necesarios en los datos. Por ejemplo puede ser para localizar los valores booleanos guardados como true/false a valores enteros 0/1, que es como se guarda en MySQL, Oracle, Postgres y SQL Server. También las fechas pueden sufrir cambios de formato, de dd/mm/aaaa al formato aaaa-mm-dd. C#.NET puede ayudarnos en la tarea de hacer este script a base de sentencias dirigidas contra la base de datos.

2.1.2 ANÁLISIS DE LOS REQUERIMIENTOS

2.1.2.1 Lo que debe hacer la herramienta

La herramienta de migración de datos de foxbase a plataformas de bases de datos relacionales, debe cumplir con los siguientes requerimientos funcionales:

- Debido a que el sistema es una herramienta para migrar datos (utilitario) y no posee las características para ser un aplicativo, necesariamente tiene que poseer una navegación sencilla e intuitiva para el usuario final. Dando como resultado un producto de fácil uso pero de alta consistencia.
- Muchas de las organizaciones que necesitan migrar los datos de foxbase poseen distintos motores de bases de datos a los cuales desean dirigir el proceso de migración requiriendo que esta trabaje con los motores de bases de datos más comerciales del mercado, siendo estos: Oracle, MySQL, SQL Server y Postgres. Por lo que se debe manejar los parámetros de

conectividad que son Nombre del Host (Nombre de la Máquina), Nombre del Servidor, Puerto (Puerto en el cual trabaja el motor de base de datos), Usuario (Usuario de la Base de Datos) y Contraseña (Contraseña del usuario de la base de datos anteriormente especificado).

- Luego de haber realizado la conectividad al motor de bases de datos previamente seleccionado, se debe poder escoger con cuál de las base de datos existentes se va a trabajar para el proceso de migración.
- La migración de datos de foxbase al motor de bases de datos seleccionado, se debe realizar de forma automática, entendiéndose por este que se debe crear la tabla en la base de datos y migrar los datos en la misma de forma transparente para el usuario.
- Luego del proceso de migración, se procederá a realizar las pruebas necesarias con la base de datos del sistema de contabilidad de la Cooperativa Politécnica de la Escuela Politécnica Nacional. Teniendo como requisito primordial el migrar el sistema de la cooperativa antes mencionada debido a que para continuar con el procesamiento de sus datos se necesita construir nuevas aplicaciones con tecnologías actuales y migrar los datos a estas nuevas tecnologías. Logrando obtener los mismos resultados que daba el sistema luego del proceso total de migración.
- Si el caso se diera de que el utilitario no proporciona la ayuda necesaria para poder migrar los datos a un motor de bases de datos que no esté implementado en la herramienta, debe poseer la posibilidad de generar un script de extensión .SQL el cual pueda ser libre de cambios o de fácil uso para ser implementado en otra base de datos.

2.1.2.2 Lo que no debe hacer la herramienta

La herramienta de migración de datos de foxbase a plataformas de bases de datos relacionales, no debe incluir los siguientes aspectos:

- El proceso de migración de datos va a realizarse de forma local, por lo que la herramienta no va a necesitar necesariamente funcionar de forma remota, en otras palabras, no migrará los datos de foxbase a un motor de base de datos el cual se encuentre en un servidor remoto conectado mediante una red al ordenador que posee la herramienta de migración.
- La herramienta de migración de datos solo trabajará con los motores de bases de datos mencionados anteriormente. Por lo que no se deberá realizar el proceso de migración con una base de datos no especificada en los requerimientos.

2.1.3 DISEÑO

2.1.3.1 Interfaz de Usuario

La presentación de la herramienta que es la que mantiene la comunicación con el usuario final, los datos de ingreso y las respuestas que el sistema brinda, son los componentes de la interfaz de usuario del sistema de migración de datos de foxbase a plataformas de bases de datos relacionales.

2.1.3.2 Diseño General de la Interfaz de Usuario

El modelo a seguir para realizar la herramienta de migración de datos esta realizada en base a los requerimientos planteados anteriormente, lo cual se puede observar en la *Figura 2* que se muestra a continuación.

Migración de Datos en Foxbase a Plataformas de BDD Relacionales	
<u>Opciones</u>	
<input type="radio"/> Crear Tabla <input type="radio"/> Generar Script	
<u>Bases de Datos</u>	
<input type="radio"/> SQL Server <input type="radio"/> Oracle <input type="radio"/> Postgres <input type="radio"/> MySQL	
<u>Conectividad</u>	
Host	<input type="text"/>
Puerto	<input type="text"/>
Usuario	<input type="text"/>
Nombre del Servidor	<input type="text"/>
Contraseña	<input type="text"/>
<input type="button" value="Conectar"/> <input type="button" value="Desconectar"/>	
Selección de la Base de Datos:	<input type="text" value="▼"/>
Selección de los Archivos a Migrar:	<input type="text" value="▼"/> <input type="button" value="Examinar"/>
<input type="button" value="Migrar"/> <input type="button" value="Cancelar"/>	
Descripción:	Pantalla que permite realizar todo el proceso de migración en Foxbase a una plataforma de bases de datos relacional.

Figura 2. Interfaz de Migración

2.1.3.3 Estructura de Foxbase

2.1.3.3.1 Definición del foxbase

El Foxbase es un lenguaje de programación gestor de Base de Datos. Se considera de alto nivel porque sus instrucciones son escritas con palabras similares a los lenguajes humanos, lo que facilita la escritura y fácil comprensión del programador.^[3]

^[3] ARIAS, Angélica; Guía de FOXBASE

El FoxPro usa tablas para almacenar datos que definen distintos tipos de archivos. Los tipos de archivos que se guardan como archivos de tabla son:

- Tabla (.dbf)
- Base de datos (.dbc)
- Formulario (.scx)
- Etiqueta (.lbx)
- Menú (.mnx)
- Proyecto (.pjx)
- Informe (.frx) ^[11]

Viendo así que el objetivo de nuestra migración es el enfocarnos en los archivos de extensión .DBF.

2.1.3.3.2 Archivos en Foxbase

Son conjuntos de registros que contienen información relativa a un conjunto de individuos u objetos y se almacenan como una unidad en los dispositivos de almacenamiento secundario de la computadora (disquetes ó disco duro).^[3]

Un archivo de tabla está formado por un **registro de encabezado** (cabecera) y por **registros de datos**.

2.1.3.3.2.1 La cabecera del .DBF. ^[9]

El registro de encabezado define la estructura de la tabla y contiene cualquier otra información relacionada con la tabla. Comienza en la posición de archivo cero.

^[11] MSDN; Estructura de archivos de tabla (.dbc, .dbf, .frx, .lbx, .mnx, .pjx, .scx, .vcx)

^[3] ARIAS, Angélica; Guía de FOXBASE

^[9] MARTÍNEZ, Luis; La Cabecera de los .DBF

De la misma forma que los ficheros DBF se componen de cabecera y registros de datos, la cabecera también se compone de dos partes.

1. Primera parte de la cabecera:

En la *Tabla 1* se puede observar cómo están distribuidos los 32 bytes de longitud. La estructura tiene la siguiente forma:

Número de Byte	Descripción/Contenido
1	Identificados del tipo de fichero
2-4	Fecha última modificación
5-8	Número de registros del fichero
9-10	Dónde empieza el principio de los registros de datos
11-12	Longitud del registro
13-28	No usado
29	Indica si tiene componentes de indexación
30	Indica el número de página de la tabla
31-32	No usado

Tabla 1.a. Estructura de la cabecera de los archivos .DBF

El **identificador del tipo de fichero** puede ser de dos tipos (byte 1):

- Si se tiene un campo memo definido en el primer byte aparecerá F5h.
- Si no contiene ningún campo memo será 03h.
- Si el valor contenido en este primer byte es distinto, FoxPro nos devolverá:
Nota database file

La **última fecha de modificación** se guardará en hexadecimal ocupando los bytes 2,3 y 4 correspondiente al año, mes y día, respectivamente.

El **número de registros** de la base de datos incluye el número de los registros de la tabla. Los bytes utilizados (del 5 al 8) están dispuestos en orden inverso de tal manera que el byte menos significativo es el primero de la izquierda. Por ejemplo: 0B 2ª 01 00. Corresponde al número 00/01/2ª/0B en hexadecimal. Puede ser que si usamos la función RECCOUNT() para contar el número de registros que contiene el fichero, el número que nos dé no coincida con este número exacto y es que contabiliza también aquellos registros marcados para ser borrados. Sólo cuando se haya realizado el correspondiente borrado físico (PACK) el resultado de la función RECCOUNT() será correcto.

La **dirección donde comienza el primer registro** de datos ocupa los bytes 9 y 10. Al igual que el anterior, los bytes están en orden inverso siendo 42 (byte 9) 1B (byte 10) igual a la dirección 1B42 en hexadecimal. A partir de la dirección indicada por dichos bytes comenzará los registros de datos.

La **longitud de un registro** es la suma de las longitudes de cada uno de los campos. Este ocupa 2 bytes (11 y 12).

El byte 29, **Flag de componente de indexación (Índices)** nos indicará la presencia de un componente de índice estructural. Si por alguna causa el fichero de índices es borrado por medio de un comando DOS como por ejemplo DEL o ERASE, FoxPro nos enviaría el siguiente mensaje: STRUCTURAL CDX FILE NOT FOUND. Para que esto no suceda la siguiente vez que queramos abrir el fichero deberemos ejecutar la siguiente orden para borrar el flag que indica que es un fichero con índices (indexado): DELETE TAG ALL

Por último, el byte 30 indica el **número de página** con el que ha sido creado el DBF. El número total de valores de códigos de página que puede soportar son 19.

Los bytes sobrantes, que no son usados por FoxPro, suelen rellenarse con el carácter 0 o nulo.

2. Segunda parte de la cabecera:

La segunda parte de la cabecera depende del número de campos del que se componga el registro. Por cada uno de los campos utiliza 32 bytes organizados de la siguiente manera:

Número Byte	Descripción/Contenido
1-11	Nombre del Campo
12	Tipo del Campo
12-14	Dirección donde comienza el campo desde el principio del registro
15-16	No usado
17-18	Longitud del campo (Campos no numéricos)
17	Longitud del campo (Campos numéricos)
18	Campos decimales (Campos numéricos)
19-32	No usado

Tabla 1.b. Estructura de la cabecera de los archivos .DBF

El **nombre del campo** (byte 1 al 11) en FoxPro está limitado a 10 caracteres, el último carácter (el más a la derecha) debe ser nulo (0). Si el nombre del campo es inferior a 10 caracteres el primer carácter de la derecha será 0 o nulo.

Pueden existir, en principio, **5 tipos de campos** definidos en el byte 12 con un carácter ASCII de la siguiente forma:

Valores de Tipos de Campos			
Tipo	Carácter ASCII	Valor Decimal	Valor Hexadecimal
Character (Carácter)	C	67	43
Date (Fecha)	D	68	44
DateTime	T	84	54
Logical (Lógico)	L	76	4C
Memo	M	77	4D
Numeric (Numérico)	N	78	4E
Float (Coma Flotante)	F	70	46
Currency (Dinero)	Y	89	59
Double	B	66	42
Integer (Entero)	I	73	49
General	G	71	47

Tabla 2. Valores de Tipos de Campos

La **dirección donde comienza el campo** (bytes 13 y 14) nos indica la posición de comienzo del campo relativo al principio del registro.

La **longitud de los campos** depende del tipo de dato que vayamos a definir. Para los campos de caracteres utiliza los bytes 17 y 18 para insertar el tamaño, eso sí, en formato invertido. Para los campos tipo Fecha, Memo, General y Lógico sólo necesita el primer byte para definir sus tamaños: 8 para el campo Fecha, 10 para el campo Memo, 10 para el campo General, 1 para el campo Lógico.

Por último, para los campos numéricos el primer byte especifica el total de número de dígitos, y el segundo el número de posiciones decimales.

Los bytes restantes no son utilizados, y su valor suele ser rellenado a nulo (0).

2.1.3.3.2.2 Registros de Datos

Los registros de datos siguen al encabezado (en bytes consecutivos) y contienen el texto real de los campos.

Los datos comienzan inmediatamente después del byte de terminación de la cabecera. El primer byte de cada registro está en el flag de borrado. Si el flag de borrado contiene 20h, el registro es activo. Si por el contrario su valor es 2Ah, significa que ese registro ha sido marcado para ser borrado.

El siguiente byte es el primer byte del primer campo. No existen separaciones entre los campos, no es necesario. La sección de definición del registro de cada campo ya especifica la longitud y la posición.

Los campos de tipo **carácter** son fáciles de leer. Cada byte, es añadido de izquierda a derecha, añadiendo un carácter. Determinar el contenido de un campo, es sólo cuestión de convertir el valor hexadecimal en su correspondiente valor ASCII.

Los campos de tipo **fecha** también son fáciles de interpretar. Cada campo ocupa 8 bytes. Los cuatro primeros son los relativos al año, los dos siguientes al mes y los dos últimos al día. Los bytes son dígitos ASCII del 30 al 39 hex que se corresponden con el 0 al 9 decimal. Al 1 de Enero de 1995 le corresponderían los siguientes valores:

A A A A M M D D → Formato

31 39 39 35 30 31 30 31 → Valor Hex.

1 9 9 5 0 1 0 1 → Valor Dec.

Los campos de tipo **lógico** usan los caracteres ASCII F y T, que traducidos a valores hexadecimales se corresponden con el 46h para el valor lógico falso (F) y 54h para el valor verdadero (T).

Los campos **memo**, como comentábamos antes, usan 10 bytes en cada registro. Si el campo resulta estar vacío, este se rellena con el carácter 20h (espacios en blanco). Si el campo memo no está vacío, los caracteres ASCII definen su posición (en bloques de 64 bytes) en el fichero .FPT. Por ejemplo, si tenemos los 10 bytes con los siguientes valores: 20 20 20 20 20 20 20 20 31 35 (Valores hex.) traducido a decimal 15. Los datos del campo memo comienzan después del décimo quinto bloque de 64 bytes del fichero FPT.

Los campos **numéricos** en FoxPro son interpretados como números ASCII individuales. Define el número de dígitos por la estructura de la base de datos. Supongamos que definimos un campo numérico de seis caracteres con tres decimales. El número 22,324 estaría de la siguiente manera:

32 32 2C 33 32 34 Hex.

2 2 , 3 2 4 Dec.

La coma decimal corresponde con el valor hexadecimal 2Ch. Si el número no ocupase las seis posiciones se toman los siguientes criterios:

- Si el espacio es por la izquierda se rellenará con un espacio en blanco (20h).
- Si el espacio no utilizado es por la derecha se pondrá un cero (30h).

Después del último registro, FoxPro añade un carácter de final de fichero (1Ah). Este byte se crea cuando se determina la longitud del fichero DBF. Tener en cuenta que es diferente al carácter de final de cabecera (0Dh).

2.1.3.3.3 Definición de Base de Datos

Una Base de Datos es un conjunto exhaustivo no redundante de datos estructurados organizados independientemente de su utilización y su implementación en máquina accesibles en tiempo real y compatibles con usuarios concurrentes con necesidad de información diferente y no predicable en tiempo.^[17]

Hay dos enfoques para el almacenamiento de datos en un sistema basado en computadora. El primero consiste en guardar los datos en archivos individuales, cada uno de ellos único para una aplicación particular. El segundo enfoque involucra la construcción de una Base de Datos.

Para comprender mejor cada uno, se comparan a través de la *Tabla-3*:

SISTEMA DE PROCESAMIENTO DE ARCHIVOS	SISTEMA DE PROCESAMIENTO DE BASE DE DATOS
Datos separados y aislados. Puesto que cada archivo almacena información específica y se procesa de manera independiente. Si se requiere relacionar dos archivos los analistas y programadores deberán decidir cuál es la relación, qué partes de cada archivo son necesarias y coordinar el	Datos integrados. Aquí existe lo que se denomina Sistema Manejador de Base de Datos (DBMS) y los datos se almacenan en un medio sencillo llamado Base de Datos. Un programa de aplicación puede ordenar al DBMS que acceda los datos del cliente, los datos de ventas o ambos. Es decir, el

^[17] TREJO, Janhil; Bases de Datos

procesamiento para que se presenten los datos correctos.	programador de la aplicación sólo específica cómo deberán combinarse los datos y el DBMS realiza las operaciones necesarias para conseguirlo.
Duplicación de los datos. Ocurre cuando la información está almacenada en archivos independientes no relacionados.	Menos duplicación de los datos, debido a que cada grupo de datos se almacena en un solo lugar (clientes, ventas, ...) resultan menos comunes los problemas de integridad de los datos; hay menor oportunidad de discrepancia entre las múltiples copias de los mismos elementos de datos.
Dependencia del programa de aplicación. Esto ocurre porque el formato físico de los archivos y registros son parte del código de aplicación; por lo tanto cuando se requiere hacer un cambio en los formatos de archivo, también deben modificarse los programas de aplicación.	Independencia programa/datos. Los formatos de los registros se almacenan en la misma Base de Datos junto con los datos y son accedidos por el DBMS y no por los programas de aplicación.
Archivos incompatibles. Cada lenguaje tiene un formato específico para sus archivos. Así los archivos de Cobol no pueden combinarse o compararse con facilidad con los archivos en C, por ejemplo.	Fácil representación de la vista de datos del usuario. Permite que los usuarios construyan su vista personal de los datos sin preocuparse por la forma en que están físicamente guardados los datos.

Tabla 3. Sistema de Archivos vs. Sistema de Base de Datos^[3]

^[3] ARIAS, Angélica; Guía de FOXBASE

2.1.3.3.4 Tipos de Datos A Utilizar

Entre los tipos de datos que vamos a utilizar para el desarrollo de nuestra herramienta de migración de datos, debemos tener en cuenta que muchos de estos son similares o idénticos en los cuatro motores de bases de datos con los cuales vamos a trabajar. Aquí tenemos una pequeña descripción de los tipos de datos que utilizaremos:

- *Char.*- Almacena una cadena de caracteres alfanuméricos de longitud fija que puede tener entre 1 y 32.767 bytes, en Oracle se permite un máximo de 2000 bytes. Un aspecto a tener en cuenta de este tipo de dato es que cuando especificamos un tamaño, aunque ingresemos una cadena menor del tamaño especificado, se rellenará al tamaño indicado inicialmente, es decir, como es de longitud fija, siempre ocupará lo mismo, independientemente del valor que contenga. Es por esta razón que preferimos almacenar los datos existentes en columnas tipo char a columnas que sean del tipo Varchar o VarChar2 en el caso de Oracle.
- *VarChar.*- Almacena una cadena de longitud variable que puede tener entre 1 y 65.535 bytes como máximo, en Oracle se permite un máximo de 4000 bytes. Dado que este tipo de dato es variable, el espacio que se almacenara será solo el necesario, a pesar de que este tamaño sea mayor al almacenado. En Oracle utilizaremos *Varchar2* ya que *VarChar* está en desuso en este motor de base de Datos.
- *Integer.*- número entero con o sin signo. Con signo el rango de valores va desde -2147483648 a 2147483647. Sin signo el rango va desde 0 a 429.4967.295.

- *Float*.- número pequeño en coma flotante de precisión simple. Los valores válidos van desde $-3.402823466E+38$ a $-1.175494351E-38$, 0 y desde $1.175494351E-38$ a $3.402823466E+38$.
- *Numeric*.- Se admiten hasta 38 dígitos de precisión. Para declarar un tipo de datos Numeric es suficiente con: nombre_columna Numeric.
Opcionalmente se le puede indicar la precisión (número total de dígitos) y la escala (número de dígitos a la derecha de la coma, decimales, los cogerá de la precisión indicada): nombre_columna Numeric (precisión, escala).
Si no se indica la precisión se tomará en función del número a guardar, si no se indica la escala se tomará escala cero.
- *Double*.- número en coma flotante de precisión doble. Los valores permitidos van desde $-1.7976931348623157E+308$ a $-2.2250738585072014E-308$, 0 y desde $2.2250738585072014E-308$ a $1.7976931348623157E+308$. Puesto que no todas las bases de datos contienen Double, vamos a utilizar el tipo de dato Numeric para los motores de bases de datos que no lo tengan. (SQL Server y Oracle), con una precisión de 2 decimales puesto que hablamos de valores monetarios.
- *Money*.- unidad monetaria que consta de tres secciones una para diferenciar los miles, otra para diferenciar los cientos y la última para diferenciar los dos decimales. Dado que como el caso anterior este tipo de dato no existe en todas las bases de datos, vamos a sustituirlo con el tipo de dato numeric con una precisión de dos decimales para los motores de base de datos que no lo contengan (Oracle y MySQL).
- *Date*.- almacena una fecha. El rango de valores va desde el 1 de enero del 1001 al 31 de diciembre de 9999. El formato de almacenamiento es de años-mes-día. En el caso de SQL Server que no posee este tipo de dato utilizaremos el tipo de dato DateTime, teniendo en cuenta que la hora será guardada como 0:00:00. Es necesario tener en cuenta que Oracle a pesar de

tener un tipo de dato *Date*, este no solo almacena la fecha sino también la hora, por lo que se tomarán las mismas precauciones que para *SQL Server*.

- *DateTime*.- Combinación de fecha y hora. El rango de valores va desde el 1 de enero del 1001 a las 0 horas, 0 minutos y 0 segundos al 31 de diciembre del 9999 a las 23 horas, 59 minutos y 59 segundos. El formato de almacenamiento es de año-mes-día horas:minutos:segundos. En el caso de *Oracle* que no posee *DateTime*, no habría ningún problema puesto que el Tipo de dato *Date* en *Oracle* es semejante a *DateTime*.
- *Bit*.- Es un tipo de dato utilizado por *SQL Server* para almacenar valores booleanos asignando 0 al valor *False* y 1 al valor *True*.
- *TinyInt*.- Es un sinónimo del tipo de dato booleano o *bool* utilizado por *MySQL*. Su utilización es igual a la del tipo de dato *Bit*.
- *Bool*.- Este tipo de dato es utilizado por la base de datos *Postgres* para almacenar los valores de verdad ya sea *True* o *False*.
- *SmallInt*.- Almacena dos bytes con signo. Puesto que *Oracle* no tiene un tipo de dato específico para almacenar valores de verdad, vamos a utilizar el tipo de dato *SmallInt* para almacenar 0 al valor *False* y 1 al valor *True*.

En la Tabla 4. Podemos ver un resumen de los tipos de datos que se utilizarán para los distintos motores de base de Datos.

Bases de Datos	FoxBase		SQL Server ^[10]	Oracle ^[1]	MySQL ^[5]	PostgreSQL ^[12]
	Tipos de Datos	C	Character	Varchar(n)	Varchar2(n)	Varchar(n)
Y		Currency	Money	Numeric	Numeric	Money
N		Numeric	Numeric	Numeric	Numeric	Numeric
F		Float	Float	Float	Float	Float
D		Date	DateTime	Date	Date	Date
T		DateTime	DateTime	Date	DateTime	DateTime
B		Double	Numeric	Numeric	Double	Double
I		Integer	Integer	Integer	Integer	Integer
L		Logical	Bit	Smallint	Tinyint	Bool
M		Memo	Varchar(n)	Varchar2(n)	Varchar(n)	Varchar(n)
G		General	Varchar(n)	Varchar2(n)	Varchar(n)	Varchar(n)

Tabla 4. Equivalencias de tipos de datos para los Motores de Bases de Datos

2.1.3.3.5 Modelo de base de datos relacional

Es una forma particular de estructurar y procesar una Base de Datos. Para ello se basa en el Modelo Entidad – Relación; el cual se emplea para interpretar, especificar y documentar los requerimientos para sistemas de procesamiento de Base de Datos y se desarrolla a partir de un análisis de requerimientos de los usuarios. Estos requerimientos se expresan en Tablas ó Entidades que requieren ser *Normalizadas* para garantizar que cada tabla ó entidad tenga información de un sólo tema.

2.1.3.3.6 Cardinalidad de Relaciones ^[20]

Partiendo de que el modelo de Base de Datos Relacional utiliza el Modelo Entidad – Relación, se explican, a continuación, los tipos de relaciones que se pueden establecer entre archivos:

^[10] MICROSOFT CORPORATION; Tipos de datos

^[1] AJPDSOFT; Tipos de Datos en Oracle

^[5] CUENCA, Carlos Luis; Tipos de Datos de Mysql

^[12] OLEA, Ismael; Tipos de datos relevantes en PostgreSQL

^[20] WIKIPEDIA, La Enciclopedia Libre; Modelo Entidad-Relación

RELACIÓN 1:1: Este tipo de relación implica que una entidad se relaciona de manera única con otra entidad. Ejemplo: si tenemos la entidad EMPLEADO y EXTENSIÓN TELEFÓNICA, la relación entre estas dos es 1:1 puesto que a un empleado le corresponde sólo una extensión telefónica.

RELACIÓN 1:N: Implica que una entidad puede relacionarse de manera única con varias ocurrencias de otra entidad. Ejemplo: supongamos dos entidades DEPARTAMENTO y EMPLEADO, establecemos que un departamento puede tener varios empleados; pero un empleado sólo puede pertenecer a un departamento por lo tanto la relación es de 1 a muchos. El orden en este tipo de relación es importante. La diferencia entre las relaciones 1:N y N:1 sucede debido a la diversa naturaleza de la entidad primaria que siempre aparece en la parte superior.

RELACIÓN N:M: En este caso varias ocurrencias de una entidad se relacionan con varias ocurrencias de otra entidad. Ejemplo: Entidades VUELO y PASAJERO; un vuelo puede llevar muchos pasajeros y un pasajero puede tener muchos vuelos en su itinerario.

2.2 PLANIFICACIÓN

En la planificación se tiene que definir el conjunto de tareas a realizar. Este listado debe contar con un responsable, tiempos de entrega y un entregable (producto final). Estas tareas pueden ser tanto automáticas como manuales. Dentro de las tareas automáticas se encuentra el desarrollo de algoritmos de programación, transformación y compilación de datos. Las tareas manuales consisten en el armado de archivos de migración realizados a partir de documentos u otra información existente que no se encuentre automatizada y se quiera incluir en la migración.

Luego de haber ejecutado los procesos de migración es preciso controlar los resultados de salida que estos devolvieron. Este control incluye la verificación de que los datos se hayan procesado en su totalidad y sin errores u omisiones. Muchas veces se controlan los datos con los existentes en el sistema anterior o con los registros escritos (por ejemplo: Datos de personal, cuentas bancarias, entre otros).

Una vez que se realizaron las verificaciones al proceso, el mismo tiene que ser aceptado para empezar a trabajar con el nuevo sistema. Esta aceptación del trabajo debe ser decidida por personas ajenas al proceso de migración que se realizó.

Para finalizar, se recomienda como medida de seguridad, que una vez finalizada la migración se guarde copia de la base de datos en un lugar seguro y sellado para prevenir problemas futuros. Esto permite distinguir entre los datos que se incorporaron al sistema mediante importación y los que fueron generados a partir del uso del mismo. Además, actúa como un respaldo del estado de los datos que fueron importados.

2.2.1 RESPONSABLES DEL PROCESO DE MIGRACIÓN

Para el proceso de migración se han tomado en cuenta los siguientes responsables: ^[7]

Administrador y Lider de Proyecto: Ing. Francisco Villavicencio

Persona que se encargará de administrar y controlar los recursos asignados a este proyecto, con el propósito de que se cumplan correctamente los planes definidos. Los recursos asignados pueden ser recursos humanos, económicos, tecnológicos, etc.

^[7] FULLER, David; Roles en el Desarrollo de Software

Analista: Estefanía Ramírez

En este rol se debe identificar las necesidades del cliente, donde se debe definir los objetivos del sistema, determinando la información que desea obtener, la información que será suministrada al sistema, la funcionalidad del sistema y el rendimiento requerido. Se debe determinar si cada uno de los requisitos especificados es o no esencial. Luego, se debe realizar la especificación de requisitos de software. Esto es, no como una especificación en lenguaje del cliente, sino que como especificación para el equipo de trabajo.

Diseñador: Danny Puente

Es el encargado de generar el diseño del sistema. Sus funciones principales son el generar el diseño arquitectónico y diseño detallado del sistema, basándose en los requisitos, generar prototipos rápidos del sistema (con analistas y programadores) para chequear los requisitos, generar el documento de diseño arquitectónico de software, y mantenerlo actualizado durante el proyecto y velar porque el producto final se ajuste al diseño realizado.

Programadores: Estefanía Ramírez y Danny Puente

Son las personas que se encargan de convertir la especificación del sistema en código fuente ejecutable utilizando uno o más lenguajes de programación, así como herramientas de software de apoyo a la programación.

Téster: Ing. Francisco Villavicencio, Estefanía Ramírez y Danny Puente

El téster es el encargado de asegurar la calidad de cada uno de los productos (documentos, prototipos, etc.). Entre sus tareas principales está el construir y aplicar los planes de prueba unitarios, de módulo, de sistema, y

aceptación parcial, manteniéndolos actualizados durante el proyecto, velar por la completitud, y exactitud (no ambigüedades) de todos los documentos del proyecto, coordinar las inspecciones, y/o caminatas, velar por la adhesión al estándar adoptado para el desarrollo, velar por la calidad del producto final (cumplimiento de los requisitos).

Ingeniero de validación y verificación: Ing. Francisco Villavicencio

El objetivo principal es el de analizar y testear en forma completa el software durante el desarrollo para determinar que ejecuta su funcionalidad correctamente, asegurarse que no ejecuta funciones no intencionalmente definidas y proveer información sobre su calidad y confiabilidad. En lo principal se encarga de comprobar los siguientes tópicos en el sistema:

- *Correctitud:* En qué grado el producto está libre de fallas.
- *Consistencia:* En qué grado el producto es consistente consigo mismo y con otros productos.
- *Necesidad:* En qué grado lo que hay en el producto es necesario.
- *Suficiencia:* En qué grado el producto es completo.
- *Rendimiento:* En qué grado el producto satisface los requisitos de rendimiento.

Documentador: Danny Puente

En este rol se trata de actuar como medio de comunicación entre los miembros del equipo, incluyendo el cliente. Además, durante el proyecto, la documentación sirve también para reducir la distorsión de ideas, ayudar al control del proyecto, almacenar la lógica de las decisiones tomadas y hacer visibles, en forma temprana, tanto las capacidades como las limitaciones del sistema.

2.3 IMPLEMENTACIÓN

En esta parte se explicara cada una de las partes de la implementación de la herramienta para la migración de datos de Foxbase a un modelo de base de datos relacional. Partiendo desde un esquema básico que involucra el diagrama de flujo o en otras palabra su algoritmo, la estructura que va a tener el mismo dentro de la programación, un poco de código fuente para entender su funcionamiento; y para terminar el uso de la herramienta en un caso práctico.

2.3.1 DIAGRAMA DE FLUJO

Como lo vemos en la *Figura 3*, utilizaremos un diagrama de flujo debido a que es la forma más tradicional de especificar los detalles algorítmicos y constituye la representación gráfica de un proceso. Lo que se trata con esta representación gráfica de los pasos de un proceso es entender de mejor manera el problema a tratar.

En el diagrama de flujo podemos ver que el usuario de la herramienta posee dos opciones, la de *Crear Tabla* y la de *Generar Script*. Si el usuario opta por la opción *Crear tabla*, este debe seleccionar el motor de Base de Datos con el que va a trabajar el proceso de migración, luego debe de establecer los parámetros de conectividad al mismo; para empezar el proceso de migración la conexión al motor de base de datos debe haber sido exitosa, quedando por último escoger la base de datos con la que desea trabajar y el archivo *.dbf* que va a migrar. Caso contrario si elige la opción *Generar Script*, el proceso es mucho más sencillo, debido a que solo debe escoger el archivo *.dbf* a migrar y empezar el proceso.

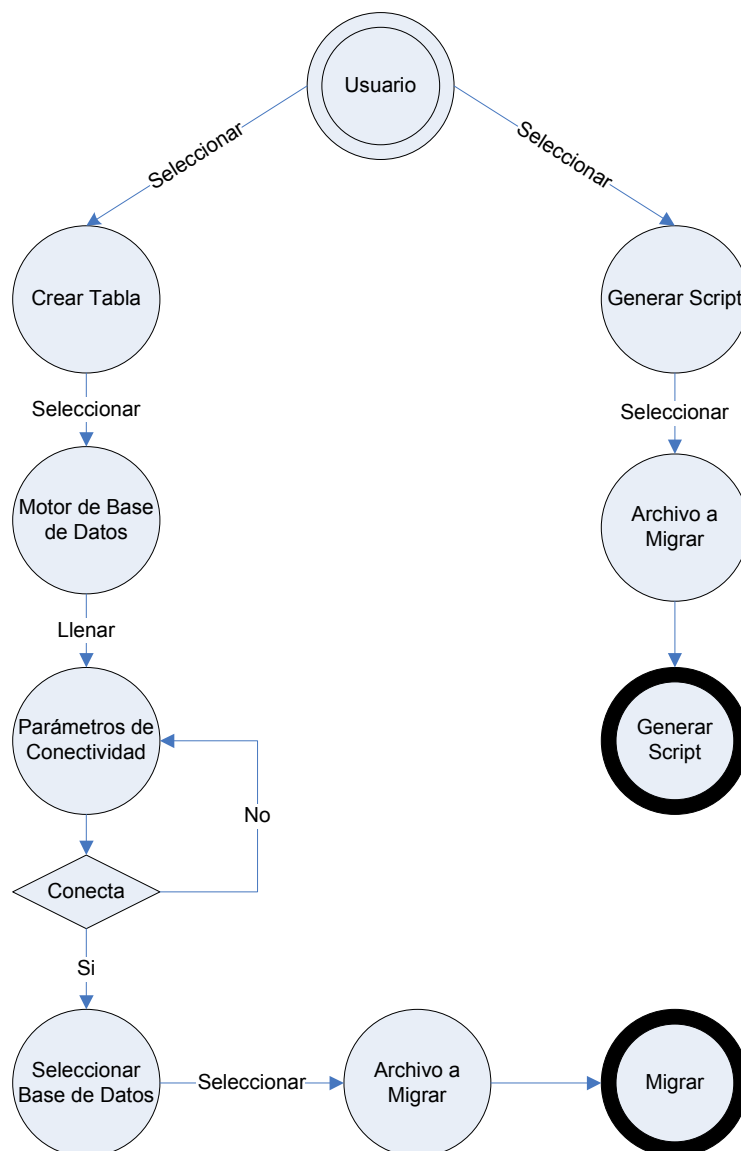


Figura 3. Diagrama de Flujo

2.3.2 ESTRUCTURA DE LA HERRAMIENTA DENTRO DE LA PROGRAMACIÓN

Dentro de la estructura en sí de la herramienta, se trabaja con dos clases fundamentales (*conexion.cs* y *DBF2BDDR.cs*) las cuales se encargan de realizar todo el proceso de migración de los datos de foxbase a un modelo de base de datos relacional.

La clase *conexion.cs* es la que se encarga en sí de realizar la conexión con todos los motores de bases de datos que se encuentran implementados en la herramienta. Siendo estos: SQL Server 2005, Oracle, PostgreSQL y MySQL. Así como también, manejar los errores que se puedan dar al tratar de conectarse con el motor de base de datos previamente seleccionado.

La clase *DBF2BDDR.cs* maneja el resto del funcionamiento de la herramienta, la cual posee varios métodos como son el de Generar Script, el método de Ingresar Datos en Base, el cual consiste en crear la tabla con el mismo nombre del archivo a migrar dentro de la base de datos previamente seleccionada, también se encarga de abrir el archivo y la base al mismo tiempo para migrar los datos de uno en uno mientras los va leyendo, entre otros.

2.3.3 CODIFICACIÓN

En esta parte mostraremos secciones del código fuente desarrollado para la herramienta con el objetivo de que se pueda entender de mejor manera el funcionamiento de la misma.

Clase: *conexion.cs*

Este método (*conectarOracle()*) es muy similar al que usamos para la conexión con todos los motores de bases de datos implementados en la herramienta; en este caso es Oracle.

```
public RetornoConexion conectarOracle()
{
    Connection con = new Connection();
    Recordset rec = new Recordset();
    conectado = false;
    List<string> listaBases = new List<string>(); try
```

```

    {
        con = new Connection();
        sqlStr = "Provider=OraOLEDB.Oracle;Data Source=" +
baseDatos.Servidor + ";User Id=" + baseDatos.Usuario +
";Password=" + baseDatos.Contraseña + ";Unicode=True";
        con.Open(sqlStr, baseDatos.Usuario, baseDatos.Contraseña,
0);

        conectado = true;
        rec = new Recordset();
        sqlStr = "SELECT username FROM all_users";
        rec.Open(sqlStr, con, CursorTypeEnum.adOpenStatic,
LockTypeEnum.adLockReadOnly, 0);
        while (!rec.EOF)
        {
            listaBases.Add(rec.Fields[0].Value.ToString());
            rec.MoveNext();
        }
        rec = null;
        MessageBox.Show("Coneccion Satisfactoria...!!",
"Felicidades", MessageBoxButtons.OK,
MessageBoxIcon.Exclamation);
        return new RetornoConexion(con, rec, listaBases, conectado);
    }
    catch (Exception ex)
    {
        MessageBox.Show("No es posible conectarse con el
servidor...!!" + "Servidor especificado no valido...!!", ex.Message,
MessageBoxButtons.OK, MessageBoxIcon.Error,
MessageBoxDefaultButton.Button1);
        con = null;
        return new RetornoConexion(con, rec, listaBases, conectado);
    }
}

```

Clase: *DBF2BDDR.cs*

En esta parte veremos cómo funciona el proceso de generación del Script dentro de la herramienta de migración de datos.

```

private void generarScript(string TableName, string DBFfilePath)
{
    try
    {
        Connection conFoxPro = new Connection();
        Recordset rsFoxPro = new Recordset(); //Recordset For
FoxPro
        string foxConStr, defaFold;

        defaFold = DBFfilePath;
        //String de Conexión para FoxPro ej. 'SourceDB'
        foxConStr = "Driver={Microsoft Visual FoxPro Driver};
SourceType=DBF;SourceDB=" + defaFold + ";Exclusive=No;";
        conFoxPro.Open(foxConStr, "", "", 0);
        sqlStr = "Select count(*) From " + DBFfilePath;
        rsFoxPro.Open(sqlStr, conFoxPro,
CursorTypeEnum.adOpenDynamic,
LockTypeEnum.adLockOptimistic, 0);
        myMax = Convert.ToInt32(rsFoxPro.Fields[0].Value);
        rsFoxPro = null;
        //Abriendo la tabla de FoxPro
        sqlStr = "Select * From " + DBFfilePath;
        rsFoxPro = new Recordset();
        rsFoxPro.Open(sqlStr, conFoxPro,
CursorTypeEnum.adOpenDynamic,
LockTypeEnum.adLockOptimistic, 0);
        //Empieza la creación del SQL. Crear tabla
        sqlStr = "Create Table " + TableName + "(";
        string foxDataType;
        for (int i = 0; i < rsFoxPro.Fields.Count; i++)
        {
            //Obtiene el Tipo de Dato de FoxPro + Adquiere el
correspondiente al Tipo de Dato en SQL
            if (obtenerTipoDato(rsFoxPro.Fields[i].Type) == "Numeric")
                foxDataType = obtenerTipoDato(rsFoxPro.Fields[i].Type)
+ "(" + rsFoxPro.Fields[i].ActualSize + ", 2)";

```

```

else if (obtenerTipoDato(rsFoxPro.Fields[i].Type) == "Bit" ||
obtenerTipoDato(rsFoxPro.Fields[i].Type) == "DateTime")
    foxDataType = obtenerTipoDato(rsFoxPro.Fields[i].Type);
else
    foxDataType = obtenerTipoDato(rsFoxPro.Fields[i].Type)
+ "(" + rsFoxPro.Fields[i].ActualSize + ")";

sqlStr = sqlStr + rsFoxPro.Fields[i].Name + " " +
foxDataType;
if (i != rsFoxPro.Fields.Count - 1)
    sqlStr = sqlStr + ", ";
else
    sqlStr = sqlStr + ");\n";
}
archivo.Text = sqlStr;
this.Cursor = Cursors.WaitCursor;
rsFoxPro.MoveFirst();
this.pbrBarra.Maximum = Convert.ToInt32(myMax);
pbrBarra.Value = 0;
foreach (Control ctl in this.Controls)
{
    ctl.Enabled = false;
}
long a = 0;
while (!rsFoxPro.EOF)
{
    //Leyendo de la Tabla de FoxPro + Añadiendo en la tabla en
SQL
    string insert = "insert into " + TableName + " values (";
    for (int i = 0; i < rsFoxPro.Fields.Count; i++)
    {
        if (obtenerTipoDato(rsFoxPro.Fields[i].Type) ==
"Numeric")
            insert += ((rsFoxPro.Fields[i].Value).ToString()).Trim();
        else if (obtenerTipoDato(rsFoxPro.Fields[i].Type) == "Bit")
            if (rsFoxPro.Fields[i].Value.ToString() == "True")
                insert += 1;
            else if (rsFoxPro.Fields[i].Value.ToString() == "False")
                insert += 0;

```

```

        else
            insert += "\\\"";
        else
            insert += "\\\"";
    ((rsFoxPro.Fields[i].Value).ToString()).Trim() + "\' ";

        if (i != rsFoxPro.Fields.Count - 1)
            insert += ", ";
        else
            insert += ");\n";
    }
    a++;
    archivo.Text += insert;
    rsFoxPro.MoveNext();
    pbrBarra.Value = pbrBarra.Value + 1;
    pnlBarra.Visible = true;
    long aux = myMax - Convert.ToInt32(pbrBarra.Value);
    gbxBases.Enabled = true;
    gbxBases.Text = "Registros Transferidos : " + pbrBarra.Value
+ " Registros Pendientes: " + aux;
    pbrBarra.Refresh();
    tlpDatosBaseArchivo.Refresh();
}
pnlBarra.Visible = false;
foreach (Control ctl in this.Controls)
{
    ctl.Enabled = true;
}
txtArchivo.Text = "";
habilitarDeshabilitarControlesPanel(tlpArchivo, true);
habilitarDeshabilitarControlesPanel(tlpBaseDatos, false);
this.Cursor = Cursors.Default;
MessageBox.Show("El script fue creado Exitosamente..!!",
"Datos Transferidos", MessageBoxButtons.OK,
MessageBoxIcon.Exclamation,
MessageBoxDefaultButton.Button1,
MessageBoxOptions.ServiceNotification);
gbxBases.Text = "";
}

```

```
catch (Exception ex)
{
    MessageBox.Show(ex.Message, "Se produjo un error al
momento de crear el script..!!", MessageBoxButtons.OK,
MessageBoxIcon.Exclamation,
MessageBoxDefaultButton.Button1,
MessageBoxOptions.ServiceNotification);
    gbxBase.Text = "";
}
}
```

2.3.4 FUNCIONAMIENTO DE LA HERRAMIENTA

Aquí describiremos paso a paso el uso de la herramienta de migración de datos utilizando un caso práctico para migrar datos de foxbase a una plataforma de bases de datos relacional.

2.3.4.1 Selección de la Opción a Realizar en la Herramienta

La herramienta nos brinda la posibilidad de crear una tabla o generar un script de los datos que vamos a migrar, las cuales las describiremos una a una a continuación.

2.3.4.1.1 Crear Tabla

Como se vio anteriormente, Foxbase maneja su estructura de bases de datos con archivos planos, por lo que esta opción (*Figura 4*) permite crear dentro de la bases de datos seleccionada posteriormente una tabla con el mismo nombre del archivo *.dbf* que contiene los datos a migrar.

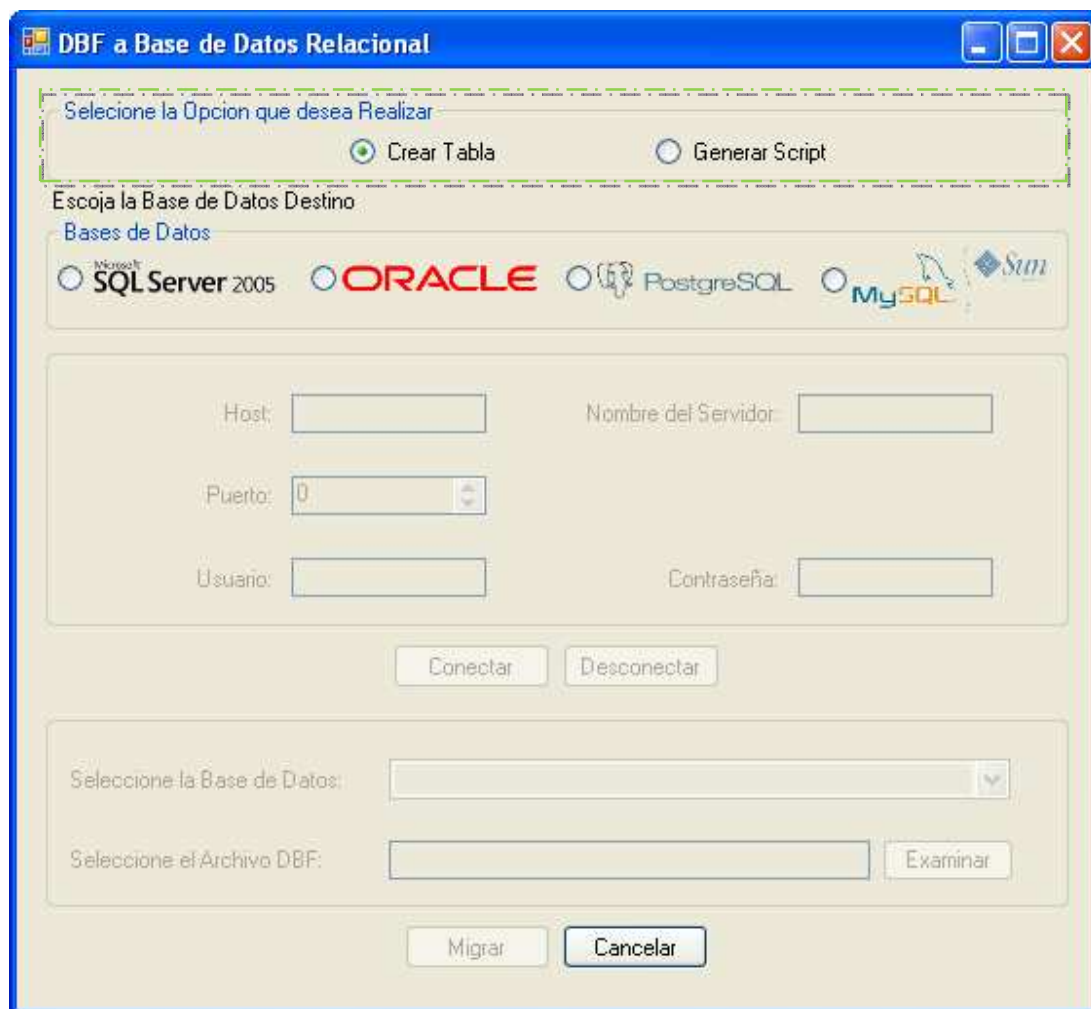


Figura 4. Migración por Selección de la Opción Tabla

2.3.4.1.2 Generar Script

En caso de que el utilitario no tenga implementado el motor de bases de datos para el cual se desea utilizar la herramienta, esta posee la opción de generar un script de extensión `.sql` el cual pueda ser libre de cambios o de fácil uso para un *DBA* (Data Base Administrator) puesto que contiene las sentencias para crear la tabla con el nombre del archivo `.dbf` y una secuencia de inserciones de los datos dentro del archivo en código SQL estándar.

Luego de seleccionar la opción *Generar Script*, la herramienta nos pide mostrar la ubicación del archivo *.dbf* (Figura 5) que posee los datos a migrar, para posteriormente empezar el proceso de generación de código en SQL (Figura 6).

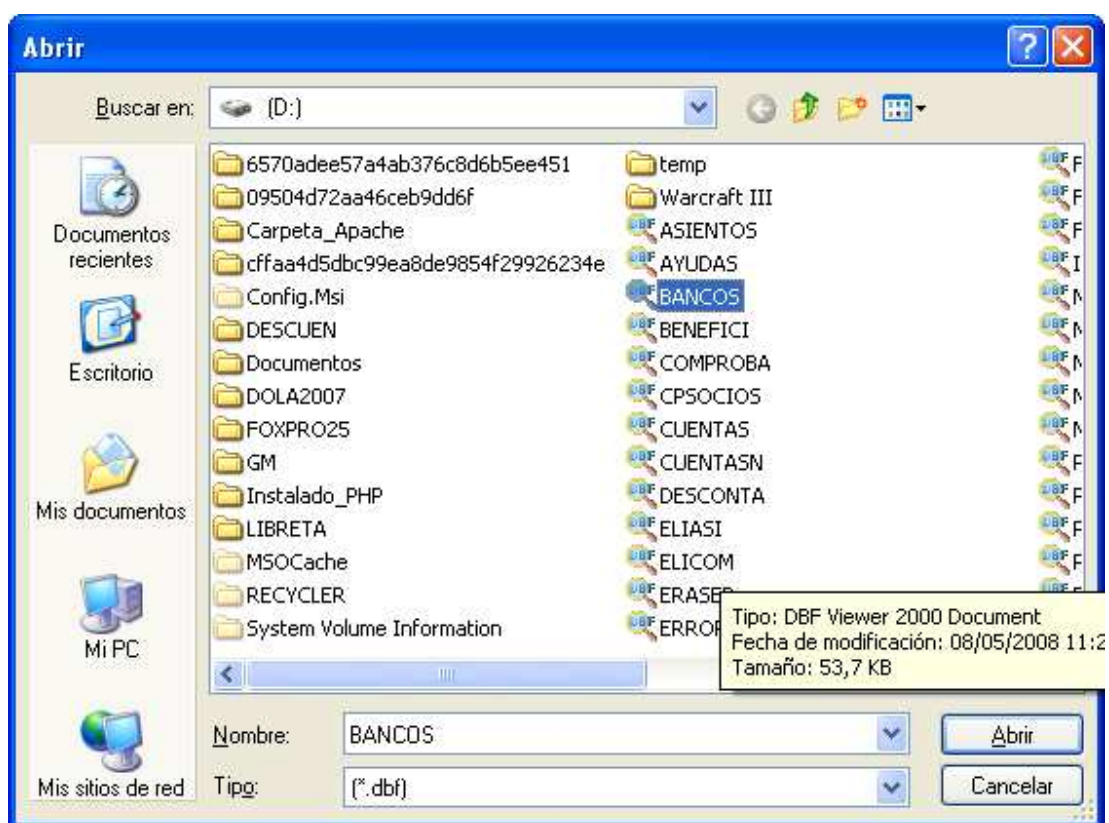


Figura 5. Selección del Archivo DBF para Generar Script en SQL

DBF a Base de Datos Relacional

Seleccione la Opción que desea Realizar

Crear Tabla Generar Script

Escoja la Base de Datos Destino

Bases de Datos

Microsoft SQL Server 2005 ORACLE PostgreSQL MySQL Sun

Host: Nombre del Servidor:

Puerto:

Usuario: Contraseña:

Seleccione la Base de Datos:

Seleccione el Archivo DBF:

Figura 6. Migración por Selección de la Opción Generar Script

Al hacer clic en la opción Migrar, el empieza la construcción del archivo *.sql* (Figura 7), para lo cual muestra se una barra de progreso junto a la cantidad de Registros Transferidos ya al *script* y la cantidad de Registros Pendientes por insertar en el mismo.

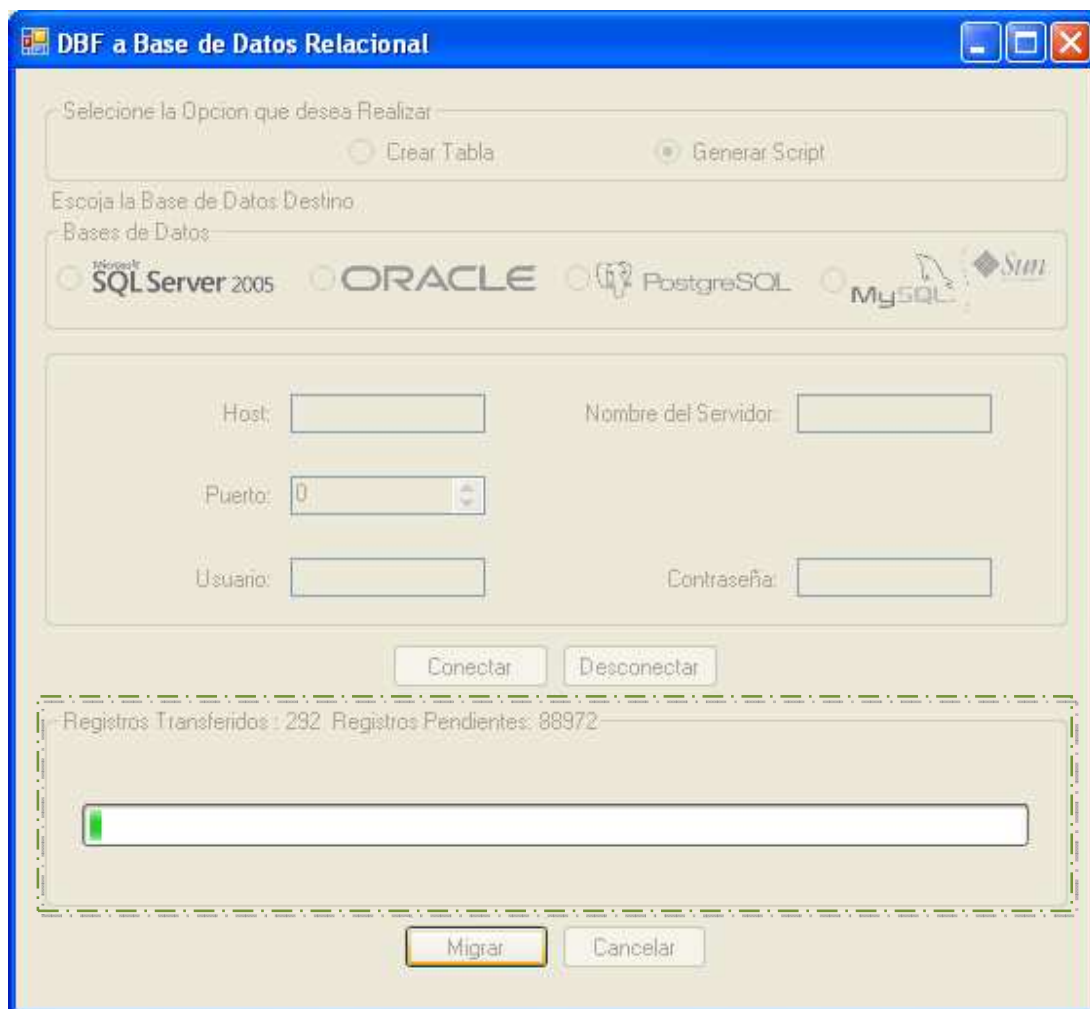


Figura 7. Progreso de Generación del Script en SQL

Si los datos son transferidos correctamente al archivo `.sql`, este mostrará un mensaje de que el script fue creado exitosamente como lo muestra la *Figura 8*.

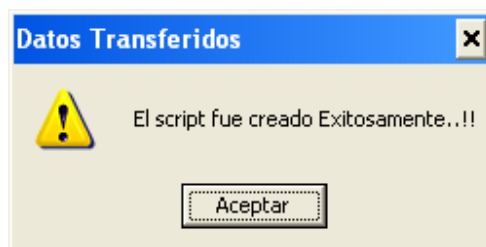


Figura 8. Generación del Script Exitosa

Para posteriormente pedir el nombre del archivo nuevo a crear (.sql) junto a la ubicación del mismo en la cual se va a almacenar en el disco duro (Figura 9).

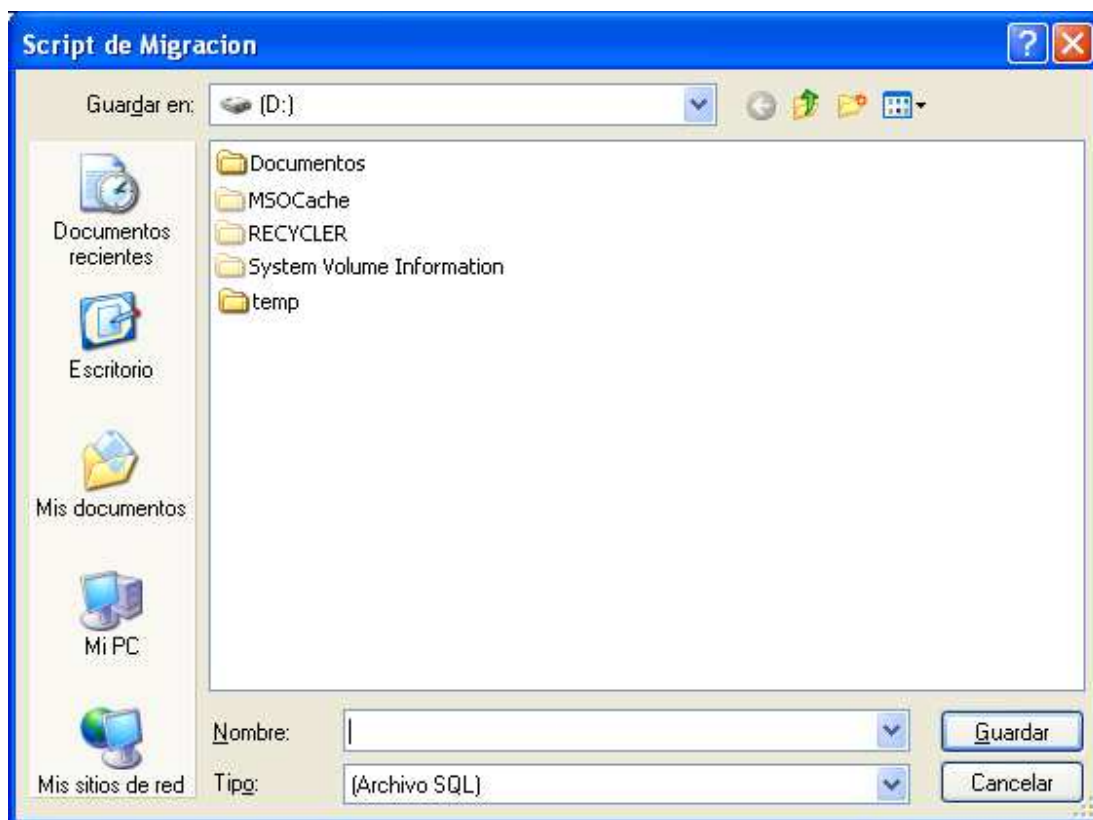


Figura 9. Selección del Nombre y Ubicación del Archivo del Script Generado

Teniendo como resultado final el script que muestra la Figura 10:

```
File Edit Fgmat View Help
Create Table .dbo.BANCOS(n_comprob varchar(7), cod_cuenta varchar(16), deb_haber Bit, valor_a Numeric(19, 2),
ordinal Numeric(19, 2), concepto varchar(254), fecha_com Date(6), n_cheque varchar(8), nota varchar(8));
insert into .dbo.BANCOS values ('196', '1110300100', 0, 6489546, 0, 'DEPOSITO EN CUENTA CORRIENTE', '03/09/1997 0:00:00', '196', '196');
insert into .dbo.BANCOS values ('197', '1110300100', 0, 42283699, 0, 'DEPOSITO EN CUENTA CORRIENTE', '08/09/1997 0:00:00', '197', '197');
insert into .dbo.BANCOS values ('198', '1110300100', 0, 16412613, 0, 'DEPOSITO EN CUENTA CORRIENTE', '09/09/1997 0:00:00', '198', '198');
insert into .dbo.BANCOS values ('199', '1110300100', 0, 44580284, 0, 'DEPOSITO EN CUENTA CORRIENTE', '10/09/1997 0:00:00', '199', '199');
insert into .dbo.BANCOS values ('200', '1110300100', 0, 17953000, 0, 'DEPOSITO EN CUENTA CORRIENTE', '11/09/1997 0:00:00', '200', '200');
insert into .dbo.BANCOS values ('201', '1110300100', 0, 99168973, 0, 'DEPOSITO EN CUENTA CORRIENTE', '12/09/1997 0:00:00', '201', '201');
insert into .dbo.BANCOS values ('202', '1110300100', 0, 6362608, 0, 'DEPOSITO EN CUENTA CORRIENTE', '12/09/1997 0:00:00', '202', '202');
insert into .dbo.BANCOS values ('203', '1110300100', 0, 3425905, 0, 'DEPOSITO EN CUENTA CORRIENTE', '15/09/1997 0:00:00', '203', '203');
insert into .dbo.BANCOS values ('204', '1110300100', 0, 5060727, 0, 'DEPOSITO EN CUENTA CORRIENTE', '16/09/1997 0:00:00', '204', '204');
insert into .dbo.BANCOS values ('205', '1110300100', 0, 3147455, 0, 'DEPOSITO EN CUENTA CORRIENTE', '17/09/1997 0:00:00', '205', '205');
insert into .dbo.BANCOS values ('206', '1110300100', 0, 17528573, 0, 'DEPOSITO EN CUENTA CORRIENTE', '18/09/1997 0:00:00', '206', '206');
```

Figura 10. Script Generado

2.3.4.2 Selección de la Base de Datos Destino

Para el proceso de migración de datos, la herramienta nos brinda la posibilidad de utilizar distintos motores de bases de datos a los cuales se desea dirigir el proceso de migración siendo estos: SQL Server, Oracle, Postgres y MySQL (Figura 11).

The screenshot shows a software window titled "DBF a Base de Datos Relacional". At the top, there are window control buttons (minimize, maximize, close). Below the title bar, the text "Seleccione la Opcion que desea Realizar" is followed by two radio buttons: "Crear Tabla" (which is selected) and "Generar Script". Underneath, the text "Escoja la Base de Datos Destino" is followed by a dashed green box containing the heading "Bases de Datos" and four radio buttons: "SQL Server 2005", "ORACLE", "PostgreSQL", and "MySQL". Below this section are five input fields: "Host", "Nombre del Servidor", "Puerto" (with a dropdown arrow and the value "0"), "Usuario", and "Contraseña". Below the input fields are two buttons: "Conectar" and "Desconectar". Further down is a section with a dropdown menu labeled "Seleccione la Base de Datos:" and a text box labeled "Seleccione el Archivo DBF:" with an "Examinar" button to its right. At the bottom of the window are two buttons: "Migrar" and "Cancelar".

Figura 11. Selección del Motor de Base de Datos para el Proceso de Migración

2.3.4.3 Parámetros de Conectividad

Para realizar la conectividad a la Base de Datos previamente seleccionada, se deben establecer los parámetros de conexión a la misma (Figura 12), las cuales

constan de: *Nombre del Host* (Nombre de la Máquina), *Nombre del Servidor* (Nombre del Servidor de Bases de Datos), *Puerto* (Puerto en el cual trabaja el motor de base de datos), *Usuario* (Usuario de la Base de Datos) y *Contraseña* (Contraseña del usuario de la base de datos anteriormente especificado).

The image shows a software dialog box titled "DBF a Base de Datos Relacional". It contains several sections for configuration. At the top, there are two radio buttons: "Crear Tabla" (selected) and "Generar Script". Below this, it asks to "Escoja la Base de Datos Destino" and lists "Bases de Datos" with radio buttons for "Microsoft SQL Server 2005", "ORACLE", "PostgreSQL", "MySQL" (selected), and "Sun". The "MySQL" section is expanded, showing fields for "Host" (localhost), "Nombre del Servidor" (mysql), "Puerto" (3306), "Usuario" (root), and "Contraseña" (masked with asterisks). There are "Conectar" and "Desconectar" buttons. At the bottom, there are fields for "Seleccione la Base de Datos:" and "Seleccione el Archivo DBF:" with an "Examinar" button, and "Migrar" and "Cancelar" buttons.

Figura 12. Parámetros de Conectividad a la Base de Datos

Si los parámetros ingresados anteriormente son correctos, se conectará correctamente al motor de bases de datos previamente seleccionado, caso contrario mostrara una falla de conexión con el servidor.

Conexión Satisfactoria (Figura 13):



Figura 13. Conexión Exitosa a la Base de Datos

Conexión Fallida (Figura 14):



Figura 14. Conexión Fallida a la Base de Datos

2.3.4.4 Selección de la Base de Datos

Luego de haber realizado la conectividad al motor de bases de datos seleccionado, se debe poder escoger con cuál de las base de datos existentes se va a trabajar para el proceso de migración (*Figura 15*).

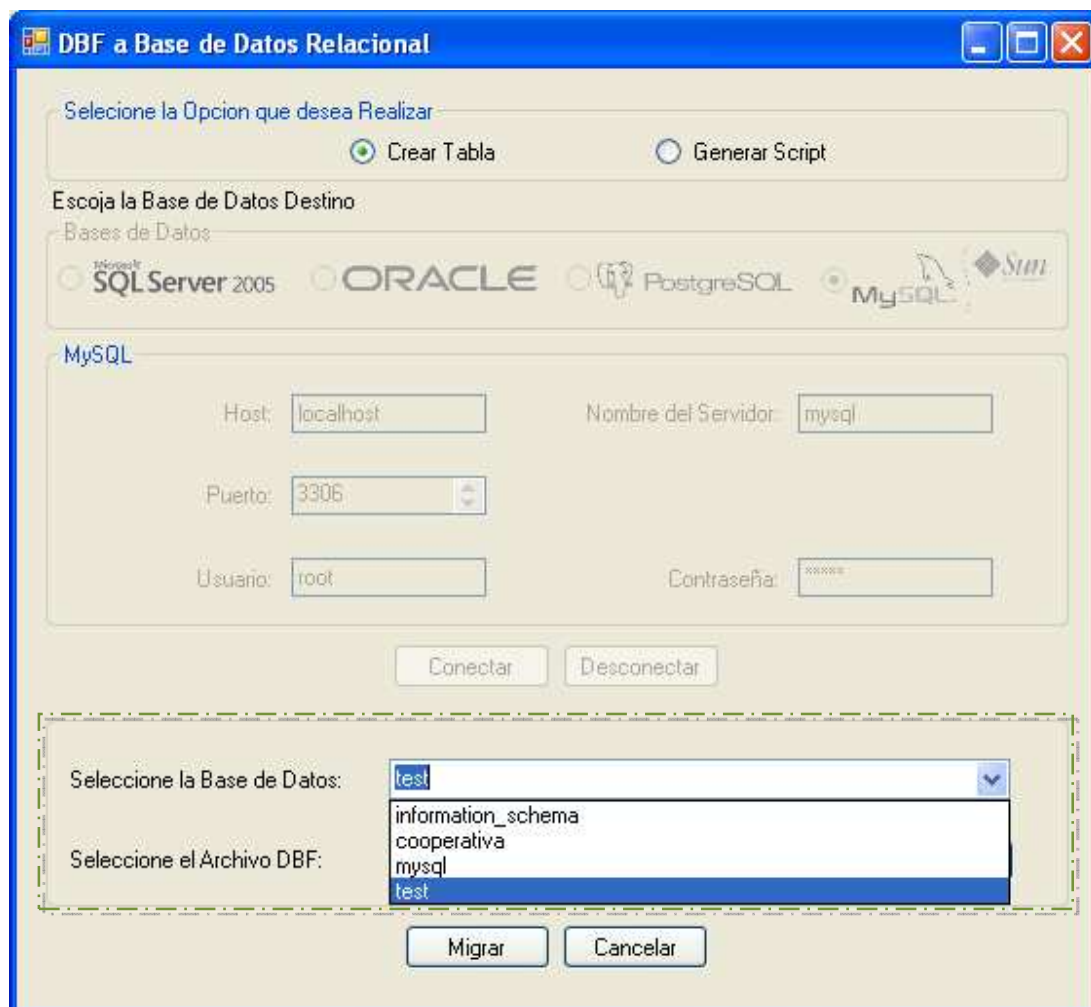


Figura 15. Selección de la Base de Datos para el Proceso de Migración

2.3.4.5 Selección del Archivo a Migrar

Luego de seleccionar la Base de Datos con la que se va a trabajar, se debe mostrar la ubicación del archivo `.dbf` (Figura 16) que posee los datos a migrar, para posteriormente empezar el proceso de migración de datos.

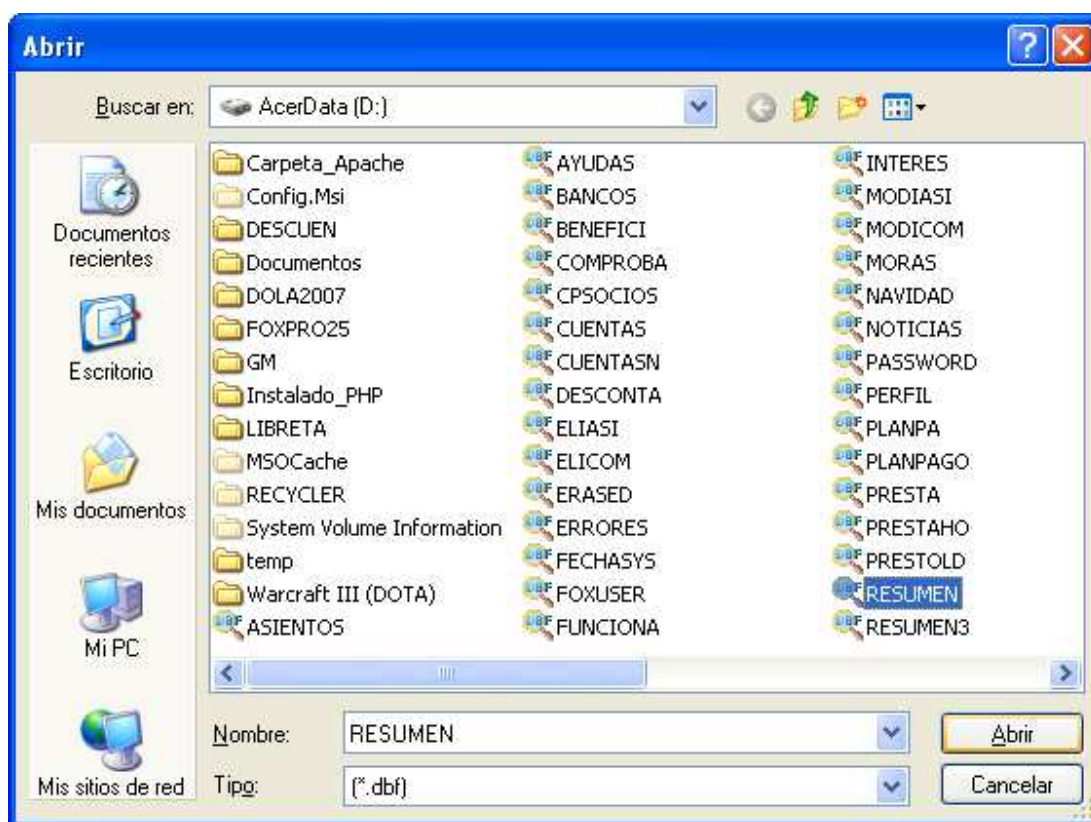


Figura 16. Selección del Archivo DBF a Migrar

Ya viendo seleccionado el archivo que deseamos migrar (Figura 17), inicia el proceso de migración de foxbase (.dbf) a un modelo de base de datos relacional, el cual se enfocaría en el motor de bases de datos previamente seleccionado.

DBF a Base de Datos Relacional

Seleccione la Opcion que desea Realizar

Crear Tabla Generar Script

Escoja la Base de Datos Destino

Bases de Datos

Microsoft SQL Server 2005 ORACLE PostgreSQL MySQL Sun

MySQL

Host: localhost Nombre del Servidor: mysql

Puerto: 3306

Usuario: root Contraseña: *****

Conectar Desconectar

Seleccione la Base de Datos: test

Seleccione el Archivo DBF: D:\RESUMEN.DBF Examinar

Migrar Cancelar

Figura 17. Previo a la Migración de los Datos

Al hacer clic en la opción *Migrar*, empieza la migración de datos (Figura 18), para lo cual muestra se una barra de progreso junto a la cantidad de Registros Transferidos ya a la base de datos y la cantidad de Registros Pendientes por insertar en la misma.

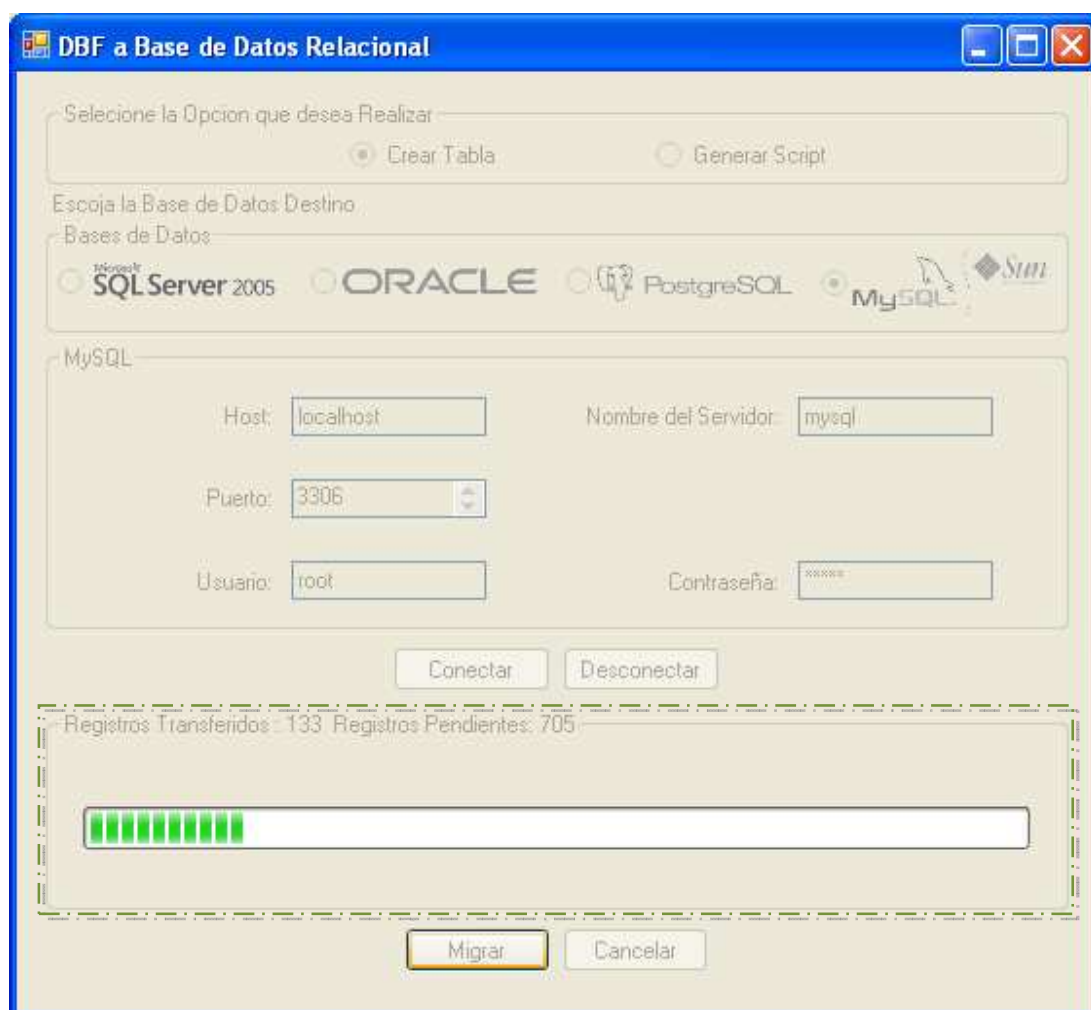


Figura 18. Proceso de Migración de Datos

Si la migración se realiza de forma exitosa, automáticamente creará la tabla en la base de datos con el mismo nombre del archivo *.dbf* y migrará los datos en la misma de forma transparente para el usuario. Caso contrario se indicará una falla en el proceso de migración.

Migración Satisfactoria (Figura 19):

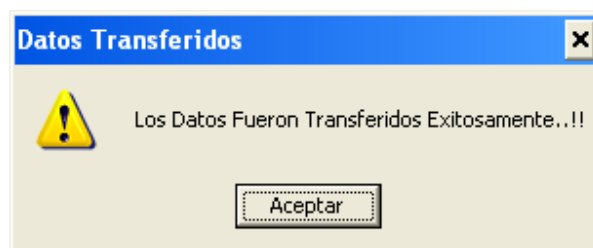


Figura 19. Mensaje de Migración Exitosa

Migración Fallida (Figura 20):



Figura 20. Mensaje de Migración Fallida

CAPÍTULO 3. PRUEBAS

En este capítulo vamos a probar la herramienta de migración de datos en un caso de la vida real. Por lo que se llevará a cabo todo el proceso de migración de los datos a un modelo de bases de datos relacional. Luego se procederá a migrar las antiguas aplicaciones desarrolladas en modo texto (*FoxPro*) a un entorno gráfico de tal forma que arrojen los mismos resultados.

3.1 ELABORACIÓN DE APLICACIONES EN MODO GRAFICO

Para la aplicación de modo gráfico, se ha planteado utilizar como caso de prueba el sistema de la Cooperativa Politécnica de Ahorro y Crédito Ltda.

3.1.1 DESCRIPCIÓN DEL CASO DE PRUEBA

3.1.1.1 Misión

La Cooperativa Politécnica de Ahorro y Crédito Ltda. Es una empresa solidaria, orientada a buscar el bienestar de sus asociados y sus familias; además de propiciar el desarrollo social, económico y cultural de la comunidad relacionada con la Escuela Politécnica Nacional.

Esta entidad de derecho privado, fue creada mediante la resolución No. 3318 del Ministerio de Bienestar Social e inscrita en la Dirección Nacional de Cooperativas con el Número de Orden 1216, en la ciudad de Quito, el 31 de Diciembre de 1971.

3.1.1.2 Objetivos

La Cooperativa cuenta actualmente con alrededor de 838 socios activos que laboran en la Escuela Politécnica Nacional en calidad de profesores, empleados administrativos y trabajadores.

Desde la doble perspectiva de Cooperación y Desarrollo, la institución trata de mejorar la calidad de vida de los individuos y la comunidad con la que trabaja; para ello, considera las siguientes áreas fundamentales: Investigación, Gestión

Cooperativa, Protección personal y patrimonial, Ahorro y Crédito; y, Intermediación Financiera.

Para el desarrollo de actividades de la cooperativa, se requiere establecer objetivos en términos medibles en un tiempo límite, de tal forma que sirvan como herramientas de dirección gerencial; por esta razón, los objetivos se clasifican básicamente en:

- **Financieros:** Son aquellos que buscan un crecimiento más rápido en lo referente a ingresos y beneficios.
- **Estratégicos:** Son los que permiten a la institución seguir compitiendo en el mercado financiero, ofreciendo a sus socios mejores servicios.

3.1.1.2.1 La Administración Financiera en la Cooperativa Politécnica

El Administrador Financiero de la cooperativa es el Gerente General quien es el encargado de resguardar y controlar los recursos financieros con los que cuenta la institución, así como la toma de decisiones necesarias para el futuro de la misma, además, analiza las ideas propuestas por la directiva que permitan invertir con la seguridad de que se obtenga un alto porcentaje de ganancias y un mínimo porcentaje de pérdidas. Una de las ideas es el crear nuevos servicios de préstamos a los socios y mejorar los beneficios que estos conllevan, con esto se intenta incrementar el número de socios que formen parte de la cooperativa.

Para lograr que la entidad alcance los objetivos que se ha propuesto, el gerente general debe conocer el Reglamento Interno y Estatutos que la rigen.

3.1.1.3 Situación Tecnológica

La Cooperativa Politécnica de Ahorro y Crédito Ltda. Hoy en día cuenta con un sistema desarrollado con Fox Pro 2.5 y la utilización de archivos planos para almacenamiento de información; es por este motivo que surge la necesidad de actualizar el manejo de esta información utilizando herramientas actuales, permitiendo que los empleados de la Cooperativa puedan acceder a la información

de sus obligaciones con la Institución desde cualquier lugar en el que se encuentren, a través del web.

3.1.2 DEFINICIÓN DEL PROBLEMA

De acuerdo a información proporcionada por directivos y personal de la Cooperativa Politécnica de Ahorro y Crédito Ltda., la institución posee actualmente un sistema desarrollado en Visual Fox 2.5 y archivos planos para el almacenamiento de información relacionada con los procesos que en ella se efectúan.

El empleo de este sistema da como resultado la subutilización tanto del recurso humano como del tiempo debido al mantenimiento constante que requiere el software, lo que genera un servicio poco eficiente a los socios.

Los costos elevados de las licencias de software propietario, han producido que las empresas no puedan contar con herramientas modernas que permitan mejorar sus servicios.

En la actualidad los conocimientos de software libre han evolucionado de gran manera, dando como resultado que la situación económica de las empresas ya no sea un problema en su crecimiento tecnológico.

3.1.3 DESCRIPCIÓN DEL NUEVO SISTEMA

El sistema desarrollado permitirá el manejo de información de socios desde su ingreso a la Cooperativa hasta su eventual retiro de la misma por motivos naturales o voluntarios.

Además el sistema admitirá el desarrollo de las siguientes actividades:

- Procesamiento de las solicitudes de los diferentes tipos de servicios financieros por parte de los socios.
- Manejo de la información de las obligaciones contraídas por los socios: Cuotas normales, préstamos de diversa índole, registro de la recaudación de

los dividendos adeudados por los socios por motivo de las obligaciones contraídas con la Cooperativa.

- Realización de consultas de los estados financieros de los socios vía WEB para lo cual se emplearan herramientas de programación modernas.

El sistema contará con distintos módulos que permitirán el manejo de la información de los diferentes servicios proporcionados a los socios. Además contará con un nivel de seguridad que controle el ingreso de los usuarios al sistema con su respectivo perfil. El sistema proveerá al usuario de una interfaz intuitiva y amigable.

3.1.4 IMPLEMENTACIÓN CON SOFTWARE LIBRE

Como se analizó anteriormente, para el proceso de creación e implementación del sistema de caso de prueba se utilizará software libre para aplicaciones Web, debido a que precio de las aplicaciones es mucho menor y la mayoría de las veces son gratuitas; aparte de que el usuario que adquiere software propietario depende al 100% de la empresa propietaria. ^[8]

A continuación se detallará cada una de las herramientas que se utilizará para la construcción del mismo.

3.1.4.1 Base de Datos PostgreSQL ^[23]



Figura 21. Logo PostgreSQL

^[8] GENTEGEEK.COM; Software Libre y Software Proprietario, ventajas y desventajas

^[23] WIKIPEDIA, La Enciclopedia Libre; PostgreSQL

Es un motor de base de datos, es servidor de base de datos relacional libre, liberado bajo la licencia BSD (*Berkeley Software Distribution*. Perteneciente al grupo de licencias de software Libre)

3.1.4.1.1 Características

Algunas de sus principales características son:

Alta concurrencia: Mediante un sistema denominado MVC (Acceso concurrente multiversión) PostgreSQL permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos. Cada usuario obtiene una visión consistente de lo último a lo que se le hizo *commit*. Esta estrategia es superior al uso de bloqueos por tabla o por filas común en otras bases, eliminando la necesidad del uso de bloqueos explícitos.

Amplia variedad de tipos nativos: PostgreSQL provee nativamente soporte para:

- Números de precisión arbitraria.
- Texto de largo ilimitado.
- Figuras geométricas (con una variedad de funciones asociadas)
- Direcciones IP (IPv4 e IPv6).
- Bloques de direcciones estilo CIDR.
- Direcciones MAC.
- Arrays.

3.1.4.1.2 Otras características

- Claves ajenas también denominadas Llaves ajenas o Llaves Foráneas (*foreign keys*).

- Disparadores (*triggers*).
- Vistas.
- Integridad transaccional.
- Herencia de tablas.
- Tipos de datos y operaciones geométricas.

3.1.4.1.3 Funciones

Bloques de código que se ejecutan en el servidor. Pueden ser escritos en varios lenguajes, con la potencia que cada uno de ellos da, desde las operaciones básicas de programación, tales como bifurcaciones y bucles, hasta las complejidades de la programación orientación a objetos o la programación funcional.

Los disparadores (*triggers* en inglés) son funciones enlazadas a operaciones sobre los datos.

Algunos de los lenguajes que se pueden usar son los siguientes:

- Un lenguaje propio llamado PL/PgSQL (similar al PL/SQL de oracle).
- C.
- C++.
- Java (vía PL/Java).
- PL/Perl.
- pI PHP.
- PL/Python.
- PL/Ruby.
- PL/sh.
- PL/Tcl.
- PL/Scheme.

- Lenguaje para aplicaciones estadísticas R through PL/R.

3.1.4.2 Servidor HTTP Apache ^[25]



Figura 22. Logo Servidor HTTP Apache

El **servidor HTTP Apache** es un software (libre) servidor HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etc.), Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual.

El servidor Apache se desarrolla dentro del proyecto HTTP Server (httpd) de la Apache Software Foundation.

Apache presenta entre otras características mensajes de error altamente configurables, bases de datos de autenticación y negociado de contenido, pero fue criticado por la falta de una interfaz gráfica que ayude en su configuración.

3.1.4.2.1 Ventajas

- Modular
- Open source
- Multi-plataforma
- Extensible
- Popular (fácil conseguir ayuda/suporte)
- Gratuito

^[25] WIKIPEDIA, La Enciclopedia Libre; Servidor HTTP Apache

3.1.4.2.2 Módulos

La arquitectura del servidor Apache es muy modular. El servidor consta de una sección *core* y mucha de la funcionalidad que podría considerarse básica para un servidor web es provista por módulos. Algunos de estos son:

- **mod_ssl** - Comunicaciones Seguras vía TLS.
- **mod_rewrite** - reescritura de direcciones servidas (generalmente utilizado para transformar páginas dinámicas como php en páginas estáticas html para así engañar a los navegantes o a los motores de búsqueda en cuanto a cómo fueron desarrolladas estas páginas).
- **mod_dav** - Soporte del protocolo WebDAV (RFC 2518).
- **mod_deflate** - Compresión transparente con el algoritmo deflate del contenido enviado al cliente.
- **mod_auth_ldap** - Permite autenticar usuarios contra un servidor LDAP.
- **mod_proxy_ajp** - Conector para enlazar con el servidor Jakarta Tomcat de páginas dinámicas en Java (servlets y JSP).

El servidor de base puede ser extendido con la inclusión de módulos externos entre los cuales se encuentran:

- **mod_perl** - Páginas dinámicas en Perl.
- **mod_php** - Páginas dinámicas en PHP.
- **mod_python** - Páginas dinámicas en Python.
- **mod_rexx** - Páginas dinámicas en REXX y Object REXX.
- **mod_ruby** - Páginas dinámicas en Ruby.
- **mod_aspdotnet** - Páginas dinámicas en .NET_de_Microsoft (Módulo retirado).
- **mod_mono** - Páginas dinámicas en Mono
- **mod_security** - Filtrado a nivel de aplicación, para seguridad.

3.1.4.3 PHP ^[22]



Figura 23. Logo PHP

PHP es un lenguaje de programación usado frecuentemente para la creación de contenido para sitios web con los cuales se puede programar las páginas html y los códigos de fuente. PHP es un acrónimo recursivo que significa “PHP Hypertext Pre-processor” (inicialmente PHP Tools, o, *Personal Home Page Tools*), y se trata de un lenguaje interpretado usado para la creación de aplicaciones para servidores, o creación de contenido dinámico para sitios web. Últimamente también para la creación de otro tipo de programas incluyendo aplicaciones con interfaz gráfica usando las librerías Qt o GTK+.

3.1.4.3.1 Usos de PHP

Los principales usos del PHP son los siguientes:

- Programación de páginas web dinámicas, habitualmente en combinación con el motor de base datos Postgres, aunque cuenta con soporte nativo para otros motores, incluyendo el estándar ODBC, lo que amplía en gran medida sus posibilidades de conexión.
- Programación en consola, al estilo de Perl o Shell scripting.
- Creación de aplicaciones gráficas independientes del navegador, por medio de la combinación de PHP y Qt/GTK+, lo que permite desarrollar aplicaciones de escritorio en los sistemas operativos en los que está soportado

^[22] WIKIPEDIA, La Enciclopedia Libre; PHP

3.1.4.3.2 *Ventajas de PHP*

Es un lenguaje multiplataforma.

- Capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad, destaca su conectividad con Postgres.
- Leer y manipular datos desde diversas fuentes, incluyendo datos que pueden ingresar los usuarios desde formularios HTML.
- Capacidad de expandir su potencial utilizando la enorme cantidad de módulos (llamados ext's o extensiones).
- Posee una amplia documentación en su página oficial, entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Permite las técnicas de Programación Orientada a Objetos.
- Permite crear los formularios para la web.
- Biblioteca nativa de funciones sumamente amplia e incluida
- No requiere definición de tipos de variables ni manejo detallado del bajo nivel.

3.1.4.4 **JavaScript** ^[19]

JavaScript es un lenguaje de programación interpretado, es decir, que no requiere compilación, utilizado principalmente en páginas web, con una sintaxis semejante a la del lenguaje Java y el lenguaje C.

^[19] WIKIPEDIA, La Enciclopedia Libre; JavaScript

Al igual que Java, JavaScript es un lenguaje orientado a objetos propiamente dicho, ya que dispone de Herencia, si bien esta se realiza siguiendo el paradigma de programación basada en prototipos, ya que las nuevas clases se generan clonando las clases base (prototipos) y extendiendo su funcionalidad.

Todos los navegadores modernos interpretan el código JavaScript integrado dentro de las páginas web. Para interactuar con una página web se provee al lenguaje JavaScript de una implementación del DOM.

Para evitar estas incompatibilidades, el World Wide Web Consortium diseñó el estándar Document Object Model (DOM, ó Modelo de Objetos del Documento en castellano), que incorporan Konqueror, las versiones 6 de Internet Explorer y Netscape Navigator, Opera versión 7, y Mozilla desde su primera versión.

3.1.4.5 Navegador: Mozilla Firefox ^[21]



Figura 24. Logo Mozilla Firefox

Mozilla Firefox es un navegador de Internet, con interfaz gráfica de usuario, desarrollado por la Corporación Mozilla y un gran número de voluntarios externos.

El programa es multiplataforma y está disponible en versiones para Microsoft Windows, Mac OS X y GNU/Linux. El código ha sido portado por terceros a FreeBSD, OS/2, Solaris, SkyOS, BeOS y más recientemente, Windows XP Professional x64 Edition.

^[21] WIKIPEDIA, La Enciclopedia Libre; Mozilla Firefox

El código fuente de Firefox es software libre, publicado bajo una triple licencia GPL/LGPL/MPL.

3.1.5 MODELOS DE BASES DE DATOS

En esta parte, trataremos los modelos de bases de datos existentes en el sistema de caso de prueba; siendo estos el modelo de base de datos basado en archivos que usa xbase para que las aplicaciones interactúen con los datos y el modelo de bases de datos implementado para el nuevo sistema con la herramienta de migración de datos, el cual usa un modelo de bases de dato relacional.

3.1.5.1 Modelo de Bases Datos basado en XBase

Xbase maneja archivos planos (.dbf) para manejar sus datos por lo que su modelo de Bases de Datos se lo puede ver a continuación en la *Figura 25*.

ASIENTOS.DBF				
Nombre	Tipo	Tamaño	Dec	
<input checked="" type="checkbox"/> ab N_COMPROB	Caracter	7	0	
<input checked="" type="checkbox"/> ab COD_CUENTA	Caracter	16	0	
<input checked="" type="checkbox"/> I DEB_HABER	Logico	1	0	
<input checked="" type="checkbox"/> 9.0 VALOR_A	Numerico	17	2	
<input checked="" type="checkbox"/> 9.0 ORDINAL	Numerico	4	0	
<input checked="" type="checkbox"/> I LIBRETA	Logico	1	0	

CUENTAS.DBF				
Nombre	Tipo	Tamaño	Dec	
<input checked="" type="checkbox"/> ab COD_CUENTA	Caracter	16	0	
<input checked="" type="checkbox"/> ab NOMBRE_CUE	Caracter	60	0	
<input checked="" type="checkbox"/> 9.0 SALDO	Numerico	17	2	
<input checked="" type="checkbox"/> 9.0 SALDO1	Numerico	17	2	
<input checked="" type="checkbox"/> 9.0 DEBE	Numerico	17	2	
<input checked="" type="checkbox"/> 9.0 HABER	Numerico	17	2	
<input checked="" type="checkbox"/> ? FECHA_INIC	Fecha	8	0	
<input checked="" type="checkbox"/> 9.0 SALDO_FINA	Numerico	17	2	
<input checked="" type="checkbox"/> I HOJA_RAMA	Logico	1	0	
<input checked="" type="checkbox"/> 9.0 SALDO_ACTU	Numerico	17	2	

COMPROBA.DBF				
Nombre	Tipo	Tamaño	Dec	
<input checked="" type="checkbox"/> ab N_PARTIDA	Caracter	15	0	
<input checked="" type="checkbox"/> ab N_COMPROB	Caracter	7	0	
<input checked="" type="checkbox"/> ab T_COMPROB	Caracter	8	0	
<input checked="" type="checkbox"/> ? FECHA_COM	Fecha	8	0	
<input checked="" type="checkbox"/> 9.0 VALOR	Numerico	17	2	
<input checked="" type="checkbox"/> ab N_CHEQUE	Caracter	8	0	
<input checked="" type="checkbox"/> ab CONCEPTO	Caracter	254	0	
<input checked="" type="checkbox"/> ? FECHA_ASIE	Fecha	8	0	
<input checked="" type="checkbox"/> ab CONTROL_US	Caracter	10	0	
<input checked="" type="checkbox"/> ab BENEFI	Caracter	30	0	
<input checked="" type="checkbox"/> I ELIMINADO	Logico	1	0	
<input checked="" type="checkbox"/> I CAMBIADO	Logico	1	0	

MODIASI.DBF				
Nombre	Tipo	Tamaño	Dec	
<input checked="" type="checkbox"/> ab N_COMPROB	Caracter	7	0	
<input checked="" type="checkbox"/> ab COD_CUENTA	Caracter	16	0	
<input checked="" type="checkbox"/> I DEB_HABER	Logico	1	0	
<input checked="" type="checkbox"/> 9.0 VALOR_A	Numerico	17	2	
<input checked="" type="checkbox"/> 9.0 ORDINAL	Numerico	4	0	
<input checked="" type="checkbox"/> ab CONTROL_US	Caracter	10	0	
<input checked="" type="checkbox"/> ? FECHACAM	Fecha	8	0	

ELIASI.DBF

Nombre	Tipo	Tamano	Dec
<input checked="" type="checkbox"/> ab N_COMPROB	Caracter	7	0
<input checked="" type="checkbox"/> ab COD_CUENTA	Caracter	16	0
<input checked="" type="checkbox"/> I DEB_HABER	Logico	1	0
<input checked="" type="checkbox"/> 9.0 VALOR_A	Numerico	17	2
<input checked="" type="checkbox"/> 9.0 ORDINAL	Numerico	4	0

ELICOM.DBF

Nombre	Tipo	Tamano	Dec
<input checked="" type="checkbox"/> ab N_PARTIDA	Caracter	15	0
<input checked="" type="checkbox"/> ab N_COMPROB	Caracter	7	0
<input checked="" type="checkbox"/> ab T_COMPROB	Caracter	8	0
<input checked="" type="checkbox"/> 7 FECHA_COM	Fecha	8	0
<input checked="" type="checkbox"/> 9.0 VALOR	Numerico	17	2
<input checked="" type="checkbox"/> ab N_CHEQUE	Caracter	8	0
<input checked="" type="checkbox"/> ab CONCEPTO	Caracter	254	0
<input checked="" type="checkbox"/> 7 FECHA_ASIE	Fecha	8	0
<input checked="" type="checkbox"/> ab CONTROL_US	Caracter	10	0
<input checked="" type="checkbox"/> ab BENEFI	Caracter	30	0
<input checked="" type="checkbox"/> I ELIMINADO	Logico	1	0

MODICOM.DBF

Nombre	Tipo	Tamano	Dec
<input checked="" type="checkbox"/> ab N_PARTIDA	Caracter	15	0
<input checked="" type="checkbox"/> ab N_COMPROB	Caracter	7	0
<input checked="" type="checkbox"/> ab T_COMPROB	Caracter	8	0
<input checked="" type="checkbox"/> 7 FECHA_COM	Fecha	8	0
<input checked="" type="checkbox"/> 9.0 VALOR	Numerico	17	2
<input checked="" type="checkbox"/> ab N_CHEQUE	Caracter	8	0
<input checked="" type="checkbox"/> ab CONCEPTO	Caracter	254	0
<input checked="" type="checkbox"/> 7 FECHA_ASIE	Fecha	8	0
<input checked="" type="checkbox"/> ab CONTROL_US	Caracter	10	0
<input checked="" type="checkbox"/> ab BENEFI	Caracter	30	0
<input checked="" type="checkbox"/> I ELIMINADO	Logico	1	0
<input checked="" type="checkbox"/> I CAMBIADO	Logico	1	0
<input checked="" type="checkbox"/> 7 FECHACAM	Fecha	8	0

ERASED.DBF

Nombre	Tipo	Tamano	Dec
<input checked="" type="checkbox"/> ab REGISTRO	Caracter	60	0
<input checked="" type="checkbox"/> 7 FECHA	Fecha	8	0
<input checked="" type="checkbox"/> ab HORA	Caracter	10	0

ERRORES.DBF

Nombre	Tipo	Tamano	Dec
<input checked="" type="checkbox"/> 7 FECHA	Fecha	8	0
<input checked="" type="checkbox"/> ab HORA	Caracter	10	0
<input checked="" type="checkbox"/> ab DESCRI	Caracter	100	0

PLANPAGO.DBF

Nombre	Tipo	Tamano	Dec
<input checked="" type="checkbox"/> ab NROSOC	Caracter	4	0
<input checked="" type="checkbox"/> ab NROPRE	Caracter	5	0
<input checked="" type="checkbox"/> ab COD_CUENTA	Caracter	10	0
<input checked="" type="checkbox"/> 9.0 NCUOTA	Numerico	3	0
<input checked="" type="checkbox"/> 7 FECHAVEN	Fecha	8	0
<input checked="" type="checkbox"/> 7 FECHACAN	Fecha	8	0
<input checked="" type="checkbox"/> I CANCELADO	Logico	1	0
<input checked="" type="checkbox"/> 9.0 INTERES	Numerico	10	2
<input checked="" type="checkbox"/> 9.0 AMORTIZA	Numerico	10	2

PASSWORD.DBF

Nombre	Tipo	Tamano	Dec
<input checked="" type="checkbox"/> ab IDENTIFICA	Caracter	11	0
<input checked="" type="checkbox"/> ab NOMBRE_USR	Caracter	36	0
<input checked="" type="checkbox"/> ab DEPART_USR	Caracter	31	0
<input checked="" type="checkbox"/> ab PwD	Caracter	15	0
<input checked="" type="checkbox"/> ab PERFIL	Caracter	11	0

INTERES.DBF

Nombre	Tipo	Tama...	Dec
<input checked="" type="checkbox"/> 9.0 INTERES	Numerico	5	2

FUNCIONA.DBF

Nombre	Tipo	Tamano	Dec
<input checked="" type="checkbox"/> ab FUNCION	Caracter	25	0
<input checked="" type="checkbox"/> ab FUNCIONAR	Caracter	40	0

BENEFICI.DBF

Nombre	Tipo	Tamaño	Dec
<input checked="" type="checkbox"/> ab NROSOC	Caracter	4	0
<input checked="" type="checkbox"/> ab NOMBREBEN	Caracter	40	0
<input checked="" type="checkbox"/> 7 FECHANACIM	Fecha	8	0
<input checked="" type="checkbox"/> ab PARENTEZCO	Caracter	10	0
<input checked="" type="checkbox"/> ab OTRO	Caracter	10	0

CPSOCIOS.DBF

Nombre	Tipo	Tamaño	Dec
<input checked="" type="checkbox"/> ab ESTSOC	Caracter	1	0
<input checked="" type="checkbox"/> ab DESSOC	Caracter	1	0
<input checked="" type="checkbox"/> ab NROSOC	Caracter	4	0
<input checked="" type="checkbox"/> ab NOMSOC	Caracter	40	0
<input checked="" type="checkbox"/> ab NOMBRE	Caracter	20	0
<input checked="" type="checkbox"/> 7 FECHANACIM	Fecha	8	0
<input checked="" type="checkbox"/> ab CODIGO	Caracter	4	0
<input checked="" type="checkbox"/> ab CEDSOC	Caracter	10	0
<input checked="" type="checkbox"/> ab DIRSOC	Caracter	30	0
<input checked="" type="checkbox"/> ab TDOSOC	Caracter	10	0
<input checked="" type="checkbox"/> ab TRABNOEPN	Caracter	30	0
<input checked="" type="checkbox"/> ab TELFNOEPN	Caracter	10	0
<input checked="" type="checkbox"/> ab BENSOC	Caracter	30	0
<input checked="" type="checkbox"/> ab IECSOC	Caracter	1	0
<input checked="" type="checkbox"/> 9.0 NHSSOC	Numerico	2	0
<input checked="" type="checkbox"/> ab ICASOC	Caracter	4	0
<input checked="" type="checkbox"/> ab ILTSOC	Caracter	2	0
<input checked="" type="checkbox"/> 7 FINSOC	Fecha	8	0
<input checked="" type="checkbox"/> 7 FAPSOC	Fecha	8	0
<input checked="" type="checkbox"/> 7 FRTSOC	Fecha	8	0
<input checked="" type="checkbox"/> ab ISXSOC	Caracter	1	0
<input checked="" type="checkbox"/> ab ICURSO	Caracter	1	0
<input checked="" type="checkbox"/> 9.0 MONTGA	Numerico	7	0
<input checked="" type="checkbox"/> 7 FECHAAP	Fecha	8	0
<input checked="" type="checkbox"/> I PROBLEMA	Logico	1	0

PRESTAH0.DBF

Nombre	Tipo	Tamaño	Dec
<input checked="" type="checkbox"/> ab NROSOC	Caracter	4	0
<input checked="" type="checkbox"/> I CANCELADO	Logico	1	0
<input checked="" type="checkbox"/> I REGISTRADO	Logico	1	0
<input checked="" type="checkbox"/> ab CUENTA	Caracter	5	0
<input checked="" type="checkbox"/> ab TIPOPRESTA	Caracter	2	0
<input checked="" type="checkbox"/> ab NROPRE	Caracter	5	0
<input checked="" type="checkbox"/> 9.0 MONTO	Numerico	9	2
<input checked="" type="checkbox"/> 7 FECHAP	Fecha	8	0
<input checked="" type="checkbox"/> 7 FECPRIM	Fecha	8	0
<input checked="" type="checkbox"/> 7 FECULTIMO	Fecha	8	0
<input checked="" type="checkbox"/> 9.0 PLAZO	Numerico	3	0
<input checked="" type="checkbox"/> 9.0 INTERES	Numerico	5	2
<input checked="" type="checkbox"/> 9.0 CUOTA	Numerico	10	2
<input checked="" type="checkbox"/> ab GAR1	Caracter	4	0
<input checked="" type="checkbox"/> ab GAR2	Caracter	4	0
<input checked="" type="checkbox"/> 9.0 ULTIMO	Numerico	2	0
<input checked="" type="checkbox"/> ab NOMSOC	Caracter	30	0
<input checked="" type="checkbox"/> ab FORMAPA	Caracter	1	0
<input checked="" type="checkbox"/> ab OBSERVA	Caracter	80	0

NOTICIAS.DBF

Nombre	Tipo	Tamaño	Dec
<input checked="" type="checkbox"/> ab NOTICIA	Caracter	60	0

PERFIL.DBF

Nombre	Tipo	Tamaño	Dec
<input checked="" type="checkbox"/> ab PERFIL	Caracter	10	0
<input checked="" type="checkbox"/> ab DESCRI	Caracter	60	0
<input checked="" type="checkbox"/> I M11	Logico	1	0
<input checked="" type="checkbox"/> I M12	Logico	1	0
<input checked="" type="checkbox"/> I M13	Logico	1	0
<input checked="" type="checkbox"/> I M14	Logico	1	0
<input checked="" type="checkbox"/> I M15	Logico	1	0
<input checked="" type="checkbox"/> I M1	Logico	1	0
<input checked="" type="checkbox"/> I M17	Logico	1	0

RESUMEN3.DBF				RESUMEN4.DBF			
Nombre	Tipo	Tamaño	Dec	Nombre	Tipo	Tamaño	Dec
<input checked="" type="checkbox"/> ab NROSOC	Caracter	4	0	<input checked="" type="checkbox"/> ab NROSOC	Caracter	4	0
<input checked="" type="checkbox"/> ab CODIGO	Caracter	4	0	<input checked="" type="checkbox"/> ab CODIGO	Caracter	4	0
<input checked="" type="checkbox"/> ab NOMSOC	Caracter	38	0	<input checked="" type="checkbox"/> ab NOMSOC	Caracter	38	0
<input checked="" type="checkbox"/> ab NOFINANCIE	Caracter	18	0	<input checked="" type="checkbox"/> ab NOFINANCIE	Caracter	18	0
<input checked="" type="checkbox"/> 9.0 APORTES	Numerico	10	2	<input checked="" type="checkbox"/> 9.0 APORTES	Numerico	10	2
<input checked="" type="checkbox"/> 9.0 DIFERENCIA	Numerico	10	2	<input checked="" type="checkbox"/> 9.0 DIFERENCIA	Numerico	10	2
<input checked="" type="checkbox"/> 9.0 INT_QUIRO	Numerico	10	2	<input checked="" type="checkbox"/> 9.0 INT_QUIRO	Numerico	10	2
<input checked="" type="checkbox"/> 9.0 INT_LIBRER	Numerico	10	2	<input checked="" type="checkbox"/> 9.0 INT_LIBRER	Numerico	10	2
<input checked="" type="checkbox"/> 9.0 INT_ANTICI	Numerico	10	2	<input checked="" type="checkbox"/> 9.0 INT_ANTICI	Numerico	10	2
<input checked="" type="checkbox"/> 9.0 INT_AUTOMA	Numerico	10	2	<input checked="" type="checkbox"/> 9.0 INT_AUTOMA	Numerico	10	2
<input checked="" type="checkbox"/> 9.0 INT_MORA	Numerico	10	2	<input checked="" type="checkbox"/> 9.0 INT_MORA	Numerico	10	2
<input checked="" type="checkbox"/> 9.0 INT_HIPO	Numerico	10	2	<input checked="" type="checkbox"/> 9.0 INT_HIPO	Numerico	10	2
<input checked="" type="checkbox"/> 9.0 AYUDAS	Numerico	10	2	<input checked="" type="checkbox"/> 9.0 AYUDAS	Numerico	10	2
<input checked="" type="checkbox"/> 9.0 INT_PRESTA	Numerico	10	2	<input checked="" type="checkbox"/> 9.0 INT_PRESTA	Numerico	10	2
<input checked="" type="checkbox"/> 9.0 AHORROS	Numerico	10	2	<input checked="" type="checkbox"/> 9.0 AHORROS	Numerico	10	2
<input checked="" type="checkbox"/> 9.0 CESANTIA	Numerico	10	2	<input checked="" type="checkbox"/> 9.0 CESANTIA	Numerico	10	2
<input checked="" type="checkbox"/> 9.0 P_QU11201	Numerico	10	2	<input checked="" type="checkbox"/> 9.0 P_QU11201	Numerico	10	2
<input checked="" type="checkbox"/> 9.0 P_QU11237	Numerico	10	2	<input checked="" type="checkbox"/> 9.0 P_QU11237	Numerico	10	2
<input checked="" type="checkbox"/> 9.0 P_LIBRERIA	Numerico	10	2	<input checked="" type="checkbox"/> 9.0 P_LIBRERIA	Numerico	10	2
<input checked="" type="checkbox"/> 9.0 P_EM11238	Numerico	10	2	<input checked="" type="checkbox"/> 9.0 P_EM11238	Numerico	10	2
<input checked="" type="checkbox"/> 9.0 P_AU11239	Numerico	10	2	<input checked="" type="checkbox"/> 9.0 P_AU11239	Numerico	10	2
<input checked="" type="checkbox"/> 9.0 P_AN11240	Numerico	10	2	<input checked="" type="checkbox"/> 9.0 P_AN11240	Numerico	10	2
<input checked="" type="checkbox"/> 9.0 P_AS11241	Numerico	10	2	<input checked="" type="checkbox"/> 9.0 P_AS11241	Numerico	10	2
<input checked="" type="checkbox"/> 9.0 P_CO11255	Numerico	10	2	<input checked="" type="checkbox"/> 9.0 P_CO11255	Numerico	10	2
<input checked="" type="checkbox"/> 9.0 P_HI11247	Numerico	10	2	<input checked="" type="checkbox"/> 9.0 P_HI11247	Numerico	10	2
<input checked="" type="checkbox"/> 9.0 MERCADERIA	Numerico	10	2	<input checked="" type="checkbox"/> 9.0 MERCADERIA	Numerico	10	2
<input checked="" type="checkbox"/> 9.0 SEGUROS	Numerico	10	2	<input checked="" type="checkbox"/> 9.0 SEGUROS	Numerico	10	2
<input checked="" type="checkbox"/> 9.0 SEGURO_V...	Numerico	10	2	<input checked="" type="checkbox"/> 9.0 SEGURO_V...	Numerico	10	2
<input checked="" type="checkbox"/> 9.0 OTROS1	Numerico	10	2	<input checked="" type="checkbox"/> 9.0 OTROS1	Numerico	10	2
<input checked="" type="checkbox"/> 9.0 OTROS2	Numerico	10	2	<input checked="" type="checkbox"/> 9.0 OTROS2	Numerico	10	2
<input checked="" type="checkbox"/> 9.0 DIFERA	Numerico	10	2	<input checked="" type="checkbox"/> 9.0 DIFERA	Numerico	10	2
<input checked="" type="checkbox"/> 9.0 SFAMILIAR	Numerico	10	2	<input checked="" type="checkbox"/> 9.0 SFAMILIAR	Numerico	10	2
<input checked="" type="checkbox"/> 9.0 ADMINISTRA	Numerico	10	2	<input checked="" type="checkbox"/> 9.0 ADMINISTRA	Numerico	10	2
<input checked="" type="checkbox"/> ab BANCO	Caracter	15	0	<input checked="" type="checkbox"/> ab BANCO	Caracter	15	0
<input checked="" type="checkbox"/> ab CUENTA	Caracter	14	0	<input checked="" type="checkbox"/> ab CUENTA	Caracter	14	0
<input checked="" type="checkbox"/> ab TIPOCUENTA	Caracter	3	0	<input checked="" type="checkbox"/> ab TIPOCUENTA	Caracter	3	0

Figura 25. Modelo de Bases Datos basado en XBase

3.1.5.2 Modelo de Base de Datos Relacional del Sistema de Caso de Prueba

Para realizar este modelo de Base de Datos, se tomaron en cuenta muchos aspectos muy importantes los cuales se enfocan en un proceso automatizado de migración y son:

- Los nombres de las tablas deben llevar los mismos nombres de los archivos *.dbf*.
- Los atributos y tipos de datos de cada uno de los archivos planos de foxbase, deben ser los mismos en el nuevo modelo de base de datos. A excepción de los tipos de datos que no existan en la Bases de Datos a implementar. Por lo que debería tomar un tipo de dato común o similar al utilizado.
- Se implementaron relaciones entre algunas tablas del modelo de bases de datos para que la misma sea más consistente. Por ende se agregaron claves primarias dentro del mismo.
- Debido a que el sistema de caso de prueba es un Sistema Contable, el cual maneja la contabilidad anualmente, se ha decidido trabajar con dos esquemas diferentes (DOLA2008 y Prestaciones) tomando en cuenta los siguientes criterios:
 - *DOLAXXXX*: En este esquema se trabajará con la información contable y variable del año. Siendo estos los asientos, comprobantes, cuentas, modificación y eliminación de los mismos. Al finalizar el periodo contable (anual) se procederá a crear un nuevo esquema DOLAXXXX donde XXXX es el año consecuente al que se manejo la contabilidad.
 - *Prestaciones*: Aquí se manejará la información no contable como es de socios, perfiles, contraseñas, noticias, funcionarios, etc.

3.1.5.2.1 Esquema DOLA2008

En esta parte se verá el modelo de bases de datos que maneja el periodo contable dentro del sistema de caso de prueba (*Figura 26*).

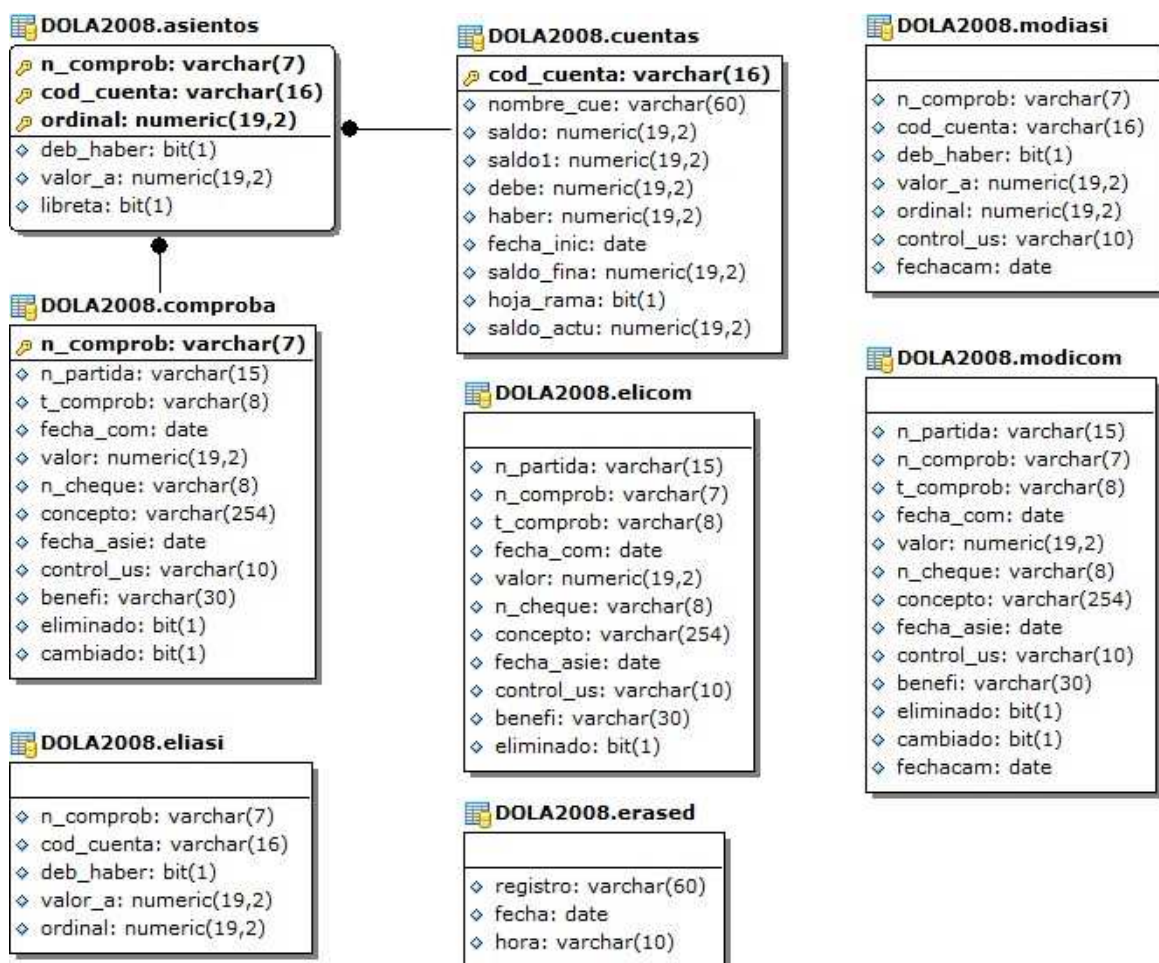


Figura 26. Esquema DOLA2008

3.1.5.2.2 Esquema Prestaciones

En este modelo se maneja la información no contable dentro del sistema de caso de prueba (Figura 27).

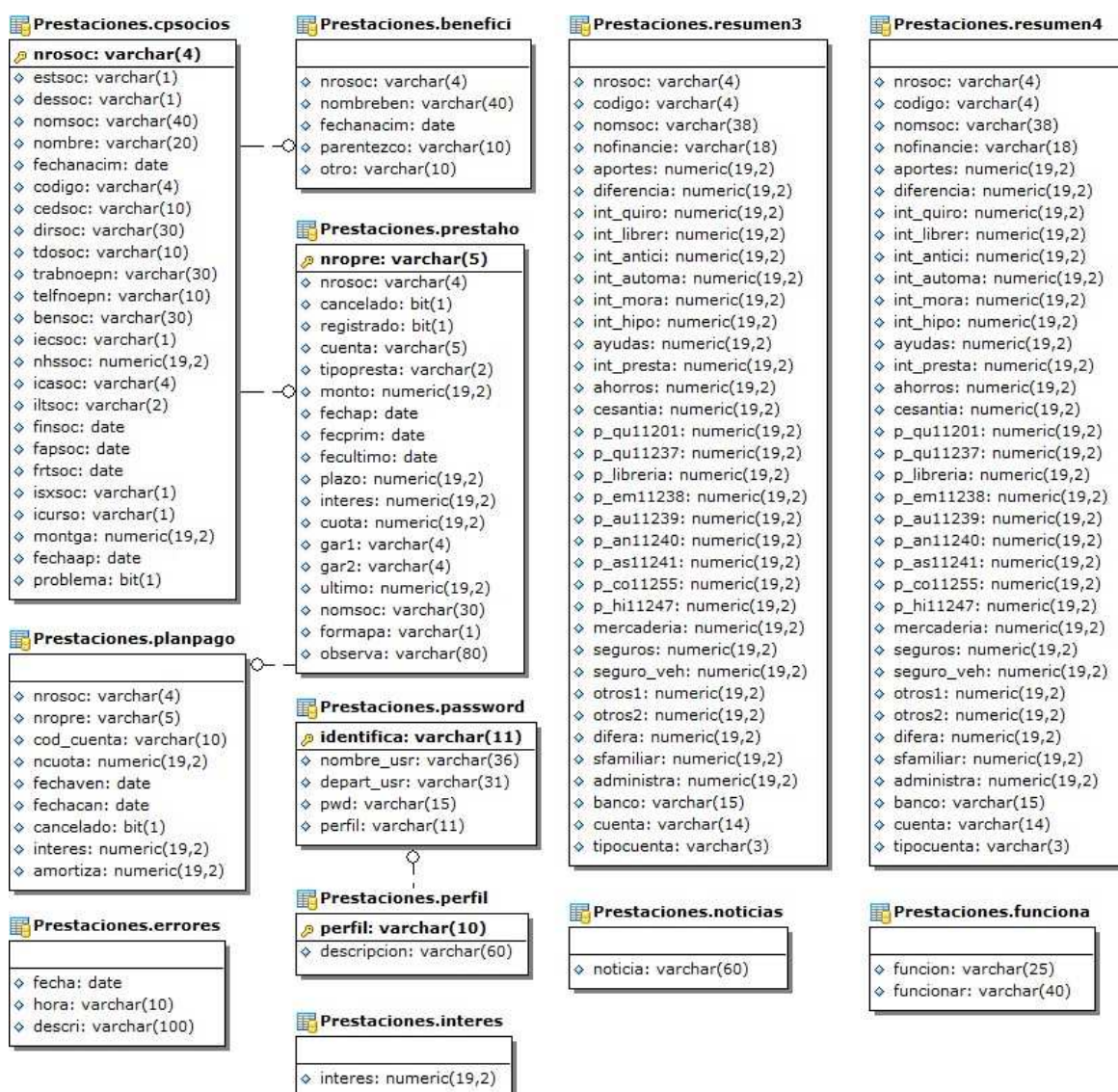


Figura 27. Esquema Prestaciones

3.1.6 INTERFAZ DEL SISTEMA DE CASO DE PRUEBA

Para realizar las interfaces del sistema de caso de prueba, se tomó como base las mismas del sistema original de la Cooperativa Politécnica de Ahorro y Crédito Ltda. debido a que es un proceso de migración y no debe existir mayor cambios del sistema anterior al sistema nuevo.

3.1.6.1 Acceso al Sistema

En la *Figura 28* y *29* se puede observar el acceso al Sistema de Caso de Prueba tanto del Sistema Antiguo como el nuevo.

Sistema Antiguo:



Figura 28. Pantalla de Acceso al Sistema (Sistema Antiguo)

Sistema Nuevo:



Figura 29. Pantalla de Acceso al Sistema (Sistema Nuevo)

3.1.6.2 Pantalla Principal

En la *Figura 30* y *31* podemos ver la pantalla principal del Sistema de la Cooperativa Politécnica, desde la cual se realizan todas las operaciones dentro del mismo.

Sistema Antiguo:



Figura 30. Pantalla Principal (Sistema Antiguo)

Sistema Nuevo:

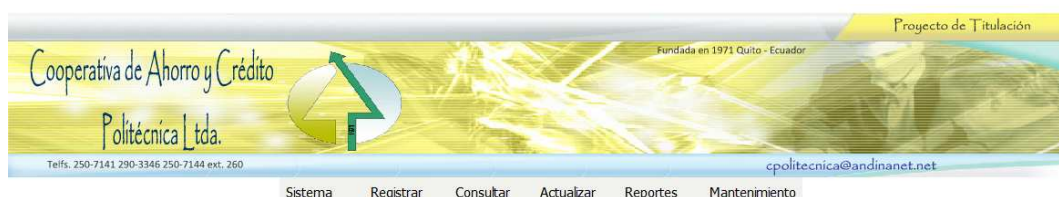


Figura 31. Pantalla Principal (Sistema Nuevo)

Las demás pantallas del sistema de caso de prueba y del sistema antiguo que se encuentra desarrollado en FoxPro 2.5 se mostrará en el *ANEXO A* dentro del *Manual de Usuario*; incluyendo ahí su funcionalidad y respuesta.

3.2 PRUEBAS DE ACOPLAMIENTO

La etapa de pruebas es una de las más significativas dentro del proceso de desarrollo de sistemas de software, ya que su objetivo es encontrar errores en las aplicaciones durante su desarrollo, lo cual juega un papel fundamental en el proceso de garantía de calidad del mismo. Esto es muy importante, porque cuanto antes se detecte un error, el costo de su reparación será menor. Es por esta razón que se hace imprescindible el empleo de pruebas sistemáticas, las cuales permiten encontrar al máximo número de errores con el menor esfuerzo posible. ^[16]

3.2.1 TIPOS DE PRUEBAS

En la orientación a objetos las pruebas que se realizan son estratégicamente similares a las pruebas de sistemas convencionales, pero tácticamente diferentes.

^[16] ROS, Nicolás Joaquín; “Métodos y Estrategias de Pruebas del Software”

Es por esto que se ha considerado a los principales tipos de pruebas:

3.2.1.1 Pruebas de Migración de Datos

Para poder cumplir con las pruebas de Migración de Datos, se ha decidido comprobar que los datos tanto antes como después de la migración son los mismos y siguen siendo consistentes con los tipos de datos.

Debido a la alta cantidad de información que posee la base de datos, se tomará al azar tres consultas a cada archivo y tabla de las bases de datos, comparando así los dos resultados; éstos se mostrarán de la *Figura 32* a la *Figura 39* y al final habrá a una descripción de los resultados obtenidos.

ESQUEMA *DOLA2008* y ESQUEMA *Prestaciones*

Resultado Esperado: Se espera que los Datos de los archivos *.dbf* (foxbase) sean iguales a los datos que se encuentran en las tablas del mismo nombre después del proceso de migración.

Prueba No. 1

Archivo asientos.dbf

Registros Existentes: 89259

ab	N_COMPROB	ab	COD_CUENTA	I	DEB_HABER	9.0	VALOR...	9.0	ORDINAL	I	LIBRETA
✓	B002296		1110100000		T		5001.10		2		F
✓	B002296		1110300400		F		5001.10		1		F
✓	B002297		1110100000		T		3255.09		2		F

Figura 32. Datos del archivo asientos.dbf

Tabla asientos

Registros Migrados: 89259

n_comprob	cod_cuenta	deb_haber	valor_a	ordinal	libr
B002296	1110100000	1	5001,1	2	0
B002296	1110300400	0	5001,1	1	0
B002297	1110100000	1	3255,09	2	0

Figura 33. Datos de la tabla asientos

Prueba No. 2

Archivo comproba.dbf

Registros Existentes: 6920

ab	ab	N...	ab	FEC...	l.0	at	ab	CONCEPTO	FEC...	ab	CONT...	ab	BE...	L	E...	L	C.
✓	B002296	...	02/01/2007	DEPOSITO BANCARIO NO. 5134110 PACIFICO	...	03/01/2007	MARIA FERN				
✓	B002297	...	03/01/2007	DEPOSITO BANCARIO NO. 4039310 PACIFICO	...	03/01/2007	MARIA FERN				
✓	B002298	...	02/01/2007	V. TAMAYO - EP. BANCARIO NO. 317324306 PA...	...	04/01/2007	MARIA FERN	VERONI...				T

Figura 34. Datos del archivo comproba.dbf

Tabla comproba

Registros Migrados: 6920

n_c	t	fecha_coi	v	n	concepto	fecha_asie	control_us	beni	elir	ca
B002296	02/01/2007	0	DEPOSITO BANCARIO NO. 5134110 PACIFICO	03/01/2007	MARIA FERN	0	0			
B002297	03/01/2007	0	DEPOSITO BANCARIO NO. 4039310 PACIFICO	03/01/2007	MARIA FERN	0	0			
B002298	02/01/2007	0	V. TAMAYO - EP. BANCARIO NO. 317324306 PAF	04/01/2007	MARIA FERN	VERON	0	1		

Figura 35. Datos de la tabla comproba

Prueba No. 3

Archivo cuentas.dbf

Registros Existentes: 25993

ab	COD_CU...	ab	NOMBRE_CUE	9.0	SAL...	9.0	SALD01	9.0	DEBE	9.0	HABER	FEC...	9.0	SAL...	I	9.0	SAL...
✓	1000000000	...	ACTIVOS	...	0.00	760734395.17	0.00	0.00	31/12/1997	0.00	F	0.00					
✓	1110000000	...	FONDOS DISPONIBLES	...	0.00	2936463.17	0.00	0.00	31/12/1997	0.00	F	0.00					
✓	1110100000	...	CAJA	...	0.00	7904537.00	5934937143.00	5942841680...	31/12/1998	0.00	T	0.00					

Figura 36. Datos del archivo cuentas.dbf

Tabla cuentas

Registros Migrados: 25993

cod_cuen	nombre_cue	saldo	saldo1	debe	haber	fecha_inic	s	h	s
1000000000	ACTIVOS	0	760734395.17	0	0	31/12/1997	0	0	0
1110000000	FONDOS DISPONIBLES	0	2936463.17	0	0	31/12/1997	0	0	0
1110100000	CAJA	0	7904537	5934937143	5942841680	31/12/1998	0	1	0

Figura 37. Datos de la tabla cuentas

Prueba No. 4

Archivo eliasi.dbf

Registros Existentes: 707

ab	N_COMPROB	ab	COD_CUENTA	L	DEB_HABER	9.0	VALOR_A	9.0	ORDINAL
✓	1064222		1110100000	F			60.00		1
✓	1064222		2110101364	T			4.00		2
✓	1064222		2113101364	T			4.00		3

Figura 38. Datos del archivo eliasi.dbf

Tabla eliasi

Registros Migrados: 707

n_comprob	cod_cuenta	deb_haber	valor_a	ordinal
1064222	1110100000	0	60	1
1064222	2110101364	1	4	2
1064222	2113101364	1	4	3

Figura 39. Datos de la tabla eliasi

Resultado Obtenido: Se puede observar que los Datos de los archivos *.dbf* efectivamente son iguales a los datos que se encuentran en las tablas del mismo nombre después del proceso de migración.

Observación: Proceso de Migración de Datos Exitoso

Las demás pruebas de migración de datos se mostrarán al final del presente documento en el *ANEXO B*. Los Archivo y tablas *password* y *perfil* no se tomaron en cuenta para realizar las pruebas de migración de datos, debido a que esta información es absolutamente confidencial y se encuentra encriptada.

3.2.1.2 Pruebas de Integración

Las pruebas de Integración son aquellas que se realizan en el ámbito del desarrollo de software una vez que se han aprobado las pruebas unitarias. Únicamente se refieren a la prueba o pruebas de todos los elementos unitarios que componen un proceso, hecha en conjunto, de una sola vez.

Consiste en realizar pruebas para verificar que un gran conjunto de partes de software funcionan juntos. Las *pruebas de integración* (algunas veces llamadas Integración y Testeo) es la fase del testeo de software en la cual módulos individuales de software son combinados y testeados como un grupo. Son las pruebas posteriores a las pruebas unitarias y preceden el testeo de sistema.^[24]

Para el cumplimiento de las pruebas de integración, se ha decidido comprobar que tanto los datos (Base de Datos) como el sistema (Caso de Prueba) funcionan correctamente y conjuntamente arrojan los mismos resultados que el sistema anterior.

Debido a que al caso de prueba es un sistema contable, éste posee muchos procesos dentro del mismo y la cantidad de información que maneja es muy alta; por lo que se tomará al azar tres procesos dentro mismo para compararlas así con el sistema antiguo. Los resultados se mostrarán de la *Figura 40* a la *Figura 46* junto a una descripción de los resultados obtenidos.

Prueba No. 1: Consulta de Comprobantes

^[24] WIKIPEDIA, La Enciclopedia Libre; Pruebas de Integración

08/10/2008 CONTABILIDAD - COOPERA 7:01:32 pm
 Sistema Registrar Consultar Actu
 COMPROBANTES ... Digite una tecla para ver las cuentas

NUMERO : B002298
 CHEQUE No :
 SUMA DE : \$0.00
 FECHA : 01/02/2007

BENEFICIARIO/DEPOSITANTE
 UERONICA TAMAYO

No PARTIDA : FECHA ASIENTO : 01/04/2007

CONCEPTO
 U. TAMAYO - EP. BANCARIO NO. 317324306 PARA CANCELACION
 24TA CUOTA P.Q. Y CUOTAS NORMALES ENE/2007

F1 < Otro > < Imprime > < Salir > ESC Salir F10 MENU
 Entre el número del comprobante

Figura 40. Consulta de Comprobantes (Ventana #1) Sistema Antiguo

08/10/2008 CONTABILIDAD - COOPERATIVA P 7:01:43 pm
 Digite una tecla para continuar

CODIGO CUENTA	NOMBRE CUENTA	DEBE	HABER
111.03.00700	BANCO PICHINCH	\$110.00	\$0.00
112.37.00487	TAMAYO TAMAYO	\$0.00	\$85.50
411.01.00100	INTERESES P. Q	\$0.00	\$9.50
211.01.01273	TAMAYO TAMAYO	\$0.00	\$4.00
211.31.01273	TAMAYO TAMAYO	\$0.00	\$4.00
221.03.00000	FONDO/AYUDAS E	\$0.00	\$5.00
421.01.00000	CUOTAS DE ADMI	\$0.00	\$2.00
		\$110.00	\$110.00

ESC Salir
 Entre el número del comprobante

Figura 41. Consulta de Comprobantes (Ventana #2) Sistema Antiguo

Resultado Esperado con el Sistema Nuevo: Se espera que el sistema de caso de prueba arroje los mismos resultados que el sistema anterior, en este caso en la consulta de comprobantes.

Número: B002298

Cheque No:

Beneficiario/Depositante:

Suma de: 0.00

VERONICA TAMAYO

Fecha: 2007-01-02

Código Cuenta	Nombre Cuenta	Debe	Haber
1110300700	BANCO PICHINCHA CTA. CTE.00559598-8	\$ 110.00	\$ 0.00
1123700487	TAMAYO TAMAYO VERONICA LILIANA	\$ 0.00	\$ 85.50
4110100100	INTERESES P. QUIROGRAFARIOS	\$ 0.00	\$ 9.50
2110101273	TAMAYO TAMAYO VERONICA LILIANA	\$ 0.00	\$ 4.00
2113101273	TAMAYO TAMAYO VERONICA LILIANA	\$ 0.00	\$ 4.00
2210300000	FONDO/AYUDAS EMERGENTES	\$ 0.00	\$ 5.00
4210100000	CUOTAS DE ADMINISTRACION	\$ 0.00	\$ 2.00
	Total:	\$ 110.00	\$ 110.00

Concepto

V. TAMAYO - EP. BANCARIO NO. 317324306 PARA
CANCELACION 24TA CUOTA P.Q. Y CUOTAS
NORMALES ENE/2007

Figura 42. Consulta de Comprobantes. Sistema Nuevo

Resultado Obtenido: El sistema de caso de prueba arroja los mismos resultados que el sistema anterior.

Observación: Existe buena integración de los datos con la aplicación respecto al modulo de consultas.

Prueba No. 2: Registro de Usuarios

Figura 43. Registro de Usuarios. Sistema Antiguo

Resultado Esperado con el Sistema Nuevo: Se espera que el sistema de caso de prueba registre usuarios de la misma manera como lo hacia el sistema anterior.

Ingreso de Usuarios

Identificación:	<input type="text" value="PRUEBA"/>
Nombre de Usuario:	<input type="text" value="PRUEBA"/>
Area de Trabajo:	<input type="text" value="SISTEMAS"/>
Perfil:	<input type="text" value="AUXILIAR"/>
Password:	<input type="password" value="••••••"/>
Confirme su password:	<input type="password" value="••••••"/>

[Nuevo Usuario](#)

[Salir](#)

Figura 44. Registro de Usuarios. Sistema Nuevo

Resultado Obtenido: Se ingreso correctamente y sin errores el nuevo usuario en ambos sistemas.

Observación: Existe buena integración de los datos con la aplicación respecto a la inserción de datos.

Prueba No. 3: Ahorros/Aportes/Navidad

AHORROS/APORTES/NAUIDAD		
NOMBRE DEL SOCIO: 0504 VILLAVICENCIO OTAÑEZ FRANCISCO	FECHA:	12/31/2007
SALDO DE AHORROS SIN ENCAJE:	\$1,379.22	
SALDO DE AHORROS EN ENCAJE:	\$0.00	
SALDO DE APORTE PARA NAUIDAD:	\$120.00	
AHORROS TOTALES Y NAUIDAD:	\$1,499.22	
SALDO DE CERTIFICADOS EMITIDOS:	\$532.00	
SALDO DE CERTIFICADOS POR EMITIR:	\$468.00	
CERTIFICADOS TOTALES :	\$1,000.00	
TOTAL AHORROS, NAUIDAD Y CERTIFICADOS:		\$2,499.22
SALDO DE SISTEMA AHORRO PLAN:		\$3,372.84
AHORROS, NAUIDAD, CERTIF. Y AHORRO PLAN:		\$5,872.06

F1 Ayuda < Imprimir > <Otro socio > < Salir > ESC Salir F10 MENU

Figura 45. Ahorros/Aportes/Navidad. Sistema Antiguo

Resultado Esperado con el Sistema Nuevo: Se espera que la información obtenida y calculada sea igual a la que desplegaría el sistema antiguo.

Ahorros/Aportes/Navidad

Nombre del Socio: 0504 VILLAVICENCIO OTAÑEZ FRANCISCO HORACIO

Fecha: 01/01/2008

Saldo de Ahorros sin Encaje:	1379.22	
Saldo de Ahorros en Encaje:	0	
Saldo de Aporte para Navidad:	120	
Ahorros Totales y Navidad:	1499.22	
Saldo de Certificados Emitidos:	532.00	
Saldo de Certificados por Emitir:	468.00	
Certificados Totales:	1000	
Total Ahorros, Navidad y Certificados:		2499.22
Saldo de Sistema Ahorro Plan:		3372.84
Ahorros, Navidad, Certificados y Ahorro Plan:		5872.06

Figura 46. Ahorros/Aportes/Navidad. Sistema Nuevo

Resultado Obtenido: La información desplegada y calculada es la misma que desplegó el sistema antiguo.

Observación: Existe buena integración de los datos con la aplicación respecto al cálculo y despliegue de valores en pantalla.

3.2.1.3 Pruebas del Sistema

Estas pruebas se basan en el acoplamiento de la aplicación con el resto de elementos del sistema como son el hardware, la información, los usuarios, etc. Este tipo de prueba para su mejor ejecución se la ha clasificado en los siguientes tipos a continuación detallados.

3.2.1.3.1 Pruebas de Recuperación

Las pruebas de recuperación permiten comprobar que el sistema se reinicie de manera correcta al instante de presentarse diferentes errores o circunstancias que trunquen su normal funcionamiento (*Tablas 5, 6 y 7*). Si la recuperación es automática (Llevada a cabo en el propio sistema) hay que evaluar la corrección de la inicialización, de los mecanismos de recuperación del estado del sistema, de la

recuperación de datos y del proceso de re arranque. Si la recuperación requiere la intervención humana, hay que evaluar los tiempos medios de reparación para determinar si están dentro de unos límites aceptables. ^[14]

PRUEBA No. 1	Creación y Actualización de la información de la Base de Datos en el caso de fallar el Suministro de Energía.
Resultado Esperado	La información no pudo guardarse
Resultado Obtenido	Perdida de la Información
Observación	La única información que no podrá recuperarse al presentarse una falla del suministro de energía será aquella que se haya ingresado antes de que se produjera el percance.

Tabla 5. Pruebas de Recuperación (Prueba No. 1)

PRUEBA No. 2	Consulta de la información de la base de datos al producirse un corte de energía eléctrica.
Resultado Esperado	La información consultada puede recuperarse
Resultado Obtenido	La información consultada es recuperada sin ningún problema
Observación	Si al producirse el corte de energía la operación que se estaba efectuando era de consulta, no existe ningún problema con la información de la base de datos; ya que esta se mantiene segura. Al restablecerse el fluido eléctrico la consulta puede nuevamente efectuarse sin ningún contratiempo.

Tabla 6. Pruebas de Recuperación (Prueba No. 2)

^[14] PRESSMAN, Roger; “Ingeniería del Software un Enfoque Práctico”

PRUEBA No. 3	Falla en el Suministro de Energía al momento de realizar borrado de información a la Base de Datos.
Resultado Esperado	La información no se pudo haberse borrado
Resultado Obtenido	La información no se borró
Observación	Si al producirse el corte de energía la operación que se estaba efectuando era de borrado de información, no se completa la misma, tampoco borra los registros anteriormente eliminados; sino que regresa al punto inicial donde empezó la transacción de borrado (<i>rollback</i>).

Tabla 7. Pruebas de Recuperación (Prueba No. 3)

3.2.1.3.2 Pruebas de Seguridad

Este tipo de pruebas tienen como meta el comprobar que los mecanismos de seguridad implantados en el sistema funcionen satisfactoriamente y lo protejan de cualquier acceso impropio. Durante este tipo de pruebas, el responsable de la prueba desempeña el papel de un individuo que desea entrar en el sistema.

Debe intentar conseguir las claves de acceso por cualquier medio, puede atacar al sistema con software a medida, diseñado para romper cualquier defensa que se haya construido.

En este caso de prueba del sistema de la Cooperativa Politécnica de Ahorro y Crédito Ltda. cuenta con mecanismos de seguridad a nivel de servidor, que controlan el acceso al sistema en base al perfil y clave de ingreso asignado a cada uno de los usuarios (responsables) del mismo.

Debido a que este sistema trabaja en un ambiente Web, la navegación entre páginas se encuentra completamente controlada y segura.

3.2.1.3.3 *Pruebas de Resistencia*

Las pruebas de resistencia o *endurance testing* permiten revisar que no hayan fugas de memoria (*memory leaks*) u otros problemas que pueden afectar el software cuando este va a correr por mucho tiempo, en otras palabras ejecuta un sistema de forma que demande recursos en cantidad, frecuencia o volúmenes anormales. Esencialmente el responsable de la prueba intenta romper el programa.^[2]

Para la ejecución de este tipo de pruebas se procedió a disminuir la memoria RAM del servidor y realiza la conexión con varios clientes del sistema, con lo que se obtuvo como resultado que su velocidad de operación fuese degenerándose cada vez más.

3.2.1.3.4 *Pruebas de Rendimiento*

A través de este tipo de pruebas se puede determinar el rendimiento del software en tiempo de ejecución dentro del contexto de un sistema integrado. La prueba de rendimiento se da durante todos los pasos del proceso de la prueba. Incluso al nivel de unidad, se debe asegurar el rendimiento de los módulos individuales a medida que se llevan a cabo las pruebas de caja blanca. Sin embargo, hasta que no están completamente integrados todos los elementos del sistema no se puede asegurar realmente el rendimiento del sistema.

El rendimiento del sistema de caso de prueba depende de ciertas circunstancias como son del hardware en el cual se encuentra instalado el servidor, debido que al contener este un volumen considerable de datos almacenados, la flexibilidad de la aplicación disminuye considerablemente por la lentitud con la que

^[2] ANDERSSON, Eve; “Los Tipos de Prueba”

se efectúan las búsquedas de información. Otro caso similar tendremos al instante en que los usuarios accedan al sistema, debido al gran número de conexiones el rendimiento del mismo se degradará rápidamente.

CAPÍTULO 4. CONCLUSIONES Y RECOMENDACIONES

4.1 CONCLUSIONES

- Hoy en día, la alta necesidad de dotar al negocio con nuevas funciones (en algunas ocasiones implementando aplicaciones nuevas o actualizando las existentes) es ahora más constante y acentuada que nunca. Por lo que la herramienta de migración de datos de Foxbase a un modelo de bases de datos relacional puede responder a esta progresiva demanda en una parte de este proceso, pero en si la más importante “Migración de los datos”.
- La selección del Modelo cascada para el desarrollo de la Herramienta de migración de datos fue apropiada debido a que solo existe un caso de uso.
- La Herramienta de Migración de Datos está orientada para usuarios que tengan conocimientos del Modelo Entidad Relación, y no para usuarios finales.
- No es posible migrar las relaciones de los archivos dado que estos datos permanecen encriptados en archivos con extensión CDX y si se pudiera abrir dichos archivos de relaciones sería muy difícil mantenerlas puesto que estos valores se toman de varias tablas y la forma de crear relaciones en Foxbase es distinta a los DBMS actuales. Es por esta razón que las relaciones de una base de Datos X que se desee migrar deberán ser creadas manualmente por el administrador del sistema.

4.2 RECOMENDACIONES

- Para la correcta ejecución de la Herramienta de Migración es necesario que se encuentre instalado .Net Framework 3.0.
- La herramienta de migración de datos debería sea utilizada no por usuarios finales, sino por una persona con conocimientos de Bases de Datos para que esta persona pueda implementar las relaciones de igual forma que en la base de datos origen.
- Antes de comenzar el proceso de migración se debe comprobar que los datos existentes en los archivos *.dbf* son correctos (fechas y valores) y que además no exista una tabla con el mismo nombre en la base de datos seleccionada para migrar los datos; puesto que podría generar errores al momento de empezar migración.
- Ubicar los archivos que se desea migrar en la raíz (C:, D:, E:, etc.), puesto que el ODBC de FoxBase no reconoce direcciones contrarias a las anteriormente indicadas y esto provocaría un error en la Herramienta de Migración de Datos.

BIBLIOGRAFÍA

- [1] AJPDSOFT; Tipos de Datos en Oracle.
<http://www.ajpdsoft.com/modules.php?name=News&file=article&sid=268> Abril 2007.
- [2] ANDERSSON, Eve; “Los Tipos de Prueba”,
<http://www.galileo.edu/wp/display/2832/2834.wimpy> Julio 2008.
- [3] ARIAS, Angélica; Guía de FOXBASE.
<http://www.angelfire.com/sc2/oliver1473/guiafox2.html> Octubre 1999.
- [4] BARRETEROS.NET; Características Introdutorias a C#.NET.
<http://www.smart-generator.com/cursos/page/Caracter%C3%ADsticas-Introdutorias-a-CNET-.aspx> Octubre 2007.
- [5] CUENCA, Carlos Luis; Tipos de Datos de MySQL.
<http://www.desarrolloweb.com/articulos/1054.php> Febrero 2007.
- [6] DACCACH, José Camilo. Artículos Delta; Seleccionando el Modelo de Desarrollo. <http://www.deltaasesores.com/prof/PRO351.html> Febrero 2007.
- [7] FULLER PADILLA, David; Roles en el Desarrollo de Software
www.eici.ucm.cl/Academicos/R_Villarroel/descargas/ing_sw_1/Roles_desarrollo_software.pdf Febrero 2003.

- [8] GENTEGEEK.COM; Software Libre y Software Propietario, ventajas y desventajas <http://www.gentegeek.com/sl-sp-ventajas-desventajas> Marzo 2006.
- [9] MARTÍNEZ, Luis. FOXPRESS, La Cabecera de los .DBF <http://www.fpress.com/revista/Num9906/Jun99.htm> Junio 1999.
- [10] MICROSOFT CORPORATION; Tipos de datos. <http://msdn2.microsoft.com/es-es/library/ms187752.aspx> Abril 2008.
- [11] MSDN; Estructura de archivos de tabla (.dbc, .dbf, .frx, .lbx, .mnx, .pjx, .scx, .vcx) http://msdn.microsoft.com/library/spa/default.asp?url=/library/SPA/fox7/help/html/contable_file_structure_lpdbfrp.asp Abril 2008.
- [12] OLEA, Ismael; Tipos de datos relevantes en PostgreSQL. <http://www.hispafuentes.com/hf-doc/temas/bd/notas-curso-BD/node134.html> Abril 2001.
- [13] POWER DATA; Migración de Datos <http://www.powerdataib.com/index.asp?idedicao=51&idSeccao=539&Action=seccao> Agosto 2007.
- [14] PRESSMAN, Roger; "Ingeniería del Software un Enfoque Práctico". Mc Graw Hill, 5ta Edición, Febrero 2002.
- [15] ROJAS, Richard; BOUCHECHTER, Israel. Ciclos de Vida de Ingeniería del Software. Enero 2005.

- [16] ROS, Nicolás Joaquín; "Métodos y Estrategias de Pruebas del Software" <http://www.um.es/giisw/docs/Prueba-sw.doc> Mayo 2008.
- [17] TREJO MARTÍNEZ, Janhil Aurora. MONOGRAFIAS.COM, Bases de Datos; <http://www.monografias.com/trabajos11/basda/basda.shtml> Abril 2008.
- [18] WIKIPEDIA, La Enciclopedia Libre; .NET. <http://es.wikipedia.org/wiki/.NET> Noviembre 2007.
- [19] WIKIPEDIA, La Enciclopedia Libre; JavaScript <http://es.wikipedia.org/wiki/Especial:Search?search=Java+Script&go=Ir> Julio 2008.
- [20] WIKIPEDIA, La Enciclopedia Libre; Modelo Entidad-Relación. http://es.wikipedia.org/wiki/Diagrama_entidad-relaci%C3%B3n Abril 2008.
- [21] WIKIPEDIA, La Enciclopedia Libre; Mozilla Firefox http://es.wikipedia.org/wiki/Mozilla_Firefox Julio 2008.
- [22] WIKIPEDIA, La Enciclopedia Libre; PHP <http://es.wikipedia.org/wiki/PHP> Julio 2008.
- [23] WIKIPEDIA, La Enciclopedia Libre; PostgreSQL <http://es.wikipedia.org/wiki/Postgres> Junio 2008.
- [24] WIKIPEDIA, La Enciclopedia Libre; Pruebas de Integración http://es.wikipedia.org/wiki/Pruebas_de_Integraci%C3%B3n Julio 2008.

- [25] WIKIPEDIA, La Enciclopedia Libre; Servidor HTTP Apache http://es.wikipedia.org/wiki/Servidor_HTTP_Apache Julio 2008.

ANEXOS

ANEXO A. Manual de Usuario del Sistema de Caso de Prueba

Este manual contiene la descripción de los procedimientos correspondientes al perfil de Autor, debido a que este posee acceso a todo el sistema y mediante el cual, podremos explicar cada uno de los componentes del mismo.

ACCESO AL SISTEMA



BIENVENIDO!!!

Servicio de Prestamos:

- Quirografario
- Automáticos Anticipos Librería
- Mercaderías
- Otros

Beneficios:

- Enfermedades
- Intervenciones Quirúrgicas
- Nacimientos
- Fallecimientos

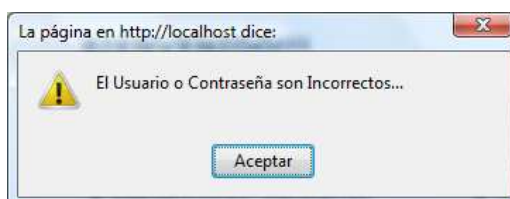
Captaciones:

- Ahorros
- Certificados
- Sistema Ahorro Plan

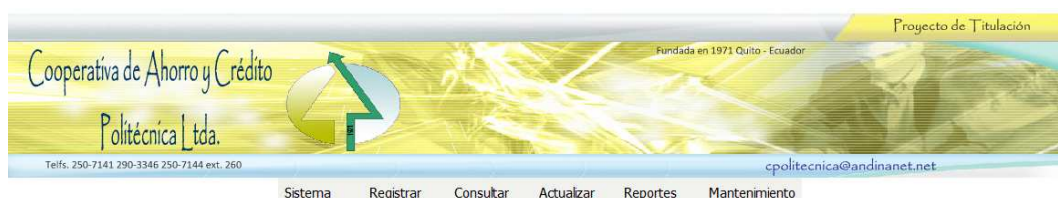
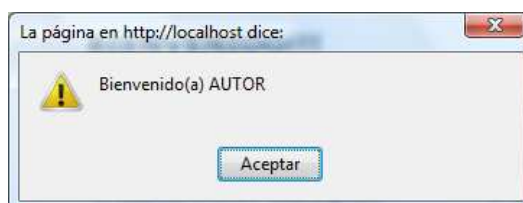
Contabilidad

Digite su Identificación:	FVILLAVI
Digite su clave:	*****
	Aceptar

Si ha digitado mal el usuario o la contraseña, el sistema muestra un mensaje de error al momento de ingresar al mismo.



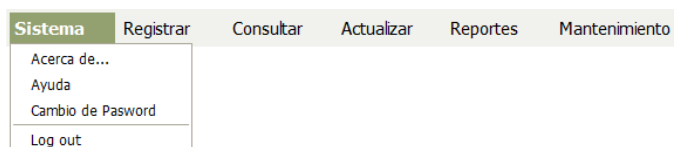
Caso contrario, el Sistema le da la Bienvenida y le muestra la pantalla de inicio del sistema con la cual el usuario puede interactuar según el perfil que posea.



A continuación detallaremos cada una de las partes del menú del sistema:

SISTEMA

Contiene información general del sistema



Acerca de...

Permite Visualizar los créditos del Sistema.

Ayuda

Despliega información de interés para el usuario.

Cambio de Password

Permite que el usuario realice el cambio de su contraseña. En el campo *Digite la Clave* debe digitar la nueva clave con la que desea ingresar al modulo de contabilidad. Su nueva clave debe contener únicamente letras o dígitos. Para asegurar que ingresó bien su nueva clave, debe volver a ingresarla en el campo *Repita la Clave*. Si estas son iguales se procederán al cambio de contraseña.

Log Out

Aquí el usuario podrá cerrar su sesión en el sistema.

REGISTRAR

Permite registrar nuevas cuentas, comprobantes, usuarios y perfiles.

Cuentas

Esta opción permite crear nuevas cuentas en el plan de cuentas del sistema. Solamente debe ingresar el código de la nueva cuenta y su nombre.

Registrar Cuenta

Código:	<input type="text" value="211.02.78660"/>
Nombre:	<input type="text" value="CUENTA DE PRUEBA"/>
<input type="button" value="Grabar"/>	

Comprobantes

Permite registrar la emisión de nuevos comprobantes.

Ingresos

Esta pantalla permite buscar un socio al cual se desea emitir el comprobante.

Registrar Comprobante Ingresos

Ingrese el nombre del depositante o beneficiario:

Beneficiario/Depositante:

Aquí se permite ingresar todos los datos para registrar la emisión de comprobantes de ingreso. Primeramente se debe ingresar el número del comprobante, el código del comprobante está codificado de la siguiente manera: I0001. Donde I indica que es un comprobante de ingreso (automáticamente colocado por el sistema), los restantes cuatro dígitos corresponden al número secuencial del comprobante. En caso de que el ingreso de dinero se realiza con un cheque, se debe ingresar el número del mismo. Seguidamente el valor del depósito y la fecha actual.

Registrar Comprobante Ingresos

Beneficiario/Depositante: VILLAVICENCIO OTAZEZ FRANCISCO HORACIO
Numero: I Socio: 0504
Cheque No:
Valor de:
Fecha: 01 ▾ 01 ▾ 2008

Antes de ingresar el concepto del comprobante, debe elija los rubros cuyas cuotas desea pagar y/o ingresar sus asientos por los códigos de cuenta, los valores en el debe y los valores en el haber.

Elija los rubros cuyas cuotas desea pagar:

Ahorros	4.00	<input type="checkbox"/>
Aportes	0.00	<input type="checkbox"/>
Ahorro Plan	34.00	<input type="checkbox"/>
Ayuda	7.00	<input type="checkbox"/>
Seguro	5.00	<input type="checkbox"/>
Navidad	10.00	<input type="checkbox"/>

Prestamo:

Prestamo	Tipo	Monto	Plazo	Vencimiento	Valor Cuota	Cancela	Cuotas
1123702547	QU	11500.00	96.00	2014-03-30	177.56	<input type="checkbox"/>	

Asientos:

Código Cuenta	Nombre Cuenta	Debe	Haber
111.01.00000	CAJA	0.00	0.00
		0.00	0.00
Total:			

Aumentar Asiento

Borrar Ultimo Asiento

Finalmente en el recuadro colocado en la parte inferior, usted debe colocar una breve descripción del motivo del depósito.

Concepto

Ok

El ingreso de los demás comprobantes se los realiza de forma similar a este comprobante de ingreso.

Usuarios

Aquí se puede crear nuevos usuarios al sistema. Para esto se debe ingresar su identificación la cual se usa para poder ingresar al sistema, el nombre completo

del usuario, el área de trabajo a la cual pertenece, el perfil que posee el nuevo usuario y su contraseña

Ingreso de Usuarios

Identificación:	<input type="text" value="PRUEBA"/>
Nombre de Usuario:	<input type="text" value="PRUEBA"/>
Area de Trabajo:	<input type="text" value="CONTABILIDAD"/>
Perfil:	<input type="text" value="CONTADOR"/>
Password:	<input type="password" value="•••••"/>
Confirme su password:	<input type="password" value="•••••"/>

Perfiles

Cada usuario posee un perfil el cual le permite navegar dentro del sistema según su rol dentro de la cooperativa. Aquí se puede crear un perfil diferente en caso de necesitarlo.

Ingreso de Perfiles

Nombre Perfil:	<input type="text" value="USUARIO"/>
Descripción:	<input type="text" value="USUARIOS PRUEBA"/>

CONSULTAR

Permite realizar consultas de información relacionada con: cuentas, comprobantes, certificados, aportes, usuarios, etc. Esta información podrá también ser imprimida.

Cuentas

Permite consultar información sobre una determinada cuenta, para ello es necesario conocer el código de la cuenta.

Consultar Cuentas

Código:

[Nueva Búsqueda](#)

[Salir](#)

En el campo *Código* deber ingresar el código o número de cuenta que desea consultar. Seguidamente usted obtiene como resultado el nombre de la cuenta y el saldo inicial del periodo contable.

Consultar Cuentas

Código: 211.01.00504

Nombre: VILLAVICENCIO OTANEZ FRANCISCO HORACIO

Saldo Inicial: 650.59

Comprobantes

Permite consultar información sobre un comprobante determinado. Para ello es necesario digitar en el campo, el código del comprobante.

Consultar Comprobantes

Número:

[Nueva Búsqueda](#)

[Salir](#)

Seguidamente se despliega toda la información del comprobante.

Consulta de Comprobantes

Número: B002296

Cheque No:

Beneficiario/Depositante:

Suma de: 0.00

Fecha: 2007-01-02

Código Cuenta	Nombre Cuenta	Debe	Haber
1110300400	BANCO DEL PACIFICO CTA.CTE. 441129-3 DOLARES	\$ 5001.10	\$ 0.00
1110100000	CAJA	\$ 0.00	\$ 5001.10
	Total:	\$ 5001.10	\$ 5001.10

Concepto

DEPOSITO BANCARIO NO. 5134110 PACIFICO

Saldo de Cuentas

Como su nombre lo indica, permite consultar los saldos finales de una cuenta, hasta una fecha determinada. Se debe indicar la fecha, a la cual se desea conocer el saldo final de la cuenta. El formato de fechas es: día/mes/año. Luego se debe ingresar el código de la cuenta.

Consultar Saldo de Cuentas

INDIQUE HASTA QUE FECHA ES EL SALDO

04 ▾ 06 ▾ 2008
Día Mes Año

Ingrese su Código:

Código:

En la siguiente pantalla se desplegará el saldo a la fecha anteriormente indicada.

Consultar Saldo de Cuentas

Código: 211.01.00504

Nombre: VILLAVICENCIO OTANEZ FRANCISCO HORACIO

Saldo Actual: 1379.22

Movimiento de Cuentas

Permite consultar los movimientos en el debe y el haber de una determinada cuenta. Para ello es necesario especificar un periodo de tiempo. En la primera pantalla que aparece al ingresar en esta opción, es necesario especificar el periodo en el cual se desea consultar el movimiento de una determinada cuenta. Junto al código de la cuenta de la cual se quiere obtener la información. El formato de las fechas es: día/mes/año.

Consultar Movimiento de Cuentas

Indique las fechas del proceso:

01 ▾	01 ▾	2008	04 ▾	03 ▾	2008
Día	Mes	Año	Día	Mes	Año

Ingrese su Código:

Código:

En la siguiente pantalla, muestra la información del movimiento de cuenta.

Consultar Movimiento de Cuentas

Código: 211.01.00504

Nombre: VILLAVICENCIO OTANEZ FRANCISCO HORACIO

Comprobante/Fecha/Concepto	Debe	Haber	Saldo
I064224 / 2007-01-08 / F. VILLAVICENCIO - CANCELA 9NA CUOTA P.Q. CTA. 30465908-04 CH. 901 PICHINCHA (403)	---	\$ 0.44	\$ 601.03
C40E207 / 2007-01-31 / REGISTRO DESCUENTOS SOLICITADOS A TESORERIA E.P.N. Y BANCOS PICHINCHA Y PACIFICO POR ENERO 2.007	---	\$ 4.00	\$ 605.03
H102ENE / 2007-01-23 / F. VILLAVICENCIO, RETIRO DE AHORROS	\$ 50.00	---	\$ 555.03
I064399 / 2007-02-01 / F. VILLAVICENCIO - CANCELA 10MA CUOTA P.Q. CTA. 30465908-04 CH. 909 PICHINCHA	---	\$ 0.44	\$ 555.47

Lista de Comprobantes

Permite consultar todos los comprobantes que se han emitido en un determinado periodo de tiempo, ya sea por ingresos o egresos. Para ello es necesario ingresar el periodo de tiempo en el cual se desea conocer el listado de comprobantes. El formato de fechas es: día/mes/año.

Consultar Lista de Comprobantes

Indique las fechas de consulta:

2008 2008
 Día Mes Año Día Mes Año

Seguidamente, se puede observar el listado de comprobantes

Reporte de Comprobantes

Número	Cheque	Fecha	Beneficiario	Eliminado	Funcionario	Valor
I064135		2007-01-02	1012 ELIZABETH VENEGAS SOLORZA	NO	MARIA FERN	218.96
I064136		2007-01-02	1107 MARCELO UZCATEGUI ANDRADE	NO	MARIA FERN	235.04
I064137		2007-01-02	0407 MARCOS RON SOASTI	NO	MARIA FERN	52.89
I064138		2007-01-02	1155 LOURDES SARZOSA VILLARROE	NO	MARIA FERN	240.61
I064139		2007-01-02	0871 NIDIA GUAYAQUIL JURADO	NO	MARIA FERN	89.06
H001ENE		2007-01-02	0138 NELSON DE LA	NO	ENRIQUE AR	44.70
I064149		2007-01-02	1068 CONSUELO SANCHEZ JACOME	NO	MARIA FERN	209.00
I064152		2007-01-02	0781 ROSA NAVARRETE RUEDA	NO	MARIA FERN	350.00
H002ENE		2007-01-02	0312 MARIA MONTERO MALDONADO	NO	ENRIQUE AR	30.00
I064160		2007-01-02	0559 JACINTO JIMENEZ SALAZAR	NO	MARIA FERN	523.16
I064140		2007-01-02	1350 CLAUDIA ZABALA YUQUILEMA	NO	MARIA FERN	130.00
I064141		2007-01-02	1411 TERESITA BURBANO IGLESIAS	NO	MARIA FERN	2850.00
I064142		2007-01-02	0850 PABLO RIVERA ARGOTI	NO	MARIA FERN	89.00
I064143		2007-01-02	1353 ALONSO GUALLI CAICHO	NO	MARIA FERN	115.00
I064144		2007-01-02	1438 JENNY ORELLANA OCAÑA	NO	MARIA FERN	4.47
I064145		2007-01-02	0523 SEGUNDO VILLEGAS CALISPA	NO	MARIA FERN	126.00
I064146		2007-01-02	1161 LUPE LOPEZ CHAVARREA	NO	MARIA FERN	350.00

Usuarios

Permite consultar a todos los usuarios del sistema.

Consultar Usuarios

Usuario	Ubicación	Perfil
ING. VILLAVICENCIO, FRANCISCO	SISTEMAS - EPN	AUTOR
MARIA FERNANDA CAADAS	RECAUDADOR	RECAUDADOR
LOURDES SARZOSA	CONTABILIDAD	PAGADOR
JUAN MERIZALDE	GERENCIA	GERENTE
CONTADOR	CONTABILIDAD	CONTADOR
RECAUDADOR	RECAUDACION	RECAUDADOR
PAGADOR	PAGADURIA	PAGADOR
AUTOR	RECAUDACION	AUTOR
MARIA JOSE	SECRETARIA	AUXILIAR
AUDITOR	AUDITORIA	GERENTE
TAMARA VASCONEZ	SECRETARIA	AUXILIAR

Perfiles

Aquí se puede consultar todos los perfiles que existen en el sistema.

Consultar Perfiles

Perfiles
AUTOR
CONTADOR
RECAUDADOR
GERENTE
PAGADOR
AUXILIAR
SISTEMAS

Certificados y Aportes

Permite consultar la lista de certificados de aportación emitidos hasta una fecha determinada. Primeramente se debe ingresar la fecha en la que se desea emitir el informe.

Consultar Certificados y Aportes

INGRESE LA FECHA DEL INFORME

01 ▼ 05 ▼ 2008
Día Mes Año

Consultar

Luego de hacer clic en el botón *Consultar*, se desplegarán los resultados de los certificados de aportación emitidos.

Certificados/Aportación Emitidos al 01/05/2008

Socio	Cta. 31101	Cta. 21131	Total
ABAD LEON WILSON ARIOSTO	310.00	690.00	1000
ACOSTA ESTRADA MARCELO GERMAN	323.00	689.00	1012
ACOSTA GALLO ERWIN ELISEO	421.00	363.00	784
ACOSTA HURTADO TANIA ALEYDA	806.00	194.00	1000
ACUNA GORDILLO WASHINGTON OSWALDO	529.00	471.00	1000
AGAMA ARIAS GERMAN GEOVANNY	374.00	523.00	897
AGUILAR BARRERA SUSANA DE LOS ANGELES	518.00	482.00	1000
AGUINAGA BARRAGAN ALVARO GONZALO	157.00	743.00	900
AGUIRRE CABRERA MARIA ELENA	227.00	272.50	499.5
AMACAYA CHUQUILLA WALTER OSWALDO	0.00	87.00	87
ALBAN RUIZ LUIS ALFREDO	135.00	865.00	1000
ALBUJA TORRES JORGE MARCELO	521.00	373.00	894
ALBUJA VITERI LUIS HUMBERTO	135.00	130.00	265
ALDAS PALACIOS HERNAN OSWALDO	279.00	130.00	409
ALMEIDA CERVANTES LUIS HOMERO	812.00	188.00	1000
ALMEIDA ESTRELLA MARIA CRISTINA	294.00	706.00	1000
ALMEIDA PAVON MARCELO GUSTAVO	135.00	865.00	1000
ALMEIDA RUEDA MAURICIO ANTONIO	0.00	4.00	4
ALMEIDA SONIA AZUCENA	535.00	130.00	665
ALMEIDA TERAN GERARDO GILBERTO	520.00	150.00	670

Lista de Saldo de Subgrupo

Esta opción le permite listar los saldos de las cuentas de subgrupo. Para ello debe ingresar la fecha a la cual desea emitir el reporte y el código de cuenta de grupo.

Consultar Lista de Saldos de Subgrupo

Ingrese la fecha del Informe:

01 ▾ 06 ▾ 2008
Día Mes Año

Ingrese la Cuenta de Mayor:

211 . 01 .00000

Consultar

A continuación muestra los resultados en pantalla.

Lista de Saldos de Subgrupo

No	Código	Cuenta	Saldo
1	2110101255	ABAD LEON WILSON ARIOSTO	188.08
2	2110100001	ACOSTA ESTRADA MARCELO GERMAN	4.71
3	2110100002	ACOSTA GALLO ERWIN ELISEO	37.54
4	2110101119	ACOSTA HURTADO TANIA ALEYDA	268.77
5	2110100003	ACUNA GORDILLO WASHINGTON OSWALDO	1448.98
6	2110101203	AGAMA ARIAS GERMAN GEOVANNY	128.54
7	2110100753	AGUILAR BARRERA SUSANA DE LOS ANGELES	9.54
8	2110101333	AGUILAR JARAMILLO JORGE ARTURO	18.52
9	2110100783	AGUINAGA BARRAGAN ALVARO GONZALO	358.8
10	2110100007	AGUIRRE CABRERA MARIA ELENA	16.09
11	2110101485	AGUIRRE FONSECA VINICIO ANTONIO	43
12	2110101405	AIMACAÑA CHUQUILLA WALTER OSWALDO	147.26
13	2110100827	ALBAN RUIZ LUIS ALFREDO	302.88
14	2110100010	ALBUJA TORRES JORGE MARCELO	12.57
15	2110100592	ALBUJA VITERI LUIS HUMBERTO	39.2
16	2110100011	ALDAS PALACIOS HERNAN OSWALDO	693.42
17	2110101490	ALDAZ DÓAZ GLADYS MARLENE	40
18	2110100719	ALMEIDA CERVANTES LUIS HOMERO	111.96
19	2110101230	ALMEIDA ESTRELLA MARIA CRISTINA	217.45
20	2110100862	ALMEIDA PAVON MARCELO GUSTAVO	952.42
21	2110101462	ALMEIDA RUEDA MAURICIO ANTONIO	0.41

Saldos de Subgrupo

Esta opción permite consultar el saldo totalizado de una cuenta de grupo. Para conocer el saldo final de una cuenta de grupo, primeramente debe ingresar la fecha hasta la cual desea conocer el informe, seguida del código de cuenta de grupo.

Consultar Saldos de Subgrupo

INGRESE LA FECHA DEL INFORME

03 ▾ 05 ▾ 2008
 Día Mes Año

INGRESE LA CUENTA DE MAYOR

211 ▾ . 01 ▾ .00000

El informe de saldos de subgrupo es el siguiente:

Cuentas del Subgrupo al 03/05/2008

Código	Cuentas	Total Subgrupo
211.01.0000	AHORROS NORMALES	298734.01

Ahorros / Aportes / Navidad

Esta opción permite conocer los saldos de: ahorros, certificados y el sistema ahorra plan. Al ingresar a esta opción primero debe buscar al socio del cual desea consultar el saldo, esto se puede hacer ingresando las primeras letras del apellido del mismo.

Consultar Ahorros/Aportes/Navidad

Ingrese el nombre del socio:

Socio:

[Otro Socio](#)[Salir](#)

Si existe más de un resultado, se debe buscar el socio del cual queremos la información, pero antes se debe indicar la fecha a la cual desea conocer los saldos de ahorro, certificados y el sistema ahorro plan.

Consultar Ahorros/Aportes/Navidad

Indique hasta que fecha es el saldo

2008
 Día Mes Año

Seleccione el nro del socio:

Nro Socio	Nombre Socio
0504	VILLAVICENCIO OTAÑEZ FRANCISCO HORACIO
1364	VILLAVICENCIO RUIZ PAUL DAVID

Otro Socio

Salir

El reporte sería el siguiente:

Ahorros/Aportes/Navidad

Nombre del Socio: 0504 VILLAVICENCIO OTAÑEZ FRANCISCO HORACIO

Fecha: 01/01/2008

Saldo de Ahorros sin Encaje:	1379.22	
Saldo de Ahorros en Encaje:	0	
Saldo de Aporte para Navidad:	120	
Ahorros Totales y Navidad:	1499.22	
Saldo de Certificados Emitidos:	532.00	
Saldo de Certificados por Emitir:	468.00	
Certificados Totales:	1000	
Total Ahorros, Navidad y Certificados:		2499.22
Saldo de Sistema Ahorro Plan:		3372.84
Ahorros, Navidad, Certificados y Ahorro Plan:		5872.06

Estados de Cuentas

Esta le permite conocer el estado de cuenta entre un rango de fechas. Al ingresar a esta opción, usted debe ingresar las fechas entre las cuales desea

conocer el estado de cuentas. Junto al código de la cuenta de la cual desea obtener la información de estado de cuentas.

Consultar Estado de Cuentas

Indique la fechas del proceso:

2008 2008
 Día Mes Año Día Mes Año

Ingrese su Código:

Código:

El estado de cuenta que desplegará es el siguiente:

Estado de Cuentas

Código: 211.01.00504

Nombre: VILLAVICENCIO OTANEZ FRANCISCO HORACIO

Comprobante	Fecha	Debe	Haber	Saldo
I064224	2007-01-08	---	\$ 0.44	\$ 601.03
C40E207	2007-01-31	---	\$ 4.00	\$ 605.03
H102ENE	2007-01-23	\$ 50.00	---	\$ 555.03
I064399	2007-02-01	---	\$ 0.44	\$ 555.47
C10F207	2007-02-09	---	\$ 64.22	\$ 619.69
C40F207	2007-02-28	---	\$ 4.00	\$ 623.69
H293FEB	2007-02-27	\$ 50.00	---	\$ 573.69
C32F207	2007-02-28	---	\$ 12.16	\$ 585.85
H011MAR	2007-03-01	\$ 50.00	---	\$ 535.85
I064593	2007-03-01	---	\$ 112.60	\$ 648.45
I064677	2007-03-07	---	\$ 0.44	\$ 648.89
C40M207	2007-03-30	---	\$ 4.00	\$ 652.89
H161MAR	2007-03-28	\$ 50.00	---	\$ 602.89
I064819	2007-03-30	---	\$ 212.60	\$ 815.49
I064858	2007-04-03	---	\$ 0.44	\$ 815.93
H018ABR	2007-04-04	\$ 50.00	---	\$ 765.93

Comprobantes Extensos

Permite consultar toda la información de un comprobante extenso. Por lo que se debe ingresar primero el código de un comprobante extenso.

Consultar Comprobantes Extensos

Número:

[Nueva Búsqueda](#)

[Salir](#)

El resultado de la consulta, sería la siguiente:

Consulta de Comprobantes Extensos

Número: C57S207

Cheque No:

Beneficiario/Depositante:

Suma de: 5226.50

Fecha: 2007-09-30

Código Cuenta	Nombre Cuenta	Debe	Haber
2112201445	ROSA ABARCA JUAN CARLOS	\$ 25.00	\$ 0.00
2112201332	MORALES JOSE GUILLERMO	\$ 19.40	\$ 0.00
2112200974	ESPIN MIGUEL ANGEL	\$ 19.40	\$ 0.00
2112201447	CLAVIJO GUASCO NELLY GERMANIA	\$ 19.40	\$ 0.00
2112201005	ARGUELLO PORTILLA J ENRIQUE	\$ 19.40	\$ 0.00
2112201300	TACO CAÑIZARES MONICA ELIZABETH	\$ 18.50	\$ 0.00
2112201298	CAÑIZARES PROAÑO SONIA GLADYS	\$ 18.50	\$ 0.00

2112200827	ALBAN RUIZ LUIS ALFREDO	\$ 5.00	\$ 0.00
2112201485	AGUIRRE FONSECA VINICIO ANTONIO	\$ 5.00	\$ 0.00
2112200007	AGUIRRE CABRERA MARIA ELENA	\$ 5.00	\$ 0.00
2112201203	AGAMA ARIAS GERMAN GEOVANNY	\$ 5.00	\$ 0.00
2112200003	ACUNA GORDILLO WASHINGTON OSWALDO	\$ 5.00	\$ 0.00
2112200001	ACOSTA ESTRADA MARCELO GERMAN	\$ 5.00	\$ 0.00
2121800000	SEGUROS DEL PICHINCHA	\$ 0.00	\$ 5226.50
Total:		\$ 5226.50	\$ 5226.50

Concepto

REGISTRO TRANSFERENCIA DE SEGURO DE VIDA
SOCIOS A SEGUROS DEL PICHINCHA POR
SEPTIEMBRE 2.007

Prestamos Vigentes

Le permite conocer los préstamos que tiene un determinado socio. Al ingresar a esta opción primero debe buscar al socio del cual desea consultar si posee o no préstamos, esto se puede hacer ingresando las primeras letras del apellido del mismo.

Consultar Préstamos Vigentes

Ingrese el nombre del socio:

Socio:

[Nueva Consulta](#)

[Salir](#)

Luego, si existe más de un resultado, se debe buscar el socio del cual queremos la información de préstamos.

Consultar Prestamos Vigentes

Seleccione el nro de socio:

Nro.Socio	Nombre Socio
0504	VILLAVICENCIO OTAÑEZ FRANCISCO HORACIO
1364	VILLAVICENCIO RUIZ PAUL DAVID

[Nueva Consulta](#)

[Salir](#)

Para así proceder a mostrar la información de préstamos del socio previamente escogido.

Prestamos Vigentes de un Socio

Sr: VILLAVICENCIO OTAÑEZ FRANCISCO

Prestamo	Tipo	Monto	Plazo	Vencimiento
1123702547	QU	11500.00	96.00	2014-03-30

ACTUALIZAR

Esta opción le permite actualizar cualquier tipo de información que usted desee en el sistema. Puede realizar cambios en: cuentas, comprobantes, usuarios y perfiles.

Cuentas

Aquí usted podrá cambiar el nombre de una cuenta. Al ingresar en esta opción, usted debe ingresar un código de la cuenta.

Actualizar Cuenta

Ingrese el código de la cuenta que desea actualizar:

Código:

Inmediatamente, usted podrá realizar los cambios en el nombre de la cuenta.

Actualizar Cuenta

Código: 211.01.00504

Nombre:

Comprobantes

Esta opción le permite modificar los datos de un comprobante. En la pantalla que se muestra a continuación debe ingresar el código del comprobante que desea modificar.

Actualizar Comprobante

Ingrese el código del comprobante que desea modificar:

Nro Comprobante :

Inmediatamente, usted puede modificar datos como: número de cheque, valor del comprobante, nombre del beneficiario, concepto y los asientos que este posea.

Cambiar de Número a Comprobantes

En caso de que haya cometido un error al ingresar un número de comprobante, usted puede hacer uso de esta opción para modificar el número de un comprobante. Primero debe buscar el comprobante que ingreso mal.

Cambiar de Número a Comprobante

Seleccione el tipo de comprobante e ingrese el número de comprobante que desea modificar:

Número de Comprobante :

En la pantalla que se despliega en esta opción usted debe ingresar el nuevo número de comprobante correcto.

Cambiar Número de Comprobante

Número Comprobante: B002296
 Nuevo número:
 Fecha: 2007-01-02
 Valor: 0.00
 Concepto: DEPOSITO BANCARIO NO. 5134110 PACIFICO

Usuarios

En esta opción, se puede cambiar la información de un usuario del sistema previamente seleccionado.

Actualizar Usuarios

Seleccione usuario que desea modificar:

Identificacion	Nombre Usuario
FRANCISCO	ING. VILLAVICENCIO, FRANCISCO
MARIFER	MARIA FERNANDA CAADAS
LOURDES	LOURDES SARZOSA
JUANJOSE	JUAN MERIZALDE
CONTADOR	CONTADOR
RECAUDADOR	RECAUDADOR
PAGADOR	PAGADOR
FVILLAVI	AUTOR
MAJOSE	MARIA JOSE
AUDITOR	AUDITOR

La información que puede ser libre de cambio en esta parte del menú de actualización, es el Departamento al que pertenece, el Perfil que posee y su contraseña.

Actualizar Usuarios

Identificación:	JUANJOSE
Usuario:	JUAN MERIZALDE
Departamento:	<input type="text" value="GERENCIA"/>
Perfil:	<input type="text" value="GERENTE"/>
Contraseña:	<input type="password" value="••••••••"/>
Repita la Contraseña	<input type="password" value="••••••••"/>
<input type="button" value="Guardar"/>	

Perfiles

Aquí se puede cambiar la descripción de un perfil existente en el sistema, el cuál debe ser previamente seleccionado de la lista que se despliega en pantalla.

Actualizar Perfiles

Seleccione el perfil que desea modificar:

Perfil
AUTOR
CONTADOR
RECAUDADOR
GERENTE
PAGADOR
AUXILIAR
SISTEMAS

La pantalla que se muestra para poder actualizar el perfil es la siguiente:

Actualizar Perfiles

Perfil: AUXILIAR
Descripcion:

REPORTES

Esta opción del menú principal, le permite conocer toda la información contable de la Cooperativa. Usted podrá generar un archivo *.pdf* de los reportes a generar o visualizar: Planes de cuenta, Auxiliares, Balances de todo tipo, Libros Diarios, entre otros. También se podrá imprimir los mismos.

Plan de Cuentas

Como su nombre lo indica, esta opción permite listar las cuentas contables que la Cooperativa utiliza.

Resumido

Esta opción lista los códigos y nombres de cuentas, exceptuando las de subgrupo.

Plan de Cuentas Resumido

Codigo Cuenta	Nombre
1000000000	ACTIVOS
1110000000	FONDOS DISPONIBLES
1110100000	CAJA
1110200000	CAJA CHICA
1110300000	BANCOS-CUENTAS CORRIENTES
1110500000	ENTREGA AYUDAS Y RETIRO AHORRO
1120000000	EXIGIBLE - PRESTAMOS
1120100000	PRESTAMOS COMPUTACION
1120200000	PRESTAMOS EMERGENTES 42%
1120300000	ORDENES DE MERCADERIA
1120500000	ANTICIPO
1120600000	CUENTAS POR COBRAR TESORERIA (EPN)
1120700000	DIFERENCIAS EN DESCUENTOS ROL
1120800000	CHEQUES PROTESTADOS Y COMISIONES
1120900000	ANTICIPOS A SUELDO
1121100000	REVOCATORIA DE CHEQUES
1121200000	LIQUIDACIONES EX SOCIOS
1121300000	CUENTAS POR LIQUIDAR
1121400000	CIRCULOS PRESTAMOS (CUOTAS)
1121500000	EMPLEADOS
1121700000	SEGURO VEHICULOS
1121800000	PRESTAMOS CIRCULOS
1121900000	CARTERA VENCIDA

Extenso

Aquí se lista los códigos y nombres de cuentas, incluyendo las cuentas de subgrupo.

Plan de Cuentas Extenso

Codigo Cuenta	Nombre
1000000000	ACTIVOS
1110000000	FONDOS DISPONIBLES
1110100000	CAJA
1110200000	CAJA CHICA
1110300000	BANCOS-CUENTAS CORRIENTES
1110300400	BANCO DEL PACIFICO CTA.CTE. 441129-3 DOLARES
1110300700	BANCO PICHINCHA CTA. CTE.00559598-8
1110500000	ENTREGA AYUDAS Y RETIRO AHORRO
1120000000	EXIGIBLE - PRESTAMOS
1120100000	PRESTAMOS COMPUTACION
1120100408	HERNANDEZ ALVAREZ MYRIAN BEATRIZ
1120100943	MONAR MONAR RICARDO ALFONSO
1120103048	ORBE MORILLO FREDDIE ERNESTO
1120103427	VELARDE GUEVARA JAIME EDISON
1120107192	GALARZA BRITO MARTHA TERESA
1120107257	HERNANDEZ ALVAREZ MYRIAN BEATRIZ
1120107321	MONAR MONAR RICARDO ALFONSO
1120161594	ABAD LEON WILSON ARIOSTO
1120161595	BRITO BALDEON AMADA ELENA
1120161596	LOPEZ PERUGACHI ROCIO DE LA GIOCONDA
1120200000	PRESTAMOS EMERGENTES 42%
1120300000	ORDENES DE MERCADERIA
1120500000	ANTICIPO

Mayor

Con esta opción usted puede visualizar el libro mayor, entre un rango de fechas. La página que se despliega inicialmente en su pantalla, es la que se muestra a continuación. En ella usted debe ingresar la fecha inicial y final del periodo contable (día/mes/año).

Reporte Mayor

Indique las Fechas de Consulta

2008 2008
 Día Mes Año Día Mes Año

Html

Pdf

Como se puede ver, este reporte puede ser generado en *HTML* o *.pdf* y si se necesita, posteriormente podrá ser impreso. El reporte sería el siguiente:

Reporte Mayor

Cuenta:		Nombre:		Saldo	
1110100000		CAJA			
Comprobante / Fecha /Concepto	Debe	Haber	Deudor	Acreedor	
1064135 / 2007-01-02 / E. VENEGAS - CANCELTA 46TA CUOTA (DIC) 5000\$ I Y 14TA CUOTA P.Q. ***	218.96		218.96		
1064135 / 2007-01-02 / E. VENEGAS - CANCELTA 46TA CUOTA (DIC) 5000\$ I Y 14TA CUOTA P.Q. ***	235.04		454		
1064135 / 2007-01-02 / E. VENEGAS - CANCELTA 46TA CUOTA (DIC) 5000\$ I Y 14TA CUOTA P.Q. ***	89.06		543.06		
1064135 / 2007-01-02 / E. VENEGAS - CANCELTA 46TA CUOTA (DIC) 5000\$ I Y 14TA CUOTA P.Q. ***	130.00		673.06		
1064135 / 2007-01-02 / E. VENEGAS - CANCELTA 46TA CUOTA (DIC) 5000\$ I Y 14TA CUOTA P.Q. ***	330.00		1003.06		
1064135 / 2007-01-02 / E. VENEGAS - CANCELTA 46TA CUOTA (DIC) 5000\$ I Y 14TA CUOTA P.Q. ***	2850.00		3853.06		
1064135 / 2007-01-02 / E. VENEGAS - CANCELTA 46TA CUOTA (DIC) 5000\$ I Y 14TA CUOTA P.Q. ***	70.00		3923.06		
1064135 / 2007-01-02 / E. VENEGAS - CANCELTA 46TA CUOTA (DIC) 5000\$ I Y 14TA CUOTA P.Q. ***	89.00		4012.06		
1064135 / 2007-01-02 / E. VENEGAS - CANCELTA 46TA CUOTA (DIC) 5000\$ I Y 14TA CUOTA P.Q. ***	52.89		4064.95		

Auxiliares

Con esta opción usted puede visualizar el movimiento de una determinada cuenta, entre un rango de fechas. La página que se despliega inicialmente en su pantalla, es la que se muestra a continuación. En ella usted debe ingresar la fecha inicial y final del periodo contable (día/mes/año), junto al código de cuenta de la cual se desea sacar el reporte.

Reporte Auxiliar

Indique las fechas del proceso:

2008 2008
 Día Mes Año Día Mes Año

Ingrese su Código:

Código:

El reporte generado sería el siguiente:

Reporte Auxiliar

Código: 211.01.00504

Nombre: VILLAVICENCIO OTANEZ FRANCISCO HORACIO

Comprobante/Fecha/Concepto	Debe	Haber	Saldo
H293FEB / 2007-02-27 / F. VILLAVICENCIO, RETIRO DE AHORROS	\$ 50.00		\$ 604.59
C32F207 / 2007-02-28 / VARIOS TRANSFERENCIA DE FONDO DE AYUDAS A AHORROS NORMALES		\$ 12.16	\$ 616.75
H011MAR / 2007-03-01 / F. VILLAVICENCIO, RETIRO DE AHORROS	\$ 50.00		\$ 566.75
I064593 / 2007-03-01 / F. VILLAVICENCIO - ABONA A SUS AHORROS CTA. 3047869004 CH. 8500PICHINCHA (212.60)		\$ 112.60	\$ 679.35
I064677 / 2007-03-07 / F. VILLAVICENCIO - CANCELA 11RA CUOTA P.Q. CTA. 30465908-04 CH. 917 PICHINCHA (238)		\$ 0.44	\$ 679.79
C40M207 / 2007-03-30 / REGISTRO DESCUENTOS SOLICITADOS A TESORERIA E.P.N. Y BANCOS PICHINCHA Y PACIFICO POR MARZO 2.007		\$ 4.00	\$ 683.79
H161MAR / 2007-03-28 / F. VILLAVICENCIO, RETIRO DE AHORROS	\$ 50.00		\$ 633.79
I064819 / 2007-03-30 / F. VILLAVICENCIO - ABONA A SUS AHORROS CTA. 3047869004 CH. 8521 PICHINCHA		\$ 212.60	\$ 846.39

Diario entre Fechas

Con esta opción usted puede visualizar el diario, entre un rango de fechas. La página que se despliega inicialmente en su pantalla, es la que se muestra a continuación. En ella usted debe ingresar la fecha inicial y final del periodo contable (día/mes año). Además debe escoger el tipo de comprobantes, del cual desea conocer el DIARIO.

- I** Comprobantes de Ingreso
- E** Comprobantes de Egreso
- C** Comprobantes de Asientos de Diario
- H** Comprobantes de Ayudas y Ahorros
- R** Distribución de Intereses Sistema Ahorro Plan
- B** Comprobantes de Caja

Reporte Diario Entre Fechas

Indique las fechas de consulta:

2008 2008
 Día Mes Año Día Mes Año

Seleccione el tipo de comprobante que desea:

I E C H R B

El reporte generado sería el siguiente:

		Reporte Diario Entre Fechas			
		Comprobates / Cuentas		Debe	Haber
Fecha					
2007-01-03	E020786	1110300700	BANCO PICHINCHA CTA. CTE.00559598-8 JOSE GUALICHICO .- AYUDA POR ASISTENCIA MEDICA FAMILIAR.		184.07
2007-01-03	E020787	2121200000	CONTRIBUCION SOLCA		7.00
		2210300000	FONDO/AYUDAS EMERGENTES		7.00
		4210100000	CUOTAS DE ADMINISTRACION		7.00
		1110300700	BANCO PICHINCHA CTA. CTE.00559598-8 ALBA RUALES.- ANTICIPO A SUELDO A 12 MESES PLAZO.		679.00
2007-01-03	E020788	1110300700	BANCO PICHINCHA CTA. CTE.00559598-8 NESTOR SALAZAR.- AYUDA POR ASISTENCIA MEDICA FAMILIAR		539.78
2007-01-04	E020789	1110300700	BANCO PICHINCHA CTA. CTE.00559598-8 SEGUROS UNIDOS.- ABONMO A FACTURA 001-002-0075877		3500.00

Balance de Situación Inicial

Esta opción le permite conocer el Balance de Situación Inicial a una determinada fecha. Existen dos tipos de balance de situación inicial, el resumido y el extenso.

Indique la fecha de Consulta

01 ▼ 01 ▼ 2008
 Día Mes Año

Resumido

Aquí se puede listar los movimientos de las cuentas de un grupo.

Reporte Balance Situacion Inicial Resumido

al 01 / 01 / 2008

1000000000	ACTIVOS			3602814.06
1110000000	FONDOS DISPONIBLES		16578.03	
1110200000	CAJA CHICA	200.00		
1110300000	BANCOS-CUENTAS CORRIENTES	15623.03		
1110500000	ENTREGA AYUDAS Y RETIRO AHORRO	755.00		
1120000000	EXIGIBLE - PRESTAMOS		3575813.35	
1120100000	PRESTAMOS COMPUTACION	1214.13		
1120300000	ORDENES DE MERCADERIA	10.00		
1120500000	ANTICIPO	32474.26		
1121500000	EMPLEADOS	228.92		
1122100000	LIQUIDACION EX SOCIOS	511.76		
1122800000	CIRCULO 5000 USD. I	1918.77		
1122900000	CIRCULO 5.000 USD. II	2892459.69		
1123900000	PRESTAMOS AUTOMATICOS	186906.88		
1124500000	GRUPO VITA ESPECIAL (60'000.000)	100		
1124700000	PRESTAMOS HIPOTECARIOS / PRENDARIOS	454435.48		
1124800000	ANTICIPO IMPUESTO A LA RENTA	39.26		
1125800000	PRESTAMO ADICIONAL LIBRERIA	727.01		
1125900000	DIFERENCIAS FINANCIERO/BANCOS	4787.19		
1220000000	INVERSIONES FINANCIERAS		71.31	
1220100000	ACCIONES FECOAC	71.31		

Extenso

En esta opción, se puede listar los movimientos de las cuentas de grupo y subgrupo.

Reporte Balance Situacion Inicial Extenso

al 01/01/2008

1000000000	ACTIVOS				3602814.06
1110000000	FONDOS DISPONIBLES			16578.03	
1110200000	CAJA CHICA		200.00		
1110300000	BANCOS-CUENTAS CORRIENTES		15623.03		
1110300400	BANCO DEL PACIFICO CTA.CTE. 441129-3 DOLARES	17569.76			
1110300700	BANCO PICHINCHA CTA. CTE.00559598-8	-1946.73			
1110500000	ENTREGA AYUDAS Y RETIRO AHORRO		755.00		
1120000000	EXIGIBLE - PRESTAMOS			3575813.35	
1120100000	PRESTAMOS COMPUTACION		1214.13		
1120103048	ORBE MORILLO FREDDIE ERNESTO	1214.13			
1120300000	ORDENES DE MERCADERIA		10.00		
1120500000	ANTICIPO		32474.26		
1121500000	EMPLEADOS		228.92		
1122100000	LIQUIDACION EX SOCIOS		511.76		
1122100057	BONES RUMBEA FRANKLIM CLEMENTE	17.04			
1122100091	CARRION ARGUELLO JOSE FRANCISCO	15.00			
1122100147	DUEÑAS CARRION EDMUNDO	47.98			
1122100224	HERDOIZA POSSO LUIS FREDERICK	10.41			
1122100289	MARROQUIN LAYEDRA FAUSTO MANUEL	57.82			
1122100759	VERA LEON RAMON HUMBERTO	18.00			
1122100901	LEON GALINDO JEANET ELIANA	63.10			

Balance de Situación

Para conocer el Balance de Situación, usted debe ingresar la fecha a la cual usted desea el reporte (día/mes/año). Existen dos tipos de Balance de Situación, el resumido y el extenso.

Reporte Balance Situacion Resumido

Indique las Fechas de Consulta

2008
 Día Mes Año

Resumido

Esta opción lista los movimientos de las cuentas de un grupo.

Reporte Balance Situacion Resumido

al 06 / 09 / 2008

1000000000	ACTIVOS			4182211.5
1110000000	FONDOS DISPONIBLES		17028.07	
1110100000	CAJA	7364.45		
1110200000	CAJA CHICA	200.00		
1110300000	BANCOS-CUENTAS CORRIENTES	8713.62		
1110500000	ENTREGA AYUDAS Y RETIRO AHORRO	750		
1120000000	EXIGIBLE - PRESTAMOS		4159424.28	
1120100000	PRESTAMOS COMPUTACION	471.88		
1120600000	CUENTAS POR COBRAR TESORERIA (EPN)	165738.99		
1121500000	EMPLEADOS	2353.63		
1121700000	SEGURO VEHICULOS	-39.76		
1122400000	CTAS POR COBRAR ANTICIPO SOCIOS	2000		
1123700000	PRESTAMO QUIROGRAFARIO ORDINARIO	3380192.4		
1123900000	PRESTAMOS AUTOMATICOS	30206.47		
1124000000	ANTICIPOS ESPECIALES	12934.5		
1124100000	CTAS. POR COBRAR ANTICIPO SUELDO SOCIOS	104597.28		
1124700000	PRESTAMOS HIPOTECARIOS / PRENDARIOS	445317.29		
1125700000	RIFAS	-3865		
1125800000	PRESTAMO ADICIONAL LIBRERIA	1141.42		

Extenso

Aquí se puede listar los movimientos de las cuentas de grupo y subgrupo.

Reporte Balance Situacion Extenso

al 06 / 09 / 2008

1000000000	ACTIVOS			4182211.5
1110000000	FONDOS DISPONIBLES		17028.07	
1110100000	CAJA	7364.45		
1110200000	CAJA CHICA	200.00		
1110300000	BANCOS-CUENTAS CORRIENTES	8713.62		
1110300400	BANCO DEL PACIFICO CTA.CTE. 441129-3 DOLARES	-13629		
1110300700	BANCO PICHINCHA CTA. CTE.00559598-8	22342.62		
1110500000	ENTREGA AYUDAS Y RETIRO AHORRO	750		
1120000000	EXIGIBLE - PRESTAMOS		4159424.28	
1120100000	PRESTAMOS COMPUTACION	471.88		
1120103427	VELARDE GUEVARA JAIME EDISON	471.88		
1120600000	CUENTAS POR COBRAR TESORERIA (EPN)	165738.99		
1121500000	EMPLEADOS	2353.63		
1121700000	SEGURO VEHICULOS	-39.76		
1121701026	MERIZALDE RIVADENEIRA JUAN JOSE	-39.76		
1122400000	CTAS POR COBRAR ANTICIPO SOCIOS	2000		
1122400556	LOZA CEDEÑO JORGE EDUARDO	2000		
1123700000	PRESTAMO QUIROGRAFARIO ORDINARIO	3380192.4		
1123700010	AGUINAGA BARRAGAN ALVARO GONZALO	1188.03		
1123700028	ALVAREZ PACHECO CLAUDIO EUGENIO	312.01		

Estado de Rendimientos

El Estado de Rendimientos le permitirá conocer si al final de un periodo contable existe utilidad o pérdida. Al ingresar en esta opción usted debe ingresar la fecha a la cual desea conocer el Estado de Rendimientos (día/mes/año). Generalmente esta acción se realiza al finalizar el periodo contable.

Reporte Estado Rendimientos

Indique la fechas de Consulta

15 ▾ 07 ▾ 2008
Día Mes Año

El Reporte de Estados de Rendimientos para este caso sería el siguiente:

Reporte Estado Rendimientos

al 15 / 07 / 2008

4000000000	INGRESOS				416966.37
4110000000	POR PRESTAMOS			377577.6	
4110100000	INTERESES RECIBIDOS		377577.6		
4110100100	INTERESES P. QUIROGRAFARIOS	317863.45			
4110100200	INTERESES P. LIBRERIAS	76.62			
4110100300	INTERESES PRESTAMO HIPOTECARIO	39505.38			
4110100400	INTERESES ANTICIPOS ESPECIALES	11242.42			
4110101100	INTERESES MORA	2375.55			
4110101200	INTERESES CIRCULOS 5.000 USD.	103.55			
4110101300	INTERESES PRESTAMOS AUTOMATICOS	6410.63			
4210000000	DIRECTOS			39388.77	
4210100000	CUOTAS DE ADMINISTRACION		32366.46		
4210300000	INGRESOS VARIOS		4649.33		
4210600000	EXCEDENTE RIFAS		2372.98		
Total Ingresos:					416966.37
5000000000	GASTOS				344553.25
5110000000	DEL PERSONAL			51335.64	
5110100000	SUELDOS		34854.1		
5110400000	BENEFICIOS SOCIALES		8536.58		
5110400100	APORTES PATRONALES, SECAP, IECE	3245.91			

Movimiento de Cuenta a Mayor

Esta opción permite desplegar el movimiento de las cuentas de subgrupo. Al ingresar en esta opción, debe ingresar la fecha inicial y final del periodo contable

(día/mes/año), junto al código de cuenta de subgrupo del cual se desea sacar el movimiento de cuentas.

Reporte Movimiento de Cuenta Mayor

Indique las fechas del proceso:

2008 2008
 Día Mes Año Día Mes Año

Ingrese su el codigo de la cuenta de mayor:

Código: .00000

El reporte generado sería el siguiente:

Reporte Movimiento Cuenta de Mayor

Código: 2110100000
 Nombre: AHORROS NORMALES
 Saldo Inicial: 256033.89

Comprobante/Fecha/Concepto	Debe	Haber	Saldo
I064136 / 2007-01-02 / M. UZCATEGUI - CANCELA 14TA CUOTA P.Q. Y 46A CUOTA (DIC) 5000\$ II CTA. 30363437-04 CH. 1274 PICHINCHA (454)		0.64	256034.53
I064139 / 2007-01-02 / N. GUAYAQUIL - ABONA A SUS AHORROS CTA. 30472318-04 CH. 444399 PICHINCHA		89.06	256123.59
I064152 / 2007-01-02 / R. NAVARRETE - ABONA A SUS AHORROS CTA. 30478690-04 CH. 8080 PICHINCHA		350.00	256473.59
I064143 / 2007-01-02 / A. GUALLI - CANCELA 10MA CUOTA P.Q. Y 10MA CUOTA P.Q. EFECTIVO		4.60	256478.19
I064153 / 2007-01-02 / M. CASA - CANCELA CUOTAS NORMALES ENE/2007 Y ABONA A SUS AHORROS EFECTIVO		254.00	256732.19
I064154 / 2007-01-02 / E. ESPIN - CANCELA TOTALIDAD DE ANTICIPO Y CUOTAS NORMALES DESDE AGO/2006 HASTA ENE/2007 EFECTIVO (PARTE CON BONO NAVIDAD)		24.97	256757.16
I064144 / 2007-01-02 / J. OELLANA - DISTRIBUCION DE EXCEDENTES 2005 EFECTIVO		4.47	256761.63
I064155 / 2007-01-02 / E. PAREDES - CANCELA 9NA CUOTA P.Q. EFECTIVO		0.48	256762.11

Balance de Comprobación

Para conocer el Balance de Comprobación, usted debe ingresar las fechas inicial y final del periodo contable (día/mes año).

Reporte Balance de Comprobación

Indique el período del Balance:

2008 2008
 Día Mes Año Día Mes Año

Inmediatamente, se despliega en pantalla el reporte de Balance de Comprobación, el cual tiene la siguiente estructura:

Reporte Balance de Comprobación

al 01 / 01 / 2008

Codigo Cuenta	Nombre Cuenta	Saldos al 01 / 01 / 2008		Movimientos		Saldos al 28 / 02 / 2008	
		Deudor	Acreedor	Deudor	Acreedor	Deudor	Acreedor
1110100000	CAJA		0	942503.94	935139.49	7364.45	
1110200000	CAJA CHICA	200.00		0	0	200	
1110300000	BANCOS-CUENTAS CORRIENTES	15623.03		2249975.24	2256884.65	8713.62	
1110500000	ENTREGA AYUDAS Y RETIRO AHORRO	755.00		88020.87	88025.87	750	
1120100000	PRESTAMOS COMPUTACION	1214.13		1366.4	2108.65	471.88	
1120300000	ORDENES DE MERCADERIA	10.00		0	10		
1120500000	ANTICIPO	32474.26		35175.74	67650		
1120600000	CUENTAS POR COBRAR TESORERIA (EPN)		0	1792428.14	1626689.15	165738.99	
1120800000	CHEQUES PROTESTADOS Y COMISIONES		0	5404.01	5404.01		
1121100000	REVOCATORIA DE CHEQUES		0	172.5	172.5		
1121500000	EMPLEADOS	228.92		19720.79	17596.08	2353.63	
1121700000	SEGURO VEHICULOS		0	269.86	309.62		39.76
1122100000	LIQUIDACION EX SOCIOS	511.76		0	511.76		
1122400000	CTAS POR COBRAR ANTICIPO SOCIOS		0	79600	77600	2000	
1122800000	CIRCULO 5000 USD. I	1918.77		0	1918.77		
1122900000	CIRCULO 5.000 USD. II	9840		0	9840		

MANTENIMIENTO

En esta sección usted podrá realizar funciones administrativas como son: Respaldos de la Base de Datos, así como también su restauración; Información de Funcionarios y Pasar saldos de las cuentas al nuevo periodo contable.

Respaldos

Realiza respaldos de toda la información del sistema y restauración a la Base de Datos en caso de existir algún problema con la misma. Esta actividad se debe

realizar mientras los usuarios no estén haciendo uso del sistema. Esta es una tarea que debe ser realizada por el administrador del sistema.

Funcionarios

Con esta opción usted puede consultar los funcionarios y los cargos de las personas que trabajan en la cooperativa. También permite modificar el nombre de funcionario en caso de existir algún error en el mismo.

Mantenimiento Funcionarios

Funcionario	Funcion
Ing. Carlos Valladares.	CONSEJO DE VIGILANCIA
Ec. Guillermo Morales	CONTADOR
Sr. Juan Merizalde	GERENTE
Ing. Jaime Calderón S.	PRESIDENTE
Sr. Enrique Argello P.	SECRETARIO

Guardar

Pasar Saldo

Esta opción le permite pasar saldos desde el balance de situación final al nuevo balance de situación. Esta operación se realiza anualmente, al cierre del periodo contable, el 31 de Diciembre de cada año.

Para ejecutar este proceso, ningún usuario debe estar haciendo uso del sistema. Al ingresar a esta opción, debe ingresar la fecha de cierre del periodo contable (día/mes/año).

Mantenimiento Pasar Saldo

Para correr este proceso se necesita que no exista ningun otro usuario se encuentre conectado

Indique hasta que fecha es el Saldo

31 ▾	12 ▾	2008
Día	Mes	Año

A continuación debe hacer clic en *Pasar Saldo*. Inmediatamente el sistema pasará los saldos de cada una de las cuentas al nuevo periodo contable.

ANEXO B. Pruebas de Migración de Datos

Resultado Esperado: Se espera que los Datos de los archivos *.dbf* (foxbase) sean iguales a los datos que se encuentran en las tablas del mismo nombre después del proceso de migración.

Prueba No. 5

Archivo elicom.dbf

Registros Existentes: 149

ab	N	ab	N_COM...	ab	T	FECHA_COM	9.0	VALOR	ab	N...	ab	CONCEPTO	FEC...	ab	CONT...	ab	BENEFI	I
✓	...	I064222				08/01/2007		60.00				P. VILLAVICENCIO - CANCELA CUOT...	08/01/2007	MARIA FER		1364 PAUL VILLAVICENCIO RUIZ	T	
✓	...	E020828				10/01/2007		133.00	003555			ESTHER NAVEDA - PAGO INTERES ...	10/01/2007	LOURDES SA		1058 DELIA NAVEDA SALAZAR	T	
✓	...	I064243				10/01/2007		336.82				V. ALVARADO - CANCELA 4/5/6/7M...	10/01/2007	MARIA FER		1101 CROKS ALVARADO OÑATE	T	

Tabla elicom

Registros Migrados: 149

n	n_co	t_cr	fecha_cc	valor	n_ch	concepto	fecha_asie	control_u	benefi	elir
	I064222		08/01/2007	60		P. VILLAVICENCIO - CANCELA CUOT	08/01/2007	MARIA FER	1364 PAUL VILLAVICENCIO RUIZ	1
	E020828		10/01/2007	133	003555	ESTHER NAVEDA - PAGO INTERES	10/01/2007	LOURDES SA	1058 DELIA NAVEDA SALAZAR	1
	I064243		10/01/2007	336.82		V. ALVARADO - CANCELA 4/5/6/7M...	10/01/2007	MARIA FER	1101 CROKS ALVARADO OÑATE	1

Prueba No. 6

Archivo erased.dbf

Registros Existentes: 619

ab	REGISTRO	FECHA	HORA
✓	C06E206 ASI 2110100753	05/01/2006	11:55:04
✓	C06E206 COM AGUILAR SUSANA 01/04/2006	05/01/2006	11:55:05
✓	I060834 ASI 1110100000	09/01/2006	11:23:11

Tabla erased

Registros Migrados: 619

registro	fecha	hora
C06E206 ASI 2110100753	05/01/2006	11:55:04
C06E206 COM AGUILAR SUSANA 01/04/2006	05/01/2006	11:55:05
I060834 ASI 1110100000	09/01/2006	11:23:11

Prueba No. 7

Archivo modiasi.dbf

Registros Existentes: 27717

ab	N_COMPROB	ab	COD_CUENTA	I	DEB_HABER	9.0	VALOR_A	9.0	ORDINAL	ab	CONTROL_US	FECHACAM
✓	C5YX99		1120600000	F			490000.00		1	LESLIE LOP		11/08/1999
✓	C5YX99		1123205622	T			154498.00		2	LESLIE LOP		11/08/1999
✓	C5YX99		1120400000	T			335147.00		3	LESLIE LOP		11/08/1999

Tabla modiasi

Registros Migrados: 27717

n_ct	cod_cuenta	de	valor_a	ordinal	control_us	fechacam
C5YX99	1120600000	0	490000	1	LESLIE LOP	11/08/1999
C5YX99	1123205622	1	154498	2	LESLIE LOP	11/08/1999
C5YX99	1120400000	1	335147	3	LESLIE LOP	11/08/1999

Prueba No. 8

Archivo modicom.dbf

Registros Existentes: 6919

ab	N.	ab	N_C...	ab	FEC...	9.0	VALOR	ab	ab	CONCEPTO	FEC...	ab	CONT...	ab	BENEFI	I	I	C.	FEC...
✓	...	B000001	23/12/2000	0.00	...	H.	CAPA - DEPOSITO BANCARIO NO. 27...	03/01/2000	MARIA FERN	HOLGER CAPA	...	T	04/01/2000				
✓	...	B000001	01/03/2000	0.00	...	H.	CAPA - DEPOSITO BANCARIO NO. 27...	04/01/2000	MARIA FERN	HOLGER CAPA	...	T	08/01/2000				
✓	...	B000001	03/01/2001	0.00	...	L.	CASTRO - DEPOSITO BANCARIO NO. 5...	03/01/2001	MARIA FERN	LUIS CASTRO S...	...	F	03/01/2001				

Tabla modicom

Registros Migrados: 6919

n	n	Δ	t	fecha_co	valor	n_che	concepto	fecha_asi	control_u	benefi	elir	ca	fechacam
B000001				03/01/2000	0		H. CAPA - DEPOSITO BANCARIO NO. 27	03/01/2000	MARIA FER	HOLGER CAPA	0	0	03/01/2000
B000001				23/12/2000	0		H. CAPA - DEPOSITO BANCARIO NO. 27	03/01/2000	MARIA FER	HOLGER CAPA	0	1	04/01/2000
B000001				03/01/2001	0		L. CASTRO - DEPOSITO BANCARIO NO.	03/01/2001	MARIA FER	LUIS CASTRO SAL	0	0	03/01/2001

Prueba No. 9

Archivo benefici.dbf

Registros Existentes: 1545

ab	NROSOC	ab	NOMBREBEN	FEC...	FECHANACIM	ab	PARENTEZCO	ab	OTRO
✓	1238		DIEGO LANDAZURI	...	23/12/1985		HERMANO		
✓	1238		MALDONADO MARTHA	...	18/06/2000		MADRE		
✓	1238		LANDAZURI JOSE	...	13/04/2000		PADRE		

Tabla benefici

Registros Migrados: 1545

n	nombreben	fechanacim	parentez	otro
1238	DIEGO LANDAZURI	23/12/1985	HERMANO	
1238	MALDONADO MARTHA	18/06/2000	MADRE	
1238	LANDAZURI JOSE	13/04/2000	PADRE	

Prueba No. 10

Archivo cpsocios.dbf

Registros Existentes: 1289

ab	ab	ab	ab	NOMSOC	ab	NOMBRE	FEC...	ab	ab	CED...	ab	DIRSOC	ab	T...
✓	A	T	1255	ABAD LEON WILSON ARIOSTO	...	WILSON ARIOSTO...	27/05/1952			1703125870		DE LOS PERALES		256804
✓	P	T	0703	ABATA CHICAIZA CELIO		CELIO	/ /			873 1701190017		RIO CENEPa 363 CDLA. AMAZ...		515442
✓	P	T	0809	ACERO LOACHAMIN RAUL ALFREDO...		RAUL ALFREDO ...	13/01/1958			647 1705678819		SAMBIZA		

Tabla cpsocios

Registros Migrados: 1289

	nro:	nomsoc	nombre	fechanac	c	cedsoc	dirsoc	tdoso	
A	T	1255	ABAD LEON WILSON ARIOSTO	WILSON ARIOSTO	27/05/1952		1703125870	DE LOS PERALES	256804
P	T	0703	ABATA CHICAIZA CELIO	CELIO	Null		873 1701190017	RIO CENEPa 363 CDLA. AMAZ	515442
P	T	0809	ACERO LOACHAMIN RAUL ALFREDO	RAUL ALFREDO	13/01/1958	647	1705678819	SAMBIZA	

Prueba No. 11

Archivo errores.dbf

Registros Existentes: 566

	FECHA	ab	HORA	ab	DESCRI
✓	18/11/2004		14:09:13		1 File 'PRNABORT' does not exist. File 'PRNABORT' does not exist. 109 EXISTE
✓	30/11/2004		09:56:49		1 File 'salida1' does not exist. File 'salida1' does not exist. 237 BALANCE
✓	30/11/2004		09:56:49		12 Variable 'COD_CUENTA' not found. Variable 'COD_CUENTA' not found. 238 BALANCE

Tabla errores

Registros Migrados: 566

fecha	hora	descri
18/11/2004	14:09:13	1 File 'PRNABORT' does not exist. File 'PRNABORT' does not exist. 109 EXISTE
30/11/2004	09:56:49	1 File 'salida1' does not exist. File 'salida1' does not exist. 237 BALANCE
30/11/2004	09:56:49	12 Variable 'COD_CUENTA' not found. Variable 'COD_CUENTA' not found. 238 BALANCE

Prueba No. 12

Archivo funciona.dbf

Registros Existentes: 5

ab	FUNCION	ab	FUNCIONAR
✓	GERENTE		Sr.Juan Merizalde
✓	PRESIDENTE		Ing. Jaime Calderón S.
✓	CONTADOR		Ec. Guillermo Morales

Tabla funciona

Registros Migrados: 5

funcion	funcionar
GERENTE	Sr.Juan Merizalde
PRESIDENTE	Ing. Jaime Calderçn S.
CONTADOR	Ec. Guillermo Morales

Prueba No. 13

Archivo interes.dbf

Registros Existentes: 1

9.0	INTERES
✓	10.50

Tabla interes

Registros Migrados: 1

interes	▼
	10,5

Prueba No. 14

Archivo noticias.dbf

Registros Existentes: 4

ab	NOTICIA
✓	SE ESTÁN OTORGANDO LOS PRESTAMOS HIPOTECARIOS
✓	CONSULTE LOS REQUISITOS EN SECRETARIA (SR. ENRIQUE ARGUELLO)
✓	HABRA PREFERENCIAS PARA VIVIENDAS: CONSTRUCCION Y ADECUACION

Tabla noticias

Registros Migrados: 4

noticia	▼
	SE ESTÁN OTORGANDO LOS PRESTAMOS HIPOTECARIOS
	CONSULTE LOS REQUISITOS EN SECRETARIA (SR. ENRIQUE ARGUELLO)
	HABRA PREFERENCIAS PARA VIVIENDAS: CONSTRUCCION Y ADECUACION

Prueba No. 15

Archivo planpago.dbf

Registros Existentes: 29172

ab	NROSOC	ab	NROPRE	ab	COD...	9.0	FECHAV...	FECHAC...	CANC...	9.0	INT...	9.0	AMOR...
✓	0783	00010	1123700010	35	30/11/2007	30/11/2007	T			12.64	93.03		
✓	0783	00010	1123700010	36	31/12/2007	/ /	F			12.11	93.56		
✓	0783	00010	1123700010	37	31/01/2008	/ /	F			11.15	94.52		

Tabla planpago.dbf

Registros Existentes: 29172

nr	nrc	cod_cuen	ncuota	fechaven	fechacan	ca	interes	amortiza
0783	00010	1123700010	35	30/11/2007	30/11/2007	1	12.64	93.03
0783	00010	1123700010	36	31/12/2007	Null	0	12.11	93.56
0783	00010	1123700010	37	31/01/2008	Null	0	11.15	94.52

Resultado Obtenido: Se puede observar que los Datos de los archivos *.dbf* efectivamente son iguales a los datos que se encuentran en las tablas del mismo nombre después del proceso de migración.

Observación: Proceso de Migración de Datos Exitoso