ANEXO 8:

CREDISYS:

MANUAL DE MANTENIMIENTO

VIERNES, 05 DE DICIEMBRE DE 2008

EL SALVADOR, CENTRO AMÉRICA

INTRODUCCIÓN

I. ESTRUCTURACIÓN DEL PROGRAMA

El sistema de control de créditos desarrollado para ACOVALLE de R.L. (denominado como CrediSys), ha sido desarrollado empleando Visual Fox Pro 9.0 Service Pack 2. A continuación se detalla información sobre los archivos que componen el código fuente y que puede ser instalado de dos maneras:

- INSTALACIÓN ESTÁNDAR: Incorpora los archivos necesarios para ejecutar el programa de créditos. No incorpora el código fuente.

- INSTALACIÓN COMPLETA: Incorpora todos los archivos del código fuente, además de los que ya incluye la instalación estándar.

En este manual se enfoca en los archivos que implementa la instalación completa.

Al instalar el programa, se crean las siguientes carpetas:

Ruta de Instalación: C:\VFPSolutions\CrediSys\

Sub-carpetas:

- Ayuda: Contiene archivos de ayuda y léame del programa.
- Classes: Los archivos de Clase definen propiedades, métodos y eventos los cuales dan funcionalidad a objetos basadas en ellas lo cual permite encapsulación del código.
- Data: Acá se almacenan los archivos de la base de datos que SQL Server crea y administra.
- Forms: Contiene todos los archivos de diseño de los formularios.
- Imgs: Carpeta de imágenes.
- Menu: Almacena los archivos creados para el menú del programa.
- Prgs: Contiene los programas y rutinas necesarias empleadas en la mayoría de formularios.
- Reports: Los archivos de los diseños de diversos reportes.
- Temp: Tablas de Visual Fox Pro para el manejo de usuarios del sistema.

La estructura del programa ha sido diseñada entorno a lo que en VFP se conoce como ‘Administrador de Proyecto’. Dicho proyecto se encuentra en C:\VFPSolutions\Credisys\Credisys.pjx. La siguiente figura muestra un listado de archivos contenidos en el apartado Docs del Administrador de Proyecto de CrediSys, los cuales aparecen ordenados alfabéticamente en orden ascendente.

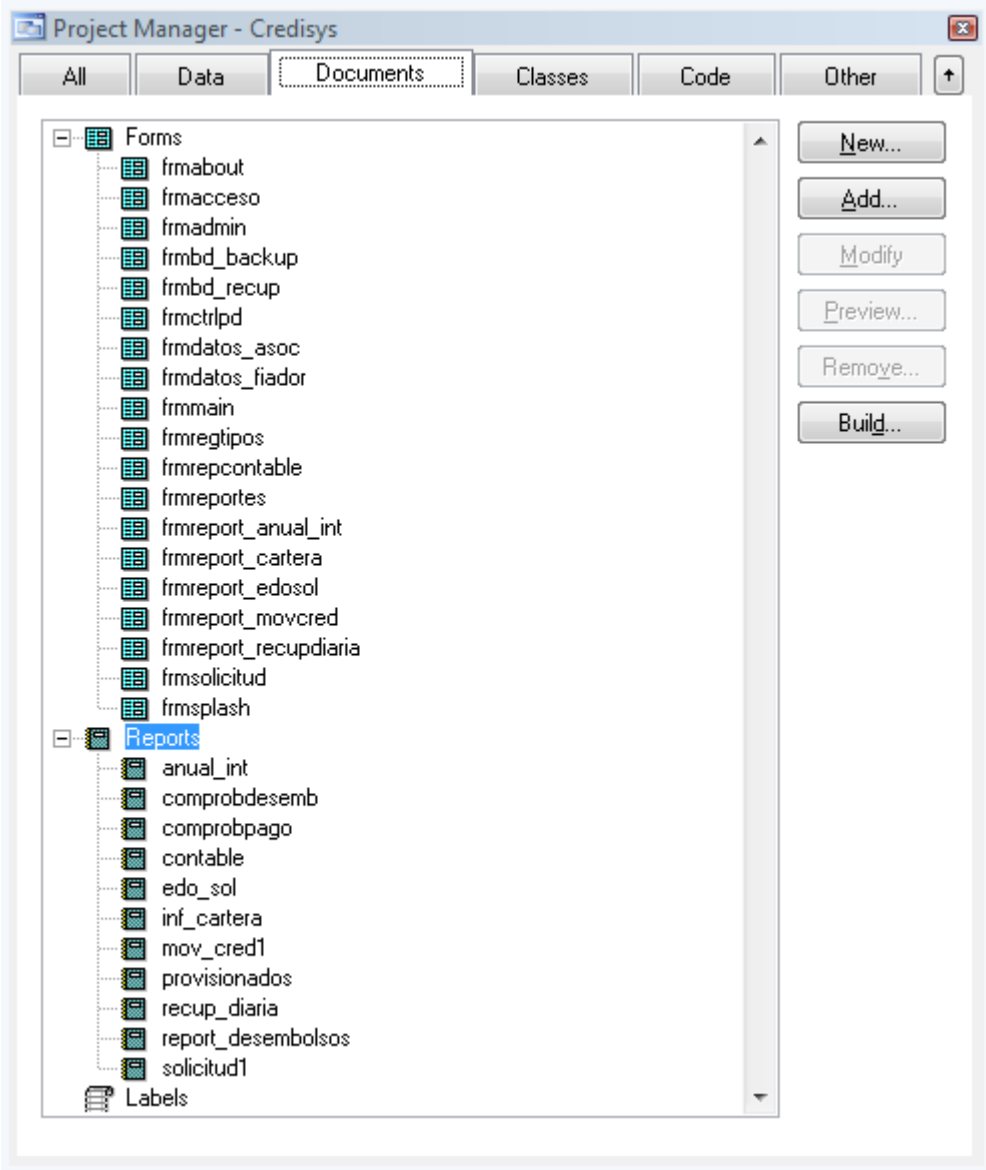


Fig. 1: Listado de formularios y reportes contenidos del diseño del programa de créditos.

Los archivos de los formularios se encuentran en C:\VFPSolutions\Credisys\Forms\. Son archivos con extensión .scx y .sct.

- Formulario frmAbout: Formulario de Acerca de CrediSys. Contiene información sobre el sistema en general. En tiempo de ejecución se puede acceder desde el menú ayuda.

- Formulario frmAcceso: Formulario que permite el acceso al sistema en tiempo de ejecución.

- Formulario frmAdmin: Es la interfaz que sirve para el control de usuarios al sistema con funciones tales como agregar, modificar o eliminar usuarios y establecer si el usuario será o no administrador del sistema CrediSys.

- Formulario frmBD_Backup: Formulario que sirve para crear respaldos de la base de datos de manera rápida y fácil.

- Formulario frmBD_Recup: Similar al formulario anterior, pero con la diferencia que se utiliza para recuperar la base de datos a partir de un respaldo previamente creado.

- Formulario frmCtrlPD: Contiene elementos necesarios para registrar operaciones de pagos y desembolsos a los créditos contratados.

- Formulario frmDatos_Asoc: Se utiliza para registrar los datos de los asociados

- Formulario: frmDatos _Fiador: Interfaz gráfica para registrar los datos de Fiadores.

- Formulario frmMain: Es la pantalla principal en la que se basan todos lo demás formularios, menús y reportes con el fin de dar prioridad al objeto que está en ejecución en ese momento.

  Los siguientes son generadores de reportes:

- Formulario frmRegTipos
- Formulario frmRepContable
- Formulario frmReportes
- Formulario frmReport_Anual_Int
- Formulario frmReport_Cartera
- Formulario frmReport_EdoSol
- Formulario frmReport_Mov_Cred
- Formulario frmReport_RecupDiaria

- Formulario frmSolicitud: Contiene lo necesario para rellenar los datos para una solicitud de crédito.

- Formulario frmSplash: Pantalla de presentación con la que inicia el programa.

Para realizar una conexión ADO (ActiveX Data Object) con la base de datos al sistema de créditos, se emplea una utilidad disponible en el sitio de descargas de Microsoft ® llamado VFPCOM9, el cual es un complemento gratuito que permite la interoperabilidad del programa creado en Visual Fox Pro con un gestor de base de datos, en este caso SQL Server por medio de un recordset[1]. La estructura para realizar la conexión en el código de VFP es el siguiente:

```
PUBLIC rs,Conn1,oCom, as Object && Declaración de variables púbicas.

&& Creando los Objetos
Conn1 = CREATEOBJECT("ADODB.Connection") &&Objeto Contenedor de la Conexión
oCom = CREATEOBJECT("vfpcom.comutil") && Objeto COM para Trabajar con ADO
                                       && Recordset desde VFP
rs = CREATEOBJECT("ADODB.Recordset") && Objeto Recordset.

&& Creamos el String de Conexión
&& lcServidor = Servidor SQL Server
&& lcUsuario = Usuario
&& lcClave = Clave del Usuario
&& lcDb = Base de Datos a Conectar Ejemplo "Northwind"
strconn = "Provider=SQL Native Client;Data Source=&lcServidor ;Initial
Catalog=&lcDb;User Id=&lcUsuario;Password=&lcClave;"

&& Abriendo la Conexión pasando el String al Objeto "ADODB.Connection"
Conn1.Open (strconn)

&& Abriendo el "ADODB.Recordset". Pasamos la Sentencia SQL a Ejecutar,
&& El objeto contenedor de la Conexión y el modo de apertura.
rs.Open ("select * from Orders", Conn1, 1, 3)
&& donde rs.Open ([Consulta SQL], [Conexión activa], [Tipo de Cursor],
[Tipo de Bloqueo])
&& [Consulta SQL]: Un SQL query transaccional cualquiera.
&& [Conexión Activa]: Es el objeto de que contiene la cadena de conexión
asociado al recorset.
&& [Tipo de Cursor]: Establece la forma de abrir el recordset (Si es 1, es
de tipo adOpenKeySet que permite ver cambios dinámicos de un registro
específico).
&& [Tipo de Bloqueo]: Establece la forma de bloquear un objeto recordset
(Si es 3, equivale a adLockOptimistic, que permite que múltiples usuarios
puedan modificar datos hasta que Update sea invocado).
```

Script 1: Estructura ADO ocupada para establecer la conexión co la base de datos.

[1] *Recordset es una estructura de datos usada en programación cuya utilidad es la de almacenar información desde una tabla de una base de datos.*