

IMPLEMENTACION DE UN SISTEMA DE INFORMACION PARA EL MANEJO DE
PAGOS DE LA COOPERATIVA TELCOOP.

INTEGRANTES

LENYS RANGEL
YUSILET PEREZ
KAREN PONZON

MANUAL DE USUARIO

UNIVERSIDAD SIMON BOLIVAR
PROGRAMA DE INGENIERIA DE SISTEMA
2006



INTRODUCCION

En el siguiente proyecto a desarrollar se describirá el funcionamiento en la cooperativa Telcoop Cta., la cual maneja su información llevando un control de los registros en el departamento de contabilidad; en la parte de pagos. Para mantener y ofrecer un mejor manejo de ésta.

Aquí se estudiara la revolución de la tecnología que vive hoy, el cual está cambiando sustancialmente la manera de trabajar de las personas, conllevando que las empresas giren en torno a ellas.

Por otra parte se observará las ventajas que ofrece la sistematización de estos módulos como son: disposición de información oportuna y exacta, para la toma de decisiones, disminuir costos y tiempo; consiguiendo, este una mayor eficiencia en el desarrollo de las actividades en la Cooperativa Telcoop Cta.

1. OBJETIVOS

1.1 OBJETIVO GENERAL

Diseñar un manual práctico para que el Usuario obtengan los conocimientos básicos acerca del sistema.

1.2 OBJETIVOS ESPECIFICOS

- Identificar los requerimientos del software.
- Especificar el motor de base de dato aplicado al sistema.
- Identificar las tablas diseñadas en el sistema.
- Analizar los procesos que conforman el sistema de información.
- Analizar los datos de entrada y salida del sistema.

2. INSTALACIÓN DEL SISTEMA DE INFORMACIÓN DE LA COOPERATIVA TELCOOP CTA

La instalación del software es explicada de forma detallada en el CD de instalación del mismo, cuando este se introduce a la unidad de CD, reproducirá automáticamente (Autorun).

Debe observar detalladamente los requerimientos básicos de Hardware y Software para darle un uso adecuado al programa.

Las herramientas para soporte de información son: motor de base de datos Mysql. La herramienta de diseño e implementación del software usada es Visual Basic versión 6.0. La herramienta de generación de reporte es Data Report de Visual Basic 6.0, ó en su defecto Crystal Report.

3. REQUERIMIENTOS BÁSICOS

Descripción General

Este software fue diseñado para la cooperativa Telcoop cta.

Para que el sistema funcione perfectamente es necesario que el equipo donde se encuentra alojado el software cuente con los siguientes programas fuentes:

- ✓ Microsoft Visual Basic 6.0, como sistema de software
- ✓ MYSQL – Front, como motor de Base de datos.

4. DEFINICIÓN DE LA BASE DE DATOS DEL SISTEMA DE INFORMACIÓN DE LA COOPERATIVA TELCOOP CTA.

Descripción Técnica

El sistema diseñado para la Cooperativa Telcoop CTA, únicamente trabaja con un motor de base de datos MYSQL – Front por su capacidad es usada para la aplicación y soporte de la información del sistema .

A continuación se indicaran las tablas creadas para el sistema de información.

Arp	Emp_salud
Ciudades	Tipo _ contrato
Fondo _ salud	Usuario
Contrato	Parámetro
Datosanuales	Aporte
Emp_arp	Liquidación
Emp_pensión	Fondo _ pensión
Empleados	Auditoria

```

CREATE TABLE `aporte` (
  `dia` int(3) NOT NULL default '0',
  `mes` varchar(20) NOT NULL default '0',
  `año` int(3) NOT NULL default '0',
  `devengado` int(50) NOT NULL default '0',
  `transporte` int(50) NOT NULL default '0',
  `salud` int(50) NOT NULL default '0',
  `pension` int(50) NOT NULL default '0',
  `arp` int(50) NOT NULL default '0',
  `sena` int(50) default '0',
  `subsidio` int(50) NOT NULL default '0',
  `icbf` int(50) default '0',
  `valor1` int(50) NOT NULL default '0',
  `valor2` int(50) NOT NULL default '0',
  `valor3` int(50) NOT NULL default '0',
  `valor4` int(50) NOT NULL default '0',
  `valor5` int(50) NOT NULL default '0',
  `valor6` int(50) NOT NULL default '0',
  `total` int(50) default '0',
  PRIMARY KEY (`mes`,`año`))

```

```

CREATE TABLE `arp` (
  `codARP` int(10) unsigned NOT NULL default '0',
  `nombre` varchar(50) NOT NULL default '0',
  PRIMARY KEY (`codARP`))

```

```

CREATE TABLE `auditoria` (
  `nomusuario` varchar(20) NOT NULL default '0',
  `fecha` varchar(50) NOT NULL default '0',
  `hora` varchar(50) NOT NULL default '0',
  `nombretabla` varchar(60) NOT NULL default '0',
  `detauditor` varchar(60) NOT NULL default '0',
  `procedimiento` varchar(50) NOT NULL default " )

```

```

CREATE TABLE `ciudades` (
  `nombre` varchar(30) NOT NULL default '0',
  PRIMARY KEY (`nombre`))

```

```
CREATE TABLE `contrato` (  
  `cedula` int(11) default NULL,  
  `nombres` varchar(50) default NULL,  
  `Apellido` varchar(50) default NULL,  
  `fecha_inicio` varchar(50) default NULL,  
  `fecha_R` varchar(50) default NULL,  
  `tipoC` varchar(50) default '0' )
```

```
CREATE TABLE `datos anuales` (  
  `año` int(30) NOT NULL default '0',  
  `salario` int(30) NOT NULL default '0',  
  `transporte` int(30) NOT NULL default '0',  
  `transporte1` int(30) NOT NULL default '0',  
  PRIMARY KEY (`año`))
```

```
CREATE TABLE `emp_pension` (  
  `identificacion` int(11) NOT NULL default '0',  
  `nombres` varchar(50) default '0',  
  `Apellido` varchar(50) default '0',  
  `codpension` varchar(50) NOT NULL default '0',  
  `fecha_afiliacion` date NOT NULL default '0000-00-00',  
  `fecha_retiro` date NOT NULL default '0000-00-00',  
  `estado` varchar(50) NOT NULL default '0',  
  `Descripcion` varchar(150) NOT NULL default '0' )
```

```
CREATE TABLE `emp_salud` (  
  `identificacion` int(11) default NULL,  
  `nombres` varchar(50) default NULL,  
  `Apellido` varchar(50) default NULL,  
  `codsalud` varchar(50) default NULL,  
  `fecha_afiliacion` date NOT NULL default '0000-00-00',  
  `fecha_retiro` date NOT NULL default '0000-00-00',  
  `estado` varchar(50) default NULL,  
  `Descripcion` varchar(50) default NULL )
```

```
CREATE TABLE `emp_arp`  
  `identificacion` int(11) default NULL,  
  `nombres` varchar(50) default NULL,  
  `Apellido` varchar(50) default NULL,  
  `codARPchar` varchar(50) default NULL,  
  `fecha_afiliacion` date NOT NULL default '0000-00-00',
```

```
`fecha_retiro` date NOT NULL default '0000-00-00',  
`estado` varchar(50) default NULL,  
`Descripcion` varchar(50) default NULL )
```

```
CREATE TABLE `empleados` (  
  `identificacion` int(11) NOT NULL default '0',  
  `nombres` varchar(50) default '0',  
  `Apellido` varchar(50) default '0',  
  `Fcha_ncto` varchar(50) default '0',  
  `sexo` char(2) default '0',  
  `ciudad` varchar(50) default '0',  
  `Direccion` varchar(50) default '0',  
  `telefono` int(30) default '0',  
  `profesion` varchar(50) default '0',  
  `Cargo` varchar(50) default '0',  
  `sueldo_basico` int(50) unsigned default '0',  
  `estado` varchar(20) default '0',  
  `descripcion` varchar(50) default '0',  
  PRIMARY KEY (`identificacion`))
```

```
CREATE TABLE `fondo_pension` (  
  `codpension` int(10) unsigned NOT NULL default '0',  
  `nombre` varchar(50) default '0',  
  PRIMARY KEY (`codpension`))
```

```
CREATE TABLE `fondo_salud` (  
  `codsalud` int(10) unsigned NOT NULL default '0',  
  `nombre` varchar(50) default '0',  
  PRIMARY KEY (`codsalud`))
```

```
CREATE TABLE `liquidacion` (  
  `dia` int(10) unsigned default '0',  
  `mes` varchar(50) NOT NULL default '0',  
  `año` int(10) unsigned NOT NULL default '0',  
  `cedula` int(50) unsigned NOT NULL default '0',  
  `cargo` varchar(50) NOT NULL default "",  
  `nombre` varchar(50) default NULL,  
  `apellido` varchar(50) default NULL,  
  `sueldo` int(50) unsigned default '0',  
  `diaslab` varchar(50) default '0',  
  `transporte` varchar(50) default '0',  
  `salario` int(50) NOT NULL default '0',  
  `auxilio` int(50) NOT NULL default '0',
```

```
`devengado` int(50) default '0',  
`salud` varchar(50) default '0',  
`pension` varchar(50) default '0',  
`deducido` varchar(50) default '0',  
`neto` varchar(50) default '0',  
PRIMARY KEY (`año`,`mes`))
```

```
CREATE TABLE `parametro` (  
  `codigo` int(11) unsigned NOT NULL default '0',  
  `descripcion` varchar(120) default '0',  
  `porcentaje1` varchar(20) default NULL,  
  `porcentaje2` varchar(11) default '0',  
  `tipoparametro` varchar(50) NOT NULL default "" )
```

```
CREATE TABLE `tipo_contrato` (  
  `codTipocontrato` tinyint(3) unsigned NOT NULL default '0',  
  `nombre` varchar(50) NOT NULL default '0',  
  PRIMARY KEY (`codTipocontrato`,`nombre`))
```

```
CREATE TABLE `usuario` (  
  `nonusuario` varchar(20) NOT NULL default '0',  
  `clave` varchar(20) NOT NULL default '0',  
  `nombre` varchar(20) NOT NULL default '0',  
  `apellido` varchar(50) NOT NULL default '0',  
  `sexo` char(2) NOT NULL default '0',  
  `acceso` varchar(30) NOT NULL default "",  
  PRIMARY KEY (`nonusuario`))
```

5. FUNCIONES QUE SE REALIZAN EN CADA PANTALLA

1. MENÚ DE OPCIONES PRINCIPALES

1.1. PANTALLA FORMULARIO PRINCIPAL

1.1.1 INICIO DE SESION

1.1.1.1 ACCESO A LA APLICACION

1.1.1.1.1 DATOS BÁSICOS

1.1.1.1.1.1 DATOS PARAMETROS

⇒ Private Sub cmbtipo_CLICK()

En este procedimiento se llena el combo box de tipo de parámetro. se evalúan las cajas de texto para que cuando un tipo no es pagado por el empleado la caja de texto correspondiente a la cooperativa se deshabilita con la función de Visible = false o Visible = true.

⇒ Private Sub cmdactualizar_Click()

Esta función permite actualizar cualquiera información que se encuentra almacenada en la base de datos, ya sea el año, el tipo, la descripción o los porcentaje correspondiente a cada tipo, teniendo en cuenta lo que es para el empleado como para la cooperativa.

⇒ Private Sub cmdcancelar_Click()

Este procedimiento permite al usuario que cierre el formulario pero sin salirse en la aplicación totalmente.

⇒ Private Sub cmdconsultar_Click()

Este procedimiento permite consultar toda la información que está registrada en la base de datos, concerniente a la tabla de parámetro.



⇒ Private Sub cmdGuardar_Click()

Este procedimiento permite almacenar en la tabla parámetro la información que el administrador o usuario asignado ingrese en las cajas de textos del formulario, en caso que todas las cajas de textos no fuesen llenadas el sistema enviará un mensaje donde le indique que la información no puede ser almacenada debido a que la información no ha sido tramitada correctamente, de lo contrario le dirá que la información fue almacenada correctamente.

⇒ Private Sub cmdLimpiar_Click()

Este procedimiento permite que las cajas de texto que contiene el formulario sea limpiado totalmente.

⇒ Private Sub Form_Load()

En este procedimiento se declaran eventos para que sean ejecutados en el formulario en modo de ejecución.

⇒ Private Sub txtporcentaje_KeyPress(KeyAscii As Integer)

Esta función valida que en la caja de texto txtporcentaje sólo se debe ingresar números.

⇒ Private Sub txtporcoop_KeyPress(KeyAscii As Integer)

Esta función valida que en la caja de texto txtporcoop sólo se debe ingresar números.

1.1.1.1.1.2 Información Empleados

⇒ Private Sub llenarcombo()

Este procedimiento permite llenar el combo box de ciudades con toda la información que está registrada en la base de datos, tabla ciudades.

⇒ Private Sub cmbestado_CLICK()

Este procedimiento llenar el combo box de estado con la palabra Habilitado o Deshabilitado. Aquí se maneja que si el empleado está o no laborando en la cooperativa, que si está activo la caja de texto descripción desaparece o si está inactivo en la caja de texto descripción muestra las razones por las cuales el empleado ya no labora en la cooperativa.

⇒ Private Sub cmbsexo_KeyPress(KeyAscii As Integer)

Este procedimiento valua que la información que está en el combo box no sea modificada con la función; en el evento keypress

```
If KeyAscii >= 0 And KeyAscii <= 255 Then
```

```
    KeyAscii = 0
```

⇒ Private Sub cmdactualizar_Click()

Este procedimiento permite actualizar la información que reposa en la base de datos en la tabla empleados. Antes de que el sistema actualice toda la información le envía un mensaje donde le pregunta si está o no seguro de actualizarla.

⇒ Private Sub cmdcancelar_Click()

Este procedimiento permite cerrar del formulario sin salirse totalmente de la aplicación.

⇒ Private Sub cmdbuscar_Click()

Este procedimiento permite hacer una consulta avanzada y más detallada acerca de todos los empleados que están registrado en la base de datos. Mostrándolo en un formulario (emergente).

⇒ Private Sub cmdconsultar_Click()

Este formulario permite consultar la información que esta almacenada en la base de datos de cada uno de los empleados sin importar su estado. En caso de que desee consultar a un empleado que no este almacenado el sistema reporta un error de consulta.

⇒ Private Sub cmdGuardar_Click()

Este formulario permite almacenar toda la información que el administrador o usuario ingrese en cada caja de texto. En caso de que todas las cajas de textos no estén llenas el sistema reporta un error donde le indica que para poder almacenar la información deben llenarse todas las cajas de textos.

⇒ Private Sub cmdLimpiar_Click()

Este procedimiento permite limpiar todas las cajas de textos que contenga información.

⇒ Private Sub Form_Load()

En este procedimiento se declaran eventos para que sean ejecutados en el formulario en modo de ejecución.

⇒ Private Sub txtapellido_KeyPress(KeyAscii As Integer)

Este procedimiento evalúa que en la caja de no se digiten números sino solo letras. El siguiente código:

```
If (KeyAscii < 63 Or KeyAscii > 122) And KeyAscii <> 32 And KeyAscii <> 8 And KeyAscii <> 241 And KeyAscii <> 209 Then  
KeyAscii = 0
```

⇒ Private Sub txtCarga_KeyPress(KeyAscii As Integer)

Este procedimiento evalúa que en la caja de no se digiten números sino solo letras. El siguiente código:

```
If (KeyAscii < 63 Or KeyAscii > 122) And KeyAscii <> 32 And KeyAscii <> 8 And KeyAscii <> 241 And KeyAscii <> 209 Then  
KeyAscii = 0
```

⇒ Private Sub txtcedula_KeyPress(KeyAscii As Integer)

Este procedimiento evalúa que en la caja de no se digiten letras sino solo número. El siguiente código:

```
If (KeyAscii >= 1 And KeyAscii <= 7 Or KeyAscii >= 9 And KeyAscii <= 12 Or KeyAscii >= 14 And KeyAscii <= 26 Or KeyAscii >= 28 And KeyAscii <= 31 Or KeyAscii >= 33 And KeyAscii <= 47 Or KeyAscii >= 65 And KeyAscii <= 122 Or KeyAscii >= 58 And KeyAscii <= 255) Then  
KeyAscii = 0
```

⇒ Private Sub txtCiudad_KeyPress(KeyAscii As Integer)

Este procedimiento evalúa que en la caja de no se digiten números sino solo letras. El siguiente código:

```
If (KeyAscii < 63 Or KeyAscii > 122) And KeyAscii <> 32 And KeyAscii <> 8 And KeyAscii <> 241 And KeyAscii <> 209 Then  
KeyAscii = 0
```

⇒ Private Sub txtProfesion_KeyPress(KeyAscii As Integer)

Este procedimiento evalúa que en la caja de no se digiten números sino solo letras. El siguiente código:

```
If (KeyAscii < 63 Or KeyAscii > 122) And KeyAscii <> 32 And KeyAscii <> 8 And KeyAscii <> 241 And KeyAscii <> 209 Then  
KeyAscii = 0
```

⇒ Private Sub txtsueldo_KeyPress(KeyAscii As Integer)

Este procedimiento evalúa que en la caja de no se digiten letras sino solo número. El siguiente código:

```
If (KeyAscii >= 1 And KeyAscii <= 7 Or KeyAscii >= 9 And KeyAscii <= 12 Or KeyAscii >= 14 And KeyAscii <= 26 Or KeyAscii >= 28 And KeyAscii <= 31 Or KeyAscii >= 33 And KeyAscii <= 47 Or KeyAscii >= 65 And KeyAscii <= 122 Or KeyAscii >= 58 And KeyAscii <= 255) Then  
KeyAscii = 0
```

⇒ Private Sub txtNombre_KeyPress(KeyAscii As Integer)

Este procedimiento evalúa que en la caja de no se digiten números sino solo letras. El siguiente código:

```
If (KeyAscii < 63 Or KeyAscii > 122) And KeyAscii <> 32 And KeyAscii <> 8 And KeyAscii <> 241 And KeyAscii <> 209 Then  
KeyAscii = 0
```

⇒ Private Sub txttelefono_KeyPress(KeyAscii As Integer)

Este procedimiento evalúa que en la caja de no se digiten letras sino solo número. El siguiente código:

```
If (KeyAscii >= 1 And KeyAscii <= 7 Or KeyAscii >= 9 And KeyAscii <= 12 Or KeyAscii >= 14 And KeyAscii <= 26 Or KeyAscii >= 28 And
```

```
KeyAscii <= 31 Or KeyAscii >= 33 And KeyAscii <= 47 Or KeyAscii >= 65 And KeyAscii <= 122 Or KeyAscii >= 58 And KeyAscii <= 255) Then  
KeyAscii = 0
```

1.1.1.1.1.3 Contrato

⇒ Private Sub cmdactualizar_Click()

Este procedimiento permite actualizar la información que reposa en la base de datos en la tabla empleados. Antes de que el sistema actualice toda la información le envía un mensaje donde le pregunta si está o no seguro de actualizarla.

⇒ Private Sub cmdadicionar_Click()

Este formulario permite almacenar toda la información que el administrador o usuario ingrese en cada caja de texto. En caso de que todas las cajas de textos no estén llenas el sistema reporta un error donde le indica que para poder almacenar la información deben llenarse todas las cajas de textos. Aquí se controla que las fechas no sean iguales ni que la fecha de inicio sea mayor que la fecha de finalización.

⇒ Private Sub cmdBuscarE_Click()

Este procedimiento permite hacer una consulta avanzada y más detallada acerca de todos los empleados que están registrado en la base de datos. Mostrándolo en un formulario (emergente).

⇒ Private Sub cmdcancelar_Click()

Este procedimiento permite cerrar del formulario sin salirse totalmente de la aplicación.

⇒ Private Sub cmdconsultar_Click()

Este formulario permite consultar la información que esta almacenada en la base de datos de cada uno de los empleados sin importar su estado. En caso de que desee consultar a un empleado que no este almacenado el sistema reporta un error de consulta.

1.1.1.1.4 Liquidación de Empleado

Los cálculos que se hacen en el sistema es básicamente en formulario de liquidación de nomina y liquidación de aporte ; los procedimiento que se utilizan son:

Private Sub cmdcalcular1_Click()

Este procedimiento es el que efectúa todos los cálculos de la liquidación. Se evalúa que si el empleado gana más de dos salario mínimo la caja de texto de transporte se deshabilita y se gana menos de dos salarios mínimo se le tiene en cuenta lo que le corresponde por auxilio de transporte, de igual manera se calcula cuanto se le descuenta dependiendo del salario por salud y pensión, para sacar el total devengado, deducido y neto a pagar.

Private Sub cmdcalcular3_Click()

Este procedimiento efectua los calculos de lo devengado y lo deducido de cada uno de los empleados, para hacer la liquidación de aporte.

Private Sub cmdsumar_Click()

Este procedimiento efectúa todos los cálculos necesarios para la liquidación de aporte para fiscales como; salud, pensión, arp, sena, icbf, subsidio comfamiliar por cada empleado.

1.1.1.1.5 Liquidación de Aporte para Fiscales

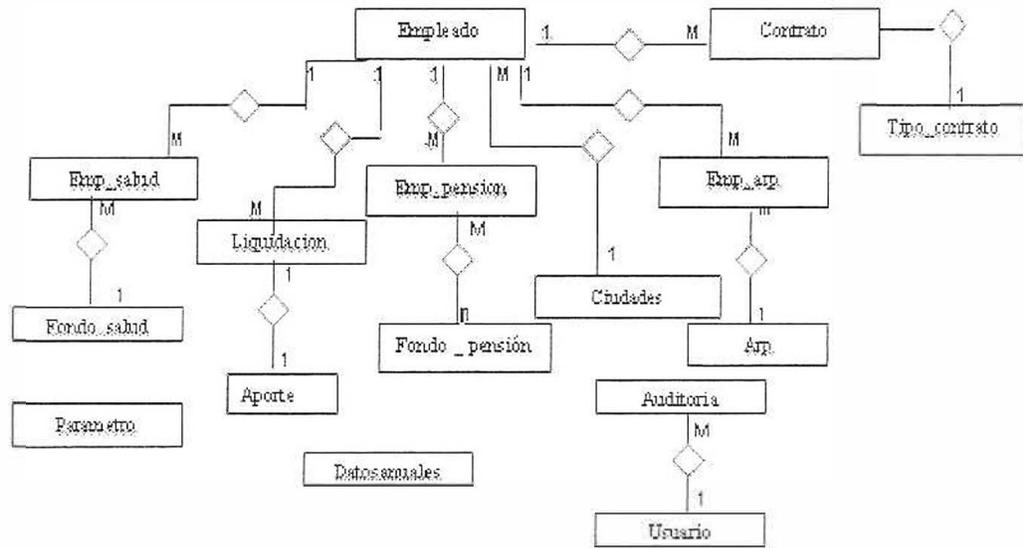
⇒ Private Sub Form_Load()

En este procedimiento se estable en los combo box los días y meses, para así tener mayor facilidad de utilizar la aplicación.

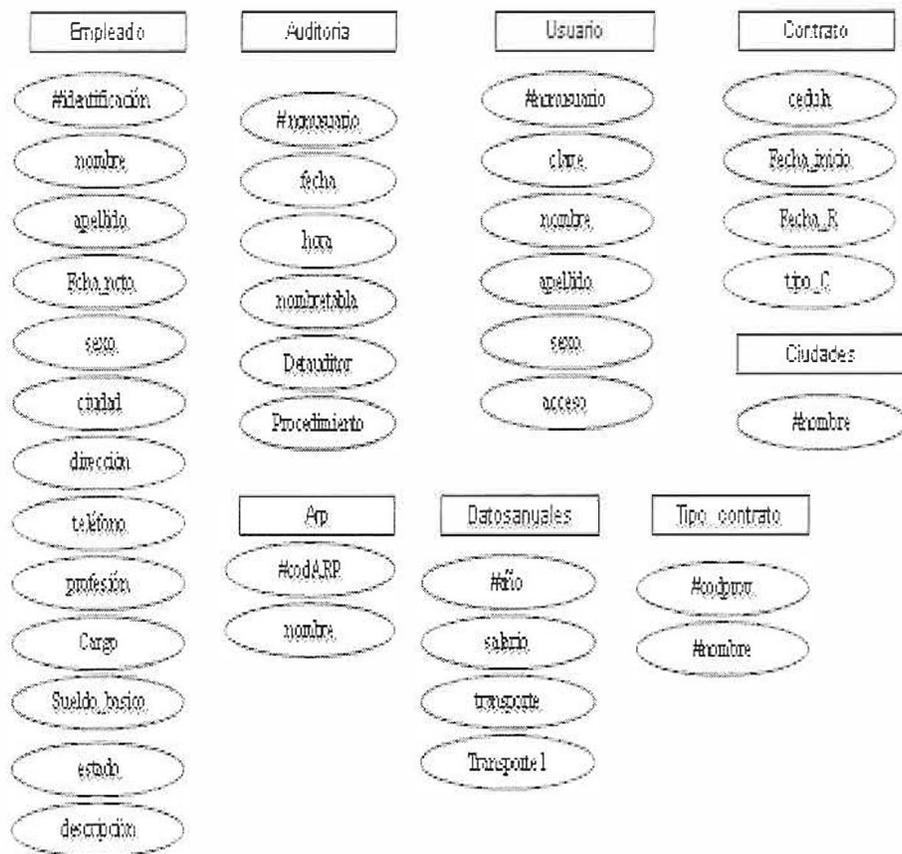
⇒ Private Sub cmdcalcular1_Click()

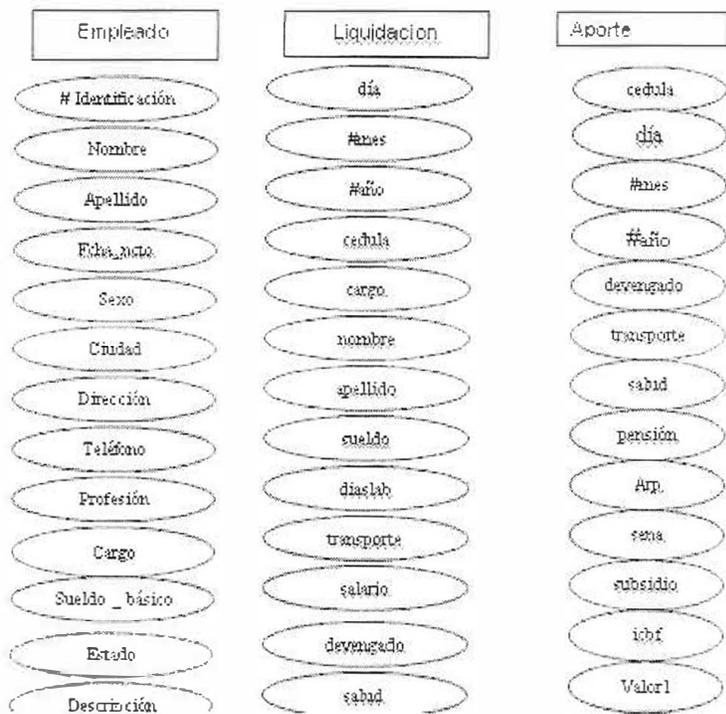
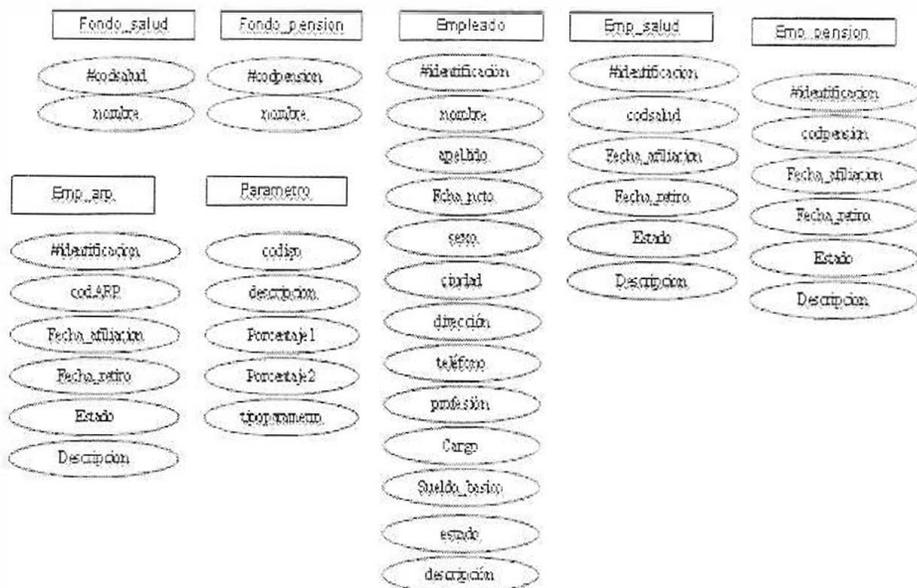
En este procedimiento se hace el calculo general de toda la liquidación de aporte, para llevarla a cabo primero se debe escoger la fecha y dependiendo de ella aparecerá el total devengado y total auxilio de Transporte, que fue calculado durante el mes para liquidar los aportes dependiendo el porcentaje que le corresponde a cada ítem de esta liquidación.

6. DIAGRAMA ENTIDAD – RELACIÓN



ATRIBUTOS DE LAS TABLAS





7 .Diagrama de flujo de Datos

