

```
        'resource_id' => $role_details['id'],
        );
        if ( $this->rule_exists( $resource_details['id'], $role_details['id'] ) ) {
            if ( $access == false ) {
                // Remove the rule as there is currently no need for it
                $details['access'] = !$access;
                $this->_sql->delete( 'acl_rules', $details );
            } else {
                // Update the rule with the new access value
                $this->_sql->update( 'acl_rules', array( 'access' => $access ) );
            }
        }
        foreach( $this->rules as $key=>$rule ) {
            if ( $details['role_id'] == $rule['role_id'] && $details['resource_id'] == $rule['resource_id'] ) {
                if ( $access == false ) {
                    unset( $this->rules[ $key ] );
                } else {
                    $this->rules[ $key ]['access'] = $access;
                }
            }
        }
    }
```

MANUAL DE PROGRAMADOR-MODULO DEPRECIACIÓN DE PLANTA Y ELEMENTOS DE LA SUPERVIGILANCIA

Tabla de Contenido

1.

FORMATO PRELIMINAR AL DOCUMENTO.....

2

2.

PRESENTACIÓN.....

4

3.

OBJETIVO DEL MANUAL

4

4.

ALCANCE.....

4

5.

MODULO DEPRECIACION

4

6.

VISTA FORMULARIO MODULO DEPRESIACIÓN

5

7.

VISTA CODIGO MODULO DEPRECIACION.....

5

7.1

Tabla Tiempo de vida de los elementos de la SuperVigilancia

6

1. *FORMATO PRELIMINAR AL DOCUMENTO*

Título: MANUAL DE PROGRAMADOR-MODULO DEPRECIACIÓN PLANTA Y ELEMENTOS DE LA SVSP

Fecha de creación: octubre 25 de 2016

Formato: Manual

Fecha de emisión: 31/10/2016

Proceso: Gestión Informática y Sistemas

Código: Versión 01

Autores Gestión Informática y Sistemas, Diego Fernando Cárdenas Hernández.
Gestión Informática y Sistemas, Mery Esperanza Sánchez Zambrano

Revisó: Claudia Marcela Ladino Herrera, Jefe Oficina Informática y Sistemas

Aprobó Claudia Marcela Ladino Herrera, Jefe Oficina Informática y Sistemas

Ubicación: Suite Visión Empresarial.

2. PRESENTACIÓN

En este documento vamos a describir el código fuente del módulo de depreciación de planta y equipos de la Supervigilancia; así como sus principales funciones y variables las cuales permiten hallar la depreciación.

Cabe resaltar que se cuenta con un motor de bases de datos de access, el desarrollo fue realizado en Microsoft Visual Basic 2008 debido a que contamos con la licencia del mismo.

Los reportes se generan por medio de crystal Report porque desde que se desarrollo el aplicativo se ha utilizado esta herramienta.

3. OBJETIVO DEL MANUAL

El presente documento tiene como objetivo explicar a los nuevos desarrolladores el código utilizado para el desarrollo del módulo de depreciación de planta y equipos de la SUPERVIGILANCIA.

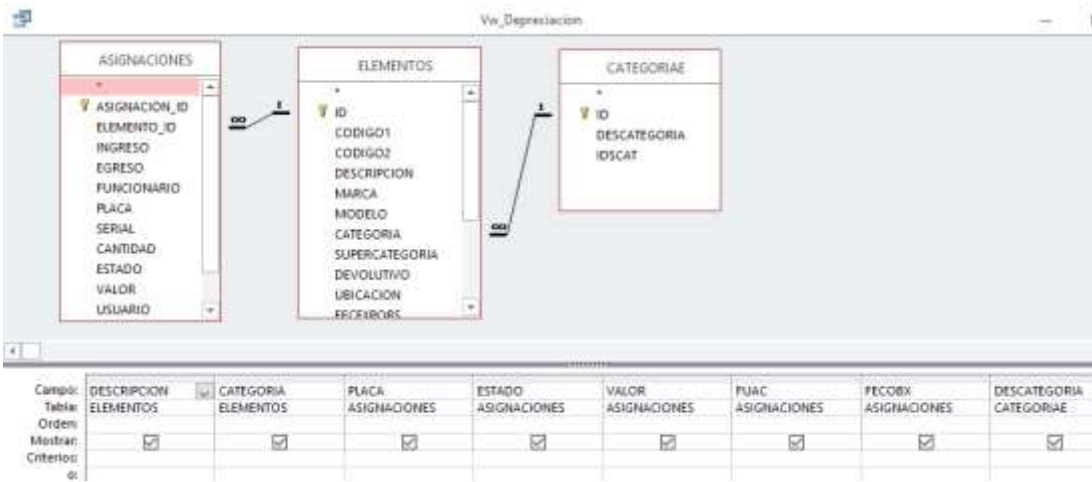
4. ALCANCE

El alcance principal de este manual es brindar una guía a los nuevos desarrolladores de la supervigilancias para realizar mejoras al módulo de depreciación de planta y equipod de la SVSP.

5. MODULO DEPRECIACION

Dentro de la depreciación de los elementos se deben tener 3 factores en cuenta para generar el reporte:

1. La base de datos Inventarios, con una vista (Vw_Depreciacion) de 3 tablas principales para el modulo depreciacion Tabla Asignaciones, Tabla Elementos, Tabla CateegoriaE y los campos que se muestran en la imagen acontnuación: (Tienen que estar en ese orden).



2. La depreciacion comienza cuando se registra un elemento a la Supervigilancia y es **asignado a un funcionario** (Solo se deprecian los elementos que han sido asignados).
3. Para generar la depreciación de los elementos se tienen en cuenta los campos definidos en la vista depreciacion (Vw_Depreciacion).

Definicion de campos:

- FUAC = Fecha inicial en la cua ingreso el elemento.
- FECOBX = Fecha de estimada de tiempo de vida.
- CATEGORIA = Id para identificar la FECOBX.
- VALOR = Valor del elemento.

6. VISTA FORMULARIO MODULO DEPRESIACIÓN

FDepreciacion

Digite la Fecha de la Depreciacion a Hallar

martes , 30 de agosto

Generar Reporte

Ingrese El Numero Del Comprobante

9

Buscar

NUMERO COMPROBANTE	SUPER CATEGORIA	VALOR
9	MUEBLES Y ENSERES	\$3.405.766,07
9	EQUIPO Y MAQUINAS DE OFICINA	\$63.691,35
9	EQUIPOS DE COMUNICACION	\$2.724.447,48
9	EQUIPOS DE COMPUTACION	\$38.815.392,28
9	TERRESTRE	\$4.079.802,75
9	EDIFICACIONES	\$0,00
9	EQUIPO DE RESTAURANTE Y CAFETERIA	\$1.439.125,05
9	LICENCIAS Y SOFTWARE	\$0,00
9	DEPRECIACIÓN DE PROPIEDADES, PLANTA Y EQUIPOS	\$50.528.224,38

Ver

Imprimir

Sección Roja: En esta sección se selecciona la fecha a hallar la depreciación; luego se oprime el botón Generar Reporte y éste se muestra en la sección naranja.

Sección Azul: La sección azul se puede consultar depreciaciones que ya se han generado, por medio del número de comprobante, esta búsqueda se muestra en la sección naranja.

Sección Verde: La sección verde nos sirve para generar una copia imprimible de la depreciación generada.

7. VISTA CODIGO MODULO DEPREDIACION

// Define el formulario

Imports System.IO
Public Class FDepreciacion

// se define el boton generar el cual genera el reporte de la depreciacion mensual.

Private Sub BtGenerar_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles BtGenerar.Click

// se definen las variables comando las cuales permiten ejecutar consultas a la base de datos

Dim comando, comando2, comando3, fechanos, placa, texto As String

// se definen lass variables parametros para controlar las consultas a a base de datos

Dim parametros, parametros2, parametros3 As New ArrayList

// se defienen variables para hallar la cantidad de meses que hay en un intervalo de dos fechas de años, al igual que la depreciacion que tiene cada elemento.

Dim contar, meses, Saldo_costo, meshallar, comprobante, mUpdate, id As Integer
Dim fec_ingreso, fec_vida, fec_hallar As Date

Página 5 de 13

SISTEMA GESTIÓN DE LA CALIDAD
Código: MAN-GSI-300-001
Fecha aprobación: 28/08/2018
Versión: 2

Superintendencia de Vigilancia y Seguridad Privada
Centro de Información al Ciudadano: Calle 24ª No. 59 – 42 Torre 4 Piso 3
Sede Administrativa: Avenida Calle 26 # 57-41 Torre 8 - Piso 11
PBX: (571) 3078038 Línea Gratuita Nacional: 01 8000 119703
www.supervigilancia.gov.co

SGS

ISO 9001

2015

SGS

ISO 14001

2015

// Se definen variables para hallar el total de la depreciacion acumulada por categorias.

Dim saldo, cat1, cat2, cat3, cat4, cat5, cat6, cat7, cat8, cat9, cat10 As Double
Dim valor, Dep_acumulado, Dep_mensual As Single

// Se definen una “copias” de las tablas de la base de datos de Inventarios que sirven para realizar la depreciacion de los elementos.

Dim mTabla As New DataTable("Vw_Depreciacion")
Dim mTabla2 As New DataTable("ASIGNACIONES")

// Se define la tabla mTabla3 para guardar y consultar la depreciacion hallada
Dim mTabla3 As New DataTable("Depreciacion")

// Se define la conexcion a la base de datos

Dim objConexion, objConexion2 As New ConexionJet

// Se asigna la consulta SQL a la variable comando
comando = "SELECT * From Vw_Depreciacion"

// Se ejecuta la consulta y se asigna a mTabla la cual trae la información
mTabla = objConexion.consultar(comando, parametros, "Vw_Depreciacion").Tables("Vw_Depreciacion").Copy

// Ciclo de repetición que recorre toda la tabla comparando registro por registro hasta que no quede ninguno por consultar.

For rex = 0 To mTabla.Rows.Count - 1

// En la tabla VW_Depreciacion se encuentra el campo categoria, donde dependiendo la catagoria va a ingresar a las sentencias IF. Este proceso lo realiza hasta que se finalice el ciclo de repetición.

2	ELEMENTOS DE USO DOMESTICO
3	EQUIPO DE COMEDOR Y COCINA
4	EQUIPO DE COMUNICACIÓN
5	EQUIPO DE COMPUTO
6	PAPELERIA
7	EQUIPO DE TRANSPORTE
8	UTILES DE ESCRITORIO
9	COMBUSTIBLES
10	INTANGIBLES
11	MUEBLES, ENSERES Y EQUIPO DE OFICINA (10)
12	MUEBLES, ENSERES Y EQUIPO DE OFICINA (12)
13	RESPUESTOS PARA MAQUINAS Y EQUIPO
14	VESTUARIO,ROPA
15	VIVERES, RANCHO Y LICORES
16	VEHICULOS
17	LICENCIAS
18	SOFTWARE

Imagen tabla categoria

Categoria	Años de vida
MUEBLES Y ENSERES	10
EQUIPO Y MAQUINAS DE OFICINA	10
EQUIPOS DE COMUNICACIÓN	10
EQUIPOS DE COMPUTACION	5
TERRESTRE	10
EDIFICACIONES	50
EQUIPO DE RESTAURANTE Y CAFETERIA	10
LICENCIAS	5
SOFTWARE	5

7.1 Tabla Tiempo de vida de los elementos de la SuperVigilancia

// Si el valor del campo categoria corresponde a uno que se muestra en la imagen anterior entra en las sentencias IF. Ejemplo. Si el registro en categoria tiene el valor de 11 entra en el IF de Muebles y enseres.

'MUEBLES Y ENSERES CAT 1

Saldo_costo = 0

If (mTabla.Rows(rex).Item(1) = 11) Or (mTabla.Rows(rex).Item(1) = 12) Then

// Asigna el valor de la columna 6 de la la vista Vw_Deprciacion en este caso y los fec_ingreso = FUAC

fec_ingreso = mTabla.Rows(rex).Item(5).ToString

// Debido a que el sistema anteriormente no solicitaba la fecha de obsolescencia o tiempo de vida, eta se debe calcular partiendo de la fec_ingreso. Para la categoria MUEBRES Y ENSERES el tiempo de vida son 10 años. En la siguiente sentencia se puede decir que toma la fec_ingreso y le suma 10 años y este valor se lo asigna a fec_vida: que es la fecha de tiempo de vida.

fec_vida = DateAdd(DateInterval.Year, 10, fec_ingreso)

'calcula numero de meses

// fec_hallar es la que selecciona el usuario para hallar la depreciacion de los elementos desde la fec_inicial hasta fec_hallar

fec_hallar = dtfecha.Value

// La siguiente sentencia proporciona el numero de meses que hay entre el intervalo entre la fecha inicial y la fecha de obsolescencia

meses = DateDiff(DateInterval.Month, fec_ingreso, fec_vida)

// esta sentecina sirve para hallar los meses que hay entre la fecha inicial y la fecha a hallar.

meshallar = DateDiff(DateInterval.Month, fec_ingreso, fec_hallar)

// asigno la columna 5 que es el valor del elemento, de la vista depreciacion (Vw_Depreciacion)

valor = mTabla.Rows(rex).Item(4).ToString

// La depreciacion mensual se halla de dividir el valor del elemento en el numero de meses para saber cuanto se deprecia mensualmente.

Dep_mensual = valor / meses

// con la variable saldo_costo se halla cuanto se ha depreciado el elemento hasta la fecha que se genera el reporte.

Saldo_costo = (Dep_mensual * meshallar)

// Si el saldo costo es mayor a 0 entonces en la variable cat1 se van sumando el acumulado mensual de todos los elementos. Si se presenta el caso que el saldo costo es menor a cero, se concidera que el elemento ha sido depreciado en su totalidad.

If Saldo_costo >= 0 Then

cat1 = cat1 + Dep_mensual

End If

End If

NOTA: LO QUE SE COMENTO EN LA SENTENCIA IF ANTERIOR OCURRE PARA LOS OTROS IF, CABE RESALTAR QUE ESTAS SENTENCIAS SON SOLO PARA HALLAR EL ACUMULADO DE DEPRECIACION HASTA LA FECHA QUE SE PRETENDE GENERAR EL REPORTE.

EN LAS SIGUIENTES SENTENCIAS IF CAMBIAN LAS VARIABLES QUE ACUMULAN POR EL TIPO DE CATEGORIA.

'EQUIPOS DE MAQUINAS DE OFICINA cat 2

```
If (mTabla.Rows(rex).Item(1) = 6) Or (mTabla.Rows(rex).Item(1) = 8) Or
(mTabla.Rows(rex).Item(1) = 13) Then
    fec_ingreso = mTabla.Rows(rex).Item(5).ToString
    fec_vida = DateAdd(DateInterval.Year, 10, fec_ingreso)
    'calcula numero de meses
    fec_hallar = dtfecha.Value
    meses = DateDiff(DateInterval.Month, fec_ingreso, fec_vida)
    meshallar = DateDiff(DateInterval.Month, fec_ingreso, fec_hallar)
    valor = mTabla.Rows(rex).Item(4).ToString
    Dep_mensual = valor / meses
    Saldo_costo = (Dep_mensual * meshallar)
    If Saldo_costo >= 0 Then
        cat2 = cat2 + Dep_mensual
    End If
End If

'EQUIPOS DE COMUNICACION cat 3
If mTabla.Rows(rex).Item(1) = 4 Then
    fec_ingreso = mTabla.Rows(rex).Item(5).ToString
    fec_vida = DateAdd(DateInterval.Year, 10, fec_ingreso)
    'calcula numero de meses
    fec_hallar = dtfecha.Value
    meses = DateDiff(DateInterval.Month, fec_ingreso, fec_vida)
    meshallar = DateDiff(DateInterval.Month, fec_ingreso, fec_hallar)
    valor = mTabla.Rows(rex).Item(4).ToString
    Dep_mensual = valor / meses
    Saldo_costo = (Dep_mensual * meshallar)
    If Saldo_costo >= 0 Then
        cat3 = cat3 + Dep_mensual
    End If
End If

'EQUIPOS DE COMPUTACION cat 4
If mTabla.Rows(rex).Item(1) = 5 Then
    fec_ingreso = mTabla.Rows(rex).Item(5).ToString
    fec_vida = DateAdd(DateInterval.Year, 5, fec_ingreso)
    'calcula numero de meses
    fec_hallar = dtfecha.Value
    meses = DateDiff(DateInterval.Month, fec_ingreso, fec_vida)
    meshallar = DateDiff(DateInterval.Month, fec_ingreso, fec_hallar)
    valor = mTabla.Rows(rex).Item(4).ToString
    Dep_mensual = valor / meses
    Saldo_costo = (Dep_mensual * meshallar)
    If Saldo_costo >= 0 Then
        cat4 = cat4 + Dep_mensual
    End If
End If

'TERRESTRE cat 5
If (mTabla.Rows(rex).Item(1) = 16) Or (mTabla.Rows(rex).Item(1) = 7) Then
    fec_ingreso = mTabla.Rows(rex).Item(5).ToString
    fec_vida = DateAdd(DateInterval.Year, 10, fec_ingreso)
    'calcula numero de meses
    fec_hallar = dtfecha.Value
    meses = DateDiff(DateInterval.Month, fec_ingreso, fec_vida)
    meshallar = DateDiff(DateInterval.Month, fec_ingreso, fec_hallar)
    valor = mTabla.Rows(rex).Item(4).ToString
    Dep_mensual = valor / meses
    Saldo_costo = (Dep_mensual * meshallar)
    If Saldo_costo >= 0 Then
        cat5 = cat5 + Dep_mensual
    End If
End If

'EDIFICACIONES cat 6
If (mTabla.Rows(rex).Item(1) = 19) Then
```

```

fec_ingreso = mTabla.Rows(rex).Item(5).ToString
fec_vida = DateAdd(DateInterval.Year, 50, fec_ingreso)
'calcula numero de meses
fec_hallar = dtfecha.Value
meses = DateDiff(DateInterval.Month, fec_ingreso, fec_vida)
meshallar = DateDiff(DateInterval.Month, fec_ingreso, fec_hallar)
valor = mTabla.Rows(rex).Item(4).ToString
Dep_mensual = valor / meses
Saldo_costo = (Dep_mensual * meshallar)
If Saldo_costo >= 0 Then
    cat6 = cat6 + Dep_mensual
End If
End If

'EQUIPO DE RESTAURANTE Y CAFETERIA cat 7
If (mTabla.Rows(rex).Item(1) = 2) Or (mTabla.Rows(rex).Item(1) = 3) Or
(mTabla.Rows(rex).Item(1) = 15) Then
    fec_ingreso = mTabla.Rows(rex).Item(5).ToString
    fec_vida = DateAdd(DateInterval.Year, 10, fec_ingreso)
    'calcula numero de meses
    fec_hallar = dtfecha.Value
    meses = DateDiff(DateInterval.Month, fec_ingreso, fec_vida)
    meshallar = DateDiff(DateInterval.Month, fec_ingreso, fec_hallar)
    valor = mTabla.Rows(rex).Item(4).ToString
    Dep_mensual = valor / meses
    Saldo_costo = (Dep_mensual * meshallar)
    If Saldo_costo >= 0 Then
        cat7 = cat7 + Dep_mensual
    End If
End If

'LICENCIAS
If (mTabla.Rows(rex).Item(1) = 17) Or (mTabla.Rows(rex).Item(1) = 18) Then
    fec_ingreso = mTabla.Rows(rex).Item(5).ToString
    fec_vida = DateAdd(DateInterval.Year, 5, fec_ingreso)
    'calcula numero de meses
    fec_hallar = dtfecha.Value
    meses = DateDiff(DateInterval.Month, fec_ingreso, fec_vida)
    meshallar = DateDiff(DateInterval.Month, fec_ingreso, fec_hallar)
    valor = Format(Val(mTabla.Rows(rex).Item(4).ToString), "##,###.#0")
    Dep_mensual = valor / meses
    Saldo_costo = Format(Val(Dep_mensual * meshallar), "##,###.#0")
    If Saldo_costo >= 0 Then
        cat8 = cat8 + Dep_mensual
    End If
End If

'SOFTWARE
If mTabla.Rows(rex).Item(1) = 18 Then
    fec_ingreso = mTabla.Rows(rex).Item(5).ToString
    fec_vida = DateAdd(DateInterval.Year, 5, fec_ingreso)
    'calcula numero de meses
    fec_hallar = dtfecha.Value
    meses = DateDiff(DateInterval.Month, fec_ingreso, fec_vida)
    meshallar = DateDiff(DateInterval.Month, fec_ingreso, fec_hallar)

    // la siguiente sentencia sirve para que el valor lo muestre en signo pesos. 0")
    valor = Format(Val(mTabla.Rows(rex).Item(4).ToString), "##,###.#0")
    Dep_mensual = valor / meses
    Saldo_costo = Format(Val(Dep_mensual * meshallar), "##,###.#0")
    If Saldo_costo >= 0 Then
        cat9 = cat9 + Dep_mensual
    End If
End If

Next

```

// la variable saldo suma todos los valores que hay en las variables de categoria para dar una depreciacion general.

saldo = cat1 + cat2 + cat3 + cat4 + cat5 + cat6 + cat7

// se realiza la consulta a la base de datos de depreciacion

```
comando3 = "select *from Depreciacion"
contar = mTabla3.Rows.Count + 100
mTabla3 = objConexion.consultar(comando3, parametros3,
"Depreciacion").Tables("Depreciacion").Copy
comprobante = mTabla3.Rows.Count
```

// sentencia que inserta la información antes hallada en la base de datos con un registro unico.

```
id = mTabla3.Rows.Count + 1
texto = "MUEBLES Y ENSERES"
comando3 = "INSERT INTO Depreciacion (ID, COMPROBANTE, CATEGORIA, VALOR)
VALUES ( '" & id & "', '" & comprobante & "', '" & texto & "', '" & cat1 & "')"
mUpdate = objConexion.consultarActualizacion(comando3, parametros3)
```

// sentencia que inserta la información antes hallada en la base de datos con un registro unico.

```
id = id + 1
texto = "EQUIPO Y MAQUINAS DE OFICINA"
comando3 = "INSERT INTO Depreciacion (ID, COMPROBANTE, CATEGORIA, VALOR)
VALUES ( '" & id & "', '" & comprobante & "', '" & texto & "', '" & cat2 & "')"
mUpdate = objConexion.consultarActualizacion(comando3, parametros3)
```

// sentencia que inserta la información antes hallada en la base de datos con un registro unico.

```
id = id + 1
texto = "EQUIPOS DE COMUNICACION"
comando3 = "INSERT INTO Depreciacion (ID, COMPROBANTE, CATEGORIA, VALOR)
VALUES ( '" & id & "', '" & comprobante & "', '" & texto & "', '" & cat3 & "')"
mUpdate = objConexion.consultarActualizacion(comando3, parametros3)
```

// sentencia que inserta la información antes hallada en la base de datos con un registro unico.

```
id = id + 1
texto = "EQUIPOS DE COMPUTACION"
comando3 = "INSERT INTO Depreciacion (ID, COMPROBANTE, CATEGORIA, VALOR)
VALUES ( '" & id & "', '" & comprobante & "', '" & texto & "', '" & cat4 & "')"
mUpdate = objConexion.consultarActualizacion(comando3, parametros3)
```

// sentencia que inserta la información antes hallada en la base de datos con un registro unico.

```
id = id + 1
texto = "TERRESTRE"
comando3 = "INSERT INTO Depreciacion (ID, COMPROBANTE, CATEGORIA, VALOR)
VALUES ( '" & id & "', '" & comprobante & "', '" & texto & "', '" & cat5 & "')"
mUpdate = objConexion.consultarActualizacion(comando3, parametros3)
```

// sentencia que inserta la información antes hallada en la base de datos con un registro unico.

```
id = id + 1
texto = "EDIFICACIONES"
comando3 = "INSERT INTO Depreciacion (ID, COMPROBANTE, CATEGORIA, VALOR)
VALUES ( '" & id & "', '" & comprobante & "', '" & texto & "', '" & cat6 & "')"
mUpdate = objConexion.consultarActualizacion(comando3, parametros3)
```

// sentencia que inserta la información antes hallada en la base de datos con un registro unico.

```
id = id + 1
texto = "EQUIPO DE RESTAURANTE Y CAFETERIA "
comando3 = "INSERT INTO Depreciacion (ID, COMPROBANTE, CATEGORIA, VALOR)
VALUES ( '" & id & "', '" & comprobante & "', '" & texto & "', '" & cat7 & "' )"
mUpdate = objConexion.consultarActualizacion(comando3, parametros3)
```

// sentencia que inserta la información antes hallada en la base de datos con un registro unico.

```
id = id + 1
texto = "DEPRECIACION DE PROPIEDADES, PLANTA Y EQUIPOS "
comando3 = "INSERT INTO Depreciacion (ID, COMPROBANTE, CATEGORIA, VALOR)
VALUES ( '" & id & "', '" & comprobante & "', '" & texto & "', '" & saldo & "' )"
mUpdate = objConexion.consultarActualizacion(comando3, parametros3)
```

// sentencia que inserta la información antes hallada en la base de datos con un registro unico.

```
id = id + 1
texto = "LICENCIAS"
comando3 = "INSERT INTO Depreciacion (ID, COMPROBANTE, CATEGORIA, VALOR)
VALUES ( '" & id & "', '" & comprobante & "', '" & texto & "', '" & cat8 & "' )"
mUpdate = objConexion.consultarActualizacion(comando3, parametros3)
```

// sentencia que inserta la información antes hallada en la base de datos con un registro unico.

```
id = id + 1
texto = "SOFTWARE"
comando3 = "INSERT INTO Depreciacion (ID, COMPROBANTE, CATEGORIA, VALOR)
VALUES ( '" & id & "', '" & comprobante & "', '" & texto & "', '" & cat9 & "' )"
mUpdate = objConexion.consultarActualizacion(comando3, parametros3)
```

// sentencia que inserta la información antes hallada en la base de datos con un registro unico.

```
id = id + 1
cat10 = cat8 + cat9
texto = "AMORTIZACION DE LICENCIAS Y SOFTWARE"
comando3 = "INSERT INTO Depreciacion (ID, COMPROBANTE, CATEGORIA, VALOR)
VALUES ( '" & id & "', '" & comprobante & "', '" & texto & "', '" & cat10 & "' )"
mUpdate = objConexion.consultarActualizacion(comando3, parametros3)
```

// sentencia que muestra la informacion hallada por pantalla.

```
comando3 = "select id, comprobante, cuentas, categoria, valor from Depreciacion where
comprobante = '" & comprobante & "' "
mTabla3 = objConexion.consultar(comando3, parametros3,
"Depreciacion").Tables("Depreciacion").Copy
Me.DvDepreciacion.DataSource = mTabla3
Me.DvDepreciacion.Columns(0).HeaderText = "ID"
Me.DvDepreciacion.Columns(0).Width = 5
Me.DvDepreciacion.Columns(0).Visible = False
Me.DvDepreciacion.Columns(1).HeaderText = "NUMERO COMPROBANTE"
Me.DvDepreciacion.Columns(1).Width = 100
Me.DvDepreciacion.Columns(2).HeaderText = "CUENTAS PUC"
Me.DvDepreciacion.Columns(2).Width = 120
Me.DvDepreciacion.Columns(3).HeaderText = "SUPER CATEGORIA"
Me.DvDepreciacion.Columns(3).Width = 310
Me.DvDepreciacion.Columns(4).HeaderText = "VALOR"
Me.DvDepreciacion.Columns(4).Width = 120
```

```
Me.DvDepreciacion.Columns(4).DefaultCellStyle.Alignment =  
DataGridViewContentAlignment.MiddleRight  
Me.DvDepreciacion.Columns(4).DefaultCellStyle.Format = "C2"  
  
End Sub
```

// Define el boton buscar

```
Private Sub btnBuscar_Click(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles btnBuscar.Click
```

// se definen variables para consultar la base de datos

```
Dim comando, columna As String  
Dim parametros As New ArrayList  
Dim mTabla As New DataTable("Depreciacion")  
Dim objConexion As New ConexionJet  
columna = ""  
mNoe = False  
Me.DvDepreciacion.DataSource = Nothing
```

// si el cuadro de texto del boton buscar se encuentra vacio, arrojará un error diciendo que debe ingresar un valor..

```
If Me.TxtBuscar.Text = "" Then  
    MessageBox.Show("Por favor debe ingresar el numero del comprobante",  
"Mostrando Reporte", MessageBoxButtons.OK, MessageBoxIcon.Error)
```

// si no esta vacio, se se ejecuta la consulta a al base de datos para verificar que ese comprobante si existe

```
Else  
    comando = "select id, comprobante, cuentas, categoria, valor From Depreciacion "  
    & "WHERE comprobante = " & Val(Me.TxtBuscar.Text)  
    mTabla = objConexion.consultar(comando, parametros,  
"Depreciacion").Tables("Depreciacion").Copy
```

// Si el numero de comprobante no existe arroja el mensaje diciendo que no existe ese numero de comprobante.

```
If mTabla.Rows.Count = 0 Then  
    MsgBox("El numero de Comprobante no existe ...", vbCritical)  
    mNoe = True
```

// si es caso contrario mostrara la informacion de ese comprobante.

```
Else  
    Me.DvDepreciacion.DataSource = mTabla  
    Me.DvDepreciacion.Columns(0).HeaderText = "ID"  
    Me.DvDepreciacion.Columns(0).Width = 5  
    Me.DvDepreciacion.Columns(0).Visible = False  
    Me.DvDepreciacion.Columns(1).HeaderText = "NUMERO COMPROBANTE"  
    Me.DvDepreciacion.Columns(1).Width = 100  
    Me.DvDepreciacion.Columns(2).HeaderText = "CUENTAS PUC"  
    Me.DvDepreciacion.Columns(2).Width = 120  
    Me.DvDepreciacion.Columns(3).HeaderText = "SUPER CATEGORIA"  
    Me.DvDepreciacion.Columns(3).Width = 310  
    Me.DvDepreciacion.Columns(4).HeaderText = "VALOR"  
    Me.DvDepreciacion.Columns(4).Width = 120  
    Me.DvDepreciacion.Columns(4).DefaultCellStyle.Alignment =  
DataGridViewContentAlignment.MiddleRight  
    Me.DvDepreciacion.Columns(4).DefaultCellStyle.Format = "C2"  
End If
```

End If
End Sub

// se define el boton imprimir sirve para imprimir el comprobante. Este reporte se genera por medio de Crystal Report

```
Private Sub BtPrint_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles BtPrint.Click
    Dim CrReport As New CrystalDecisions.CrystalReports.Engine.ReportDocument
    mNoe = False
    Try
```

// verifica que el formato del reporte exista en la ruta.

```
If File.Exists("C:\Inventarios\REPORTES\Depreciacion.Rpt") Then
    CrReport = New CrystalDecisions.CrystalReports.Engine.ReportDocument()
    CrReport.Load("C:\Inventarios\REPORTES\Depreciacion.Rpt")
    CrReport.Database.Dispose()
    CrReport.SetParameterValue("Idn", Val(Me.TxtBuscar.Text))
    CrReport.SetDatabaseLogon("", "Inventarios")
    FPrint.CrVisor.ReportSource = CrReport
    FPrint.Show()
```

// si no existe arroja el mensaje que ese formato no existe.

```
Else
    MessageBox.Show("Error: No existe el reporte Depreciacion.Rpt, Por favor consulte Administrador", "Mostrando Reporte", MessageBoxButtons.OK, MessageBoxIcon.Error)
End If
```

// Imprime el reporte.

```
Catch ex As Exception
    MessageBox.Show("Excepción: " & ex.Message, "Mostrando Reporte")
End Try
End Sub
End Class
```