



CORP. SOLUSGT -PROYECTO 1

Grupo # 13

Contenido

Introducción	2
Objetivos.....	3
Objetivos Generales	3
Objetivos Específicos.....	3
Contenido técnico	0
Diagrama entidad relación	0
Diseño del programa desarrollado en Visual Basic	1

Introducción

La Corporación GT solicita un diseño e implementación de un sistema administrativo y financiero que pueda realizar la creación y el manejo de las cuentas de sus clientes, utilizando una base de datos, Formularios, basados en Microsoft Office.

Objetivos

Objetivos Generales

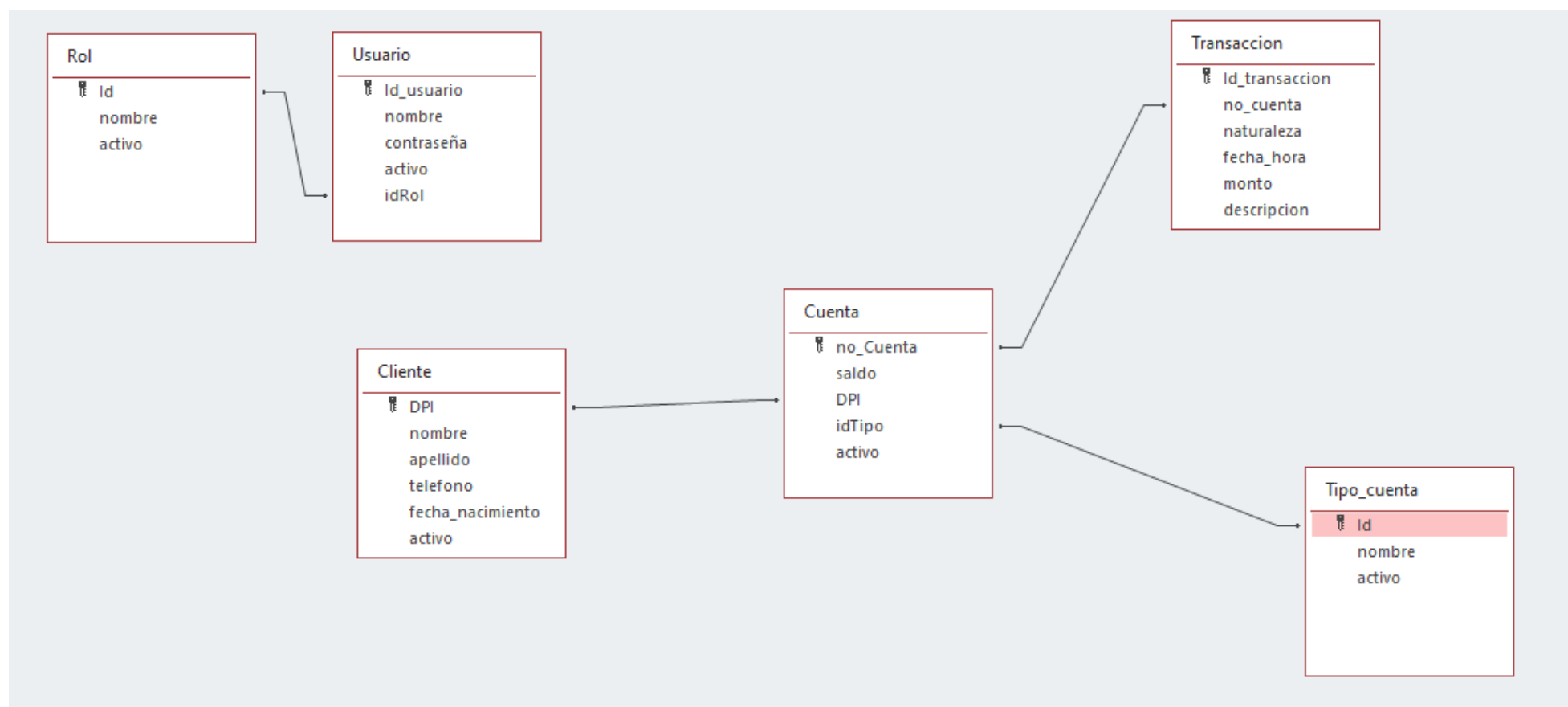
- Realizar un sistema administrativo y financiero optimo para el uso de la empresa

Objetivos Específicos

- Realizar una base de datos que tenga el control del sistema
- Implementar la base de datos para el manejo del sistema Administrativo y Financiero
- La aplicación debe ser sencilla y fácil para el usuario.

Contenido técnico

Diagrama entidad relación



Diseño del programa desarrollado en Visual Basic

Funcionalidad de la aplicación

El Usuario se encargara del uso y manejo del manejo de las cuentas en la cual va registrado y va dirigido a la base de datos de la aplicación.

Conexión conexión a la base de datos pat->ruta del archivo, dba-> nombre de la base de datos

```
Sub conectar()  
Call obtener  
  
Set cone = New ADODB.Connection  
On Error GoTo Salir  
With cnn  
    .Provider = "Microsoft.ACE.OLEDB.12.0"  
    .ConnectionString = "Data Source=" & pat & dba:  
  
    C_Error = True  
    .Open  
End With  
Exit Sub  
  
Salir:  
    C_Error = False  
    MsgBox Err.Description  
  
End Sub
```

Desconectar

```
Sub cerrar()  
    If cone.State = 1 Then  
        cnn.Close  
        Set cnn = Nothing  
    End If  
  
End Sub
```

Mostrar Clientes

Muestra todos los registros de clientes, en la hoja clientes, recorre uno por uno y lo ingresa en una columna

```
Sub MostrarClientes()  
    Dim cone As New ADODB.Connection  
    Dim rst As New ADODB.Recordset  
    Dim qry As String  
    Dim datos As Object  
  
    qry = "SELECT * FROM Cliente"  
  
    Set datos = cone.Execute(qry)  
  
    cont = 4  
    Do While Not datos.EOF  
        Hoja4.Cells(cont, 1) = datos.Fields(0)  
        Hoja4.Cells(cont, 2) = datos.Fields(1)  
        Hoja4.Cells(cont, 3) = datos.Fields(2)  
        Hoja4.Cells(cont, 4) = datos.Fields(3)  
        Hoja4.Cells(cont, 5) = datos.Fields(4)  
        cont = cont + 1  
        datos.MoveNext  
    Loop  
    datos.Close  
    cone.Close  
End Sub
```

Mostrar Cuenta

Muestra todos los registros de cuenta, en la hoja cuentas, recorre uno por uno y lo ingresa en una columna.

```
Sub MostarCuentas()  
    Dim cone As New ADODB.Connection  
    Dim rst As New ADODB.Recordset  
    Dim qry As String  
    Dim datos As Object  
  
    |  
  
    qry = "SELECT Cuenta.no_Cuenta, Cuenta.DPI, Cliente.nombre, Cliente.apellido, Cuenta.saldo FROM Cliente INNER JOIN Cuenta ON Cliente."  
  
    Set datos = cone.Execute(qry)  
  
    cont = 4  
    Do While Not datos.EOF  
        Hoja5.Cells(cont, 1) = datos.Fields(0)  
        Hoja5.Cells(cont, 2) = datos.Fields(1)  
        Hoja5.Cells(cont, 3) = datos.Fields(2)  
        Hoja5.Cells(cont, 4) = datos.Fields(3)  
        Hoja5.Cells(cont, 5) = datos.Fields(4)  
        cont = cont + 1  
        datos.MoveNext  
    Loop  
    datos.Close  
    cone.Close  
End Sub
```

Consultas importantes

1. Nuevo cliente

```
qry = "INSERT INTO Cliente VALUES (" & RegistrarClientes.TextBox1.Value & "," & RegistrarClientes.TextBox2.Value & |  
    "," & RegistrarClientes.TextBox3.Value & "," & RegistrarClientes.TextBox4.Value & "," & RegistrarClientes.DTPicker1.Value & ",true)"
```

2. Nueva cuenta

```
qry = "INSERT INTO Cuenta(saldo,DPI,idTipo,activo) VALUES ('" & CrearCuenta.TextBox2 & "','" & CrearCuenta.TextBox1 & "',2,TRUE)"
```

3. Registrar transacción: Registra el tipo de transacción realizada (Retiro, Deposito, Transferencia). Ejemplo:

```
qry = "INSERT INTO Transaccion ( no_cuenta,naturaleza,fecha_hora,monto) VALUES (" & noC & "','" & naturaleza & "',NOW(),' & monto & '"
```

a. Retiro: modifica la cuenta, restándole el monto retirado

```
qry = " UPDATE cuenta SET cuenta.saldo = cuenta.saldo-' & Saldo & "' WHERE (((cuenta.[no_Cuenta])=" & nCuenta & "));"
```

b. Deposito: modifica la cuenta, sumandole el monto depositado

```
qry = " UPDATE cuenta SET cuenta.saldo = cuenta.saldo+' & Saldo & "' WHERE (((cuenta.[no_Cuenta])=" & nCuenta & "));"
```


- 4. Comprobante Deposito:** para generar el documento a imprimir comprobante se solicitan estos datos

```
qry = "SELECT Cliente.nombre, Cliente.apellido FROM Cliente INNER JOIN Cuenta ON Cliente.DPI = Cuenta.DPI WHERE No_cuenta =" & nCuenta
```

- 5. Comprobante Retiro:** para generar el documento a imprimir comprobante se solicitan estos datos

```
qry = "SELECT Cliente.nombre, Cliente.apellido FROM Cliente INNER JOIN Cuenta ON Cliente.DPI = Cuenta.DPI WHERE No_cuenta =" & nCuenta
```

6. Reporte 1

```
SELECT TOP 5 Cliente.nombre, Cliente.apellido, Sum(Transaccion.monto) AS SumaDemonto,  
Cuenta.DPI FROM (Cliente INNER JOIN Cuenta ON Cliente.DPI = Cuenta.DPI) INNER JOIN  
Transaccion ON Cuenta.no_Cuenta = Transaccion.no_cuenta WHERE  
(((Transaccion.naturaleza)='Deposito')) GROUP BY Cliente.nombre, Cliente.apellido, Cuenta.DPI  
ORDER BY Sum(Transaccion.monto) DESC;"
```

7. Reporte 2

```
qry = "SELECT TOP 5 Cliente.nombre, Cliente.apellido, Sum(Cuenta.saldo) AS SumaDesaldo,  
Cliente.DPI FROM Cliente INNER JOIN Cuenta ON Cliente.DPI = Cuenta.DPI GROUP BY  
Cliente.nombre, Cliente.apellido, Cliente.DPI ORDER BY Sum(Cuenta.saldo), Cliente.DPI;"
```

Validaciones

1. Texto

Verifica si el texto ingresado es de tipo texto

```
If Not (KeyAscii >= 97 And KeyAscii <= 122 Or KeyAscii >= 65 And KeyAscii <= 90) Then
    KeyAscii = 0
    MsgBox "en el campo Nombre Solo se aceptan Letras"

End If
```

2. Numero entero

Verifica si el texto ingresado es de tipo numero

```
If Not (KeyAscii >= 48 And KeyAscii <= 57) Then
    KeyAscii = 0
    MsgBox "en el campo DPI Solo se aceptan numeros"

End If
```

3. Numero Decimal

Verifica si el texto ingresado es de tipo numero, si no lo es, el texto se remplaza por un vacio, si es punto y solo hay uno lo ingresa, si no remplaza un vacio

```
Dim caracter As Variant
Dim largo As Integer
Dim punto As Integer
Dim i As Integer
punto = 0
largo = VBA.Len(Texto)
For i = 1 To largo
    caracter = VBA.Mid(CStr(Texto), i, 1)
    If caracter <> "" Then
        If caracter = VBA.Chr(44) Then
            punto = punto + 1
            If punto > 1 Then
                Texto = Application.WorksheetFunction.Replace(Texto, i, 1, "")
                SoloDecimal = Texto
                punto = 0
                MsgBox "los decimales se ingresa con una coma"

            End If
        Else
            If caracter < VBA.Chr(48) Or caracter > VBA.Chr(57) Then
                Texto = VBA.Replace(Texto, caracter, "")
                SoloDecimal = Texto
                MsgBox "solo se pueden ingresar decimales"

            End If
        End If
    End If
Next i

SoloDecimal = Texto
```

Reportes:

Reporte 1: después de realizar la consulta se ingresan los datos uno a uno por columna el cual se ingresa en la columna 2, luego de terminar de llenar datos se selecciona el rango y se procede a realizar la grafica de barras

```
cont = 2
Do While Not datos.EOF
    Hoja6.Cells(cont, 1) = datos.Fields(0) & " " & datos.Fields(1)
    Hoja6.Cells(cont, 2) = datos.Fields(2)

    cont = cont + 1
    datos.MoveNext
Loop
datos.Close
cone.Close

Hoja6.Shapes.AddChart.Select
ActiveChart.SetSourceData Source:=Hoja6.Range("A2:B6")
```

Reporte 2: después de realizar la consulta se ingresan los datos uno a uno por columna el cual se ingresa en la columna 2, luego de terminar de llenar datos se selecciona el rango y se procede a realizar la grafica de pie

```
cont = 2
Do While Not datos.EOF
    Hoja7.Cells(cont, 1) = datos.Fields(0) & " " & datos.Fields(1)
    Hoja7.Cells(cont, 2) = datos.Fields(2)

    cont = cont + 1
    datos.MoveNext
Loop
datos.Close
cone.Close

Dim chrt As ChartObject

Set chrt = Hoja7.ChartObjects.Add(Left:=300, Width:=270, Top:=7, Height:=210)
chrt.Chart.SetSourceData Source:=Hoja7.Range("A1:B5")
chrt.Chart.ChartType = xlPie
```