



Universidad Autónoma de Querétaro
Facultad de Ingeniería
Licenciatura en Matemáticas
Aplicadas

Aplicación para el control de ventas y análisis de información en Microsoft
Excel con desarrollo en VBA (Visual Basic For Applications)

TESIS

Que como parte de los requisitos para obtener el grado de
Licenciado en Matemáticas Aplicadas

Presenta:

Briann Coronado Trejo

Dirigido por:

MMA Ivan González García

Centro Universitario

Querétaro, Qro. 2021

México



Dirección General de Bibliotecas y Servicios Digitales
de Información



Aplicación para el control de ventas y análisis de
información en Microsoft
Excel con desarrollo en VBA (Visual Basic For
Applications)

por

Briann Coronado Trejo

se distribuye bajo una [Licencia Creative Commons
Atribución-NoComercial-SinDerivadas 4.0
Internacional](https://creativecommons.org/licenses/by-nc-nd/4.0/).

Clave RI: IGLIN-219568



Universidad Autónoma de Querétaro
Facultad de Ingeniería
Licenciatura en Matemáticas
Aplicadas

Aplicación para el control de ventas y análisis de información en Microsoft
Excel con desarrollo en VBA (Visual Basic For Applications)

TESIS

Que como parte de los requisitos para obtener el grado de
Licenciado en Matemáticas Aplicadas

Presenta: Briann Coronado Trejo

Dirigido por: MMA Ivan González García

MMA Ivan González García	
_____	_____
Presidente	Firma
Dr. Víctor Antonio Aguilar Arteaga	
_____	_____
Secretario	Firma
Dr. Roberto Augusto Gomez Loenzo	
_____	_____
Vocal	Firma
M en C Juan Pablo Ramirez Vazquez	
_____	_____
Suplente	Firma

Centro Universitario
Querétaro, Qro. 2021

México

Resumen

Esta tesis se enfoca en crear una solución a los problemas posteriormente mencionados de la empresa Duran & Duran Consulting mediante Microsoft Excel y con el desarrollo de VBA (Visual Basic for Applications). Existen 3 principales problemas a solucionar:

1. La generación y manipulación de una base de datos estructurada y consistente de clientes y proveedores.
2. El desarrollo de una interfaz gráfica para la correcta y controlada captación de información del área de ventas.
3. Generar algoritmos para el análisis de la información obtenida y crear reportes automatizados que ayuden en la toma de decisiones.

Desarrollando todo lo anterior con una interacción amigable y de fácil razonamiento para los usuarios y o capturistas.

A mis padres —.

Aplicación para el control de ventas y
análisis de información en Microsoft
Excel con desarrollo en VBA
(Visual Basic For Applications)

Briann Coronado

Una persona que nunca cometió un error nunca intentó algo nuevo

Albert Einstein

Agradecimientos

A mi madre Gudelia y a mi padre Jorge por haberme dado el apoyo incondicional. A la Universidad Autónoma de Querétaro y a la Facultad de Ingeniería por ser mi casa durante cinco años. A la empresa Duran & Duran Consulting por haberme permitido realizar la investigación.

Contenido

Resumen.....	4
Agradecimientos	8
Introducción a Microsoft Excel y VBA	11
Capítulo 1	
Generación y manipulación de una base de datos estructurada de clientes y proveedores.....	18
Capítulo 2	
Desarrollo de una interfaz gráfica para la correcta y controlada captación de información del área de ventas	27
Capítulo 3	
Generación de algoritmos para el análisis de la información obtenida y creación reportes automatizados que ayuden en la toma de decisiones	46
Capítulo 4	
Conclusiones	48
Bibliografía	50

Introducción a Microsoft Excel y VBA

¿Qué es Microsoft Excel?

Microsoft Excel es un software de aplicación publicado por la empresa Microsoft, que brinda soporte digital a los labores contables, financieros, organizativos y de programación mediante hojas de cálculo. Forma parte del paquete Microsoft Office, que contiene diversos programas de oficina como Microsoft Word,

Microsoft PowerPoint, etc. Excel es posiblemente el programa más popular de su área en el mundo entero, publicado en diferentes versiones y actualizaciones desde su primera aparición en 1985. Entre su oferta de capacidades está la gestión de tablas, formatos y fórmulas matemáticas, así como funciones preprogramadas y su correspondiente conversión a gráficos de diversa naturaleza. Este programa pertenece a Microsoft bajo licencia comercial de uso y trabaja en los sistemas operativos de Windows, Macintosh y algunos otros, en diversos idiomas. Sin embargo, existen versiones paralelas como el paquete de código abierto OpenExcel o LibreExcel, pertenecientes a OpenOffice y LibreOffice respectivamente.

Historia de Microsoft Excel

El antecesor directo de Microsoft Excel fue un programa de manejo de hojas de cálculo publicado por Microsoft en 1982, de nombre Multiplan, que fue sumamente popular en ciertos sistemas, pero no lo suficiente para superar al Lotus 1-2-3 Development, una empresa competidora comprada por IBM en 1996. En 1985 apareció la primera versión de Excel para Mac, y la primera para Windows en 1987, dándole a la aplicación ventaja sobre el Lotus 1-2-3 que tardó mucho en migrar hacia dicho entorno virtual. Así, Excel se convirtió paulatinamente en el programa de hoja de cálculo por excelencia para PC, superando incluso al popular Quattro Pro de la empresa Borland.

¿Qué es Visual Basic?

Visual Basic es una aplicación y un lenguaje de programación desarrollados por Alan Cooper para Microsoft. Se origina en el clásico lenguaje BASIC. La primera versión salió en 1991 en un entorno relativamente sencillo para facilitar la creación de programas gráficos. Visual Basic, como su nombre lo indica, utiliza una interfaz totalmente visual. Actualmente, los programas creados en Visual Basic solo funcionan en Windows. La aplicación visual, permite crear ventanas, botones, menús, etc. De forma sencilla con solo arrastrar y soltar los elementos.

Luego se definen las apariencias, posiciones y comportamientos tanto de forma visual como utilizando código de programación. Este lenguaje toma elementos de diferentes paradigmas como el orientado a objetos y el orientado a eventos.

Visual Basic suele considerarse un sistema RAD (Rapid Application Development) porque permite crear aplicaciones de forma rápida, especialmente para prototipos, empleando Data Access Object, Remote Data Objects o ActiveX Data Objects, la creación de controles ActiveX y objetos. Los programas escritos en Visual Basic también pueden usar el API de Windows. Su última versión estable, la versión 6, fue lanzada en 1998. En abril de ese año Microsoft dejó de dar soporte al IDE de Visual Basic. De todas maneras, mantiene compatibilidad con las aplicaciones desarrolladas en Visual Basic para ser usadas en Windows Vista, Server 2008, Windows 7, Windows 8.1, Windows Server 2012 y Windows 10. El sucesor de Visual Basic es Visual Basic.Net.

Historia de Visual Basic

En 1990 se inicia el proyecto de nombre Basic Thunder. De hecho, Thunder estuvo presente hasta el último lanzamiento de Visual Basic con el nombre de la función interna primaria ThunderRTMain. En mayo 1991 se lanza Visual Basic 1.0 para DOS. El lenguaje no era completamente compatible con Visual Basic para Windows, utilizaba una interfaz de texto que simulaba una interfaz gráfica. En noviembre 1992 se lanza Visual Basic 2.0. El entorno de programación se hizo más fácil de utilizar y mejoró su velocidad. En 1993 se lanza Visual Basic 3.0 en dos versiones estándar y profesional. Visual Basic 3.0 incluía la versión 1.1 de Microsoft Jet Database Engine que podía leer y escribir bases de datos Jet (o Access). En agosto de 1995 se lanza Visual Basic 4.0, primera versión que podía crear programas para Windows tanto en 32 bits como en 16 bits. Tenía tres ediciones: estándar, profesional y empresarial. Introdujo la posibilidad de escribir clases que no eran GUI (Graphical User Interface). Tenía ciertas incompatibilidades con versiones diferentes de Visual Basic mientras que las versiones anteriores utilizaban controles VBX (Visual Basic Extension), este Visual Basic comenzó a usar controles OLE

(Object Linking and Embedding). Estos luego fueron renombrados como controles ActiveX. En febrero 1997 se lanza la versión Visual Basic 5.0, exclusivo para versiones de 32 bits de Windows. Aquellos programas escritos en Visual Basic 4.0 de 16 bits podían ser importados a Visual Basic 5.0. Se introdujo la posibilidad de crear controles de usuario personalizables, como también la habilidad de compilar en código ejecutable nativo de Windows, acelerando así la ejecución en ciertas tareas de cálculos intensivos. A mediados de 1998 se lanza Visual Basic 6.0, mejorando múltiples áreas, incluyó la habilidad de crear aplicaciones basadas en la web. El 31 de marzo de 2005 se termina el soporte para Visual Basic 6.0. En marzo de 2008 se finalizó el soporte extendido para Visual Basic 6.0, de todas maneras, la compatibilidad de los programas creados en Visual Basic está garantizada al menos hasta Windows 10.

Justificación

La empresa Duran & Duran Consulting cuyo giro se centraliza en las áreas de asesoramiento financiero, contable y de construcción se vio beneficiada con el proyecto al poder analizar, observar y modificar información desde las áreas de las bases de datos de clientes y proveedores, ventas y hasta la de análisis y retroalimentación de los datos.

El propósito del proyecto se vio generado por la necesidad de un programa propio en la empresa. Al ver la oportunidad de desarrollo en el proyecto utilizando herramientas como estadística, programación y optimización se tomó la decisión de desarrollar un programa para resolver los problemas cotidianos en la empresa. Al observar que la información de Duran & Duran Consulting se encontraba en su mayoría en Microsoft Excel, se empezó desarrollando formularios, clases y macros en Visual Basic para la manipulación y el análisis de esta información. Se consideró el diseño y los botones de la manera que fueran amigables y de fácil aprendizaje para el usuario.

Descripción del problema

Todo comienza con la necesidad de la empresa Duran & Duran Consulting de un programa que pudiera resolver 3 principales problemas.

- 1.- La generación y manipulación de una base de datos estructurada consistente de clientes y proveedores.
- 2.- La captura correcta de información en el área de ventas.
- 3.- El análisis de dicha información para generar reportes que ayude a la retroalimentación de la empresa y para poder tomar decisiones concretas con base en la información obtenida.

Desarrollando todo lo anterior con una interacción amigable y de fácil razonamiento para los usuarios y o capturistas.

Fundamentación teórica

La visión de este proyecto se limitará a desarrollar matemáticamente las siguientes áreas:

- a) Desarrollo de una base de datos estructurada y relacional de clientes y proveedores con algoritmos para la correcta creación, modificación y eliminación de elementos.
- b) Diseño de una estructura estándar para un historial equivalente a las cuentas bancarias con su debida creación de algoritmos para el correcto llenado de éste, así como la generación de controles de seguridad para efectos monetarios.
- c) Desarrollo de la automatización de reportes de la información obtenida para la retroalimentación.

Hipótesis y objetivos

Hipótesis.

Es posible desarrollar una interfaz gráfica de usuario en VBA con los requerimientos solicitados por la empresa para el control adecuado de la información y la gestión de bases de datos.

Objetivo general:

Crear una interfaz gráfica de usuario que permita el uso adecuado y la gestión a través de un software de fácil manipulación con Microsoft Excel de los diferentes tópicos administrativos de una empresa.

Objetivos particulares:

- 1.- Creación de una bases de datos estructurada y relacional de clientes y proveedores.
- 2.- Desarrollo de un modelo para el control de las ventas que a la vez sea un histórico de sus cuentas bancarias.
- 3.- Automatizar la obtención de reportes para el análisis de la información obtenida y brindar retroalimentación a la empresa para la toma de decisiones.
- 4.-Desarrollar algoritmos mediante VBA (Visual Basic for Applications) para la obtención fundamentada de información y posteriormente analizarla.

Metodología

Cabe resaltar que el proyecto ya se ha llevado a cabo, y se dividió en tres etapas descritas a continuación:

1.- Desarrollo de una base de datos estructurada de clientes y proveedores.

En esta etapa del proyecto se construyó en un libro de Excel habilitado para macros una estructura estándar en una hoja de cálculo bloqueada de Microsoft Excel tanto para clientes, proveedores y los elementos eliminados. Se dio la estructura donde la primera columna de la hoja de cálculo representaba el ID del elemento y las siguientes columnas representaban información importante para la empresa tales como razón social, registro federal del contribuyente, dirección, teléfono, email, vendedor, etc. Los renglones de las hojas representan los registros de clientes o proveedores.

Se desarrolló un menú de opciones para poder ejecutar las funciones tales como guardar un cliente, modificar un cliente, buscar un cliente, eliminar un cliente, agregar a proveedores, ordenar los clientes por razón social y ordenar los clientes por ID. Cada una de estas funciones despliega un formulario en donde se ingresa información, después se corren algoritmos que inicializan validando la información ingresada y posteriormente desbloquea la hoja de cálculo ejecuta la función y al final bloquea de nuevo la hoja de cálculo.

2.- Desarrollo de un modelo para el control de las ventas que a la vez sea un histórico de sus cuentas bancarias.

En la segunda etapa se desarrolló en otro libro de Excel habilitado para macros diferente al de la base de datos una estructura estándar para llevar a cabo un registro y un control de las cuentas bancarias de la empresa, cada hoja de cálculo representa una cuenta bancaria en donde las columnas representan elementos tales como un identificador para saber si es un depósito, retiro o una transferencia interna, un ID de procedencia cuando es una transferencia interna, un ID general, un número de operación, un estado del dinero depositado que puede estar en varios modos como disponible, cobrado parcialmente y cobrado, un indicador de vendedores, un estado de la comisión del vendedor, fecha de movimiento, número de cliente, razón social del cliente, un registro de cheque, depósito, retiro, saldo instantáneo, saldo total, etc.

Además, se generaron consultas al libro de base de datos que lee la información de los clientes y proveedores, se desarrolló un menú con distintas funciones tales como ingresar depósito, ingresar retiro, ingresar retiro pago de vendedores, actualizar pendientes, transferencia interna, generar orden de compra, generar recibo de efectivo, etc. Cada una de las funciones despliega un formulario en donde se ingresa información, después se corren los algoritmos para validar la información ingresada posteriormente se desbloquea la hoja de cálculo, se genera el registro y se bloquea la hoja de cálculo.

3.- Automatizar la obtención de reportes para el análisis de la información obtenida y brindar retroalimentación a la empresa para la toma de decisiones.

En esta última etapa se desarrollaron en un libro de Excel habilitado para macros consultas al libro de ventas para poder leer la información en tiempo real de manera independiente y se creó un menú con funciones tales como calcular estadísticas de ingresos, calcular estadísticas de egreso, generar reporte de gastos, generar reporte de pago de comisiones, etc. Cada una de las funciones tiene como únicos argumentos un intervalo de tiempo (fecha inicial y fecha final) para poder generar los reportes. Los reportes se presentaron mediante tablas con gráficos para un mejor entendimiento.

Capítulo 1

Generación y manipulación de una base de datos estructurada de clientes y proveedores

En este capítulo vamos a definir la información requerida por la empresa de sus clientes y proveedores para dimensionar la base de datos, así mismo vamos a definir un conjunto de funciones para la manipulación de dicha información y se desarrollara una interfaz en forma de menú que las contenga para su interacción con el capturista.

Se va a crear un libro de Microsoft Excel en el cual las bases de datos tanto de clientes como de proveedores serán consideradas como tablas independientes desarrolladas en una hoja de cálculo, vistas de la forma como un arreglo de filas y columnas para alojar los datos. A continuación, listaremos algunas definiciones de los conceptos básicos de Microsoft Excel.

Definición 1.0.1 Un libro de trabajo es un archivo de Microsoft Excel con extensión .XLSX el cual está formado por una o varias hojas de cálculo, el número de hojas dentro del libro dependerá de la cantidad de memoria disponible. El nombre del libro de trabajo es por definición “Libro1”.

Definición 1.0.2 Una hoja de cálculo es una tabla formada por 1,048,576 filas y 16,384 columnas, en donde las filas son enumeradas y las columnas se nombran en orden alfabético.

Definición 1.0.3 Una celda es la intersección de una fila y una columna en una hoja de cálculo, en donde cada celda se denomina mediante el nombre de la columna seguida por el número de fila.

Una vez creado el libro de trabajo, lo llamaremos BASE DE DATOS, crearemos una hoja de cálculo llamada Clientes y otra hoja llamada Proveedores. A continuación, definiremos la información relevante de un cliente en donde cada categoría representará una columna y cada cliente será una fila. En la imagen siguiente enlistaremos los campos de la hoja.

NÚMERO
RAZÓN SOCIAL
RFC
VENDEDOR
PORCENTAJE
CONTACTO
NOMBRE COMERCIAL
RESIDENCIA FISCAL
NÚMERO DE REGISTRO
TRIBUTARIO
USO CFDI
TELÉFONO 1
CONTACTO 1
TELÉFONO 2
CONTACTO 2
E-MAIL 1
E-MAIL 2
E-MAIL 3
PAÍS
CP
ESTADO
MUNICIPIO
LOCALIDAD
COLONIA
CALLE
NÚMERO EXTERIOR
NÚMERO INTERIOR
REFERENCIA

Obteniendo un total de 27 campos en donde a cada columna se le asignará el tipo de dato que contendrá de acuerdo con el campo en la pestaña de inicio en el apartado de formato de celdas. Le daremos formato de tabla para distinguir de manera visual las diferentes filas y le pondremos color. La siguiente imagen muestra la hoja de clientes.

	A	B	C	D	E	F	G	H	I	J	K	L
1	NÚMERO	RAZÓN SOCIAL	RFC	VENDEDOR	PORCENTAJE	CONTACTO	NOMBRE COMERCIAL	RESIDENCIA FISCAL	NÚMERO DE REGISTRO	USO CFDI	TELÉFONO 1	CONTACTO 1
2												
3												
4												
5												
6												
7												
8												

De esta manera concluimos con la estructura de la base de datos de clientes, análogamente creamos una hoja de proveedores y eliminados. Esta última nos servirá para tener un registro de los clientes eliminados en la base de datos.

1.1. Desarrollo del menú de funciones



Se diseñó la hoja principal del libro llamada Menú la cual aloja 8 botones en donde cada uno activa una función específica por medio de macros.

Descripción de funciones:

a) GUARDAR CLIENTE:

Esta función sirve para capturar un nuevo cliente. Al dar clic en el botón se nos abrirá un formulario de Excel con los campos que muestra la imagen siguiente, los cuales marcados con * son obligatorios

GUARDAR CLIENTE

NOTA: AL INGRESAR UNA NUEVA PERSONA FISICA POR FAVOR INTRODUCE LA RAZÓN SOCIAL EN EL SIGUIENTE ORDEN: NOMBRE-APELLIDOS.

*R.F.C.: |

*RAZÓN SOCIAL: |

*NOMBRE COMERCIAL: |

*CALLE: |

*NÚMERO EXTERIOR: |

NÚMERO INTERIOR: |

LOCALIDAD: |

*COLONIA: |

MUNICIPIO: |

*ESTADO: |

*PAÍS: |

*C.P.: |

*EMAIL-1: |

EMAIL-2: |

EMAIL-3: |

*TELÉFONO-1: |

*CONTACTO-1: |

TELÉFONO-2: |

CONTACTO-2: |

*VENDEDOR: |

RESIDENCIA FISCAL: |

NÚMERO REGISTRO TRIBUTARIO: |

USO CFDI: |

REFERENCIA: |

*PORCENTAJE: |

CONTACTO: |

*TIPO: |

GUARDAR FINALIZAR

Dentro del desarrollo del formulario se especifica diseño, colores y cuadros de texto que aparecen en el mismo y se definen los procedimientos a ejecutar al inicializar el formulario:

```

1 Private Sub UserForm_Initialize()
2 TextBox1.SetFocus
3 With ComboBox1
4 .AddItem "CLIENTE"
5 .AddItem "PROVEEDOR"
6 .AddItem "AMBOS"
7 End With
8 With ComboBox2
9 .AddItem "Vendedor 1"
10 .AddItem "Vendedor 2"
11 .AddItem "Vendedor 3"
12 .AddItem "Vendedor 4"
13 .AddItem "Vendedor 5"
14
15 End With
16
17 End Sub

```

El código anterior posiciona el puntero en el primer cuadro de texto del formulario que es el RFC y llena las listas correspondientes a los cuadros de vendedor y tipo.

Al dar clic en el botón de guardar se activa la macro que ejecuta el código para guardar un cliente, esté primero revisa que la información ingresada en el formulario sea correcta y que los campos obligatorios estén llenos.

A continuación, parte del código de la macro en VBA.

```

1 Private Sub GUARDAR_Click()
2 If (ComboBox1 <> "CLIENTE") And (ComboBox1 <> "PROVEEDOR") And (ComboBox1 <> "AMBOS") Then
3 ComboBox1 = Empty
4 End If
5 Dim bandera5 As Integer
6 bandera5 = 0
7 If (TextBox1 = Empty) Or (TextBox2 = Empty) Or (TextBox3 = Empty) Or (TextBox4 = Empty) Or (TextBox5 = Empty) Or (TextBox8 = Empty) Or (TextBox10 = Empty) Or (TextBox11 = Empty) Or
8 (TextBox12 = Empty) Or (TextBox13 = Empty) Or (TextBox16 = Empty) Or (TextBox17 = Empty) Or (TextBox25 = Empty) Or (ComboBox2 = Empty) Or (ComboBox1 = Empty) Then
9 MsgBox "PARA GUARDAR UN NUEVO CLIENTE EN NECESARIO INTRODUCIR TODOS LOS CAMPOS MARCADOS CON (*), GRACIAS.", , "REQUISITO"
10 Else
11 ThisWorkbook.Sheets("CLIENTES").Unprotect ("CLIENTES")
12 ThisWorkbook.Sheets("PROVEEDORES").Unprotect ("PROVEEDORES")
13
14 TextBox1 = UCase(TextBox1)
15 ultimo_renglon = ThisWorkbook.Sheets("CLIENTES").Cells(Rows.Count, "A").End(xlUp).Row
16 If InStr(TextBox25, ",") > 0 Then
17 TextBox25.Value = Replace(TextBox3, ",", ".")
18 End If
19 If (IsNumeric(TextBox25) = False) Or ((TextBox25 >= 0) = False) Or ((TextBox25 <= 100) = False) Then
20 MsgBox "PORCENTAJE SOLO ADMITE UN NUMERO ENTRE 0 Y 100 GRACIAS.", , "ERROR"
21 bandera5 = 1
22 End If

```

Una vez revisando que los cuadros de texto obligatorios vengan no vacíos se declaran las banderas a usar en el procedimiento y se desbloquean las hojas de cálculo tanto de

```

1 |'Declaración de las variables
2 Dim rango_numeros As Range
3 Dim ultima_fila As Long
4 Dim max_numero As Long
5 ultima_fila = ultimo_renglon
6 Set rango_numeros = Range(ThisWorkbook.Sheets("CLIENTES").Cells(2, 1), ThisWorkbook.Sheets("CLIENTES").Cells(ultima_fila, 1))
7 max_numero = Application.WorksheetFunction.Max(rango_numeros)
8 Dim bandera, bandera4 As Integer
9 bandera = 0
10 Dim i As Long
11 Dim RFC_C As String
12 For i = 2 To ultimo_renglon
13 RFC_C = ThisWorkbook.Sheets("CLIENTES").Cells(i, 3).Value
14 If (RFC_C = TextBox1) Then
15 MsgBox "EL CLIENTE A GUARDAR YA EXISTE EN LA BASE DE DATOS", , "ERROR"
16 bandera = 1
17 Exit For
18 End If
19 Next i

```

clientes como de proveedores y se verifica que el RFC no sea uno existente. Se encuentra el número del renglón del último registro para escribir la información a guardar en uno posterior.

A continuación, se muestra el código que calcula el número máximo de cliente y le aumenta uno para asignárselo al nuevo cliente y escribe en la hoja de cálculo la información ingresada en el formulario insertando un renglón entre el 1 y 2.

```
1 Dim numero As Integer
2 numero = max_numero + 1
3 Application.Goto ThisWorkbook.Sheets("CLIENTES").Range("A2")
4 Selection.EntireRow.Insert
5 With ThisWorkbook.Sheets("CLIENTES")
6   .Cells(2, 1).Value = numero &"número"
7   .Cells(2, 2).Value = UCase(TextBox2) &"razón social"
8   .Cells(2, 3).Value = UCase(TextBox1) &"rfc"
9   .Cells(2, 4).Value = UCase(ComboBox2) &"vendedor"
10  .Cells(2, 5).Value = UCase(TextBox25) &"porcentaje"
11  .Cells(2, 6).Value = UCase(TextBox26) &"contacto"
12  .Cells(2, 7).Value = UCase(TextBox3) &"nombre comercial"
13  .Cells(2, 8).Value = UCase(TextBox21) &"residencia fiscal"
14  .Cells(2, 9).Value = UCase(TextBox22) &"número de registro tributario"
15  .Cells(2, 10).Value = UCase(TextBox23) &"uso cfdi"
16  .Cells(2, 11).Value = UCase(TextBox16) &"teléfono 1"
17  .Cells(2, 12).Value = UCase(TextBox17) &"contacto 1"
18  .Cells(2, 13).Value = UCase(TextBox18) &"teléfono 2"
19  .Cells(2, 14).Value = UCase(TextBox19) &"contacto 2"
20  .Cells(2, 15).Value = TextBox13 &"email 1"
21  .Cells(2, 16).Value = TextBox14 &"email 2"
22  .Cells(2, 17).Value = TextBox15 &"email 3"
23  .Cells(2, 18).Value = UCase(TextBox11) &"pais"
24  .Cells(2, 19).Value = UCase(TextBox12) &"c.p"
25  .Cells(2, 20).Value = UCase(TextBox10) &"estado"
26  .Cells(2, 21).Value = UCase(TextBox9) &"municipio"
27  .Cells(2, 22).Value = UCase(TextBox7) &"localidad"
28  .Cells(2, 23).Value = UCase(TextBox8) &"colonia"
29  .Cells(2, 24).Value = UCase(TextBox4) &"calle"
30  .Cells(2, 25).Value = UCase(TextBox5) &"número exterior"
31  .Cells(2, 26).Value = UCase(TextBox6) &"número interior"
32  .Cells(2, 27).Value = UCase(TextBox24) &"referencia"
```

Posteriormente en la última parte del proceso se boquean todas las hojas del libro.

```
1 ThisWorkbook.Sheets("CLIENTES").Protect ("CLIENTES")
2 ThisWorkbook.Sheets("MENÚ").Protect ("MENU")
3 ThisWorkbook.Sheets("ELIMINADOS").Protect ("ELIMINADOS")
4 ThisWorkbook.Sheets("PROVEEDORES").Protect ("PROVEEDORES")
```

En caso de querer cerrar el formulario está el botón de finalizar que manda ejecutar el siguiente código.

```
1 Private Sub FINALIZAR_Click()
2   Sheets("CLIENTES").Protect ("CLIENTES")
3   Sheets("PROVEEDORES").Protect ("PROVEEDORES")
4   End
5   End Sub
```

El cual bloquea las páginas de clientes y proveedores y cierra el formulario.

b) MODIFICAR CLIENTE:

Esta función sirve para modificar la información existente de un cliente.

MODIFICAR ×

**NOTA: SOLO SE PODRÁ MODIFICAR UN CLIENTE BUSCÁNDOLO POR MEDIO DE LA RAZÓN SOCIAL.
PASO 1: INTRODUCE LA RAZÓN SOCIAL.
PASO 2: ACTUALICE LOS DATOS Y PRESTONE EL BOTÓN MODIFICAR PARA GUARDAR LOS CAMBIOS.
NOTA: PARA LIMPIAR LAS CASILLAS DE TEXTO PRESIONE EL BOTÓN LIMPIAR.**

*RAZÓN SOCIAL:

*R.F.C.:

*NOMBRE COMERCIAL:

*CALLE:

*NÚMERO EXTERIOR:

NÚMERO INTERIOR:

LOCALIDAD:

*COLONIA:

MUNICIPIO:

*ESTADO:

*PAÍS:

*C.P.:

*EMAIL-1:

EMAIL-2:

EMAIL-3:

*TELÉFONO-1:

*CONTACTO-1:

TELÉFONO-2:

CONTACTO-2:

*VENDEDOR:

RESIDENCIA FISCAL:

NÚMERO REGISTRO TRIBUTARIO:

USO CFDI:

REFERENCIA:

PORCENTAJE:

CONTACTO:

En este formulario de igual forma se ejecutan códigos al inicializarlo para posicionar el puntero en el primer cuadro de texto que es la razón social, este cuadro de texto se inicializa como una lista y se llena con el código siguiente.

```
1 Private Sub UserForm_Initialize()  
2  
3     ComboBox1.SetFocus  
4     With ComboBox2  
5         .AddItem "Vendedor 1"  
6         .AddItem "Vendedor 2"  
7         .AddItem "Vendedor 3"  
8         .AddItem "Vendedor 4"  
9         .AddItem "Vendedor 5"  
10    End With  
11  
12    Dim rango_rfc, rango_rz, rz, rfcl As Range  
13    Dim ultimo_renglon As Long  
14    ultimo_renglon = ThisWorkbook.Sheets("CLIENTES").Cells(Rows.Count, "A").End(xlUp).Row  
15    Set rango_rfc = Range(ThisWorkbook.Sheets("CLIENTES").Cells(2, 2), ThisWorkbook.Sheets("CLIENTES").Cells(ultimo_renglon, 2))  
16  
17    For Each rfcl In rango_rfc  
18        ComboBox1.AddItem rfcl.Value  
19    Next rfcl  
20 End Sub
```

El cuadro de texto de razón social hace una búsqueda inteligente de forma que al escribir las primeras letras muestra las razones sociales que empaten con lo escrito y llena el formulario para visualizar la información completa del cliente seleccionado en la lista.

El siguiente código revisa si la razón social está vacía entonces limpia el formulario, de lo contrario desbloquea la hoja de clientes y localiza el último renglón con información.

```

1 Private Sub ComboBox1_Change()
2     If (ComboBox1 = Empty) Then
3         TextBox1 = Empty 'RFC
4         ComboBox2 = Empty 'vendedor
5         TextBox25 = Empty 'porcentaje
6         TextBox26 = Empty 'contacto
7         TextBox3 = Empty 'nombre comercial
8         TextBox21 = Empty 'residencia fiscal
9         TextBox22 = Empty 'número de registro tributario
10        TextBox23 = Empty 'uso cfdi
11        TextBox16 = Empty 'teléfono 1
12        TextBox17 = Empty 'contacto 1
13        TextBox18 = Empty 'teléfono 2
14        TextBox19 = Empty 'contacto 2
15        TextBox13 = Empty 'email 1
16        TextBox14 = Empty 'email 2
17        TextBox15 = Empty 'email 3
18        TextBox11 = Empty 'pais
19        TextBox12 = Empty 'c.p
20        TextBox10 = Empty 'estado
21        TextBox9 = Empty 'municipio
22        TextBox7 = Empty 'localidad
23        TextBox8 = Empty 'colonia
24        TextBox4 = Empty 'calle
25        TextBox5 = Empty 'número exterior
26        TextBox6 = Empty 'número interior
27        TextBox24 = Empty 'referencia
28        ComboBox1.SetFocus
29     Else
30         ThisWorkbook.Sheets("CLIENTES").Unprotect ("CLIENTES")
31         ComboBox1 = UCase(ComboBox1)
32         ultimo_renglon = ThisWorkbook.Sheets("CLIENTES").Cells(Rows.Count, "A").End(xlUp).Row

```

Posteriormente hace una búsqueda mediante el RFC y si lo encuentra llena el formulario con la información del cliente y rompe el ciclo de búsqueda.

```

33 'Declaración de las variables
34 Dim bandera As Integer
35 bandera = 0
36 Dim i As Long
37 Dim RZ_C As String
38 For i = 2 To ultimo_renglon
39     RZ_C = ThisWorkbook.Sheets("CLIENTES").Cells(i, 2)
40     If (RZ_C = ComboBox1) Then
41         With ThisWorkbook.Sheets("CLIENTES")
42             TextBox1 = .Cells(i, 3).Value 'RFC
43             ComboBox2 = .Cells(i, 4).Value 'vendedor
44             TextBox25 = .Cells(i, 5).Value 'porcentaje
45             TextBox26 = .Cells(i, 6).Value 'contacto
46             TextBox3 = .Cells(i, 7).Value 'nombre comercial
47             TextBox21 = .Cells(i, 8).Value 'residencia fiscal
48             TextBox22 = .Cells(i, 9).Value 'número de registro tributario
49             TextBox23 = .Cells(i, 10).Value 'uso cfdi
50             TextBox16 = .Cells(i, 11).Value 'teléfono 1
51             TextBox17 = .Cells(i, 12).Value 'contacto 1
52             TextBox18 = .Cells(i, 13).Value 'teléfono 2
53             TextBox19 = .Cells(i, 14).Value 'contacto 2
54             TextBox13 = .Cells(i, 15).Value 'email 1
55             TextBox14 = .Cells(i, 16).Value 'email 2
56             TextBox15 = .Cells(i, 17).Value 'email 3
57             TextBox11 = .Cells(i, 18).Value 'pais
58             TextBox12 = .Cells(i, 19).Value 'c.p
59             TextBox10 = .Cells(i, 20).Value 'estado
60             TextBox9 = .Cells(i, 21).Value 'municipio
61             TextBox7 = .Cells(i, 22).Value 'localidad
62             TextBox8 = .Cells(i, 23).Value 'colonia
63             TextBox4 = .Cells(i, 24).Value 'calle
64             TextBox5 = .Cells(i, 25).Value 'número exterior
65             TextBox6 = .Cells(i, 26).Value 'número interior
66             TextBox24 = .Cells(i, 27).Value 'referencia
67         End With
68     Exit For

```

De no encontrar al cliente vuelve a limpiar el formulario y termina el procedimiento.

```
69 Else
70     TextBox1 = "" 'RFC
71     TextBox20 = "" 'vendedor
72     TextBox25 = "" 'porcentaje
73     TextBox26 = "" 'contacto
74     TextBox3 = "" 'nombre comercial
75     TextBox21 = "" 'residencia fiscal
76     TextBox22 = "" 'número de registro tributario
77     TextBox23 = "" 'uso cfdi
78     TextBox16 = "" 'teléfono 1
79     TextBox17 = "" 'contacto 1
80     TextBox18 = "" 'teléfono 2
81     TextBox19 = "" 'contacto 2
82     TextBox13 = "" 'email 1
83     TextBox14 = "" 'email 2
84     TextBox15 = "" 'email 3
85     TextBox11 = "" 'pais
86     TextBox12 = "" 'c.p
87     TextBox10 = "" 'estado
88     TextBox9 = "" 'municipio
89     TextBox7 = "" 'localidad
90     TextBox8 = "" 'colonia
91     TextBox4 = "" 'calle
92     TextBox5 = "" 'número exterior
93     TextBox6 = "" 'número interior
94     TextBox24 = "" 'referencia
95 End If
96 Next i
97 End If
98 End Sub
```

El botón modificar hace el mismo procedimiento de búsqueda y reemplaza la información del cliente existente con la nueva en el formulario y concluye bloqueando las hojas y mandando un mensaje de texto de “cliente modificado satisfactoriamente”. Por otro lado, el botón finalizar de igual forma bloquea las hojas y cierra el procedimiento.

c) **BUSCAR CLIENTE:**

Esta función sirve para mostrar en un formulario la información existente de un cliente buscándolo por su RFC o la razón social.

d) **AGREGAR A PROVEEDORES:**

Esta función copia la información de un cliente buscándolo por su RFC o la razón social y lo agrega a la hoja de proveedores.

e) **ELIMINAR CLIENTE:**

Esta función busca a un cliente por su RFC o la razón social y lo elimina de la hoja de clientes agregándolo a la hoja de eliminados.

f) **BASE DE CLIENTES:**

Esta función nos mueve del menú a la hoja de clientes.

g) **ORDENAR CLIENTES POR RAZÓN SOCIAL:**

Esta función ordena los clientes de la hoja de clientes de forma ascendente por la razón social.

h) **ORDENAR CLIENTES POR NÚMERO DE FORMA:**

Esta función ordena los clientes por el número de cliente de forma ascendente.

Capítulo 2

Desarrollo de una interfaz gráfica para la correcta y controlada captación de información del área de ventas

En este capítulo vamos a definir las reglas de negocio y los algoritmos que se ejecutaran en cada una de las funciones a desarrollar dentro del menú.

Se va a crear un libro de Microsoft Excel llamado VENTAS en el cual se tendrán las hojas de CLIENTES y PROVEEDORES conectadas al libro de BASE DE CLIENTES creado anteriormente mediante la red de internet para la lectura de información, además se crearán las hojas de orden de compra, póliza de cheque, retiro de vendedores, pendientes, información general, menú, y otro conjunto de hojas en donde cada una representará una cuenta bancaria.

Procederemos a definir primero la estructura que tendrán las hojas que representan una cuenta bancaria en donde cada hoja tendrá el nombre del banco seguido con su apodo de la cuenta, ejemplo "BANCO APODO".

Se dejarán 16 renglones disponibles para la clasificación de los estados de un ingreso o egreso, los vendedores, la información de la cuenta bancaria y el saldo total. En el renglón 17 a partir de la columna A colocaremos los siguientes conceptos columna por columna; Para mostrarlos se ponen de manera transpuesta:

	A
1	DEPOSITO (1) RETIRO (0) TRANSF (-1)
2	IDENTIFICADOR DE PROCEDENCIA
3	IDENTIFICADOR
4	NO.OPERACIÓN
5	ESTADO DEL CLIENTE, RETIRO O TRANSFERENCIA
6	VENDEDOR
7	PORCENTAJE % INTERVALO (2.5,100]
8	ESTADO DEL VENDEDOR
9	COMISIÓN DEL VENDEDOR \$
10	COMISIÓN RESTANTE DEL VENDEDOR \$
11	ESTADO DEL DESPACHO
12	COMISIÓN DEL DESPACHO (2.5 %) \$
13	COMISIÓN RESTANTE DEL DESPACHO \$
14	FECHA DE MOVIMIENTO
15	NO.CLIENTE
16	CLIENTE
17	CHEQUE
18	DEPOSITO
19	RETIRO
20	MÁXIMO A RETIRAR (CLIENTE)
21	SALDO INSTANTANEO \$
22	REFERENCIA
23	FECHA DE CAPTURA
24	TIPO DE RETIRO (CHEQUE O TRANSFERENCIA)
25	PERSONA QUE COBRÓ EL CHEQUE
26	RFC DEL CLIENTE
27	CATEGORÍA DEL RETIRO

En el sentido de que cada concepto representa una columna en donde inicia a partir del renglón 17, se definen los tipos de datos de todas las columnas como son cadenas, flotantes, enteros, etc.

De la misma forma se ponen los colores de los estados con su descripción en las columnas B y C a partir del renglón 2

	A	B	C
1			
2	ESTADOS		DISPONIBLE
3			COBRADO (SIN SALDO)
4			CANCELADO
5			TRANSFERENCIA
6			COBRADO PARCIALMENTE
7			RETIRO (CLIENTE)
8			RETIRO (VENDEDOR)
9			RETIRO (DESPACHO)
10			RETIRO (PAGO SERVICIOS)
11			PENDIENTE

A continuación, una breve explicación de los estados que puede tener un movimiento.

COLOR	ESTADO	DESCRIPCIÓN
	DISPONIBLE	Se le asigna el estado a un ingreso que tiene el 100% saldo para retirar
	COBRADO (SIN SALDO)	Se le asigna el estado cuando se retira todo el saldo de un ingreso
	CANCELADO	Se le asigna el estado cuando un movimiento es cancelado
	TRANSFERENCIA	Es un ingreso que hace referencia a una transferencia interna
	COBRADO PARCIALMENTE	Se le asigna el estado a un ingreso que aún tiene saldo para retirar pero que ya se le ha retirado antes
	RETIRO (CLIENTE)	Es un egreso de cliente
	RETIRO (VENDEDOR)	Es un egreso de vendedor
	RETIRO (DESPACHO)	Es un egreso de despacho
	RETIRO (PAGO SERVICIOS)	Es un egreso como un pago a servicios
	PENDIENTE	Es un ingreso que no se ha identificado

En la columna I y J se definen los nombres de los vendedores abreviados junto con sus comisiones inicializadas en 0 a partir del renglón 2 en donde también se define una comisión de despacho en la celda I16 inicializada en 0.

	I	J
	VENDEDOR	TOTAL \$
	1	0.00
✓	2	0.00
✓	3	0.00
✓	4	0.00
✓	5	0.00
✓	6	0.00
✓	7	0.00
✓	8	0.00
✓	9	0.00
✓	10	0.00
✓	11	0.00
✓	12	0.00
	TOTAL DESPACHO \$	0.00

A continuación, una imagen de la estructura de la hoja:

De esta forma para concluir con la estructura de la hoja también se inmoviliza el panel hasta el renglón 17, y se agrega una macro asociada al botón **volver al menú** que ejecuta el siguiente código con objetivo de movernos a la hoja de menú:

```

1 Sub VOLVER_AL_MENU ()
2 ThisWorkbook.Sheets ("MENÚ") .Activate
3 ThisWorkbook.Sheets ("MENÚ") .Cells (1, 1) .Activate
4 End Sub

```

Todas las hojas relacionadas a una cuenta bancaria deben tener el mismo formato y respetar la ubicación de los conceptos.

Definiremos ahora la hoja de pendientes que tiene la misma estructura que la de una hoja de cuenta bancaria excepto por la información de los estados, los vendedores, el saldo del despacho y el saldo total.

La hoja de retiro de vendedores se detalla la semana correspondiente al pago de comisiones ya que son semanales, la fecha del retiro, cuenta bancaria de los depósitos, importe, el número de cheque y el banco. Donde abajo se desglosa el vendedor que se le asigna el depósito, la fecha del depósito, cliente, porcentaje, importe del depósito y la comisión obtenida, esta estructura es más sencilla como se muestra a continuación:

	A	B	C	D	E	F	G	
1								
2	DISTRIBUCIÓN DE COMISIONES DE VENDEDORES							
3		SEMANA:		FECHA:				
4								
5		EMPRESA:						
6								
7		CHEQUE:		TOTAL:				
8		BANCO:						
9								
10								
11	VENDEDOR	FECHA DE DEPOSITO	CLIENTE	PORCENTAJE	DEPOSITO	COMISIÓN		
12								
13								

La hoja póliza de cheque se ejecuta cuando se realizan pagos a proveedores por este medio o para hacer retiros bancarios para compra de insumos en efectivo. A continuación, los conceptos incluidos en esta hoja.

	A	B	C	D	E	F	G	H	I	J	K
1										FOLIO:	329
2	PÓLIZA DE CHEQUE					COPIA DEL CHEQUE					
3									FECHA:		
4											
5	PAGUESE ESTE CHEQUE A:								\$		
6											
7	LA CANTIDAD DE:										
8						MONEDA NACIONAL					
9		CHEQUE NO.									
10		BANCO:									
11		CUENTA:									
12											
13											
14	CONCEPTO DE PAGO						QUIEN RECIBE CHEQUE				
15	R.CLIENTE						NOMBRE:		FECHA:		
16											
17							FIRMA:				
18											
19											
20	CUENTA		NOMBRE				DEBE		HABER		
21											
22											
23			COMISIÓN DE VENTA								
24			DESPACHO								
25											
26	ENTREGA:										
27	SALIDA:										
28	LLEGADA:										
29						SUMAS IGUALES:					
30											
31	CHEQUE ELABORADO POR:		PORCENTAJE:		VENDEDOR:			PERSONA QUE COBRO:			
32											
33											
34											
35	NÚMERO DE PÓLIZA:						DEPOSITO:				
36											
37											
38											
39	REFERENCIA:										
40											
41											
42											
43											
44										RFC:	

A continuación, se define la hoja orden de compra.

En esa hoja se llena información relevante sobre los artículos, cantidad, precio e importe a pagar a los proveedores por cada orden de compra y así tener un registro físico, el cual se le entrega a los directores para ser aprobado y posteriormente pagado.

	A	B	C	D	E	F	G		
1	ORDEN DE COMPRA								
2						FECHA	<input type="text"/>		
3									
4	PROVEEDOR								
5									
6									
7	ARTICULO	DESCRIPCIÓN	CANTIDAD	PRECIO/UNIDAD	TOTAL				
8				0.00	0.00				
9				0.00	0.00				
10				0.00	0.00				
11				0.00	0.00				
12				0.00	0.00				
13				0.00	0.00				
14				0.00	0.00				
15				0.00	0.00				
16				0.00	0.00				
17				0.00	0.00				
18				0.00	0.00				
19				0.00	0.00				
20				0.00	0.00				
21				0.00	0.00				
22				0.00	0.00				
23				0.00	0.00				
24				0.00	0.00				
25				0.00	0.00				
26				0.00	0.00				
27				0.00	0.00				
28				0.00	0.00				
29				0.00	0.00				
30				0.00	0.00				
31				0.00	0.00				
32				0.00	0.00				
33				0.00	0.00				
34				0.00	0.00				
35						SUBTOTAL	0.00		
36	COMENTARIOS O INSTRUCCIONES ESPECIALES								
37	<input type="text"/>								
38									
39								TOTAL	<input type="text"/>
40									
41									
42									
43									
44									
45	Director 1			Director 2					

Ahora la hoja de información general, en esta hoja se tiene una macro que hace un barrido sobre la información de los saldos totales, como comisiones de despacho y vendedores en todas las cuentas bancarias que va sumando esta información por concepto para que al finalizar la plame en los renglones correspondientes, así de esta forma se conoce el saldo de todas las cuentas, las comisiones disponibles, etc.

EMPRESA	SALDO TOTAL	SALDO DESPACHO	SALDO COMISIONES	VENDEDOR	ACUMULADAS HASTA EL MOMENTO DE TODAS LAS CUENTAS BANCARIAS
BANCO APODO 1	0.00	0.00	0.00	VENDEDOR 1	0.00
BANCO APODO 2	0.00	0.00	0.00	VENDEDOR 2	0.00
BANCO APODO 3	0.00	0.00	0.00	VENDEDOR 3	0.00
BANCO APODO 4	0.00	0.00	0.00	VENDEDOR 4	0.00
BANCO APODO 5	0.00	0.00	0.00	VENDEDOR 5	0.00
BANCO APODO 6	0.00	0.00	0.00	VENDEDOR 6	0.00
BANCO APODO 7	0.00	0.00	0.00	VENDEDOR 7	0.00
BANCO APODO 8	0.00	0.00	0.00	VENDEDOR 8	0.00
BANCO APODO 9	0.00	0.00	0.00	VENDEDOR 9	0.00
BANCO APODO 10	0.00	0.00	0.00	VENDEDOR 10	0.00
BANCO APODO 11	0.00	0.00	0.00	VENDEDOR 11	0.00
BANCO APODO 12	0.00	0.00	0.00	VENDEDOR 12	0.00
BANCO APODO 13	0.00	0.00	0.00		
BANCO APODO 14	0.00	0.00	0.00		
BANCO APODO 15	0.00	0.00	0.00		
BANCO APODO 16	0.00	0.00	0.00		

SALDO DE TODAS LAS CUENTAS	0.00
DESPACHO : COMISIONES ACUMULADAS HASTA EL MOMENTO	0.00

ACTUALIZAR INFORMACIÓN

A continuación, se muestra el código que ejecuta la macro de actualizar información:

Primeramente, se declaran e inicializan todas las variables y se desbloquea la hoja de información general:

```

1 Sub ACTUALIZAR_INFORMACION()
2
3
4 Dim hoja As String
5 Dim vendedor1, vendedor2, vendedor3, vendedor4, vendedor5, vendedor6, vendedor7, vendedor8, vendedor9, vendedor10, vendedor11, vendedor12,
6 despacho, comisiones As Double
7 vendedor1 = 0
8 vendedor2 = 0
9 vendedor3 = 0
10 vendedor4 = 0
11 vendedor5 = 0
12 vendedor6 = 0
13 vendedor7 = 0
14 vendedor8 = 0
15 vendedor9 = 0
16 vendedor10 = 0
17 vendedor11 = 0
18 vendedor12 = 0
19 despacho = 0
20 comisiones = 0
21 ThisWorkbook.Sheets("INFORMACIÓN GENERAL").Unprotect ("BRIANN_SISTEMA_2")
22 Range(ThisWorkbook.Sheets("INFORMACIÓN GENERAL").Cells(3, 3), ThisWorkbook.Sheets("INFORMACIÓN GENERAL").Cells(27, 4)).ClearContents
23 Range(ThisWorkbook.Sheets("INFORMACIÓN GENERAL").Cells(3, 8), ThisWorkbook.Sheets("INFORMACIÓN GENERAL").Cells(13, 9)).ClearContents
24 ThisWorkbook.Sheets("INFORMACIÓN GENERAL").Cells(8, 8).ClearContents

```

Después se corre un ciclo para recorrer las cuentas bancarias e ir sumando las comisiones y saldos:

```

24
25 For i = 10 To ThisWorkbook.Sheets.Count
26     For j = 1 To 16
27         hoja = ThisWorkbook.Sheets("INFORMACIÓN GENERAL").Cells(2 + j, 2)
28         If (hoja = ThisWorkbook.Sheets(i).Name) Then
29             ThisWorkbook.Sheets("INFORMACIÓN GENERAL").Cells(2 + j, 3) = CDb1(ThisWorkbook.Sheets(i).Cells(15, 21))
30             ThisWorkbook.Sheets("INFORMACIÓN GENERAL").Cells(2 + j, 4) = CDb1(ThisWorkbook.Sheets(i).Cells(15, 10))
31             vendedor1 = vendedor1 + CDb1(ThisWorkbook.Sheets(i).Cells(2, 10))
32             vendedor2 = vendedor2 + CDb1(ThisWorkbook.Sheets(i).Cells(3, 10))
33             vendedor3 = vendedor3 + CDb1(ThisWorkbook.Sheets(i).Cells(4, 10))
34             vendedor4 = vendedor4 + CDb1(ThisWorkbook.Sheets(i).Cells(5, 10))
35             vendedor5 = vendedor5 + CDb1(ThisWorkbook.Sheets(i).Cells(6, 10))
36             vendedor6 = vendedor6 + CDb1(ThisWorkbook.Sheets(i).Cells(7, 10))
37             vendedor7 = vendedor7 + CDb1(ThisWorkbook.Sheets(i).Cells(8, 10))
38             vendedor8 = vendedor8 + CDb1(ThisWorkbook.Sheets(i).Cells(9, 10))
39             vendedor9 = vendedor9 + CDb1(ThisWorkbook.Sheets(i).Cells(10, 10))
40             vendedor10 = vendedor10 + CDb1(ThisWorkbook.Sheets(i).Cells(11, 10))
41             vendedor11 = vendedor11 + CDb1(ThisWorkbook.Sheets(i).Cells(12, 10))
42             vendedor12 = vendedor12 + CDb1(ThisWorkbook.Sheets(i).Cells(13, 10))
43
44             despacho = despacho + CDb1(ThisWorkbook.Sheets(i).Cells(15, 10))
45             comisiones = 0
46             For k = 2 To 13
47                 comisiones = Round(CDb1(comisiones + ThisWorkbook.Sheets(i).Cells(k, 10).Value), 2)
48             Next k
49             ThisWorkbook.Sheets("INFORMACIÓN GENERAL").Cells(2 + j, 5) = comisiones
50             Exit For
51         End If
52     Next j
53
54 Next i

```

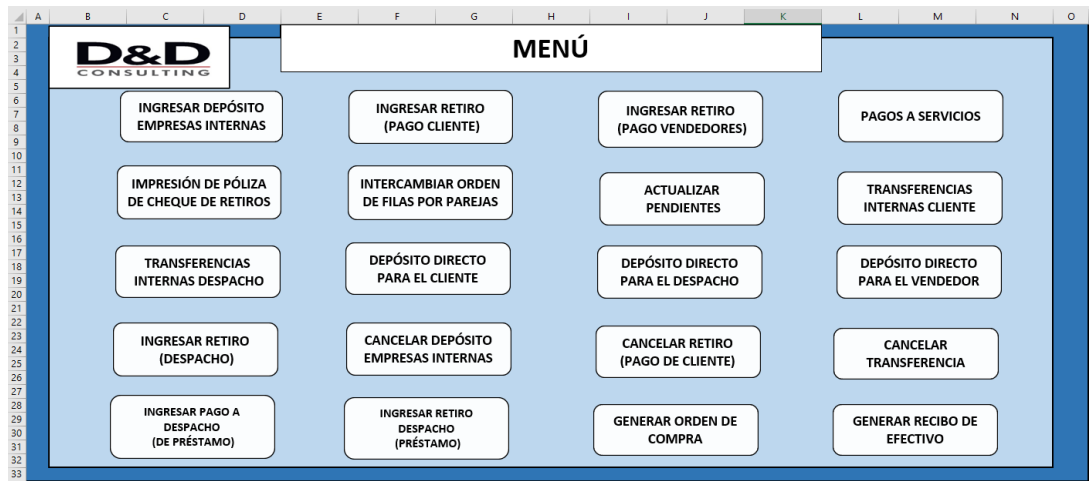
Por último, se escribe la información de las variables en la hoja de información general y se bloquea.

```

56 ThisWorkbook.Sheets("INFORMACIÓN GENERAL").Cells(3, 8) = vendedor1
57 ThisWorkbook.Sheets("INFORMACIÓN GENERAL").Cells(4, 8) = vendedor2
58 ThisWorkbook.Sheets("INFORMACIÓN GENERAL").Cells(5, 8) = vendedor3
59 ThisWorkbook.Sheets("INFORMACIÓN GENERAL").Cells(6, 8) = vendedor4
60 ThisWorkbook.Sheets("INFORMACIÓN GENERAL").Cells(7, 8) = vendedor5
61 ThisWorkbook.Sheets("INFORMACIÓN GENERAL").Cells(8, 8) = vendedor6
62 ThisWorkbook.Sheets("INFORMACIÓN GENERAL").Cells(9, 8) = vendedor7
63 ThisWorkbook.Sheets("INFORMACIÓN GENERAL").Cells(10, 8) = vendedor8
64 ThisWorkbook.Sheets("INFORMACIÓN GENERAL").Cells(11, 8) = vendedor9
65 ThisWorkbook.Sheets("INFORMACIÓN GENERAL").Cells(12, 8) = vendedor10
66 ThisWorkbook.Sheets("INFORMACIÓN GENERAL").Cells(13, 8) = vendedor11
67 ThisWorkbook.Sheets("INFORMACIÓN GENERAL").Cells(14, 8) = vendedor12
68
69 ThisWorkbook.Sheets("INFORMACIÓN GENERAL").Cells(8, 10) = despacho
70
71 ThisWorkbook.Sheets("INFORMACIÓN GENERAL").Protect ("BRIANN_SISTEMA_2")
72
73 End Sub
74

```

Ahora definiremos la hoja del menú de funciones



Se diseñó la hoja principal del libro llamada Menú la cual aloja 20 botones en donde cada uno activa una función específica por medio de macros.

Descripción de funciones:

a) **INGRESAR DEPÓSITO A EMPRESAS INTERNAS:**

Esta función activa el siguiente formulario en donde se solicita la información del depósito, como lo son el banco, apodo de la cuenta bancaria, la cuenta, leyendo del libro creado anteriormente BASE DE DATOS el cliente y obteniendo el vendedor y su porcentaje asociado al mismo, la cantidad del depósito y si es del tipo depósito extraordinario, una vez teniendo toda la información completa al dar clic en el botón guardar se crea un registro por un ingreso en donde se distribuye la comisión del vendedor y se crea un historial de movimientos bancarios muy parecido a un estado de cuenta. Nota un depósito extraordinario es aquel que no genera comisión alguna.

The image shows a web form titled "INGRESAR DEPÓSITO" with the D&D CONSULTING logo on the left. The form contains the following fields and elements:

- NOTE: PARA GUARDAR UN DEPÓSITO ES NECESARIO INTRODUCIR LOS CAMPOS MARCADOS CON (*).
- *APODO: [dropdown]
- *BANCO: [dropdown]
- CUENTA: [dropdown]
- CHEQUE: [text input]
- REFERENCIA: [text input]
- *DÍA: [dropdown] *MES: [dropdown] *AÑO: [dropdown]
- *CLIENTE: [text input]
- *VENDEDOR: [dropdown]
- *PORCENTAJE: [text input]
- * DEPÓSITO: [text input]
- NOTE: SI QUIERES MODIFICAR O ACTUALIZAR EL VENDEDOR O PORCENTAJE PUEDES INGRESARLO. NOTE: EL PORCENTAJE VA DE 2.5 A 100.
- NOTE: SI ES DEPÓSITO EXTRAORDINARIO
- TIPO DE DEPÓSITO: [dropdown]
- *MES: [dropdown] *AÑO: [dropdown]
- GUARDAR [button]
- FINALIZAR [button]

De igual manera el formulario se inicializa como los códigos anteriores llenando los campos de los cuadros de texto del tipo lista y posiciona el cursor en apodo. En el siguiente código se muestra el paso en el cual se introduce el registro a la hoja de la cuenta bancaria especificada.

```

1  Id = "" & CStr(0) & CStr(ComboBox2.ListIndex + 1) & CStr(empresal) & CStr(cuental) & CStr(numero_cliente) & CStr(usuario1) & CStr(Left(Date, 2)) & CStr(Right(Left(Date, 5), 2)) & CStr(Right(Date, 2)) & CStr(operacion) & ""
2  ThisWorkbook.Sheets(1).Cells(indice + 1, 1) = 1 'saber que fue un deposito
3  ThisWorkbook.Sheets(1).Cells(indice + 1, 3) = id
4  ThisWorkbook.Sheets(1).Cells(indice + 1, 4) = operacion 'numero de operacion
5  ThisWorkbook.Sheets(1).Cells(indice + 1, 5).Interior.Color = xwbolor 'estado del cliente
6  ThisWorkbook.Sheets(1).Cells(indice + 1, 5) = letraon 'estado del cliente
7  ThisWorkbook.Sheets(1).Cells(indice + 1, 6) = ComboBox5.Text 'vendedor
8  ThisWorkbook.Sheets(1).Cells(indice + 1, 7) = TextBox3 'porcentaje
9  ThisWorkbook.Sheets(1).Cells(indice + 1, 8).Interior.Color = xwbolor 'estado del vendedor
10 ThisWorkbook.Sheets(1).Cells(indice + 1, 9) = letraores
11 ThisWorkbook.Sheets(1).Cells(indice + 1, 9) = Round(CDbl(((TextBox3 - 2.5) / (100)) * (TextBox1)), 2) 'porcentaje del vendedor
12 ThisWorkbook.Sheets(1).Cells(indice + 1, 10) = Round(CDbl(((TextBox3 - 2.5) / (100)) * (TextBox1)), 2) 'porcentaje del vendedor restante
13 ThisWorkbook.Sheets(1).Cells(indice + 1, 11).Interior.ColorIndex = 4 'estado del despacho
14 ThisWorkbook.Sheets(1).Cells(indice + 1, 11) = "DESCUENTO"
15 ThisWorkbook.Sheets(1).Cells(indice + 1, 11).Font.ColorIndex = 1 'estado del cliente
16 ThisWorkbook.Sheets(1).Cells(indice + 1, 12) = Round(CDbl(((2.5) / (100)) * (TextBox1)), 2) 'porcentaje del despacho
17 ThisWorkbook.Sheets(1).Cells(indice + 1, 13) = Round(CDbl(((2.5) / (100)) * (TextBox1)), 2) 'porcentaje del despacho restante
18 ThisWorkbook.Sheets(1).Cells(15, 10) = Round(CDbl(((2.5) / (100)) * (TextBox1) + Round(ThisWorkbook.Sheets(1).Cells(15, 10).Value, 2)), 2) 'total del despacho
19 For i = 2 To 13
20     vendedor1 = ThisWorkbook.Sheets(1).Cells(1, 9).Text
21     If (vendedor1 = ComboBox5.Text) Then
22         vendedor_efectivo = Round(ThisWorkbook.Sheets(1).Cells(1, 10).Value, 2)
23         ThisWorkbook.Sheets(1).Cells(1, 10) = Round(CDbl(((TextBox3 - 2.5) / (100)) * (TextBox1) + vendedor_efectivo), 2)
24     Exit For
25     End If
26 Next i
27 ThisWorkbook.Sheets(1).Cells(indice + 1, 14) = dia & "/" & mes & "/" & ComboBox6 'Fecha
28 ThisWorkbook.Sheets(1).Cells(indice + 1, 15) = numero_cliente 'numero_cliente
29 ThisWorkbook.Sheets(1).Cells(indice + 1, 16) = ComboBox4.Text 'cliente
30 ThisWorkbook.Sheets(1).Cells(indice + 1, 17) = UCase(TextBox2.Text) 'cheque
31 ThisWorkbook.Sheets(1).Cells(indice + 1, 18) = CDbl(TextBox1) 'deposito
32 ThisWorkbook.Sheets(1).Cells(indice + 1, 20) = Round(CDbl(TextBox1 - ((2.5) / (100)) * (TextBox1) - ((TextBox3 - 2.5) / (100)) * (TextBox1)), 2) 'restante
33 *****
34 saldo_total = Round(ThisWorkbook.Sheets(1).Cells(15, 21).Value, 2)
35 ThisWorkbook.Sheets(1).Cells(15, 21) = Round(CDbl(saldo_total + (TextBox1)), 2) 'saldo total
36 *****
37 ThisWorkbook.Sheets(1).Cells(indice + 1, 21) = Round(ThisWorkbook.Sheets(1).Cells(15, 21).Value, 2) 'saldo instantaneo
38 ThisWorkbook.Sheets(1).Cells(indice + 1, 22) = UCase(TextBox5) 'referencia
39 ThisWorkbook.Sheets(1).Cells(indice + 1, 22).EntireRow.AutoFit
40 ThisWorkbook.Sheets(1).Cells(indice + 1, 23) = Date 'fecha captura
41 ThisWorkbook.Sheets(1).Cells(indice + 1, 24) = rfc1 'rfc

```

Se muestra el id, que es una variable del tipo texto que almacena la información del apodo, banco, cuenta, cliente, vendedor, fecha, número de operación de forma que cada concepto es una cadena de números con el cual podemos definir una llave para el rastreo de cada movimiento, se calcula el número de operación dependiendo de los movimientos bancarios y el mes en el cual se hizo el registro, se asigna la comisión del vendedor y la comisión del despacho como método de fondo de ahorro del 2.5% fijo sobre el ingreso.

b) INGRESAR RETIRO (PAGO CLIENTE):

En esta función se ingresa un registro de egreso en la cuenta bancaria con la finalidad de documentar los gastos asociados a cada venta de un cliente, pueden ser honorarios, insumos, costos, etc.

The screenshot shows a web form for 'INGRESAR RETIRO PAGO CLIENTE' with the D&D CONSULTING logo. The form contains several input fields and dropdown menus:

- *EMPRESA: [dropdown]
- *BANCO: [dropdown]
- CUENTA: [dropdown]
- BUSCAR IDENTIFICADOR: [button]
- REFERENCIA: [text input]
- *DÍA: [dropdown]
- *MES: [dropdown]
- *AÑO: [dropdown]
- *IDENTIFICADOR: [dropdown]
- MONTO MÁXIMO NO INCLUYE COMISIONES DEL VENDEADOR NI DEL DESPACHO, GRACIAS. [text]
- # CLIENTE: [text input]
- #VENDEADOR: [text input]
- #MONTO MÁXIMO A RETIRAR DEL CLIENTE: [text input]
- CLIENTE Y VENDEADOR MUESTRAN LA INFORMACIÓN ASOCIADA AL IDENTIFICADOR. [text]
- *RETIRO PARA EL CLIENTE: [text input]
- *TIPO DE RETIRO: [dropdown]
- CHEQUE: [text input]
- *PERSONA QUE COBRARÁ EL CHEQUE: [dropdown]
- GUARDAR RETIRO: [button]
- FINALIZAR: [button]

En esta función se introduce la información de la cuenta bancaria, un texto de referencia del egreso, la fecha del mismo, un identificador, este identificador es una llave para encontrar el ingreso del cual se quiere extraer algún importe y así poder verificar si tiene monto disponible a retirar, se cuenta con el botón de buscar identificador, más abajo desglosaremos su código y su función, una vez seleccionado el identificador se llenan automáticamente la información marcada con #, que es cliente, el vendedor, el monto máximo a retirar, después se captura el importe a retirar, el tipo de retiro y si es cheque el número del mismo y la persona que lo cobrará.

La función del botón buscar identificador ejecuta una macro en la cuál va y busca en la cuenta bancaria especificada todos aquellos ingresos que tengan un estado disponible o parcialmente cobrado, es decir, que tienen saldo para retirar, posteriormente se llena la lista de identificador, a continuación, se muestra el código.

Primeramente, si la información de la cuenta bancaria es vacía manda un mensaje de error solicitando ingresarla, de lo contrario de declaran e inicializan las variables y se hace una búsqueda por nombre de la hoja hasta que coincida con la cuenta bancaria, una vez que identifica la cuenta bancaria hace un barrido desde el último registro al primero y agregando a la lista los identificadores que cumplan con el estado adecuado

```

1 Private Sub BUSCAR_IDENTIFICADOR_Click()
2   ComboBox4 = Empty
3   ComboBox4.Clear
4   If (ComboBox1 = Empty) Or (ComboBox2 = Empty) Then 'Si la empresa o el banco son vacios no podras buscar el identificador
5     MsgBox "PARA BUSCAR EL IDENTIFICADOR ES NECESARIO PRIMERO INTRODUCIR AL MENOS BANCO Y EMPRESA.", , "ERROR"
6   Else
7     Dim hoja_useform, hoja_libro As String
8     If (ComboBox3 <> Empty) Then
9       hoja_useform = ComboBox2.Text + " " + ComboBox1.Text + " " + ComboBox3.Text
10    Else
11      hoja_useform = ComboBox2.Text + " " + ComboBox1.Text
12    End If
13    'verificar que exista una hoja del libro con ese nombre
14    Dim id As String
15    Dim ultimo_renglon, ultima_filas As Long
16    Dim i, exito, j As Integer
17    Dim indice, k, xcolor As Long
18    Dim bandera As Integer
19    Dim bandera2, etiqueta As String
20    Dim depositos As Double
21    exito = 0
22    bandera = 0
23    For i = 1 To ThisWorkbook.Sheets.Count
24      hoja_libro = ThisWorkbook.Sheets(i).Name
25      If (hoja_useform = hoja_libro) Then
26        bandera = 1 'si existe hoja_useform en el libro
27        'Vamos a cargar los identificadores en el combobox4
28        indice = ThisWorkbook.Sheets(i).Cells(Rows.Count, "A").End(xlUp).Row 'encontramos el indice del número de operación
29        For k = indice To 18 Step -1
30          etiqueta = ThisWorkbook.Sheets(i).Cells(k, 1).Value
31          If (etiqueta = "1") Then
32            xcolor = ThisWorkbook.Sheets(i).Cells(k, 5).Interior.Color 'color del estado del cliente
33            If (xcolor = 65280) Or (xcolor = 14478595) Then 'solo muestra los identificadores que aún les puedes retirar
34              ComboBox4.AddItem ThisWorkbook.Sheets(i).Cells(k, 3).Value
35              exito = 1
36            End If
37          End If
38        Next k
39      Exit For
40    End If 'cierra el if donde encuentra la hoja de la empresa en el libro
41  Next i
42

```

Después se define un mensaje de error en caso de no encontrar la cuenta bancaria con el nombre ingresado o en caso de éxito manda un mensaje de que lo identificadores se cargaron correctamente.

```

42
43
44     If (bandera <> 1) Then
45         MsgBox "ERROR EN LOS DATOS DE EMPRESA, BANCO Y CUENTA." & vbCrLf & "ESA CUENTA NO EXISTE." & vbCrLf & "REVISALOS, GRACIAS.", , "ERROR"
46     Else
47         If (exito = 1) Then
48             MsgBox "IDENTIFICADORES CARGADOS CORRECTAMENTE", , "INFORMACIÓN"
49         Else
50             MsgBox "NO HAY DEPOSITOS DE LOS CUALES SE PUEBAN HACER RETIROS.", , "INFORMACIÓN"
51         End If
52     End If
53
54 End If
55 End Sub

```

Una vez cargada toda la información en el formulario se ejecuta el proceso del botón guardar retiro en el cual de igual forma se busca la cuenta bancaria, se busca el identificador y se genera el registro con su nuevo identificador asociado a la salida de efectivo.

A continuación, se muestra el código referente a la parte de la creación del número de operación y los casos para el retiro.

```

1 | indice = CSng(ThisWorkbook.Sheets(i).Cells(Rows.Count, "A").End(xlUp).Row) 'encontramos el indice del último renglón en esa hoja
2 | For j = 18 To indice
3 |     id_hoja = CStr(ThisWorkbook.Sheets(i).Cells(j, 3).Text)
4 |     If (id_hoja = ComboBox4.Text) And (CStr(ThisWorkbook.Sheets(i).Cells(j, 1)) = "1") Then
5 |         banderal = 1
6 |         restantel = Round(CDbl(ThisWorkbook.Sheets(i).Cells(j, 20).Value), 2) 'restante del cliente
7 |         If (retirul <= restantel) Then 'se puede realizar el retiro
8 |             If (CInt(Month(ThisWorkbook.Sheets(i).Cells(indice, 23).Value)) <> CInt(Month(Date))) Then
9 |                 MsgBox ";EL NÚMERO DE OPERACIÓN CAMBIÓ POR SER UN MES DIFERENTE!" & vbCrLf & "EL SIGUIENTE ES SER: 0001",
10 |                    & vbCrLf & "SE GUARDARA CON ESE NÚMERO DE OPERACIÓN, GRACIAS.", , "INFORMACIÓN"
11 |                 max_numero = 0
12 |             Else
13 |                 Dim operaciones() As Integer
14 |                 ReDim operaciones(0 To (indice - 17))
15 |                 For p = 18 To indice
16 |                     If (Month(ThisWorkbook.Sheets(i).Cells(p, 23).Text) = Month(Date)) Then
17 |                         operaciones(p - 18) = CInt(ThisWorkbook.Sheets(i).Cells(p, 4).Value)
18 |                     Else
19 |                         operaciones(p - 18) = 0
20 |                     End If
21 |                 Next p
22 |                 If (indice > 18) Then
23 |                     For p = 0 To (indice - 19)
24 |                         If (operaciones(p) > operaciones(p + 1)) Then
25 |                             c = operaciones(p + 1)
26 |                             operaciones(p + 1) = operaciones(p)
27 |                             operaciones(p) = c
28 |                         End If
29 |                     Next p
30 |                     max_numero = operaciones(indice - 18)
31 |                 Else
32 |                     max_numero = 1
33 |                 End If
34 |             End If
35 |             numero_operacion = CInt(max_numero) + 1
36 |             If Len(CStr(numero_operacion)) = 1 Then
37 |                 operacion = "000" + CStr(numero_operacion)
38 |             End If
39 |             If Len(CStr(numero_operacion)) = 2 Then
40 |                 operacion = "00" + CStr(numero_operacion)
41 |             End If

```


Después se ingresa la información del registro en el caso en el cual el retiro es estrictamente menor a lo disponible.

```

1 If Len(CStr(numero_operacion)) = 3 Then
2     operacion = "0" + CStr(numero_operacion)
3 End If
4 If Len(CStr(numero_operacion)) = 4 Then
5     operacion = CStr(numero_operacion)
6 End If
7 If (retiro1 < restantel) Then 'se va a retirar pero aún va a quedar saldo y cambia a color 14478595 (cobrado parcialmente)
8     'se modifica el deposito original
9     ThisWorkbook.Sheets(i).Cells(j, 20) = Round(restantel - retiro1, 2)
10    ThisWorkbook.Sheets(i).Cells(j, 5).Interior.Color = 14478595 'se queda como cobrado parcialmente
11    ThisWorkbook.Sheets(i).Cells(j, 5) = "C.PARCIAL" 'se queda como cobrado parcialmente
12    ThisWorkbook.Sheets(i).Cells(j, 5).Font.ColorIndex = 1 'estado del cliente
13
14    'para el retiro
15    ThisWorkbook.Sheets(i).Cells(indice + 1, 1) = 0 ' porque es un retiro
16    ThisWorkbook.Sheets(i).Cells(indice + 1, 3) = ThisWorkbook.Sheets(i).Cells(j, 3) 'id mismo que el del deposito
17    ThisWorkbook.Sheets(i).Cells(indice + 1, 4) = operacion ' número de operación
18    ThisWorkbook.Sheets(i).Cells(indice + 1, 5).Interior.Color = 6299648 'se queda como retiro cliente
19    ThisWorkbook.Sheets(i).Cells(indice + 1, 5) = "R.CLIENTE" 'estado del cliente
20    ThisWorkbook.Sheets(i).Cells(indice + 1, 5).Font.ColorIndex = 2 'estado del cliente
21
22
23    ThisWorkbook.Sheets(i).Cells(indice + 1, 6) = ThisWorkbook.Sheets(i).Cells(j, 6) 'usuario mismo que el deposito
24
25    ThisWorkbook.Sheets(i).Cells(indice + 1, 14) = dial & "/" & mes1 & "/" & ComboBox8 'Fecha
26    ThisWorkbook.Sheets(i).Cells(indice + 1, 15) = ThisWorkbook.Sheets(i).Cells(j, 15) 'número cliente mismo que el deposito
27    ThisWorkbook.Sheets(i).Cells(indice + 1, 16) = ThisWorkbook.Sheets(i).Cells(j, 16) 'cliente mismo que el deposito
28    ThisWorkbook.Sheets(i).Cells(indice + 1, 17) = UCase(TextBox3.Text) 'Cheque
29    ThisWorkbook.Sheets(i).Cells(indice + 1, 19) = retiro1 'retiro
30    *****
31    saldo_total = CDBl(ThisWorkbook.Sheets(i).Cells(15, 21).Value)
32    ThisWorkbook.Sheets(i).Cells(15, 21) = Round(saldo_total - retiro1, 2) 'saldo total
33    *****
34    ThisWorkbook.Sheets(i).Cells(indice + 1, 21) = Round(saldo_total - retiro1, 2) 'saldo instantáneo
35    ThisWorkbook.Sheets(i).Cells(indice + 1, 22) = UCase(TextBox4.Text) 'referencia
36    ThisWorkbook.Sheets(i).Cells(indice + 1, 22).EntireRow.AutoFit
37    ThisWorkbook.Sheets(i).Cells(indice + 1, 23) = Date 'fecha captura
38    ThisWorkbook.Sheets(i).Cells(indice + 1, 24) = UCase(ComboBox9.Text) 'tipo de retiro
39    ThisWorkbook.Sheets(i).Cells(indice + 1, 25) = ComboBox10 'persona que cobra el cheque
40    exito = 1
41 Exit For

```

Ahora se incluye el caso en donde se retira todo y una bandera en donde si el retiro es mayor a lo disponible se manda un mensaje informando de nuevo con lo disponible y rompe el ciclo.

```

1 Else 'se está cobrando lo exacto, se vuelve cobrado color 255
2     'se modifica el deposito original
3     ThisWorkbook.Sheets(i).Cells(j, 20) = Round(restantel - retiro1, 2)
4     ThisWorkbook.Sheets(i).Cells(j, 5).Interior.Color = 255 'se queda como cobrado parcialmente
5     ThisWorkbook.Sheets(i).Cells(j, 5) = "COBRADO" 'se queda como cobrado parcialmente
6     ThisWorkbook.Sheets(i).Cells(j, 5).Font.ColorIndex = 1 'estado del cliente
7
8
9
10    'para el retiro
11    ThisWorkbook.Sheets(i).Cells(indice + 1, 1) = 0 ' porque es un retiro
12    ThisWorkbook.Sheets(i).Cells(indice + 1, 3) = ThisWorkbook.Sheets(i).Cells(j, 3) 'id mismo que el del deposito
13    ThisWorkbook.Sheets(i).Cells(indice + 1, 4) = operacion ' número de operación
14    ThisWorkbook.Sheets(i).Cells(indice + 1, 5).Interior.Color = 6299648 'se queda como retiro cliente
15    ThisWorkbook.Sheets(i).Cells(indice + 1, 5) = "R.CLIENTE" 'estado del cliente
16    ThisWorkbook.Sheets(i).Cells(indice + 1, 5).Font.ColorIndex = 2
17
18
19    ThisWorkbook.Sheets(i).Cells(indice + 1, 6) = ThisWorkbook.Sheets(i).Cells(j, 6) 'usuario mismo que el deposito
20    ThisWorkbook.Sheets(i).Cells(indice + 1, 14) = dial & "/" & mes1 & "/" & ComboBox8 'Fecha
21    ThisWorkbook.Sheets(i).Cells(indice + 1, 15) = ThisWorkbook.Sheets(i).Cells(j, 15) 'número cliente mismo que el deposito
22    ThisWorkbook.Sheets(i).Cells(indice + 1, 16) = ThisWorkbook.Sheets(i).Cells(j, 16) 'cliente mismo que el deposito
23    ThisWorkbook.Sheets(i).Cells(indice + 1, 17) = UCase(TextBox3.Text) 'Cheque
24    ThisWorkbook.Sheets(i).Cells(indice + 1, 19) = retiro1 'retiro
25    *****
26    saldo_total = CDBl(ThisWorkbook.Sheets(i).Cells(15, 21).Value)
27    ThisWorkbook.Sheets(i).Cells(15, 21) = Round(saldo_total - retiro1, 2) 'saldo total
28    *****
29    ThisWorkbook.Sheets(i).Cells(indice + 1, 21) = Round(saldo_total - retiro1, 2) 'saldo instantáneo
30    ThisWorkbook.Sheets(i).Cells(indice + 1, 22) = UCase(TextBox4.Text) 'referencia
31    ThisWorkbook.Sheets(i).Cells(indice + 1, 22).EntireRow.AutoFit
32    ThisWorkbook.Sheets(i).Cells(indice + 1, 23) = Date 'fecha captura
33    ThisWorkbook.Sheets(i).Cells(indice + 1, 24) = UCase(ComboBox9.Text) 'tipo de retiro
34    ThisWorkbook.Sheets(i).Cells(indice + 1, 25) = ComboBox10 'persona que cobra el cheque
35
36    exito = 1
37 Exit For
38 End If
39 Else
40     MsgBox "NO SE PUEDE REALIZAR EL RETIRO DE CLIENTE PORQUE ES MAYOR QUE EL MONTO MÁXIMO A RETIRAR." & vbNewLine & "EL MONTO MÁXIMO A RETIRAR ES DE: ",
41     & (restantel), , "ERROR"
42 Exit For
43 End If 'if del retiro de cliente mayor que el máximo a retirar

```

Luego, en caso de cumplir todo para el retiro se manda envía un mensaje de retiro registrado correctamente y se bloquea la hoja de la cuenta bancaria y se finaliza el formulario.

c) INGRESAR RETIRO (PAGO VENDEDORES):

En esta función se introducían los datos de la cuenta bancaria de la cual se quería realizar el retiro, se introducía la fecha del retiro y se daba clic en el botón actualizar comisiones disponibles de vendedores, el cual llena la información respecto a las comisiones disponibles. Posteriormente se ingresa el retiro de cada vendedor, el número de cheque en caso de ser así, la persona que lo cobra y la semana correspondiente.

The screenshot shows a software interface titled "RETIRO DE COMISIONES DE MÚLTIPLES VENDEDORES". On the left is the "D&D CONSULTING" logo. The main area contains several input fields: "*EMPRESA:", "*BANCO:", "CUESTA:", "REFERENCIA:", "*DÍA:", "*MES:", and "*AÑO:". A button labeled "ACTUALIZAR COMISIONES DISPONIBLES DE VENDEDORES" is positioned to the right of the bank and account fields. Below these is a table with three columns: "VENDEDORES:", "COMISIONES DISPONIBLES:", and "RETIRO DE VENEDOR:". The table has 12 rows, each for a vendor from "#VENDEDOR 1:" to "#VENDEDOR 12:". At the bottom of the table are two summary fields: "#SUMA DE COMISIONES DISPONIBLES:" and "#SUMA DE RETIROS:". Below the table are four more fields: "*TIPO DE RETIRO:", "CHEQUE:", "*PERSONA QUE COBRARÁ EL CHEQUE:", and "*SEMANA NÚMERO:". At the very bottom are two buttons: "GUARDAR RETIRO" and "FINALIZAR".

El botón actualizar comisiones disponibles de vendedores realizaba la función de ir a buscar la cuenta bancaria, declarar e inicializar las variables de los montos de las comisiones.

```

1 Private Sub ACTUALIZAR_COMISIONES_Click()
2 ThisWorkbook.Application.ScreenUpdating = False
3 Dim hoja_useform, hoja_libro, vendedor(11), vendedor2, vendedor1 As String
4 Dim i, j, etiqueta, d, bandera_hoja_libro, exito, etiquetal As Integer
5 Dim indice, k, p, s, xcolor, xcolor_comisiones As Long
6 Dim acumulado, acumulado_comisiones As Double
7 bandera_hoja_libro = 0
8
9
10 etiqueta = 0
11 exito = 0
12 'limpiamos los textbox de las comisiones disponibles
13 TextBox5 = Empty
14 TextBox8 = Empty
15 TextBox10 = Empty
16 TextBox12 = Empty
17 TextBox14 = Empty
18 TextBox16 = Empty
19 TextBox19 = Empty
20 TextBox21 = Empty
21 TextBox23 = Empty
22 TextBox24 = Empty
23 TextBox27 = Empty
24 TextBox28 = Empty
25 TextBox2 = Empty
26 vendedor(0) = "VENDEDOR 1"
27 vendedor(1) = "VENDEDOR 2"
28 vendedor(2) = "VENDEDOR 3"
29 vendedor(3) = "VENDEDOR 4"
30 vendedor(4) = "VENDEDOR 5"
31 vendedor(5) = "VENDEDOR 6"
32 vendedor(6) = "VENDEDOR 7"
33 vendedor(7) = "VENDEDOR 8"
34 vendedor(8) = "VENDEDOR 9"
35 vendedor(9) = "VENDEDOR 10"
36 vendedor(10) = "VENDEDOR 11"
37 vendedor(11) = "VENDEDOR 12"

```

Posteriormente de todo ser correcto hace un barrido de las comisiones restantes de cada uno de los vendedores.

```
1 indice = CSng(ThisWorkbook.Sheets(i).Cells(Rows.Count, "A").End(xlUp).Row) 'encontramos el indice del número de operación
2 For j = 2 To 13 'for que busca en los vendedores
3   vendedor1 = CStr(ThisWorkbook.Sheets(i).Cells(j, 9).Text)
4   For k = 0 To 11
5     If (vendedor1 = vendedor(k)) Then 'encontró el vendedor en el vector de vendedores
6       acumulado = 0
7       acumulado_comisiones = 0
8       'primero vamos a ver si si esta bien los números de la suma de las comisiones con el acumulado
9       acumulado_comisiones = Round(CDbl(ThisWorkbook.Sheets(i).Cells(j, 10).Value), 2)
10      For s = 18 To indice
11        etiquetal = CInt(ThisWorkbook.Sheets(i).Cells(s, 1).Value)
12        vendedor2 = CStr(ThisWorkbook.Sheets(i).Cells(s, 6).Text)
13        If (etiquetal = 1) Then
14          If (vendedor2 = vendedor(k)) Then
15            xcolor_comisiones = ThisWorkbook.Sheets(i).Cells(s, 8).Interior.Color 'color del estado del vendedor
16            If (xcolor_comisiones = 65280) Or (xcolor_comisiones = 14478595) Then ' si el color es verde o azul claro
17              acumulado = acumulado + Round(CDbl(ThisWorkbook.Sheets(i).Cells(s, 10).Value), 2)
18            End If
19          End If
20        End If
21      Next s
22      If (Round((acumulado - acumulado_comisiones), 2) <> 0) Then
23        ThisWorkbook.Sheets(i).Cells(j, 10) = acumulado
24        MsgBox "HABÍA UNA DIFERENCIA CON EL ACUMULADO DE LAS COMISIONES Y LA SUMA DE SUS COMISIONES DEL VENDEDOR.",
25          & vbNewLine & "SE ACTUALIZARÓN LOS DATOS, INTÉNTALO DE NUEVO, GRACIAS.", , "ERROR"
26      End If
27      Exit For
28    End If
29  Next k
30 Next j
```

Y al final se escriben en los cuadros del formulario.

Una vez llenada la información del formulario al dar clic en el botón guardar retiro se hace un registro de egreso con todas las reglas respecto a los egresos, se llena la hoja de retiro de vendedores en donde se especifica los montos de las comisiones de cada uno y se manda imprimir.

d) PAGOS A SERVICIOS:

En esta función despliega el siguiente formulario:

RETIRO PARA PAGAR SERVICIOS

NOTA: LOS CAMPOS MARCADOS CON (*) SON OBLIGATORIOS.

*EMPRESA:

*BANCO:

CUENTA:

REFERENCIA:

*DÍA: *MES: *AÑO:

*PROVEEDOR:

MONTO MÁXIMO A RETIRAR: *CATEGORÍA DEL RETIRO:

*RETIRO:

*TIPO DE RETIRO:

CHEQUE:

*PERSONA QUE COBRARÁ EL CHEQUE:

GUARDAR RETIRO FINALIZAR

En donde se tiene que llenar la información de la cuenta bancaria de la cual se realizara el retiro, el proveedor que proviene desde la base de datos de clientes y proveedores creada en el primer capítulo, el monto máximo a retirar que es obtenido de las comisiones del 2.5% que genera el despacho que es llenado una vez que se selecciona el proveedor, el monto del retiro, la categoría del retiro para hacer la clasificación de gastos y la forma en la que será el retiro, regularmente en cheque para obtener efectivo. Al dar clic en el botón guardar retiro se ejecuta una macro que crea un registro de egreso en la cuenta bancaria con todos sus elementos.

e) IMPRESIÓN DE PÓLIZA DE CHEQUE DE RETIROS:

IMPRESIÓN DE PÓLIZA DE CHEQUE PARA RETIROS

CAMPOS MARCADOS CON (*) SON OBLIGATORIOS
CAMPOS MARCADOS CON (#) SON LLENADOS AUTOMÁTICAMENTE

*EMPRESA: BUSCAR IDENTIFICADOR

*BANCO:

CUENTA:

*IDENTIFICADOR:

*NO. OPERACIÓN: #IDENTIFICADOR DE PROCEDENCIA:

#CLIENTE:

#RETIRO: #VENDEDOR:

#CHEQUE: #PORCENTAJE:

#DEPÓSITO: #PERSONA QUE COBRARÁ EL CHEQUE:

*NOMBRE DEL PORTADOR DEL CHEQUE:

*PERSONA QUE REALIZA LA PÓLIZA: #TIPO:

REFERENCIA:

#REFERENCIA DEL RETIRO:

IMPRIMIR PÓLIZA DE RETIRO FINALIZAR

NOTA: CLIENTE, RETIRO, CHEQUE, DEPOSITO VENDEDOR, PORCENTAJE MUESTRAN LA INFORMACIÓN ASOCIADA AL IDENTIFICADOR RESPECTIVAMENTE CON ESE NÚMERO DE OPERACIÓN.

En esta función se desglosa el formulario anterior, la función nos ayuda a llenar los datos de la hoja de póliza de cheque y hace la impresión de la hoja, este cheque corresponde a un retiro generado anteriormente, éste es ubicado por su identificador y el número de operación, para poder localizarlo se llenan los datos de la cuenta bancaria, y al ingresar el identificador y el número de operación se llenan los cuadros marcados con #, después solo se llena la persona que realiza la póliza.

f) INTERCAMBIAR ORDEN DE FILAS POR PAREJAS:

INTERCAMBIO DE FILAS POR PAREJAS

D&D CONSULTING

NOTA: PARA INTERCAMBIAR FILAS ES NECESARIO INTRODUCIR LOS CAMPOS MARCADOS CON (*), LOS CAMPOS MARCADOS CON (#) SE LLENAN AUTOMATICAMENTE.
1.- INTRODUCE LOS DATOS DE EMPRESA, BANCO Y CUENTA.
2.- BUSCA LOS IDENTIFICADORES.
3.- OBSERVA QUE SEA LA INFORMACIÓN CORRECTA DE CADA IDENTIFICADOR.
4.- OPRIME EL BOTÓN INTERCAMBIAR.

*EMPRESA: [dropdown]
*BANCO: [dropdown] BUSCAR IDENTIFICADORES
CUENTA: [dropdown]
*IDENTIFICADOR 1: [dropdown]
*NO. OPERACIÓN 1: [dropdown]
#CLIENTE 1: [text]
#RETIRO 1: [text]
#DEPOSITO 1: [text]
*IDENTIFICADOR 2: [dropdown]
*NO. OPERACIÓN 2: [dropdown]
#CLIENTE 2: [text]
#RETIRO 2: [text]
#DEPOSITO 2: [text]

INTERCAMBIAR FINALIZAR

Esta función desglosa el formulario anterior que nos solicita indicar la cuenta bancaria, los identificadores y los números de operación que se van a intercambiar, posteriormente al dar clic en el botón buscar identificadores se cargan todos aquellos que existan en la cuenta bancaria junto con sus números de operación, al dar clic en intercambiar se activa la macro que busca las filas en donde se ubican y las intercambia, esto con fin de tener los registros igual que el estado de cuentas del banco.

g) ACTUALIZAR PENDIENTES:

The screenshot shows a web application window titled 'ACTUALIZAR PENDIENTES' with a close button (X) in the top right corner. On the left side, there is a logo for 'D&D CONSULTING'. The main area contains a form with the following fields and controls:

- * EMPRESA: [dropdown menu]
- * BANCO: [dropdown menu]
- CUENTA: [dropdown menu]
- * IDENTIFICADOR: [dropdown menu]
- * CLIENTE: [dropdown menu]
- * VENDEDOR: [dropdown menu]
- * PORCENTAJE: [text input]
- DEPOSITO: [text input]
- REFERENCIA: [text input]
- BUSCAR IDENTIFICADOR: [button]
- NOTA: DEPOSITO MUESTRA LA CANTIDAD ASOCIADA AL IDENTIFICADOR. [text box]
- GUARDAR: [button]
- FINALIZAR: [button]

An 'INSTRUCCIONES' button is located in the top right corner of the form area.

Esta función despliega el formulario anterior. Este nos ayuda a actualizar la información de los depósitos en las cuentas bancarias que en el momento de su registro no se tenía conocimiento completo de la procedencia y se ingresaron como deposito con cliente pendiente, una vez ingresada la información de la cuenta bancaria al dar clic en el botón buscar identificador se nos llenara la lista con todos aquellos ingresos que cumplen con la característica de ser pendientes. Al seleccionar algún identificador nos aparecerá el monto del depósito y se seleccionará el cliente al que pertenece y su porcentaje asignado. Al dar clic en guardar se actualiza la información de ese registro y se asignan las comisiones al vendedor y al despacho procediendo a actualizar el estado del ingreso como disponible. Nota: En los retiros de cliente no se puede retirar alguno que proceda de un identificador pendiente, de hecho, no se cargan en la lista de identificadores.

Las siguientes funciones son combinaciones de las anteriores. Se sigue el mismo razonamiento en cada una de estas.

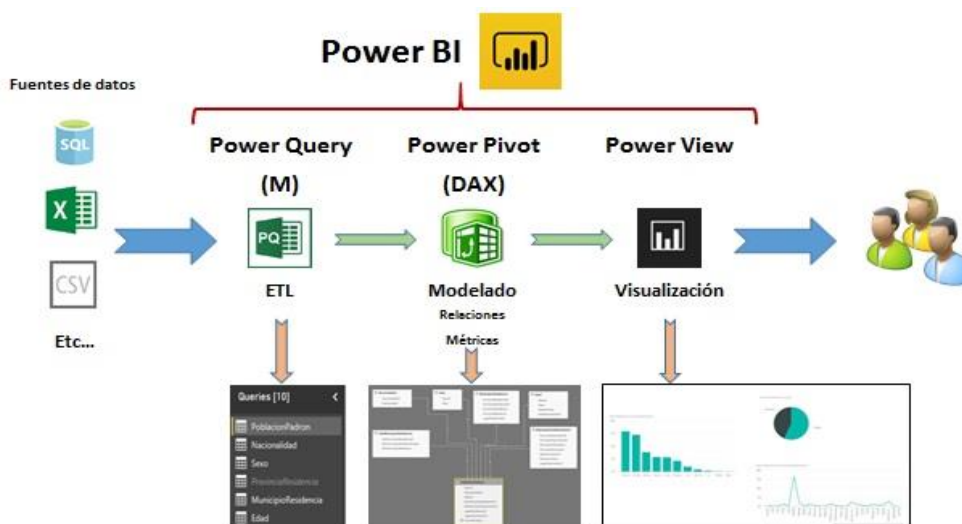
Capítulo 3

Generación de algoritmos para el análisis de la información obtenida y creación reportes automatizados que ayuden en la toma de decisiones

En esta parte del proyecto se desarrollo un libro de Microsoft Excel llamado REPORTEES conectado al libro de VENTAS en el cual se tenía acceso a toda la información de las cuentas bancarias, posteriormente se crearon gráficas e información resumida de todas las cuentas en el libro de ventas que servían para tomar decisiones y éste era presentado a los directores de la empresa. Se tuvo la oportunidad de actualizar ese libro de REPORTEES usando una herramienta especializada para la visualización de información que es Microsoft Power BI.

A continuación, veremos la aplicación de Microsoft Power BI en el proyecto.

En el siguiente diagrama veremos la funcionalidad de Power BI:



La manera de trabajar en Power BI es primero tener una fuente de datos, que esta ya se había creado en el capítulo 3 que es el libro de VENTAS, una vez teniendo los datos se hace la conexión a Power BI mediante Power Query, posteriormente se hace el modelado de datos en Power BI Desktop y se desarrolla también la reportería junto con los roles de usuario del modelo. Una vez

creado el modelo y los reportes se publican en la nube de Power BI para que los usuarios tengan acceso a los reportes desde cualquier dispositivo conectado a internet. Para poder tener los reportes actualizados en la nube se debe tener una puerta de enlace en la máquina con conexión a la red de la empresa para que funcione como servidor y la puerta de enlace ejecute las consultas hacia el libro de VENTAS y actualice el conjunto de datos del modelo de la nube.

Se desarrollo un modelo de datos en Power BI con el uso de pronósticos e indicadores estadísticos, métricas, e informes especializados en ventas, por la privacidad no se puede mostrar el reporte, pero a continuación mencionaremos algunas gráficas e indicadores utilizados.

- a) Gráfica Z
- b) Pipeline
- c) Pronósticos a 1 año
- d) Sección de preguntas y respuestas de los datos mediante lenguaje natural
- e) Distribución de las ventas por cliente y la regla de Pareto

Capítulo 4

Conclusiones

La investigación se llevó a cabo de manera exitosa, se tuvieron juntas mensuales con los directores de la empresa posteriores a la entrega del proyecto con el fin de tener retroalimentación de éste, se lograron solucionar los problemas planteados de manera concreta y completa usando habilidades de programación, estadística y razonamiento matemático. En los reportes entregados a los directores se pudo apreciar los rubros que constituían los mayores gastos de la compañía, así como los principales vendedores, clientes y la distribución del flujo de efectivo en bancos. Se tuvo la posibilidad de actualizar el desarrollo implementando información de pronósticos, desarrollo de una nueva interfaz web y un análisis de inteligencia de negocios en la herramienta de Microsoft Power BI.

Bibliografía

[1] (Alegsa, 2018)

[http://www.alegsa.com.ar/Dic/visual basic.php](http://www.alegsa.com.ar/Dic/visual%20basic.php)

[2] (PICKELNY & TARRACHANO, 2018)

<http://www.mxmedia.com.ar/capacitacion/teoria/Excel%202016%20-%20Completo.pdf>

[3] (Raffino, 2018)

<https://concepto.de/excel>

