

SQL	4
VISUAL STUDIO	6
Aplicación WPF	6
Clase Conexión	6
Llenar combo box	6
En la clase que se necesite:	7
Alta	7
Baja	7
Modificación	7
Búsqueda	7
Llenar dataGrid	8
Sitio Web	8
Clase Conexión	8
En la página que se necesite:	8
Iniciar sesión	8
Alta	9
Baja	9
Modificación	9
Llenar gridView-Búsqueda	10
Llenar dropDrownList	10
Llenar radioButton	11
Llenar checkBox	11
Session	11
Cookies	11
Checar que no se dupliquen datos al llenar la página web	12
EXCEL	12
Referencias	12
Relativa	12
Absoluta	12
Funciones	12
Reemplazar	12
Seleccionar toda la fila o columna	13
Operaciones con fechas	13
Buscar objetivo	13
Selección simple	13
Operadores lógicos	13
Búsqueda en tabla	13
Filtros	13
Filtro simple	13
Filtro avanzado	13
Tablas dinámicas	14
Hoja de informe	14
Excel con Visual Basic	14

Function()	15
Sub()	16
Ejemplo de macro 1	16
Interfaces (UserForm)	16
Método Cerrar UserForm	16
Método Limpiar UserForm	17
Método Resultado UserForm	17
Llenar ComboBox	17
Llenar ComboBox a partir de otro ComboBox	17
Utilizar funciones en los botones	18
Instrucciones de selección	18
Selección simple	18
Selección múltiple	18
Selección por casos	18
Instrucciones de repetición	18
For	18
For each	18
Do (while)	18
InputBox y MsgBox	19
InputBox()	19
MsgBox()	19
Escritura en las celdas	19
DLL	19
Android	21
Clase DatabaseHelper o AdminSQLiteHelper	21
Otras clases	21
Enlazar botones con métodos	22
En la clase:	22
Iniciar sesión	23
Alta	23
Baja	24
Modificación	24
Llenar spinner con base de datos	25
Si es una lista	25
Página Web	25
Realizar una acción cuando se selecciona un elemento	26
Enlazar ventanas	26
Ejemplo	26
public class SQLiteOpenHelper extends android.database.sqlite.SQLiteOpenHelper{	26
public class MainActivity extends AppCompatActivity {	27
public class Info extends AppCompatActivity {	28

SQL

SQL Statement	Syntax
AND / OR	SELECT column_name(s) FROM table_name WHERE condition AND/OR condition
ALTER TABLE	ALTER TABLE table_name ADD column_name datatype or ALTER TABLE table_name DROP COLUMN column_name
AS (alias)	SELECT column_name AS column_alias FROM table_name or SELECT column_name FROM table_name AS table_alias
BETWEEN	SELECT column_name(s) FROM table_name WHERE column_name BETWEEN value1 AND value2
CREATE DATABASE CREATE TABLE	CREATE DATABASE database_name CREATE TABLE table_name (column_name1 data_type, column_name2 data_type, column_name3 data_type...)
DELETE	DELETE FROM table_name WHERE some_column=some_value or DELETE FROM table_name (Note: Deletes the entire table!!) DELETE * FROM table_name (Note: Deletes the entire table!!)
DROP DATABASE DROP TABLE GROUP BY	DROP DATABASE database_name DROP TABLE table_name SELECT column_name, aggregate_function(column_name) FROM table_name WHERE column_name operator value GROUP BY column_name
HAVING	SELECT column_name, aggregate_function(column_name) FROM table_name WHERE column_name operator value GROUP BY column_name HAVING aggregate_function(column_name) operator value
IN	SELECT column_name(s) FROM table_name WHERE column_name IN (value1, value2, ...)

INSERT INTO	INSERT INTO table_name VALUES (value1, value2, value3, ...) <i>or</i>
	INSERT INTO table_name (column1, column2, column3, ...) VALUES (value1, value2, value3,...)
INNER JOIN	SELECT column_name(s) FROM table_name1 INNER JOIN table_name2 ON table_name1.column_name=table_name2.column_name
LEFT JOIN	SELECT column_name(s) FROM table_name1 LEFT JOIN table_name2 ON table_name1.column_name=table_name2.column_name
RIGHT JOIN	SELECT column_name(s) FROM table_name1 RIGHT JOIN table_name2 ON table_name1.column_name=table_name2.column_name
LIKE	SELECT column_name(s) FROM table_name WHERE column_name LIKE pattern
ORDER BY	SELECT column_name(s) FROM table_name ORDER BY column_name [ASC DESC]
SELECT	SELECT column_name(s) FROM table_name
SELECT *	SELECT * FROM table_name
SELECT DISTINCT	SELECT DISTINCT column_name(s) FROM table_name
SELECT TOP	SELECT TOP number percent column_name(s) FROM table_name
TRUNCATE TABLE	TRUNCATE TABLE table_name
UNION	SELECT column_name(s) FROM table_name1 UNION SELECT column_name(s) FROM table_name2
UPDATE	UPDATE table_name SET column1=value, column2=value, ... WHERE some_column=some_value
WHERE	SELECT column_name(s) FROM table_name WHERE column_name operator value

VISUAL STUDIO

Aplicación WPF

1. Clase Conexión

```
public static SqlConnection agregarConexion()
{
    SqlConnection con;
    try
    {
        con = new SqlConnection("Data Source=112SALAS29;Initial Catalog=Salas2;Persist
Security Info=True;User ID=sa;Password=sqladmin");
        con.Open();
        MessageBox.Show("Conectado");
    }
    catch (Exception ex)
    {
        con = null;
        MessageBox.Show("No se pudo conectar --" + ex);
    }
    return con;
}
```

a. Llenar combo box

```
public static void llenarCombo (ComboBox cb) {
    try {
        SqlConnection con = agregarConexion();
        SqlCommand cmd = new SqlCommand("select nombre from programa",
con);
        SqlDataReader dr = cmd.ExecuteReader();
        while (dr.Read())
        {
            cb.Items.Add(dr["nombre"].ToString());
        }
        cb.SelectedIndex = 0;
        dr.Close();
    }
    catch (Exception ex)
    {
        MessageBox.Show("no se pudo llenar el combo");
    }
}
```

2. En la clase que se necesite:

a. Alta

```
public static int agregaAlumno(Alumno a)
{
    SqlConnection con = Conexion.agregarConexion();
    SqlCommand cmd = new SqlCommand(String.Format("insert into
alumno(claveUnica, nombre, sexo, correo, semestre, idPrograma)values('{0}', '{1}',
'{2}', '{3}', '{4}', '{5}')"
a.claveUnica, a.nombre, a.sexo, a.correo, a.semestre, a.programa), con);
    int res = cmd.ExecuteNonQuery();
    con.Close();
    return res;
}
```

b. Baja

```
public static int eliminaAlumno(Int16 cu)
{
    SqlConnection con = Conexion.agregarConexion();
    SqlCommand cmd = new SqlCommand(String.Format("delete from alumno where
claveUnica='{0}'", cu), con);
    int res = cmd.ExecuteNonQuery();
    con.Close();
    return res;
}
```

c. Modificación

```
public static int modificar(Int16 f, String a)
{
    SqlConnection con = Conexion.agregarConexion();
    SqlCommand cmd = new SqlCommand(String.Format("update alumno set correo
='{0}' where claveUnica='{1}'", a, f), con);
    int res = cmd.ExecuteNonQuery();
    con.Close();
    return res;
}
```

d. Búsqueda

```
public static List<Alumno> buscar(String a)
{
    Alumno al;
    List<Alumno> lis = new List<Alumno>();
    SqlConnection con = Conexion.agregarConexion();
    SqlCommand cmd = new SqlCommand(String.Format("select * from alumno where
nombre like '%{0}%", a), con);
    SqlDataReader lec = cmd.ExecuteReader();
    while (lec.Read())
    {
        al = new Alumno();
        al.claveUnica = lec.GetInt16(0);
        al.nombre = lec.GetString(1);
    }
}
```

```

        al.sexo = lec.GetString(2);
        al.correo = lec.GetString(3);
        al.semestre = lec.GetInt16(4);
        al.programa = lec.GetInt16(5);
        lis.Add(al);
    }
    con.Close();
    return lis; }

```

3. Llenar dataGrid

```

private void btBuscar_Click(object sender, RoutedEventArgs e)
{
    if (tbNombre.Text.Equals("",""))
        MessageBox.Show("DON'T DO IT");
    else
        dgDatos.ItemsSource = Alumno.buscar(tbNombre.Text);
}

```

Sitio Web

1. Clase Conexión

```

protected OdbcConnection conectarBD()
{
    String stringConexion = "Driver={SQL Server Native Client
11.0};Server=localhost;Uid=sa;Pwd=sqladmin;Database=JuegosX";
    try
    {
        OdbcConnection conexion = new OdbcConnection(stringConexion);
        conexion.Open();
        return conexion;
    }
    catch (Exception ex)
    {
        return null;
    }
}

```

2. En la página que se necesite:

a. Iniciar sesión

```

protected void btPagina2_Click(object sender, EventArgs e)
{
    OdbcConnection miConexion = conectarBD();
    if (miConexion != null)
    {
        string query = "select claveU from usuario where email="
+ txUsuario.Text + "and password = " + txContraseña.Text + """;
        OdbcCommand sql = new OdbcCommand(query, miConexion);
        OdbcDataReader lector = sql.ExecuteReader();
    }
}

```

```

        if (lector.HasRows)
        {
            lector.Read();
            Session["CU"] = lector.GetInt16(0).ToString();
            Response.Redirect("Pagina2.aspx");
        }
    }
    else
    {
        lbContador.Text = "El usuario no se encuentra";
    }
}

```

b. Alta

```

protected void btnAlta_Click(object sender, EventArgs e)
{
    OdbcConnection con = conectarBD();
    if(con != null)
    {
        OdbcCommand sql1 = new OdbcCommand("select max(IdProf) from Prof",
con);

        OdbcDataReader lector = sql1.ExecuteReader();
        int id = 0;
        if (lector.Read()) { id = lector.GetInt16(0) + 1; }
        lector.Close();
        OdbcCommand sql = new OdbcCommand("insert into Prof values("+id+",
        '"+ tbNombre.Text + "', '"+ddProfesores.SelectedValue+"' )", con);
        int res = sql.ExecuteNonQuery();
        if(res > 0)
        {
            Response.Redirect("Reporte.aspx");
        }
    }
}

```

c. Baja

```

public static int eliminaAlumno(Int16 cu)
{
    OdbcConnection miConexion = conectarBD();
    if (miConexion != null)
    {
        OdbcCommand cmd = new SqlCommand(String.Format("delete from alumno where
        claveUnica='{0}'", cu), con);
        int res = cmd.ExecuteNonQuery();
        con.Close();
        return res;
    }
}

```

d. Modificación

```

public static void modifica( Int16 cu)
{
    OdbcConnection miConexion = conectarBD();

```



```

        if (miConexion != null)
        {
            OdbcCommand sql = new OdbcCommand("update trabajador set salario =
            300 where idTrabajador= '"+ cu +"' ", miConexion);
            int res = cmd.ExecuteNonQuery();
        }
    }
}

```

e. Llenar gridView-Búsqueda

```

protected void Page_Load(object sender, EventArgs e)
{
    OdbcConnection con = agregarConexion();
    if(con!= null)
    {
        OdbcCommand cmd = new OdbcCommand("select * from
informacion$", con);

        OdbcDataReader lector = cmd.ExecuteReader();
        GridView1.DataSource = lector;
        GridView1.DataBind();
    }
}

```

f. Llenar dropDownList

```

protected void Page_Load(object sender, EventArgs e)
{
    if (!IsPostBack)
    {
        OdbcConnection miConexion = conectarBD();
        if (miConexion != null)
        {
            OdbcCommand sql = new OdbcCommand("select juegos.nombre
            from juegos,escriben where escriben.claveU ="
            + Session["CU"].ToString() + " and escriben.claveJ=juegos.claveJ",
            miConexion);
            OdbcDataReader lector = sql.ExecuteReader();
            ddJuegos.Items.Clear();
            while (lector.Read())
            {
                ddJuegos.Items.Add(lector.GetString(0));
            }
            lector.Close();
        }
    }
}

protected void Page_Load(object sender, EventArgs e)
{
    if (DropDownList1.Items.Count == 0)
    {
        DropDownList1.Items.Add("Rojo");
        DropDownList1.Items.Add("Verde");
        DropDownList1.Items.Add("Azul");
    }
}

```

```

        DropDownList1.Items.Add("Amarillo");
        DropDownList1.Items.Add("Morado");
    }
}

```

g. Llenar radioButton

```

protected void Page_Load(object sender, EventArgs e)
{
    if (RadioButtonList1.Items.Count == 0)
    {
        RadioButtonList1.Items.Add("Vainilla");
        RadioButtonList1.Items.Add("Fresa");
        RadioButtonList1.Items.Add("Limón");
        RadioButtonList1.Items.Add("Chocolate");
    }
}

```

h. Llenar checkBox

```

protected void Page_Load(object sender, EventArgs e)
{
    if (CheckBoxList1.Items.Count == 0)
    {
        CheckBoxList1.Items.Add("Americano");
        CheckBoxList1.Items.Add("Capuchino");
        CheckBoxList1.Items.Add("Late");
        CheckBoxList1.Items.Add("Mocha");
    }
}

```

4. Session

```

protected void btPag2_Click(object sender, EventArgs e)
{
    Session["sesion"] = txUsuario.Text;
}

protected void Page_Load(object sender, EventArgs e)
{
    lblSession.Text = Session["sesion"].ToString();
}

```

5. Cookies

```

protected void btPag2_Click(object sender, EventArgs e)
{
    HttpCookie cookie1 = new HttpCookie("contraseña", txContraseña.Text);
    cookie1.Expires = new DateTime(2018, 12, 25);
    Response.Cookies.Add(cookie1);
    Response.Redirect("Pagin2.aspx");
}

```

```
protected void Page_Load(object sender, EventArgs e)
{
    lbCookie.Text = Request.Cookies["contraseña"].Value;
}
```

6. Checar que no se dupliquen datos al llenar la página web

```
if (CheckBoxList1.Items.Count == 0){ }
if(!IsPostBack){}
```

EXCEL

1. Referencias

a. Relativa

Son la combinación de una letra y un número sin caracteres o espacios entre ellos.

b. Absoluta

Permanecen fijas sin importar la cantidad de veces que sean copiadas. Se debe anteponer el símbolo \$ a la columna y fila de la referencia.

\$A7	Columna fija
A\$7	Fila fija
\$A\$7	Celda fija

2. Funciones

- MAX(), MIN(), SUMA(), CONTAR()
- PROMEDIO() → Se calcula considerando las celdas que tienen un número
- CONTAR.SI() → Una sola condición
- CONTAR.SI.CONJUNTO() → Muchas condiciones
- SUMAR.SI.CONJUNTO(), PROMEDIO.SI.CONJUNTO()
- ALEATORIO(), ALEATORIO.ENTRE()
- ELEGIR(numIndice, valor1, valor2)
- BUSCARV() → Busca verticalmente en una tabla

3. Reemplazar

- Seleccionar tabla

- b. Buscar y seleccionar→ Reemplazar

4. Seleccionar toda la fila o columna

- a. Control + Shift + Flechita hacia donde necesites

5. Operaciones con fechas

- a. HOY(), AHORA()
- b. AÑO(fecha), MES(fecha), DÍA(fecha)
- c. DIASEM(fecha)

6. Buscar objetivo

- a. Datos→ Análisis de hipótesis→ Buscar objetivo
- b. Definir la celda: celda que tenga una función.
- c. Con el valor: el valor final que necesitas.
- d. Cambiando celda: la celda que va a ajustar para dar el valor final.

7. Selección simple

- a. =SI(Condición, Proceso Verdadero, Proceso Falso)

8. Operadores lógicos

- a. Y(), O(), NO()

9. Búsqueda en tabla

- a. =BUSCARV(Valor que desea buscar, rango en el que quiere buscar el valor, el número de columna en el rango que contiene el valor devuelto, Coincidencia exacta o Coincidencia aproximada indicado como 0/FALSO o 1/VERDADERO).
- b. La tabla de selección no incluye encabezados

10. Filtros



a. Filtro simple

- i. Seleccionar la tabla con títulos
- ii. Datos
- iii. Filtrar

b. Filtro avanzado

- i. Seleccionar la tabla con títulos

- ii. Datos
- iii. Filtros avanzados

11. Tablas dinámicas

- a. Seleccionar tabla con títulos
- b. Insertar → Tabla Dinámica

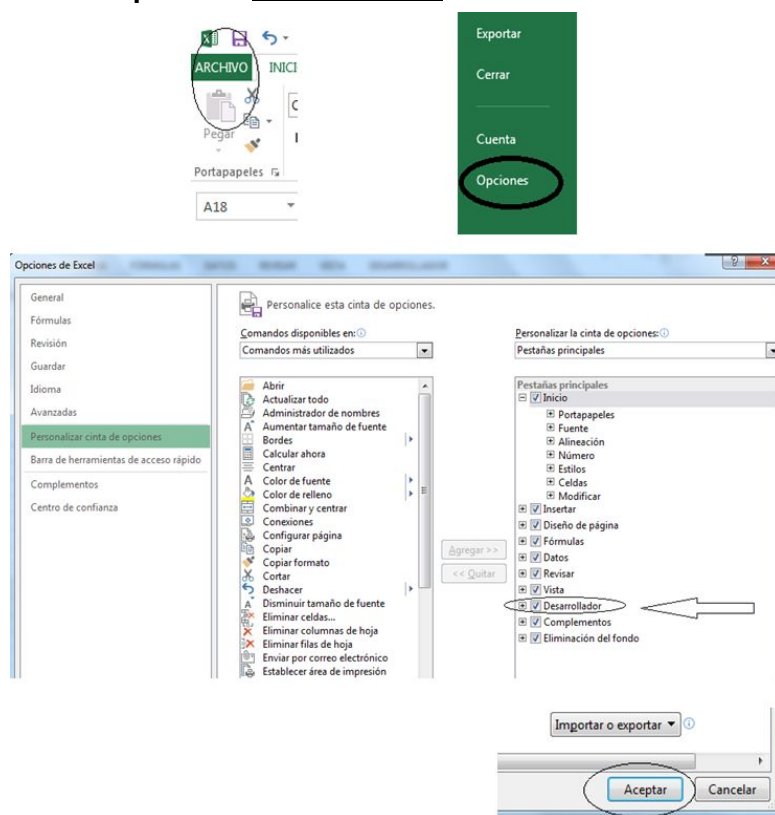
Para un filtro en específico se pone el campo en filtros y se selecciona en la tabla el dato deseado.

12. Hoja de informe

- a. Herramientas de Tabla → Analizar → Tabla Dinámica → Opciones → Mostrar páginas de filtro de informes

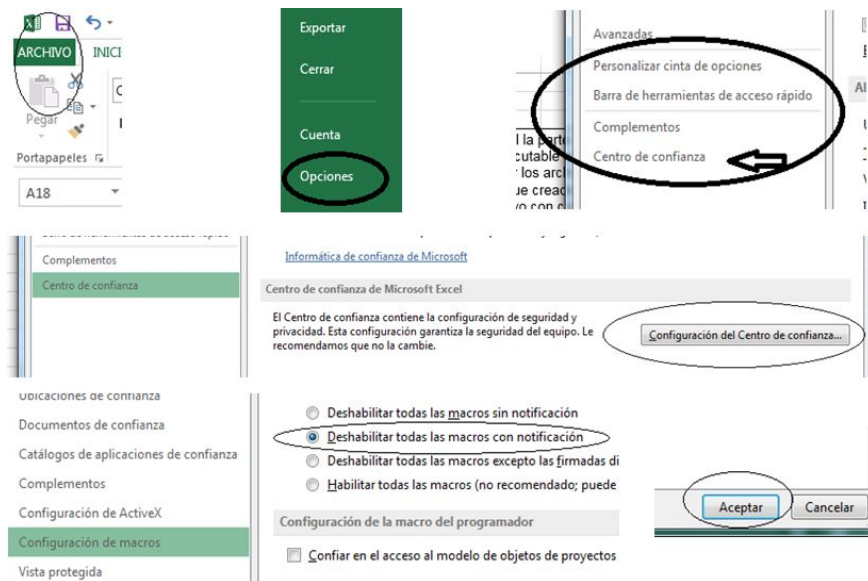
13. Excel con Visual Basic

- a. Mostrar la pestaña Desarrollador

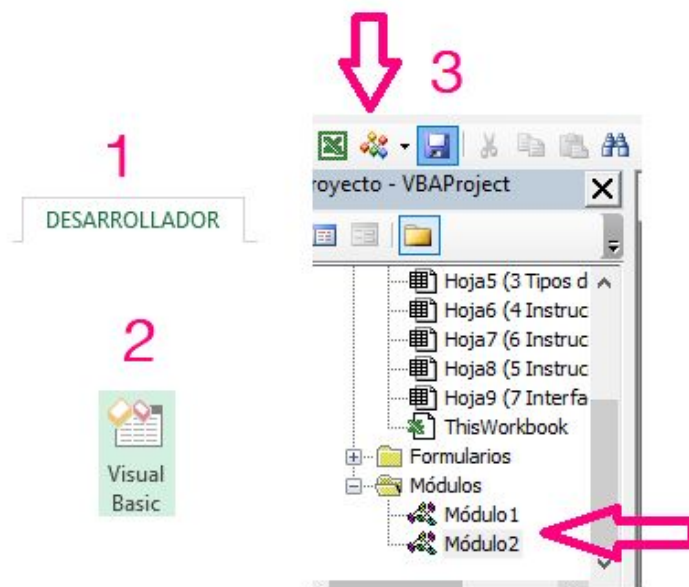


Para que el código de un libro pueda ser ejecutado, debe habilitarse o activarse, después de reconocer que el archivo proviene de un origen seguro.

- b. Habilitar el uso de Macros



c. Código ejecutable



Funciones `function()` y procedimientos `sub()`
 Las funciones regresan valores, los `sub()` no

i. `Function()`

nombre	parámetros	tipo del resultado
<code>Function Area(x</code>	<code>As Double, y As Double)</code>	<code>As Double</code>
<code>Area = x * y</code>	instrucciones	nombre de la función para indicar el resultado
<code>End Function</code>		

ii. Sub()

nombre **parámetros**

```
Sub PerimetroRectangulo(x As Double, y As Double)

    MsgBox x * 2 + y * 2      indicaciones

End Sub
```

iii. Ejemplo de macro 1

```
Function contarDato(dato As Integer, celdas As Range) As Integer
    Dim res As Integer, valor As Range
    res = 0
    For Each valor In celdas
        If valor.Value = dato Then
            res = res + 1
        End If
    Next
    contarDato = res
    MsgBox "hay : " & res
End Function
```

```
Function impuesto() As Integer
    Dim celda As Integer
    celda = Range("J1").Value
    impuesto = celda * 1.16
End Function
```

iv. Ejemplo de macro 2

```
Sub rutina()
    Dim n1, n2 As Integer                      → Declarar variables del método
    n1 = Range("A1").Value                    → Selecciona el valor de una celda en específico
    n2 = Range("B1").Value                    → Selecciona el valor de una celda en específico
    Range("A2").Value = n1 * n2              → Asigna el valor a una celda en específico
    Range("C17").Activate                    → Mueve el cursor a una celda en específico
End Sub
```

→ Yo lo hice Padilla <3 atte Carlitos

v. Interfaces (UserForm)

Visual Basic → Insertar → UserForm

Los métodos se hacen al dar doble click en un botón de la interfaz

1. Método Cerrar UserForm

```
Private Sub btCerrar_Click()
    Unload Me
End Sub
```


2. Método Limpiar UserForm

Ejemplo de la calculadora

```
Private Sub btLimpiar_Click()  
    txtNum1.Value = ""  
    txtNum2.Value = ""  
    rbSuma.Value = False  
    rbResta.Value = False  
    rbMult.Value = False  
    rbDivision.Value = False  
End Sub
```

3. Método Resultado UserForm

```
Private Sub btResultado_Click()  
    Dim n1, n2, ans As Double  
    n1 = CDb1(txtNum1.Value)————→ Con CDb1(cosa) haces cast de String a Double  
    n2 = CDb1(txtNum2.Value)————→ Con Int(cosa) haces cast de string a Integer  
    If obSuma.Value = True Then  
        ans = n1 + n2  
    Else  
        If obResta.Value = True Then  
            ans = n1 - n2  
        Else  
            If obMult.Value = True Then  
                ans = n1 * n2  
            Else  
                ans = n1 / n2  
            End If  
        End If  
    End If  
    MsgBox "El resultado es: " & ans  
End Sub
```

4. Llenar ComboBox

```
Private Sub UserForm_Initialize() → tiene que ser en el UserForm en modo  
    cbCategoria.AddItem ("Pastel")  
    cbCategoria.AddItem ("Tarta")  
End Sub
```

5. Llenar ComboBox a partir de otro ComboBox

```
Private Sub cbCategoria_Change()  
    cbNombre.Clear  
    If cbCategoria.Value = "Pastel" Then  
        cbNombre.AddItem ("Chocolate")  
        cbNombre.AddItem ("Vainilla")  
        cbNombre.AddItem ("Tres Leches")  
    Else  
        cbNombre.AddItem ("Limón")  
        cbNombre.AddItem ("Oreo")  
        cbNombre.AddItem ("Vainilla")  
    End If  
End Sub
```


6. Utilizar funciones en los botones

```
Private Sub CommandButton1_Click()  
    Call multiplica  
End Sub  
  
Sub multiplica()  
    Dim celda As Range  
    For Each celda In Range("A1", "A10")  
        celda.Value = celda.Value * 10  
    Next  
  
End Sub
```

Instrucciones de selección

Selección simple

```
If condición o proposición Then  
    instrucciones_V  
End If
```

Selección múltiple

```
If proposición_1 Then  
    instrucciones_V1  
Elif proposición_2 Then  
    instrucciones_V2  
...  
  
Elif proposición_N Then  
    instrucciones_VN
```

Else

```
    instrucciones_FN  
End If
```

Selección por casos

```
Select Case expresión  
    Case valor o valores_1  
        instrucciones_1  
    Case valor o valores_2  
        instrucciones_2  
    ...  
    ...  
    Case Else  
        instrucciones_else  
End Select
```

Instrucciones de repetición

For

```
For contador = inicio To final [Step  
incremento]  
    instrucciones  
Next
```

For each

```
For Each elemento In grupo  
    instrucciones  
Next
```

Do (while)

```
Do [{While | Until} condición]  
    instrucciones  
Loop
```

InputBox y MsgBox

InputBox()

Inputbox() es una función que despliega una ventana para recibir información como texto. El tipo de valor que entrega la función Inputbox() es String, un texto a su vez. Sin embargo, el valor entregado puede ser transformado a un valor con otra representación (número entero o decimal, fecha, hora). Es decir, los textos "56", "1.7e9", "13-oct-2015 16:35" pueden ser convertidos (implícita o explícitamente) a tipo Integer, Double, Date, respectivamente.

MsgBox()

MsgBox es un comando que despliega una ventana para mostrar información. La función MsgBox() puede entregar un valor constante entero predefinido dado por la acción de un usuario como el oprimir un botón.

Escritura en las celdas

- indicando la celda activa, mediante el objeto ActiveCell
- indicando la ubicación absoluta de la celda, mediante la función Range()
- indicando la ubicación relativa con respecto a una celda dada o a la celda activa, mediante el método Offset()

El método celdaRef.Offset(r,c) debe aplicarse a una celda de referencia celdaRef, especificando la cantidad de renglones hacia abajo r y la cantidad de columnas a la derecha c para la celda que se quiere ubicar; valores negativos de las cantidades descritas, implican las direcciones opuestas: arriba e izquierda, respectivamente. (Ver Figura R)

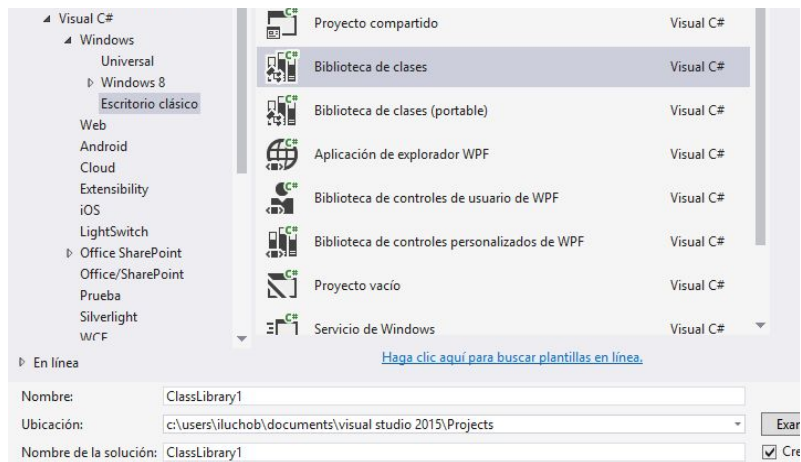
La escritura puede hacerse cambiando el valor de la propiedad Value de la celda considerada.

La lectura del valor escrito en la celda puede hacerse a partir del valor dado por la propiedad Value.

Nota: si no se especifica la hoja del libro donde quiere realizarse la escritura o lectura, la operación se aplica a la hoja activa.

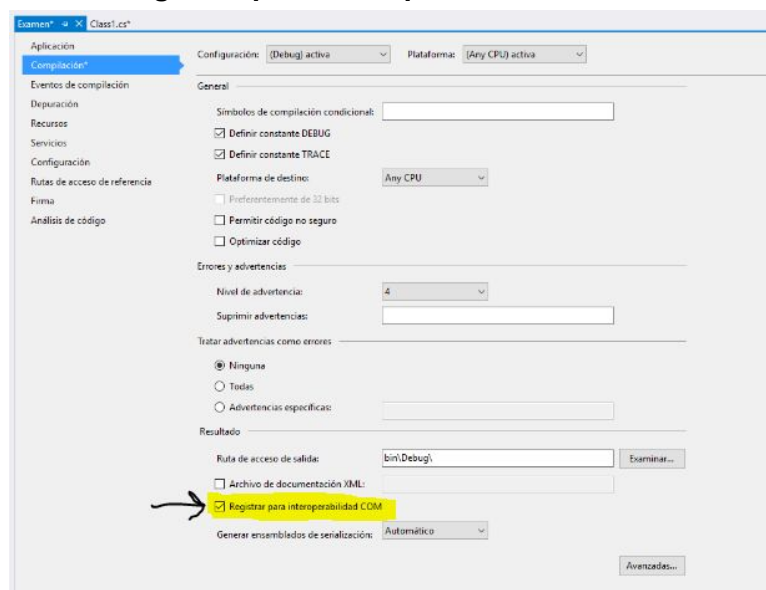
DLL

1. Visual Studio → Click derecho → Ejecutar como administrador
2. Nuevo Proyecto → Biblioteca de clases (.NET Standard)

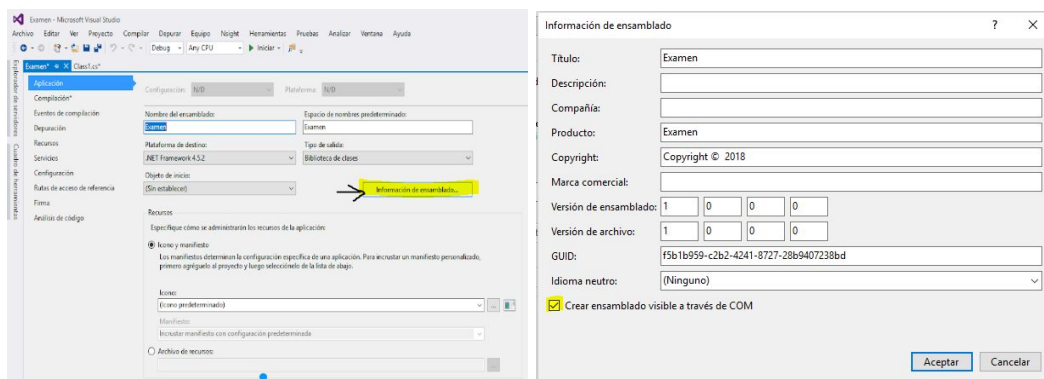


3. Proyecto → Propiedades *nombre del proyecto*

4. Compilación → Registrar para interoperabilidad COM



5. Aplicación → Información de ensamblado → Marcar “Crear ensamblado visible a través de COM”



6. Compilar → Compilar solución
FALTA IGUAL CÓMO USARLO EN EXCEL

El método en Excel debe llamarse igual que el método en el DLL

Android

1. Clase DatabaseHelper o AdminSQLiteHelper

En esta clase se crea la base de datos. Hereda de la clase SQLiteOpenHelper los métodos onCreate y onUpgrade

Utilizaremos esta clase para realizar cambios, leer o extraer información la base de datos

O AdminSQLiteHelper

```
public class DatabaseHelper extends SQLiteOpenHelper {  
  
    public DatabaseHelper(Context context) {  
        super(context, name: "ListFood", factory: null, version: 1);  
    }  
}
```

Nombre de la base de datos

En el método onCreate creamos la base de datos

```
// Creamos las tablas de nuestra base de datos y le insertamos valores a las tablas  
public void onCreate(SQLiteDatabase sqLiteDatabase) {  
    sqLiteDatabase.execSQL("create table REGISTRO( idUsuario text primary key, " +  
        " contra text);");  
    sqLiteDatabase.execSQL("create table INGREDIENTES( idIngrediente text primary key, " +  
        "nombreIngrediente text, " +  
        "calorias integer, " +  
        "cantidad integer, " +  
        "caducidad date , " +  
        "idU text references REGISTRO(idUsuario) );");  
    sqLiteDatabase.execSQL("create table RECETAS( idReceta text primary key, " +  
        "nombre text, " +  
        "ingredientes text, " +  
        "procedimiento text);");  
}
```

En el método onUpgrade actualiza la base de datos

```
// Este método elimina las tablas que ya se encuentren en la base de datos y las actualiza  
@Override  
public void onUpgrade(SQLiteDatabase sqLiteDatabase, int oldVersion, int newVersion) {  
    sqLiteDatabase.execSQL("drop table if exists REGISTRO");  
    sqLiteDatabase.execSQL("drop table if exists INGREDIENTES");  
    sqLiteDatabase.execSQL("drop table if exists RECETAS");  
  
    onCreate(sqLiteDatabase);  
}
```

2. Otras clases

Cada vez que se despliega la pantalla se realizan las acciones dentro del Oncreate.

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_alta_alimento);
    // Asocia atributos de la clase con elementos gráficos
    bundle = this.getIntent().getExtras();
    nom = (EditText) findViewById(R.id.eTNombreAlimento);
    cal = (EditText) findViewById(R.id.eTCalorias);
    cant = (EditText) findViewById(R.id.eTCantidad);
    cad = (EditText) findViewById(R.id.eTCaducidad);
}
```

a. Enlazar botones con métodos

i. En la clase:

```
public void crearCuenta (View v){
    Intent intent = new Intent( packageContext: Inicio.this, CrearCuenta.class);
    startActivity(intent);
}
```

ii. En la interfaz

```
<Button
    android:id="@+id/btCrearCuenta"
    android:layout_width="130dp"
    android:layout_height="39dp"
    android:layout_alignParentBottom="true"
    android:layout_centerHorizontal="true"
    android:layout_marginBottom="98dp"
    android:background="@android:color/holo_gre
    android:onClick=crearCuenta"
    android:text="Crear Cuenta"
    app:layout_constraintEnd_toEndOf="parent"
```

Crear un Intent (Redirect) y Bundle (Session)

a. En la página en donde estoy

Objeto que nos lleva a otra página	En donde estamos	A donde queremos ir
<pre>Intent intent = new Intent(packageContext: Inicio.this, Principal.class);</pre>		
<pre>Bundle bundle = new Bundle(); bundle.putString("usuario", usuario.getText().toString());</pre>		
	Nombre del bundle	Información del bundle
<pre>intent.putExtras(bundle);</pre>		
<pre>startActivity(intent);</pre>		Se lleva al bundle a la nueva página
Ejecuta la acción		

b. En la página a la que se redirige

Recuperamos el bundle

```
Bundle bundle = this.getIntent().getExtras();
```

```
String usuario = bundle.getString( key: "usuario");
```

Recuperamos la información con el nombre del bundle

b. Iniciar sesión

```
public void iniciarSesion(View view) {
```

Donde esta nuestra base

```
DatabaseHelper db = new DatabaseHelper( context: this);
```

```
SQLiteDatabase sq = db.getReadableDatabase();
```

Nos permite hacer cambios

Trae información de la base de datos

```
Cursor cursor = sq.rawQuery( sq: "select contra from REGISTRO where idUsuario = " +  
    "'" + usuario.getText().toString() + "';", selectionArgs: null);
```

```
//Si cursor.getCount()>0, entonces el usuario
```

```
// proporcionado si existe en la base de datos
```

```
if(cursor.getCount()>0) {
```

```
    cursor.moveToFirst(); Necesario para empezar a leer la info
```

```
    // Verifica si la contraseña es correcta
```

```
    if(cursor.getString( columnIndex 0).equals(password.getText().toString())) {
```

```
        Intent intent = new Intent( packageContext Inicio.this, Principal.class);
```

```
        Bundle bundle = new Bundle();
```

```
        bundle.putString("usuario",usuario.getText().toString());
```

```
        intent.putExtras(bundle);
```

```
        startActivity(intent);
```

```
    } else {
```

```
        Toast.makeText( context: Inicio.this, text: "Contraseña incorrecta", Toast.LENGTH_LONG);
```

```
    }
```

```
// Si el cursor.getCount()=0 es por que no existe el usuario
```

```
} else {
```

```
    Toast.makeText( context: Inicio.this, text: "Usuario incorrecto", Toast.LENGTH_LONG);
```

```
}
```

c. Alta

```
public void iniciarSesion(View view) {
```

Donde esta nuestra base

```
DatabaseHelper db = new DatabaseHelper( context: this);
```

```
SQLiteDatabase sq = db.getReadableDatabase();
```

Nos permite hacer cambios

Trae información de la base de datos

```
Cursor cursor = sq.rawQuery( sq "select contra from REGISTRO where idUsuario = " +
    "'" + usuario.getText().toString() + "';", selectionArgs: null);

// Si cursor.getCount() > 0, entonces el usuario
// proporcionado si existe en la base de datos
if(cursor.getCount() > 0) {
    cursor.moveToFirst(); → Necesario para empezar a leer la info
    // Verifica si la contraseña es correcta
    if(cursor.getString( columnIndex 0 ).equals(password.getText().toString())) {
        Intent intent = new Intent( packageContext Inicio.this, Principal.class);
        Bundle bundle = new Bundle();
        bundle.putString("usuario", usuario.getText().toString());
        intent.putExtras(bundle);
        startActivity(intent);
    } else {
        Toast.makeText( context Inicio.this, text "Contraseña incorrecta", Toast.LENGTH_LONG);
    }
}

// Si el cursor.getCount() = 0 es por que no existe el usuario
} else {
    Toast.makeText( context Inicio.this, text "Usuario incorrecto", Toast.LENGTH_LONG);
}
```

d. Baja

```
public void baja(View view) {
    DatabaseHelper db = new DatabaseHelper(this);
    SQLiteDatabase sq = db.getWritableDatabase();

    long res = sq.delete("INGREDIENTES", → Nombre de la tabla
        "nombreIngrediente = '" + nombre.getText().toString() + "' and idU
        = '" + bundle.getString("usuario") + "'", null); } Condición

    if(res > 0) {
        Toast.makeText(this, "Se elimino el alimento", Toast.LENGTH_LONG);
    }
    else {
        Toast.makeText(this, "No se encontró el alimento", Toast.LENGTH_LONG);
    }
}
```

e. Modificación

```
// Actualiza la cantidad del alimento en la base de datos
public void modificar(View view) {
    DatabaseHelper db = new DatabaseHelper( context this);
    SQLiteDatabase sq = db.getWritableDatabase();
    sq.execSQL("update INGREDIENTES set cantidad = "+cant.getText().toString()+
        " where nombreIngrediente like '"+spinner.getSelectedItem().toString()+
        "' and idU like '"+bundle.getString( key: "usuario")+"'");
}
```

f. Llenar spinner con base de datos

Para hacer eso hace uso de una Lista implementada con un ArrayList

```
//Llena el spinner con los alimentos de la base de datos asociados al usuario
List<String> alimentos = new ArrayList<String>();
DatabaseHelper db = new DatabaseHelper( context: this);
SQLiteDatabase sq = db.getReadableDatabase();

String usuario = bundle.getString( key: "usuario");
Cursor cursor = sq.rawQuery( sql: "select nombreIngrediente from INGREDIENTES " +
    "where INGREDIENTES.idU = '"+usuario+"';", selectionArgs: null);

while(cursor.moveToNext()){
    alimentos.add(cursor.getString( columnIndex: 0));
}

cursor.close();
ArrayAdapter<String> adapter = new
ArrayAdapter<String>(this,android.R.layout.simple_spinner_dropdown_item,alimentos);
spinner.setAdapter(adapter);
```

Usamos una lista para llenar el spinner

Almacena la info de la base de datos en la lista

El Adapter permite que la lista se despliegue con el formato del Spinner

i. Si es una lista

```
ArrayAdapter adapter = new
ArrayAdapter(this,android.R.layout.simple_list_item_1,lista_recetas);
lista.setAdapter(adapter);
```

g. Página Web

Activar los permisos dentro del Manifest:

```
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
```

Luego en su clase:

```
public class RecetasWeb extends AppCompatActivity {
    WebView w;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_recetas_web);
        // Asocia atributos de la clase con elementos gráficos
        w = (WebView) findViewById(R.id.webView);
        // Carga la pagina web seleccionada
        w.loadUrl("https://www.kiwilimon.com/recetas");
        w.getSettings().setJavaScriptEnabled(true);
        w.setWebViewClient(new WebViewClient());
    }
}
```


h. Realizar una acción cuando se selecciona un elemento

Si se carga desde que abres la pagina, va dentro del Oncreate:

```
lista.setOnItemClickListener(new AdapterView.OnItemClickListener() {  
    @Override  
    public void onItemClick(AdapterView<?> parent, View view, int position, long id)  
    {  
        Identifica el elemento seleccionado  
        String titulo = parent.getItemAtPosition(position).toString();  
  
        Intent intent = new Intent(RecetasApp.this, VerRecetas.class);  
        Bundle bundle= new Bundle();  
        bundle.putString("titulo", titulo.toString());  
        intent.putExtras(bundle);  
        startActivity(intent);  
    }  
});
```

i. Enlazar ventanas

```
public void setBtInfo(View view) {  
    Intent intent = new Intent(MainActivity.this, Info.class);  
    startActivity(intent);  
    //NO OLVIDAR PONER EL ONCLICK EN EL DISEÑO  
}  
}
```

3. Ejemplo

EXAMEN FINAL - PRÁCTICA DOS

Realiza una aplicación móvil para el sistema de adopción de perros de un refugio. La aplicación debe tener dos registros uno de las personas que adoptan a los perros con su nombre, correo electrónico y teléfono, el segundo registro debe de ser de los datos de los perros: nombre, talla (grande, mediana, chica) y edad. En una segunda pantalla se debe de seleccionar a un perro por nombre (de una lista) y debe de dar la información: adoptado y el nombre del dueño o no esta adoptado.

```
public class SQLiteOpenHelper extends android.database.sqlite.SQLiteOpenHelper{  
    public SQLiteOpenHelper (Context context) {  
        super(context, "Perros", null, 2);  
    }  
  
    @Override  
    public void onCreate(SQLiteDatabase sqLiteDatabase) {  
        sqLiteDatabase.execSQL("create table Perro (CU integer primary key  
autoincrement, nombre text, talla text, edad int, adoptado int, dueño text);");  
        sqLiteDatabase.execSQL("insert into Perro values (0,'Aldo', 'chica', 2,  
0,'');");  
        sqLiteDatabase.execSQL("insert into Perro values (1,'Tocino', 'mediana', 1,  
1,'Padilla');");  
    }  
}
```

```

@Override
public void onUpgrade(SQLiteDatabase sqLiteDatabase, int i, int i1) {
    sqLiteDatabase.execSQL("drop table if exists Perro;");
    onCreate(sqLiteDatabase);
}
}

public class MainActivity extends AppCompatActivity {

    Button btAlta, btInfo;
    EditText etNombre, etEdad, etAdoptado, etDuenio;
    Spinner spTalla;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        btAlta = (Button) findViewById(R.id.btAlta);
        btInfo = (Button) findViewById(R.id.btInfo);
        etNombre = (EditText) findViewById(R.id.etNombre);
        etEdad = (EditText) findViewById(R.id.etEdad);
        etAdoptado = (EditText) findViewById(R.id.etAdoptado);
        etDuenio = (EditText) findViewById(R.id.etDuenio);
        spTalla = (Spinner) findViewById(R.id.spTalla);

        List<String> h = new ArrayList<String>();
        h.add("Chica");
        h.add("Mediana");
        h.add("Grande");
        ArrayAdapter<String> ad = new
ArrayAdapter<String>(this, android.R.layout.simple_spinner_dropdown_item, h);
        spTalla.setAdapter(ad);
    }

    public void alta(View v) {
        try {
            SQLiteOpenHelper db = new SQLiteOpenHelper(this);
            SQLiteDatabase nuevoPerro = db.getWritableDatabase();
            ContentValues contentValues = new ContentValues();
            contentValues.put("nombre", etNombre.getText().toString());
            contentValues.put("talla", spTalla.getSelectedItem().toString());
            contentValues.put("edad",
Integer.parseInt(etEdad.getText().toString()));
            contentValues.put("adoptado",
Integer.parseInt(etAdoptado.getText().toString()));
            contentValues.put("dueño", etDuenio.getText().toString());
            long res = nuevoPerro.insert("Perro", null, contentValues);

            if (res == -1) {
                Toast toast = Toast.makeText(this, "No se pudo añadir el perro",
Toast.LENGTH_LONG);
                toast.show();
            } else {

```

```

        Toast toast = Toast.makeText(this, "Se añadió el perro",
Toast.LENGTH_LONG);
        toast.show();
    }
} catch (Exception e){

        Toast toast = Toast.makeText(this,"No se pudo añadir el
perro",Toast.LENGTH_LONG);
        toast.show();
    }
}
public void setBtInfo(View view) {
    Intent intent = new Intent(MainActivity.this, Info.class);
    startActivity(intent);
}
}

public class Info extends AppCompatActivity {

    Button regresar,mostrar;
    ListView lista;
    Spinner perros;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_info);

        perros=(Spinner)findViewById(R.id.spPerros);
        mostrar=(Button)findViewById(R.id.btMostrar);
        regresar=(Button)findViewById(R.id.btRegresar);
        lista=(ListView)findViewById(R.id.listLista);

        List<String> p = new ArrayList<String>();
        SQLiteOpenHelper db = new SQLiteOpenHelper(this);
        SQLiteDatabase sq = db.getReadableDatabase();
        Cursor cursor = sq.rawQuery("select nombre from Perro;",null);
        while(cursor.moveToNext()){
            p.add(cursor.getString(0));
        }
        cursor.close();
        ArrayAdapter<String> ad = new
ArrayAdapter<String>(this,android.R.layout.simple_spinner_dropdown_item,p);
        perros.setAdapter(ad);
    }

    public void setRegresar(View v){
        Intent intent = new Intent(Info.this, MainActivity.class);
        startActivity(intent);
    }

    public void setMostrar(View v){
        List<String> p = new ArrayList<String>();
        SQLiteOpenHelper db = new SQLiteOpenHelper(this);
        SQLiteDatabase sq = db.getReadableDatabase();

```

```

        Cursor cursor = sq.rawQuery("select dueño from Perro where nombre like
'" + perros.getSelectedItemAt().toString() + "'";", null);
        while(cursor.moveToNext()) {
            p.add(cursor.getString(0));
        }
        cursor.close();
        ArrayAdapter<String> ad = new
ArrayAdapter<String>(this, android.R.layout.simple_list_item_1, p);
        lista.setAdapter(ad);
    }
}

```