```
//  +++++++++  PLATAFORMA BASE DE DATOS PARA TARIFACIÓN DE PRODUCTOS  +++++++++

/*
 El código que precede es el responsable de ejecutar funciones que permiten interactuar
 a la Base de datos en Microsoft Access con un dispositivo externo para tarifación de
 productos a través de la interfaz serial hacia un módulo inalámbrico XBEE que cuenta
 con el protocolo de comunicación IEEE 802.15.4

 La presente plataforma dispone de una sección para redacción y difusión de ofertas o
 mensajes que serán transmitidos por medio del puerto de comunicación serial. Por otra
 parte permite visualizar las adquisiones tanto del coche A como del coche B únicamente
 con fines de comprobación de la operación del tarifador de productos inalámbrico.
 */

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using System.Threading;
using My = Microsoft.VisualBasic.Devices; //Espacio nombres para relación con dispositivos
using System.IO;        // Permite leer y escribir en los archivos y secuencias de datos

namespace Identificador_Prod
{
    public partial class Form1 : Form
    {
        // Declaración de variables globales
        string codigo, detalle, dirOrig;
        int cont, aux, ini_fact, n, aux_eliminar;
        double sumtotalCh1, sumtotalCh2, precio;
        int[] buffer = new int[50];
        string datos_recibidos;
        string[] cadena_Acceso = new string[6];

        string coche_A = "A01";     // Direccionamiento coches
        string coche_B = "A02";

        // Requerimientos Comunicación Serial UART para uso de los puertos
        // bajo software con el uso de librerías de VisualBasic.
        System.IO.Ports.SerialPort o = new System.IO.Ports.SerialPort();
        public delegate void WriteDataDelegate(string str);
        My.Ports ports = new Microsoft.VisualBasic.Devices.Ports();

        public Form1()
        {
            InitializeComponent();
            precio = sumtotalCh1 = sumtotalCh2 = 0.00;          // Inicialización Precios
            txtMensaje.Text = "* Dscto. 10% en Perfumeria *";  // Mensaje u ofertas por ↙
    defecto
            txtMensaje2.Text = "* Hoy 2X1 Recargas CLARO *";
        }
        private void Form1_Load(object sender, EventArgs e)
        {
            try
            {
                Identificador_Prod.Conexion.Conectar();    // Llamado a la clase Conexión ↙
    para Base datos
                //MessageBox.Show("Conexion establecida con éxito, COM5 2400", "CONEXION", ↙
     MessageBoxButtons.OK, MessageBoxIcon.Information);
                Cargar();
                CargarContador();

                //Configuracion del Puerto UART
                o = ports.OpenSerialPort("COM5", 2400, System.IO.Ports.Parity.None, 8,     ↙
    System.IO.Ports.StopBits.One);
                o.DataReceived += new System.IO.Ports.SerialDataReceivedEventHandler       ↙
    (o_DataReceived);
                txtFecha.Text = DateTime.Now.ToString();  // Hora y fecha sistema
            }
            catch (Exception)
            {
```

```
            }
        }
        private void Cargar()
        {       // Subrutina para disponer de los datos de productos en la aplicación
            try
            {
                Identificador_Prod.Conexion.CargarTabla("select *from PRODUCTOS",
    dataGrid4);
            }
            catch (Exception e)
            {
                MessageBox.Show(e.ToString());    // Visualiza mensaje en caso de error
            }
        }
        private void CargarContador()
        {           // Permite disponer del número de registros de productos en la BDD
            try
            {
                Identificador_Prod.Conexion.CargarTabla("SELECT COUNT (*) AS CONT FROM
    PRODUCTOS", dgTabla);
                cont = Convert.ToInt32(dgTabla[0, 0].ToString());
            }
            catch (Exception e)
            {
                MessageBox.Show(e.ToString());  // Visualiza mensaje en caso de error
            }
        }

        void o_DataReceived(object sender, System.IO.Ports.SerialDataReceivedEventArgs e)
        {
            string strData = o.ReadExisting();  // Adquisición de datos externos

            if (this.textBox1.InvokeRrequired)
            {
                WriteDataDelegate WriteInvoke = new WriteDataDelegate(this.WriteData);  //
     Reserva espacio memoria
                this.Invoke(WriteInvoke, strData);
            }
            else
            {
                this.txtCodigo.Text = strData;
                txtCodigo.Update();
                this.textBox1.Text = strData;    // Carga de datos obtenidos
                textBox1.Update();
            }
        }
        public void WriteData(string strDatos)
        {
            this.textBox1.Text += strDatos;     //  Integración de datos leídos
        }

        public void Resp_Uart()
        {                                       // sentencia trama respuesta UART
            try
            {
                o.Write("+");                   // Caracter inicio trama
                o.Write(dirOrig.Trim());        // Dirección coche
                o.Write("^");                   // Delimitador trama
                o.WriteLine(detalle.Trim());    // Descripción producto
                o.Write("_");                   // Delimitador trama
                if (aux_eliminar == 1)          // Verificación para desglose precio
                    o.WriteLine(("-" + precio.ToString()).Trim());
                else
                    o.WriteLine((precio.ToString()).Trim());
                o.Write("[");                   // Delimitador trama
                if (dirOrig == coche_A)
                {                           // Envío compra total acorde a Dir. coche
                    o.WriteLine((sumtotalCh1.ToString()).Trim());
                }
                if (dirOrig == coche_B)
                {
                    o.WriteLine((sumtotalCh2.ToString()).Trim());
                }
                o.Write("]".Trim());            // Caracter fin de trama
```

```
            }
            catch (Exception e)
            {
                MessageBox.Show(e.ToString()); // Visualiza mensaje en caso de error
            }
        }

        private void timer1_Tick_1(object sender, EventArgs e)
        {
            datos_recibidos = this.textBox1.Text; // almacenamiento datos en variable local
            textBox1.Text = "";               // borrado casillero de recepción de datos

            if (datos_recibidos.ToString() != "")  // Verifica si existen datos recibidos
            {
                Discriminar_Datos();        // Llamado a subrutina caracterización datos
                txtOrigen.Text = dirOrig;   // Almacenamiento en variable de Dirección coche

                if (txtOrigen.Text == coche_A)  // Comprobación Identificación coche
                {
                    listBox3.Items.Add(codigo);
                    txtCodigo.Text = codigo;    // Visualización variable código barras
                }
                if (txtOrigen.Text == coche_B)  // Comprobación Identificación coche
                {
                    listBCod2.Items.Add(codigo);
                    txtCodigo2.Text = codigo;   // Visualización variable código barras
                }
            }

            if (codigo != "" & codigo != null) // Verifica disponibilidad de código
            {
                aux = 0;
                for (int i = 0; i < cont; i++) // Método búsqueda matriz de datos
                {
                    // Comparación registro para Coche "A1"
                    if (Convert.ToString(dataGrid4[i, 0]) == txtCodigo.Text)
                    {          // Registro datos localizados
                        detalle = (Convert.ToString(dataGrid4[i, 1]).Trim());
                        listBox1.Items.Add(detalle);
                        precio = Convert.ToDouble((dataGrid4[i, 2]));

                        if (aux_eliminar == 1)     // Desglose precio producto
                        {
                            listBox2.Items.Add("-" + precio);
                            sumtotalCh1 = sumtotalCh1 - precio;
                        }
                        else
                        {
                            listBox2.Items.Add(precio);  // visualización precio
                            sumtotalCh1 = sumtotalCh1 + precio;  // Cálculo total compra
                        }
                        txtTotal.Text = Convert.ToString(sumtotalCh1); // visualización total
                        aux = 1;
                        //////
                        Resp_Uart();   // Llamado subrutina transmisión de datos

                        txtCodigo.Text = "";
                    }

                    // Comparación registro para Coche "A2"
                    if (Convert.ToString(dataGrid4[i, 0]) == txtCodigo2.Text)
                    {           // Registro datos localizados
                        detalle = (Convert.ToString(dataGrid4[i, 1]).Trim());
                        listBDetalle2.Items.Add(detalle);
                        precio = Convert.ToDouble((dataGrid4[i, 2]));
                        if (aux_eliminar == 1)     // Desglose precio producto
                        {
                            listBPrecio2.Items.Add("-" + precio);// visualización precio
                            sumtotalCh2 = sumtotalCh2 - precio; // Cálculo total compra
                        }
                        else
```

```
                    {
                        listBPrecio2.Items.Add(precio);  // visualización precio
                        sumtotalCh2 = sumtotalCh2 + precio;  // Cálculo total compra
                    }
                    txtTotal2.Text = Convert.ToString(sumtotalCh2);
                    aux = 1;

                    //////
                    Resp_Uart();   // Llamado subrutina transmisión de datos
                    txtCodigo2.Text = "";
                }
            }
            codigo = "";
            if (aux == 0)   // Auxiliar para verificar existencia producto
            {                    //   Envío trama
                o.Write("+");
                o.Write(dirOrig.Trim());       // Dirección coche
                o.Write("^");
                o.Write("PRODUCTO NO REGISTRADO");
                o.Write(";");
            }
            aux_eliminar = 0;   // Reset variable para eliminar producto
        }
    }
    public void Discriminar_Datos()   // Método caracterización datos
    {
        codigo = "";
        dirOrig = "";
        if (datos_recibidos.ToString() != "")  // Verificación de datos en buffer
        {
            // Comprobación de fin de recepción datos
            if (datos_recibidos[datos_recibidos.Length - 1].ToString() == "^")
            {
                try
                {
                    cadena_Acceso = datos_recibidos.Split('^'); // Delimitación dados ↙
en vector
                    dirOrig += cadena_Acceso[1];                // Identificación     ↙
Dirección coche
                    if (datos_recibidos[0].ToString() == "P")   // "P" indicador para ↙
publicidad
                    {
                        Publicar();       //  Llamado subrutina para difusión ofertas/↙
mensajes
                    }

                    // caracter "@" indica q se inicializa nuevamente la venta
                    if (datos_recibidos[0].ToString() == "@" && dirOrig == coche_A)
                    {
                        ini_fact = 1;            // auxiliar indicador nueva          ↙
tarifación
                        listBox1.Items.Clear();
                        listBox2.Items.Clear();  // Borrado de datos visualizados
                        listBox3.Items.Clear();
                        txtTotal.Text = "";
                        sumtotalCh1 = 0;         // Reset valor compra total
                    }
                    if (datos_recibidos[0].ToString() == "@" && dirOrig == coche_B)
                    {
                        ini_fact = 1;            // auxiliar indicador nueva          ↙
tarifación
                        listBCod2.Items.Clear();
                        listBDetalle2.Items.Clear(); // Borrado de datos visualizados
                        listBPrecio2.Items.Clear();
                        txtTotal2.Text = "";
                        sumtotalCh2 = 0;            // Reset valor compra total
                    }

                    if (datos_recibidos[0].ToString() == "*")
                        aux_eliminar = 1;    // Activación método para desglose       ↙
producto

                    if (ini_fact == 1)
                    {
```

```
                            codigo += cadena_Acceso[2].Trim();  // Carga del código de barras
                        }
                    }
                    catch (Exception)
                    {
                    }
                }
            }
        }

        private void salirToolStripMenuItem1_Click(object sender, EventArgs e)
        {
            Form1.ActiveForm.Close();     // Desactivación o cierre ventana tarifación
        }

        // Método para difusíon de mensajes u ofertas de manera continua
        public void Publicar()
        {
            try
            {
                switch (n)
                {
                    case 1:
                        o.Write("^");   // trama envío mensaje 1ra instancia
                        o.Write(txtMensaje.Text.Trim());
                        o.Write(";");
                        n = 2;
                        break;
                    case 2:
                        o.Write("^");   // trama envío mensaje 2da instancia
                        o.Write(txtMensaje2.Text.Trim());
                        o.Write(";");
                        n++;
                        break;

                    default:            // trama envío mensaje por defecto
                        o.Write("^");
                        o.Write("Pensamos en tu comodidad");
                        o.Write(";");
                        n = 1;
                        break;
                }
            }
            catch (Exception)
            {
            }
        }

        private void btnPublicar_Click(object sender, EventArgs e)
        {
            Publicar();   // Tras pulsar botón, se publica el mensaje
        }
    }
}  //  _____________________ ----   F  I  N   ---- _____________________
```