Universidad de Puerto Rico Recinto de Rio Piedras

Facultad de Administración de Empresas

Instituto de Estadísticas y Sistemas Computadorizados de Información

Desktop Application:

Wholesaler Warehouse Management System

Alexis Omar Rodríguez Rodríguez
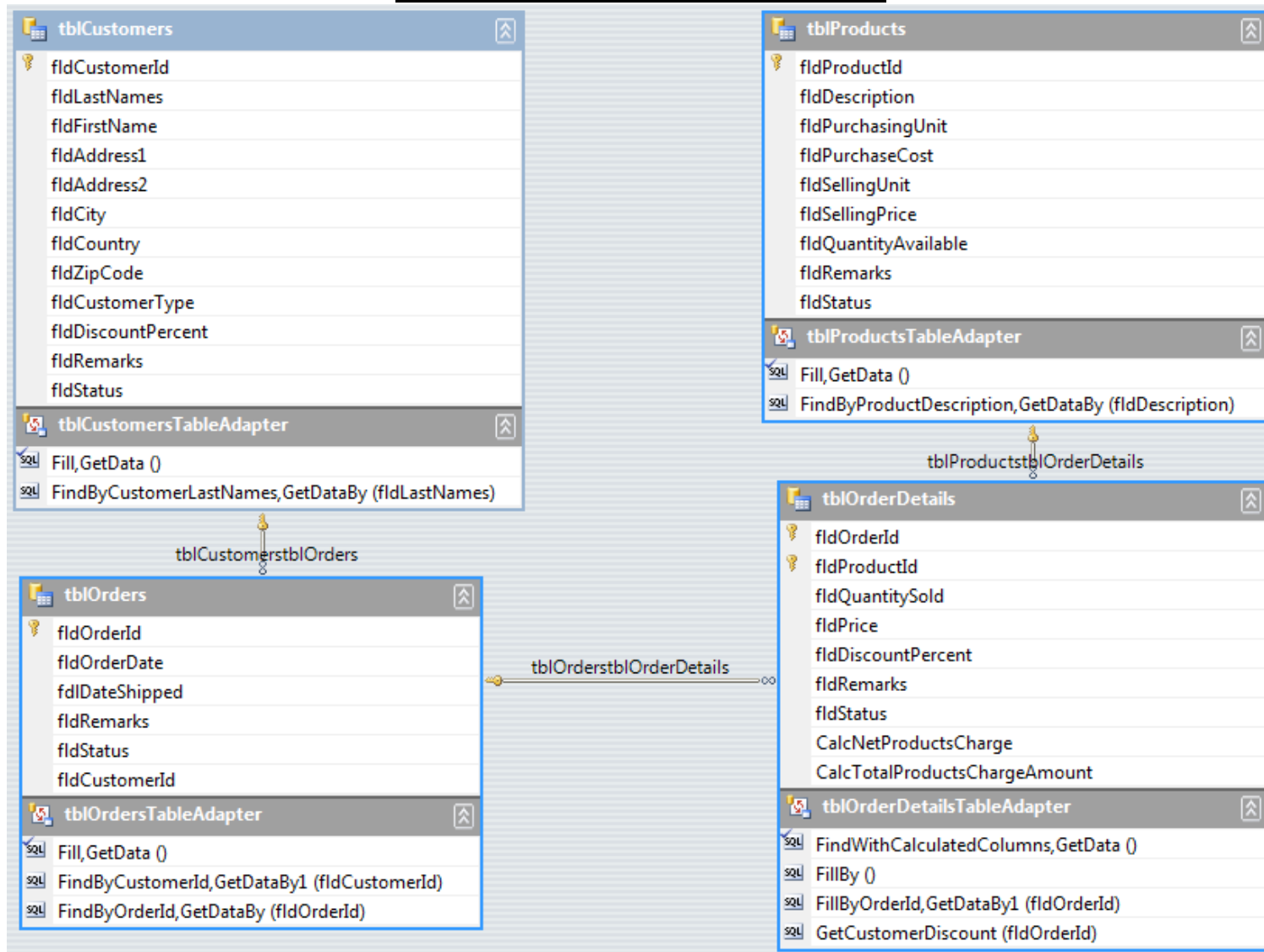
801-06-7192

5 de abril de 2012

SICI 4266 Sec. 0U1

Prof. Arnaldo I. Ramos

Functional requirements

A wholesaler warehouse manager wants a system to support its daily operations. Specifically the manager wants the system to do the following:
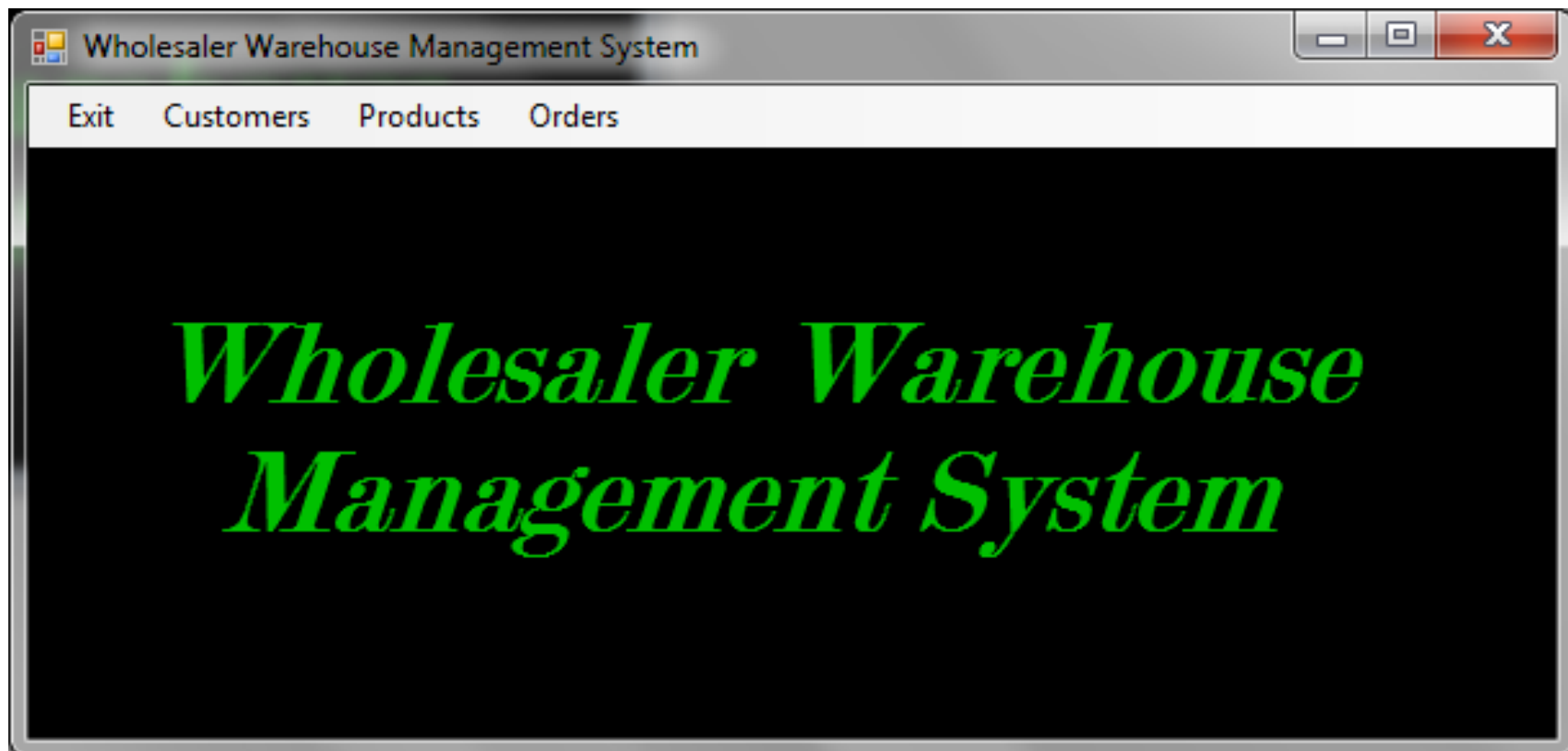
1. Allow salesmen to take customer orders, by selecting products already defined in a table.

2. Keep track of all the orders placed by customers.

3. Allow special discounts by customer and product.

4. Support the production of shipping documents (to pick up and deliver the merchandise), customer invoices, and payment receipts.

Entity Relationship Diagram (ERD)

tblCustomers
- fldCustomerId
- fldLastNames
- fldFirstName
- fldAddress1
- fldAddress2
- fldCity
- fldCountry
- fldZipCode
- fldCustomerType
- fldDiscountPercent
- fldRemarks
- fldStatus

tblCustomersTableAdapter
- Fill,GetData ()
- FindByCustomerLastNames,GetDataBy (fldLastNames)

tblProducts
- fldProductId
- fldDescription
- fldPurchasingUnit
- fldPurchaseCost
- fldSellingUnit
- fldSellingPrice
- fldQuantityAvailable
- fldRemarks
- fldStatus

tblProductsTableAdapter
- Fill,GetData ()
- FindByProductDescription,GetDataBy (fldDescription)

tblProductstblOrderDetails

tblCustomerstblOrders

tblOrders
- fldOrderId
- fldOrderDate
- fdlDateShipped
- fldRemarks
- fldStatus
- fldCustomerId

tblOrdersTableAdapter
- Fill,GetData ()
- FindByCustomerId,GetDataBy1 (fldCustomerId)
- FindByOrderId,GetDataBy (fldOrderId)

tblOrderstblOrderDetails

tblOrderDetails
- fldOrderId
- fldProductId
- fldQuantitySold
- fldPrice
- fldDiscountPercent
- fldRemarks
- fldStatus
- CalcNetProductsCharge
- CalcTotalProductsChargeAmount

tblOrderDetailsTableAdapter
- FindWithCalculatedColumns,GetData ()
- FillBy ()
- FillByOrderId,GetDataBy1 (fldOrderId)
- GetCustomerDiscount (fldOrderId)

MAIN MENU FORM

```vb
Module Module1

    Public row As Integer
    Public strCustomerId As String
    Public strOrderId As String

End Module

Public Class frmMain

    Private Sub Exit_Click(sender As System.Object, e As System.EventArgs) Handles mnuExit.Click

        'Salir de la aplicacion.
        Me.Close()

    End Sub

    Private Sub AllCustomers_Click(sender As System.Object, e As System.EventArgs) Handles mnuAllCustomers.Click

        'Abrir la forma CLIENTES.
        frmAllCustomers.ShowDialog()

    End Sub

    Private Sub AllProducts_Click(sender As System.Object, e As System.EventArgs) Handles mnuAllProducts.Click

        'Abrir la forma PRODUCTOS.
        frmAllProducts.ShowDialog()

    End Sub

    Private Sub OrderPerCustomer_Click(sender As System.Object, e As System.EventArgs) Handles mnuOrderPerCustomer.Click

        'Abrir la forma ORDENES.
        frmCustomerOrders.ShowDialog()

    End Sub

    Private Sub OrderDetails_Click(sender As System.Object, e As System.EventArgs) Handles mnuOrderDetails.Click
```

```
        'Abrir la forma DETALLES DE LA ORDEN.
        frmOrderDetails.ShowDialog()

    End Sub

End Class
```

CUSTOMERS FORM

Query -Fill-
SELECT fldCustomerId, fldLastNames, fldFirstName, fldAddress1, fldAddress2, fldCity, fldCountry,
fldZipCode, fldCustomerType, fldDiscountPercent, fldRemarks, fldStatus
FROM tblCustomers

Query -FindByLastNames-
SELECT fldAddress1, fldAddress2, fldCity, fldCountry, fldCustomerId, fldCustomerType,
fldDiscountPercent, fldFirstName, fldLastNames, fldRemarks, fldStatus, fldZipCode
FROM tblCustomers
WHERE (fldLastNames LIKE ? + '%')
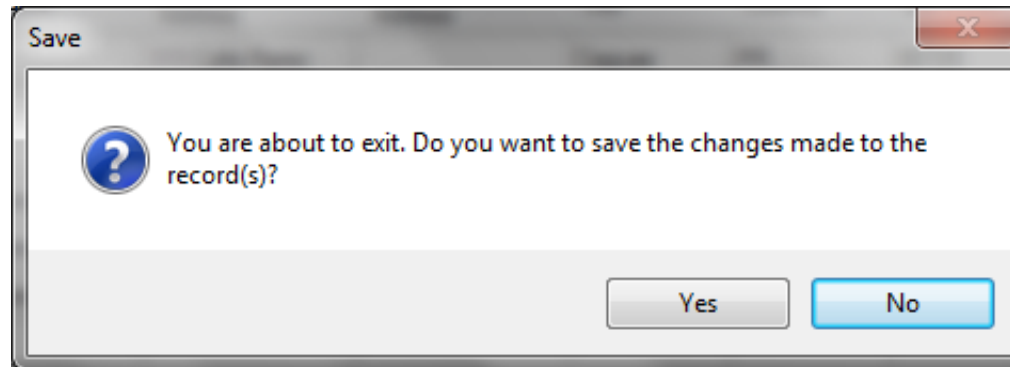

```vb
Public Class frmAllCustomers

    Private Sub Return_Click(sender As System.Object, e As System.EventArgs) Handles mnuReturn.Click

        ' Cuando se ejecuta "Me.Close()" se hace referencia al evento "FormClosing" para pedir almacenar si es necesario y
cerrar la forma.
        Me.Close()

    End Sub
```

CUSTOMERS SAVE MESSAGEBOXDIALOG


```vbnet
Private Sub FormClosing_FormClosing(sender As Object, e As System.Windows.Forms.FormClosingEventArgs) Handles Me.FormClosing

        'Cotejar cambios en el DataSet de clientes, grabar si existen cambios, y salir de la forma. De existir un error al
    almacenar se despliega un mensaje de error.
        Try
            If CustomersProductsOrdersDataSet.HasChanges() Then
                If MessageBox.Show("You are about to exit. Do you want to save the changes made to the record(s)?", "Save",
                                   MessageBoxButtons.YesNo,
                                   MessageBoxIcon.Question,
                                   MessageBoxDefaultButton.Button2) = Windows.Forms.DialogResult.Yes Then
                    TblCustomersTableAdapter.Update(Me.CustomersProductsOrdersDataSet.tblCustomers)
                End If
            End If
        Catch ex As Exception
            MessageBox.Show("An error ocurred while trying to save. Make sure your entering all the required filds before
    attempting to save.", "Error found")
        End Try

    End Sub

    Private Sub Load_Load(sender As System.Object, e As System.EventArgs) Handles MyBase.Load

        'Cargar la forma con todos los records de clientes.
```

```vbnet
        Me.TblCustomersTableAdapter.Fill(Me.CustomersProductsOrdersDataSet.tblCustomers)

        'Llamar al evento "CalculateTotalCustomers()" y dar el foco a txtLastNames.
        CalculateTotalCustomers()
        txtLastNames.Focus()

    End Sub

    Private Sub CalculateTotalCustomers()

        'Calcular total de clientes y desplegar resultado en un label.
        Dim row As Integer
        row = dgvAllCustomers.RowCount - 1
        lblTotalCustomers.Text = row.ToString

    End Sub

    Private Sub Save_Click(sender As System.Object, e As System.EventArgs) Handles mnuSave.Click

        'Almacenar aun sin haber perdido el foco, llenar nuevamente la tabla de Customers, recalcular totales y dar el foco
a "txtLastNames".
        'De existir un error al almacenar o cargar se despliega un mensaje de error.
        Try
            dgvAllCustomers.EndEdit()
            TblCustomersTableAdapter.Update(CustomersProductsOrdersDataSet.tblCustomers)
            Me.TblCustomersTableAdapter.Fill(Me.CustomersProductsOrdersDataSet.tblCustomers)
            CalculateTotalCustomers()
            txtLastNames.Focus()
        Catch ex As Exception
            MessageBox.Show("An error ocurred while trying to save. Make sure your entering all the required filds before
attempting to save.", "Error found")
        End Try

    End Sub
```
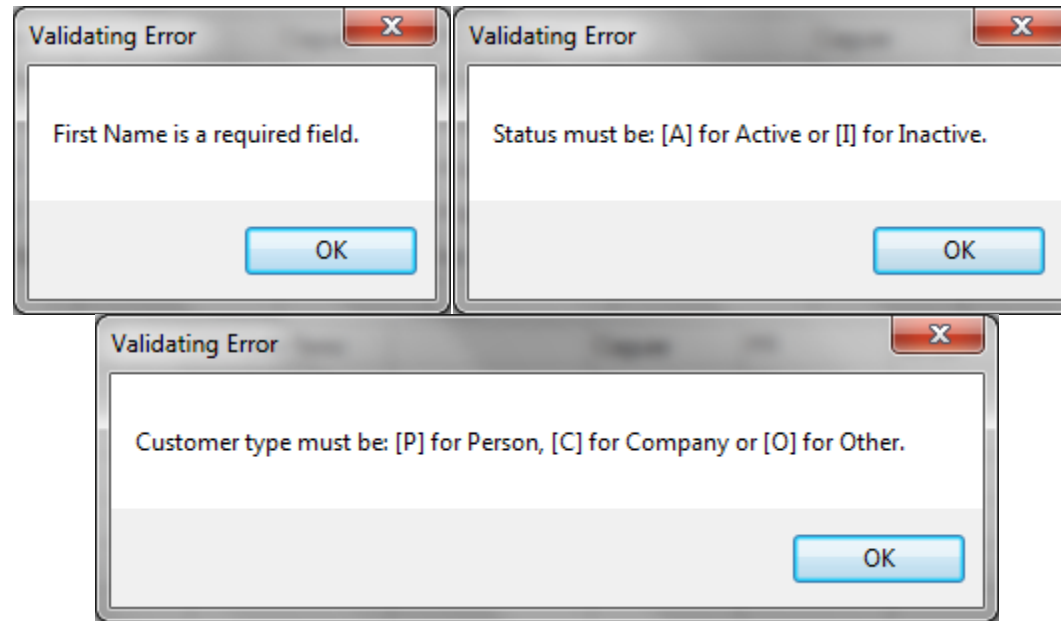
CUSTOMERS DATAGRIDVIEW VALIDATNG MESSAGEBOXDIALOGS

Validating Error

First Name is a required field.

[OK]

Validating Error

Status must be: [A] for Active or [I] for Inactive.

[OK]

Validating Error

Customer type must be: [P] for Person, [C] for Company or [O] for Other.

[OK]

```vbnet
    Private Sub CellValidating_CellValidating(sender As Object, e As
System.Windows.Forms.DataGridViewCellValidatingEventArgs) Handles dgvAllCustomers.CellValidating

        'Validar los datos entrados en los campos del DataGridView.
        Select Case e.ColumnIndex

            Case Is = 0
                If String.IsNullOrEmpty(e.FormattedValue.ToString()) Then
                    MessageBox.Show("Customer Id is a required field.", "Validating Error")
                    e.Cancel = True
                End If

            Case Is = 1
                If String.IsNullOrEmpty(e.FormattedValue.ToString()) Then
                    MessageBox.Show("Last Names is a required field.", "Validating Error")
```

```vb
                e.Cancel = True
            End If

        Case Is = 2
            If String.IsNullOrEmpty(e.FormattedValue.ToString()) Then
                MessageBox.Show("First Name is a required field.", "Validating Error")
                e.Cancel = True
            End If

        Case Is = 3
            If String.IsNullOrEmpty(e.FormattedValue.ToString()) Then
                MessageBox.Show("Postal Address is a required field.", "Validating Error")
                e.Cancel = True
            End If

        Case Is = 5
            If String.IsNullOrEmpty(e.FormattedValue.ToString()) Then
                MessageBox.Show("City is a required field.", "Validating Error")
                e.Cancel = True
            End If

        Case Is = 6
            If String.IsNullOrEmpty(e.FormattedValue.ToString()) Then
                MessageBox.Show("Country is a required field.", "Validating Error")
                e.Cancel = True
            End If

        Case Is = 7
            If String.IsNullOrEmpty(e.FormattedValue.ToString()) Then
                MessageBox.Show("Zip code is a required field.", "Validating Error")
                e.Cancel = True
            End If

        Case Is = 8
            If String.IsNullOrEmpty(e.FormattedValue.ToString()) Then
                MessageBox.Show("Customer type must be: [P] for Person, [C] for Company or [O] for Other.", "Validating
Error")
                e.Cancel = True
            End If
```

```vb
            Case Is = 11
                If String.IsNullOrEmpty(e.FormattedValue.ToString()) Then
                    MessageBox.Show("Status must be: [A] for Active or [I] for Inactive.", "Validating Error")
                    e.Cancel = True
                End If

        End Select

End Sub
```

CUSTOMERS DATAGRIDVIEW DEFAULT VALUES NEDDED


```vbnet
    Private Sub DefaultsValues_DefaultValuesNeeded(sender As Object, e As System.Windows.Forms.DataGridViewRowEventArgs)
Handles dgvAllCustomers.DefaultValuesNeeded

        'Asignar valores predetermiandos del DataGridView.
        With e.Row
            .Cells("fldCityDataGridViewTextBoxColumn").Value = "Guaynabo"
            .Cells("fldCountryDataGridViewTextBoxColumn").Value = "Puerto Rico"
            .Cells("fldZipCodeDataGridViewTextBoxColumn").Value = "00926"
            .Cells("fldCustomerTypeDataGridViewTextBoxColumn").Value = "P"
            .Cells("fldDiscountPercentDataGridViewTextBoxColumn").Value = "0"
            .Cells("fldStatusDataGridViewTextBoxColumn").Value = "A"
        End With
    End Sub
```

Customers

Save   Customer Order   Return

Search customer

Last Names: p| [Find Customer]

Customers Information

	Customer Id	Last Names	First Name	Postal Address	Physical Address	City	Country	Zip Code	Custome Type	Discount Percent	Remarks	Status
▶	C111	Perez Perez	Juan	111 Calle Perez		Caguas	PR	00725	P	10	Es mi amigo.	A
	C222	Perez Perez	Maria	222 Avenida Perez		Ponce	PR	00832	P	0		A
*									%			

Total customers: 2

*Customer type = P (Person), C (Company) u O (Other)   *Status = A (Active) or I (Inactive)

```
    Private Sub FindByCustomerLastNames_Click(sender As System.Object, e As System.EventArgs) Handles btnFind.Click

        'Buscar por apellidos, recalcular totales y dar el foco a txtLastNames.
        Me.TblCustomersTableAdapter.FindByCustomerLastNames(Me.CustomersProductsOrdersDataSet.tblCustomers,
txtLastNames.Text)
        CalculateTotalCustomers()
        txtLastNames.Focus()

    End Sub
```

CUSTOMERS –GO TO CUSTOMER ORDERS-


```vbnet
Private Sub GoToCustomerOrders_Click(sender As System.Object, e As System.EventArgs) Handles mnuCustomerOrders.Click

        'Almacenar el valor de la primera celda del record selecionado en la variables publica strCustomerId declarada en el
"Modulo1" y abrir la forma frmCustomerOrders.
        'Abrir la forma frmCustomerOrders de todas maneras si no se ha selecionado un record.
        If dgvAllCustomers.Rows.Count > 1 Then
            strCustomerId = dgvAllCustomers.CurrentRow.Cells(0).Value.ToString()
            Dim currentForm As New frmCustomerOrders
            currentForm.ShowDialog()
        Else
            frmCustomerOrders.ShowDialog()
        End If
    End Sub
```

```vbnet
    Private Sub dgvAllCustomers_UserAddedRow(sender As Object, e As System.Windows.Forms.DataGridViewRowEventArgs) Handles
dgvAllCustomers.UserAddedRow

        'Calcular justo cuando se va a agregar un nuevo record.
        CalculateTotalCustomers()

    End Sub
```

CUSTOMERS DELETE MESSAGEBOXDIALOG


```vbnet
    Private Sub UserDeletedRow_UserDeletedRow(sender As Object, e As System.Windows.Forms.DataGridViewRowEventArgs) Handles
dgvAllCustomers.UserDeletedRow

        'Actualizar base de datos y recalcular luego de haber eliminado un record. De existir un error al almacenar o cargar
se despliega un mensaje de error.
        Try
            Me.TblCustomersTableAdapter.Update(Me.CustomersProductsOrdersDataSet.tblCustomers)
            CalculateTotalCustomers()
        Catch ex As Exception
            MessageBox.Show("An error ocurred while trying to save.", "Error found")
        End Try
```

```vbnet
    End Sub

    Private Sub UserDeletingRow_UserDeletingRow(sender As Object, e As System.Windows.Forms.DataGridViewRowCancelEventArgs)
Handles dgvAllCustomers.UserDeletingRow

        'Pedir al usuario si desea borrar un record, y salir de la forma. Si dice que "NO" se cancela el evento.
        Dim intSelectedRow As Integer
        intSelectedRow = dgvAllCustomers.CurrentRow.Index
            If MessageBox.Show("Are you sure you want to permanently delete customer " + CStr(dgvAllCustomers.Item(0,
intSelectedRow).Value) + "?", "Delete",
                                MessageBoxButtons.YesNo,
                                MessageBoxIcon.Warning,
                                MessageBoxDefaultButton.Button2) = Windows.Forms.DialogResult.No Then
                e.Cancel = True
            End If
    End Sub

End Class
```

PRODCUTS FORM

Products

Save   Return

Search Product

Description: [   ]   [Find]

Products List

	Product Id	Description	Purchasing Unit	Purchase Cost	Selling Unit	Selling Price	Quantity Avaliable	Remarks	Status
▶	P444	Lenovo	U	$ 9.99	U	$ 499.99	8	Quad-Core	A
	P555	CISCORouter	U	$ 20.00	U	$ 74.99	23		A
	p666	Presentation Board	U	$ 2.00	U	$ 16.79	13	Black	A
	P777	PaperMate Ballpoint P...	D	$ 1.00	U	$ 11.49	47		A
	p888	HP Pavillon	U	$ 199.99	U	$ 699.99	8		A
	p999	Epson 125, 3-pack	D	$ 499.99	U	$ 59.99	7	For Stylus printer	I
	P100	Binders 1"	D	$ 9.99	D	$ 29.99	6		A
	P200	At-A-Glance 2012	D	$ 120.00	U	$ 17.89	10	Clearance	I
	P300	File Cabinet	U	$ 39.99	U	$ 129.99	4		A

Total Products: **15**

 * Purchasing Unit = U(Uno), C(Caja), D(Docena), o R(Resma)   * Selling Unit = U(Uno), C(Caja), D(Docena), o R(Resma)

Query -Fill-
SELECT fldProductId, fldDescription, fldPurchasingUnit, fldPurchaseCost, fldSellingUnit,
fldSellingPrice, fldQuantityAvailable, fldRemarks, fldStatus
FROM tblProducts

Query -FindByProductDescription-
SELECT fldProductId, fldDescription, fldPurchasingUnit, fldPurchaseCost, fldSellingUnit,
fldSellingPrice, fldQuantityAvailable, fldRemarks, fldStatus
FROM tblProducts
WHERE fldDescription LIKE ? + '%'

```vbnet
Option Strict On

Public Class frmAllProducts

    Private Sub Return_Click(sender As System.Object, e As System.EventArgs) Handles mnuReturn.Click

        'Cuando se ejecuta "Me.Close()" se hace referencia al evento "FormClosing" para pedir almacenar si es necesario y
cerrar la forma.
        Me.Close()

    End Sub
```
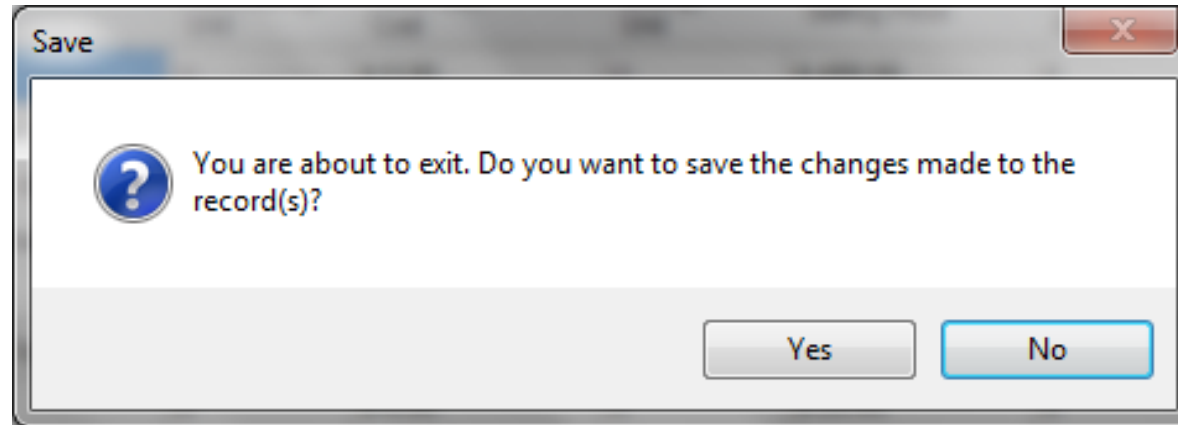
```vb
    Private Sub FormClosing_FormClosing(sender As Object, e As System.Windows.Forms.FormClosingEventArgs) Handles
Me.FormClosing

        'Cotejar cambios en el DataSet de clientes, grabar si existen cambios, y salir de la forma. De existir un error al
almacenar se despliega un mensaje de error.
        Try
            If CustomersProductsOrdersDataSet.HasChanges() Then
                If MessageBox.Show("You are about to exit. Do you want to save the changes made to the record(s)?", "Save",
                                MessageBoxButtons.YesNo,
                                MessageBoxIcon.Question,
                                MessageBoxDefaultButton.Button2) = Windows.Forms.DialogResult.Yes Then
                    TblProductsTableAdapter.Update(Me.CustomersProductsOrdersDataSet.tblProducts)
                End If

            End If
        Catch ex As Exception
            MessageBox.Show("An error ocurred while trying to save. Make sure your entering all the required filds before
attempting to save.", "Error found")
        End Try

    End Sub

    Private Sub Load_Load(sender As System.Object, e As System.EventArgs) Handles MyBase.Load
```

```vbnet
        'Cargar la forma con todos los records de productos.
        Me.TblProductsTableAdapter.Fill(Me.CustomersProductsOrdersDataSet.tblProducts)

        'Llamar al evento "CalculateTotalProducts()" y dar el foco a txtLastNames.
        CalculateTotalProducts()
        txtProductDescription.Focus()

    End Sub

    Private Sub CalculateTotalProducts()

        'Calcular total de productos y desplegar resultado en un label.
        Dim row As Integer
        row = dgvProducts.RowCount - 1
        lblTotalProducts.Text = row.ToString

    End Sub

    Private Sub Save_Click(sender As System.Object, e As System.EventArgs) Handles mnuSave.Click

        'Almacenar aun sin haber perdido el foco, llenar nuevamente la tabla de productos, recalcular totales y dar el foco
a "txtProductDescription".
        'De existir un error al almacenar o cargar se despliega un mensaje de error.
        Try
            dgvProducts.EndEdit()
            TblProductsTableAdapter.Update(CustomersProductsOrdersDataSet.tblProducts)
            Me.TblProductsTableAdapter.Fill(Me.CustomersProductsOrdersDataSet.tblProducts)
            CalculateTotalProducts()
            txtProductDescription.Focus()
        Catch ex As Exception
            MessageBox.Show("An error ocurred while trying to save. Make sure your entering all the required filds before
attempting to save.", "Error found")
        End Try

    End Sub
```
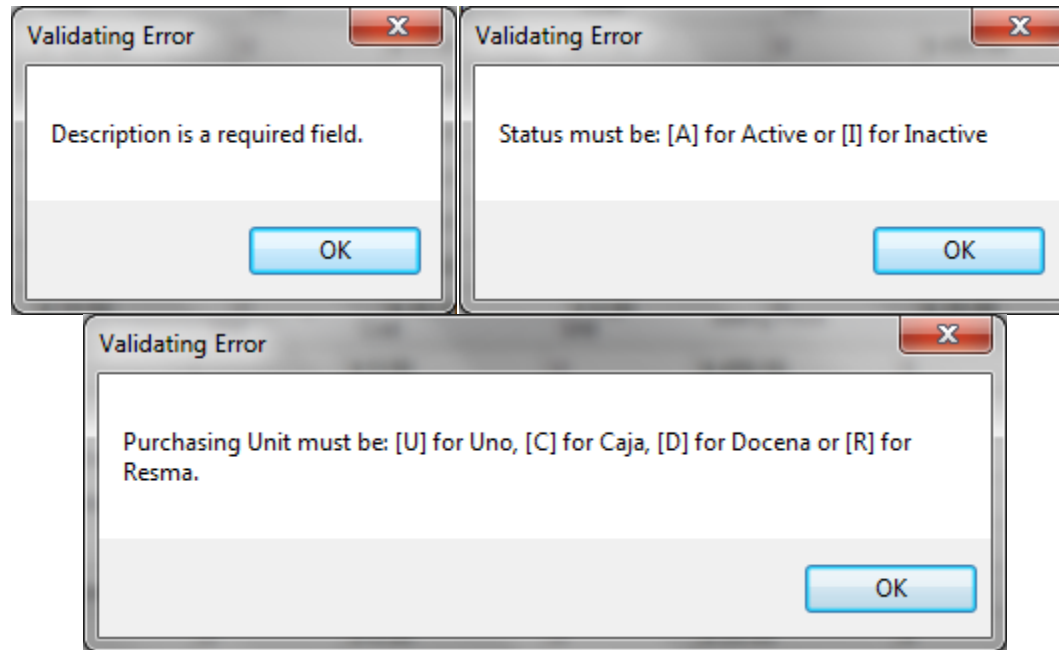
PRODUCTS DATAGRIDVIEW CELL VALIDATING


```vbnet
    Private Sub CellValidating_CellValidating(sender As Object, e As
System.Windows.Forms.DataGridViewCellValidatingEventArgs) Handles dgvProducts.CellValidating

        'Validar los datos entrados en los campos del DataGridView.
        Select Case e.ColumnIndex

            Case Is = 0
                If String.IsNullOrEmpty(e.FormattedValue.ToString()) Then
                    MessageBox.Show("Product Id is a required field.", "Validating Error")
                    e.Cancel = True
                End If

            Case Is = 1
                If String.IsNullOrEmpty(e.FormattedValue.ToString()) Then
```

```vb
                MessageBox.Show("Description is a required field.", "Validating Error")
                e.Cancel = True
            End If

        Case Is = 2
            If String.IsNullOrEmpty(e.FormattedValue.ToString()) Then
                MessageBox.Show("Purchasing Unit must be: [U] for Uno, [C] for Caja, [D] for Docena or [R] for Resma.",
"Validating Error")
                e.Cancel = True
            End If

        Case Is = 3
            If String.IsNullOrEmpty(e.FormattedValue.ToString()) Then
                MessageBox.Show("Purchasing cost is a required field.", "Validating Error")
                e.Cancel = True
            End If

        Case Is = 4
            If String.IsNullOrEmpty(e.FormattedValue.ToString()) Then
                MessageBox.Show("Selling Unit must be: [U] for Uno, [C] for Caja, [D] for Docena or [R] for Resma.",
"Validating Error")
                e.Cancel = True
            End If

        Case Is = 5
            If String.IsNullOrEmpty(e.FormattedValue.ToString()) Then
                MessageBox.Show("Selling price is a required field.", "Validating Error")
                e.Cancel = True
            End If

        Case Is = 6
            If String.IsNullOrEmpty(e.FormattedValue.ToString()) Then
                MessageBox.Show("Quantity available is a required field.", "Validating Error")
                e.Cancel = True
            End If

        Case Is = 8
            If String.IsNullOrEmpty(e.FormattedValue.ToString()) Then
                MessageBox.Show("Status must be: [A] for Active or [I] for Inactive", "Validating Error")
```

```vb
                e.Cancel = True
            End If

    End Select

End Sub
```

PRODUCTS DEFAULT VALUES NEDEED


```vb
    Private Sub DefaultValuesNeeded_DefaultValuesNeeded(sender As Object, e As
System.Windows.Forms.DataGridViewRowEventArgs) Handles dgvProducts.DefaultValuesNeeded

        'Asignar valores predetermiandos del DataGridView.
        With e.Row
            .Cells("fldPurchasingUnitDataGridViewTextBoxColumn3").Value = "U"
            .Cells("fldSellingUnitDataGridViewTextBoxColumn5").Value = "U"
            .Cells("fldSellingPriceDataGridViewTextBoxColumn6").Value = "0.00"
            .Cells("fldQuantityAvailableDataGridViewTextBoxColumn7").Value = "0"
            .Cells("fldStatusDataGridViewTextBoxColumn9").Value = "A"
        End With
    End Sub


    Private Sub Find_Click(sender As System.Object, e As System.EventArgs) Handles btnFind.Click
```

```vb
        'Buscar por descripcion, recalcular totales y dar el foco a "txtProductDescription".
        Me.TblProductsTableAdapter.FindByProductDescription(Me.CustomersProductsOrdersDataSet.tblProducts,
txtProductDescription.Text)
        CalculateTotalProducts()
        txtProductDescription.Focus()
    End Sub

    Private Sub dgvProducts_UserAddedRow(sender As Object, e As System.Windows.Forms.DataGridViewRowEventArgs) Handles
dgvProducts.UserAddedRow

        'Calcular justo cuando se va a agregar un nuevo record.
        CalculateTotalProducts()

    End Sub
```

PRODUCTS DELETE MESSAGEBOXDIALOG
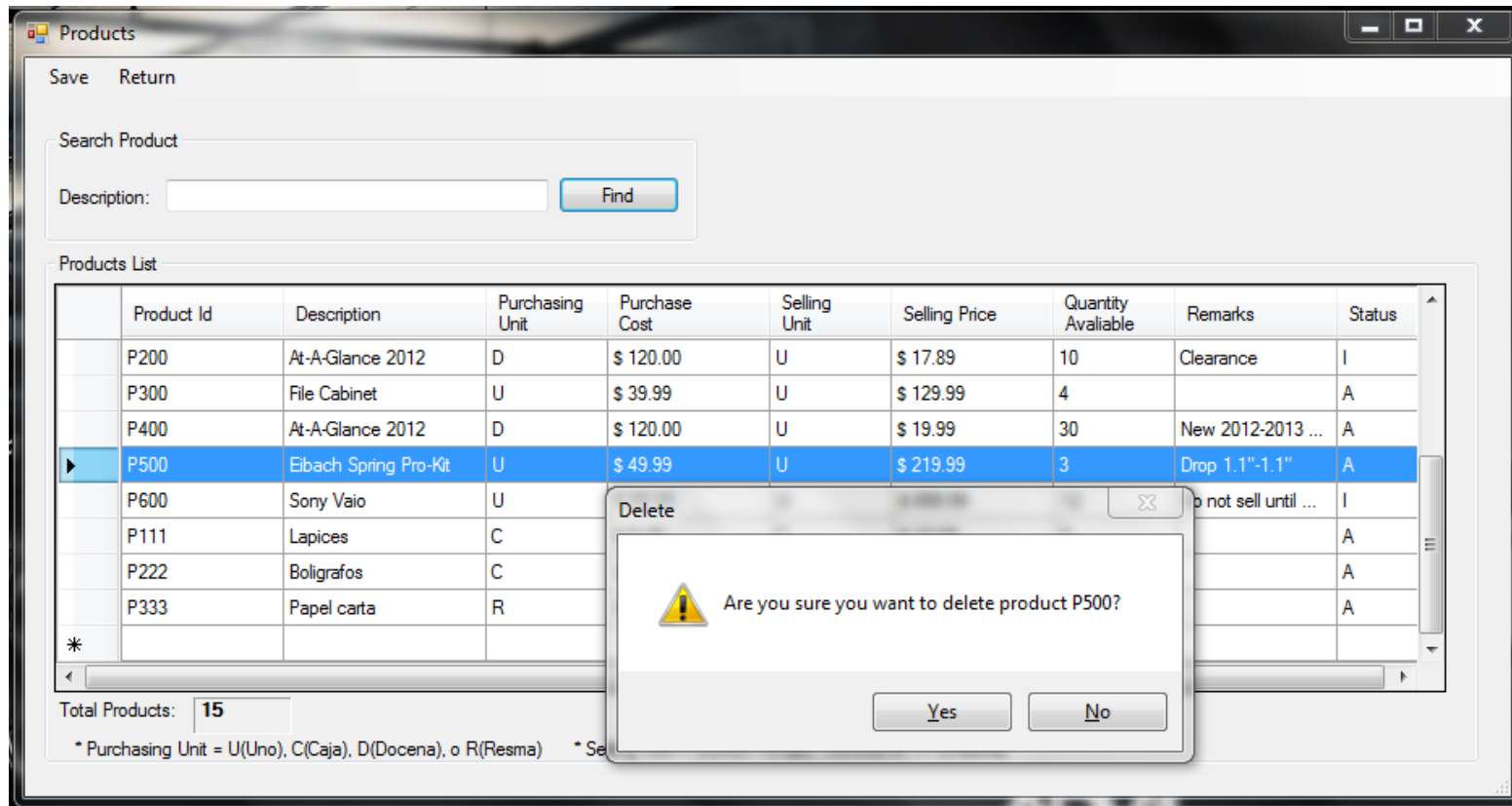

```vb
    Private Sub UserDeletedRow_UserDeletedRow(sender As Object, e As System.Windows.Forms.DataGridViewRowEventArgs) Handles
dgvProducts.UserDeletedRow

        'Actualizar base de datos y recalcular luego de haber eliminado un record. De existir un error al almacenar o cargar
se despliega un mensaje de error.
        Try
            Me.TblProductsTableAdapter.Update(Me.CustomersProductsOrdersDataSet.tblProducts)
            CalculateTotalProducts()
```

```vbnet
        Catch ex As Exception
            MessageBox.Show("An error ocurred while trying to save.", "Error found")
        End Try

    End Sub

    Private Sub UserDeletingRow_UserDeletingRow(sender As Object, e As System.Windows.Forms.DataGridViewRowCancelEventArgs)
Handles dgvProducts.UserDeletingRow


        'Pedir eliminar un record, y salir de la forma. Si indica "NO" se cancela el evento.
        Dim intSelectedRow As Integer
        intSelectedRow = dgvProducts.CurrentRow.Index
        If MessageBox.Show("Are you sure you want to delete product " + CStr(dgvProducts.Item(0, intSelectedRow).Value) +
"?", "Delete",
                            MessageBoxButtons.YesNo,
                            MessageBoxIcon.Warning,
                            MessageBoxDefaultButton.Button2) = Windows.Forms.DialogResult.No Then
            e.Cancel = True
        End If
    End Sub

End Class
```

CUSTOMER ORDERS FORM

Customer Orders

Save   Order Details   Return

Search Order

Customer Id: C111

Customer order(s)

	Order Id	Date Shipped	Order Date	Remarks	Status
▶	O111		01/01/2001	Store Pickup	A
	O333		10/10/2011	Delivery	A
	O555		10/10/2011	Near Plaza Guay.	A
✱					

Total Orders: 3

*Status = A (Active) or I (Inactive)

Query –Fill,GetData()-
SELECT fldOrderId, fldOrderDate, fdlDateShipped, fldRemarks, fldStatus, fldCustomerId
FROM tblOrders

Query -FindByCustomerId-
SELECT fldOrderId, fldOrderDate, fdlDateShipped, fldRemarks, fldStatus, fldCustomerId
FROM tblOrders
WHERE fldCustomerId LIKE ?

```vbnet
Public Class frmCustomerOrders

    Private Sub Return_Click(sender As System.Object, e As System.EventArgs) Handles mnuReturn.Click

        'Cuando se ejecuta "Me.Close()" se hace referencia al evento "FormClosing" para pedir almacenar si es necesario y
cerrar la forma.
        Me.Close()

    End Sub
```



```vbnet
    Private Sub FormClosing_FormClosing(sender As Object, e As System.Windows.Forms.FormClosingEventArgs) Handles
Me.FormClosing
```
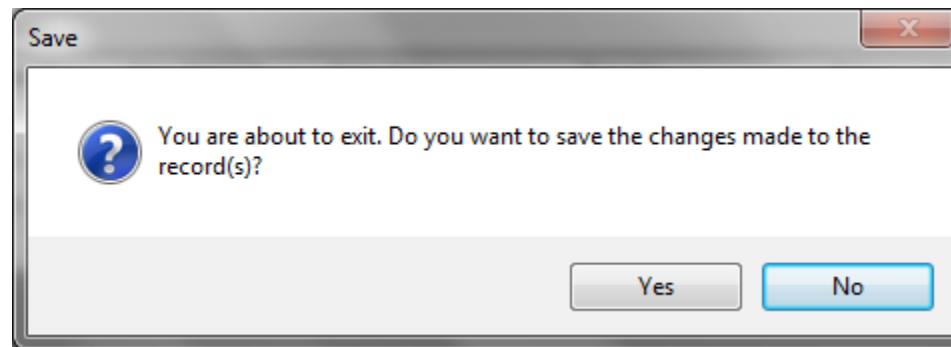
```vbnet
        'Cotejar cambios en el DataSet de Ordenes, almacenar si existen cambios, y salir de la forma. De existir un error al
    almacenar se despliega un mensaje de error.
        Try
            If CustomersProductsOrdersDataSet.HasChanges() Then
                If MessageBox.Show("You are about to exit. Do you want to save the changes made to the record(s)?", "Save",
                            MessageBoxButtons.YesNo,
                            MessageBoxIcon.Question,
                            MessageBoxDefaultButton.Button2) = Windows.Forms.DialogResult.Yes Then
                    TblOrdersTableAdapter.Update(Me.CustomersProductsOrdersDataSet.tblOrders)
                End If

            End If
        Catch ex As Exception
            MessageBox.Show("An error ocurred while trying to save. Make sure your entering all the required filds before
    attempting to save.", "Error found")
        End Try

    End Sub

    Private Sub Load_Load(sender As System.Object, e As System.EventArgs) Handles MyBase.Load

        'Cargar la forma con todos los records de Ordenes y llamar el evento "CalculateTotalCustomerOrders".
        Me.TblCustomersTableAdapter.Fill(Me.CustomersProductsOrdersDataSet.tblCustomers)
        CalculateTotalCustomerOrders()

        ' El valor de la variable strcustomerId declarada en el "Module1" se seleciona en el combobox de la forma
    CustomerOrders si el usuario viene
        'desde la forma frmCustomers.
        cboCustomerId.SelectedValue = strCustomerId.ToString()
        cboCustomerId.Focus()

    End Sub

    Private Sub CalculateTotalCustomerOrders()

        'Calcular total de ordenes y desplegar resultado en un label.
        Dim row As Integer
        row = dgvCustomerOrders.RowCount - 1
```

```vb
            lblTotalCustomerOrders.Text = row.ToString

    End Sub

    Private Sub Save_Click(sender As System.Object, e As System.EventArgs) Handles mnuSave.Click

        'Almacenar aun sin haber perdido el foco, llenar nuevamente la tabla de ordenes, recalcular totales y dar el foco a
"cboCustomerId".
        'De existir un error al almacenar o cargar se despliega un mensaje de error.
        Try
            dgvCustomerOrders.EndEdit()
            TblOrdersTableAdapter.Update(CustomersProductsOrdersDataSet.tblOrders)
            Me.TblOrdersTableAdapter.Fill(Me.CustomersProductsOrdersDataSet.tblOrders)
            CalculateTotalCustomerOrders()
            cboCustomerId.Focus()
        Catch ex As Exception
            MessageBox.Show("An error ocurred while trying to save. Make sure your entering all the required filds before
attempting to save.", "Error found")
        End Try

    End Sub
```
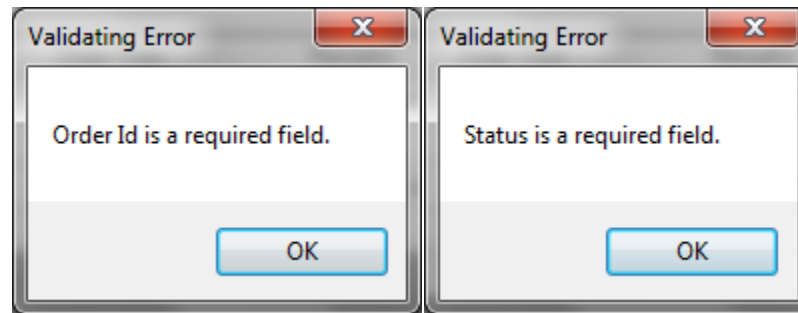
CUSTOMER ORDERS CELL VALIDATING


```vb
    Private Sub CellValidating_CellValidating(sender As Object, e As
System.Windows.Forms.DataGridViewCellValidatingEventArgs) Handles dgvCustomerOrders.CellValidating
```

```vbnet
        'Validar los datos entrados en los campos del DataGridView.
        Select Case e.ColumnIndex

            Case Is = 0
                If String.IsNullOrEmpty(e.FormattedValue.ToString()) Then
                    MessageBox.Show("Order Id is a required field.", "Validating Error")
                    e.Cancel = True
                End If

            Case Is = 1
                If String.IsNullOrEmpty(e.FormattedValue.ToString()) Then
                    MessageBox.Show("Order Date is a required field.", "Validating Error")
                    e.Cancel = True
                End If

            Case Is = 4
                If String.IsNullOrEmpty(e.FormattedValue.ToString()) Then
                    MessageBox.Show("Status is a required field.", "Validating Error")
                    e.Cancel = True
                End If

        End Select

    End Sub
```

CUSTOMER ORDER DEFAULT VALUES NEEDED

Customer Orders

Save   Order Details   Return

Search Order

Customer Id: C111 ▾

Customer order(s)

	Order Id	Date Shipped	Order Date	Remarks	Status
	O111		01/01/2001	Store Pickup	A
	O333		10/10/2011	Delivery	A
	O555		10/10/2011	Near Plaza Guay.	A
▸*		05/04/2012	05/04/2012		A

Total Orders: 3

*Status = A (Active) or I (Inactive)

```vbnet
    Private Sub DefaultValuesNeeded_DefaultValuesNeeded(sender As Object, e As
System.Windows.Forms.DataGridViewRowEventArgs) Handles dgvCustomerOrders.DefaultValuesNeeded

        'Asignar valores predetermiandos del DataGridView.
        With e.Row
            .Cells("fldDateShippedDataGridViewTextBoxColumn").Value = Date.Today
            .Cells("fldOrderDateDataGridViewTextBoxColumn2").Value = Date.Today
            .Cells("fldStatusDataGridViewTextBoxColumn5").Value = "A"
        End With

    End Sub

    Private Sub cboOrderId_SelectedIndexChanged(sender As System.Object, e As System.EventArgs) Handles
cboCustomerId.SelectedIndexChanged

        'Buscar por orderId y recalcular totales.
        Me.TblOrdersTableAdapter.FindByCustomerId(CustomersProductsOrdersDataSet.tblOrders, cboCustomerId.SelectedValue)
        CalculateTotalCustomerOrders()

    End Sub


    Private Sub cboOrderId_SelectedIndexChanged(sender As System.Object, e As System.EventArgs) Handles
cboCustomerId.SelectedIndexChanged

        'Buscar por orderId y recalcular totales.
        Me.TblOrdersTableAdapter.FindByCustomerId(CustomersProductsOrdersDataSet.tblOrders, cboCustomerId.SelectedValue)
        CalculateTotalCustomerOrders()

    End Sub
```

CUSTOMER ORDERS –GO TO ORDER DETAILS-

Customer Orders

Save   Order Details   Return

Search Order

Customer Id: C777

Customer order(s)

	Order Id	Date Shipped	Order Date	Remarks	Status
▶	O888		27/03/2012	Orden especial	A
*					

Total Orders: 1

*Status = A (Active) or I (Inactive)

Order Details

Save   Customer Order   Return

Search Order Details

Order Id: O888

Order Details

	Order Id	Product Id	Quantity Sold	Price	Discount Percent	Remarks	Status	Products Charge Amount	Net Products Charge
▶	O888	P444	50	$ 10.00	10		A	$ 500.00	$ 450.00
	O888	P111	20	$ 50.00	100		A	$ 1,000.00	$ 0.00
	O888	P222	33	$ 19.00	10		A	$ 627.00	$ 564.30
	O888	P333	1	$ 100.00	5		A	$ 100.00	$ 95.00
*									

Total Products: 4

Total Products Charge Amount: $ 2,227.00

Total Net Charges: $ 1,109.30

Customer Discount: |

*Status = A (Active) or I (Inactive)

Grand Total Amount: |

```vb
    Private Sub GoToOrderDetails_Click(sender As System.Object, e As System.EventArgs) Handles mnuOrderDetails.Click

        'Almacenar el valor de la primera celda del record selecionado en la variables publica strOrderId declarada en el
"Modulo1" y abrir la forma frmOrderDetails.
        'Abrir la forma frmCustomerOrders de todas maneras si no se ha selecionado un record.
        If dgvCustomerOrders.Rows.Count > 1 Then
            strOrderId = dgvCustomerOrders.CurrentRow.Cells(0).Value.ToString()
            Dim currentForm As New frmOrderDetails
            currentForm.ShowDialog()
        Else
            frmOrderDetails.ShowDialog()
        End If

    End Sub




    Private Sub dgvCustomerOrders_UserAddedRow(sender As Object, e As System.Windows.Forms.DataGridViewRowEventArgs) Handles
dgvCustomerOrders.UserAddedRow

        'Calcular justo cuando se va a agregar un nuevo record.
        CalculateTotalCustomerOrders()

    End Sub
```

CUSTOMER ORDERS USER DELETING ROW

Customer Orders

Save   Order Details   Return

Search Order

Customer Id: C111 ▼

Customer order(s)

	Order Id	Date Shipped	Order Date	Remarks	Status
▶	O111		01/01/2001	Store Pickup	A
	O333		10/10/2011	Delivery	A
	O555		10/10/2011	Near Plaza Guay.	A
*					

Total Orders: 3

*Status = A (Active) or I (Inactive)

Delete

⚠ Are you sure you want to delete customer order O111?

Yes   No

```vb
    Private Sub UserDeletedRow_UserDeletedRow(sender As Object, e As System.Windows.Forms.DataGridViewRowEventArgs) Handles
dgvCustomerOrders.UserDeletedRow

        'Actualizar base de datos y recalcular luego de haber eliminado un record. De existir un error al almacenar o cargar
se despliega un mensaje de error.
        Try
            Me.TblOrdersTableAdapter.Update(Me.CustomersProductsOrdersDataSet.tblOrders)
            CalculateTotalCustomerOrders()
        Catch ex As Exception
            MessageBox.Show("An error ocurred while trying to save.", "Error found")
        End Try

    End Sub

    Private Sub UserDeletingRow_UserDeletingRow(sender As Object, e As System.Windows.Forms.DataGridViewRowCancelEventArgs)
Handles dgvCustomerOrders.UserDeletingRow

        'Pedir eliminar un record, y salir de la forma. Si indica "NO" se cancela el evento.
        Dim intSelectedRow As Integer
        intSelectedRow = dgvCustomerOrders.CurrentRow.Index

        If MessageBox.Show("Are you sure you want to delete customer order " + CStr(dgvCustomerOrders.Item(0,
intSelectedRow).Value) + "?", "Delete",
                           MessageBoxButtons.YesNo,
                           MessageBoxIcon.Warning,
                           MessageBoxDefaultButton.Button2) = Windows.Forms.DialogResult.No Then
            e.Cancel = True
        End If
    End Sub

End Class
```

ORDER DETAILS FORM

Order Details

Save   Customer Order   Return

Search Order Details

Order Id: O111 ▼

Order Details

	Order Id	Product Id	Quantity Sold	Price	Discount Percent	Remarks	Status	Products Charge Amount	Net Products Charge
▶	O111	P111	10	$ 100.00	10	Liquidación	A	$ 1,000.00	$ 900.00
	O111	P222	5	$ 200.00	20	Productos dañados	A	$ 1,000.00	$ 800.00
	O111	P333	1	$ 100.00	10		A	$ 100.00	$ 90.00
	O111	P500	1	$ 20.00	50		A	$ 20.00	$ 10.00
	O111	P444	3	$ 10,000...	0	mbbmnn	A	$ 30,000.00	$ 30,000.00
*									

Total Products: 5

Total Products Charge Amount: $ 32,120.00

Total Net Charges: $ 31,800.00

Customer Discount:

*Status = A (Active) or I (Inactive)

Grand Total Amount:

```sql
                        Query -FindWithCalculatedColumn-
SELECT fldOrderId, fldProductId, fldQuantitySold, fldPrice, (fldQuantitySold * fldPrice) As
        CalcTotalProductsChargeAmount, fldDiscountPercent, fldRemarks, fldStatus,
        (fldPrice * (1-fldDiscountPercent/100)) * fldQuantitySold As CalcNetProductsCharge
                            FROM tblOrderDetails
            WHERE tblOrderDetails.fldOrderId = tblOrderDetails.fldOrderId


                                Query -FillBy-
SELECT fldPrice * (1 - fldDiscountPercent / 100) * fldQuantitySold AS CalcNetProductsCharge,
fldQuantitySold * fldPrice AS CalcTotalProductsChargeAmount, fldDiscountPercent, fldOrderId,
            fldPrice, fldProductId,  fldQuantitySold, fldRemarks, fldStatus
                            FROM tblOrderDetails
                            WHERE fldOrderId LIKE ?


                            Query -FillByOrderId-
SELECT fldOrderId, fldProductId, fldQuantitySold, fldPrice, (fldQuantitySold * fldPrice) As
        CalcTotalProductsChargeAmount, fldDiscountPercent, fldRemarks, fldStatus,
        (fldPrice * (1-fldDiscountPercent/100)) * fldQuantitySold As CalcNetProductsCharge
                            FROM tblOrderDetails
                            WHERE fldOrderId LIKE ?


                        Query -GetCustomerDiscount-
                SELECT       tblCustomers.fldDiscountPercent
                FROM           ((tblOrderDetails INNER JOIN
                tblOrders ON tblOrderDetails.fldOrderId = tblOrders.fldOrderId) INNER
                                    JOIN
                tblCustomers ON tblOrders.fldCustomerId = tblCustomers.fldCustomerId)
                WHERE        (tblOrderDetails.fldOrderId LIKE ? + '%')
```
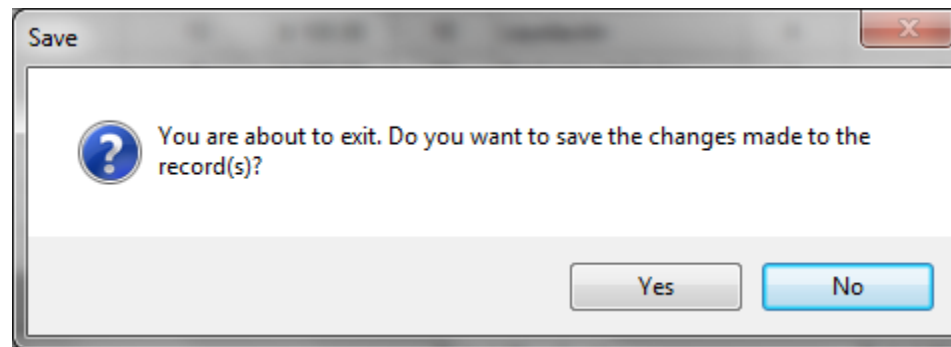
```vbnet
Public Class frmOrderDetails

    Private Sub Return_Click(sender As System.Object, e As System.EventArgs) Handles mnuReturn.Click

        'Cuando se ejecuta "Me.Close()" se hace referencia al evento "FormClosing" para pedir almacenar si es necesario y
cerrar la forma.
        Me.Close()

    End Sub
```



```vbnet
    Private Sub FormClosing_FormClosing(sender As Object, e As System.Windows.Forms.FormClosingEventArgs) Handles
Me.FormClosing

  'Cotejar cambios en el DataSet de Order Details, grabar si existen cambios, y salir de la forma. De existir un error al
almacenar se despliega un mensaje de error.
Try
            If CustomersProductsOrdersDataSet.HasChanges() Then
                If MessageBox.Show("You are about to exit. Do you want to save the changes made to the record(s)?", "Save",
                                 MessageBoxButtons.YesNo,
                                 MessageBoxIcon.Question,
                                 MessageBoxDefaultButton.Button2) = Windows.Forms.DialogResult.Yes Then
                Me.TblOrderDetailsTableAdapter.Update(Me.CustomersProductsOrdersDataSet.tblOrderDetails)
                End If
            End If
        Catch ex As Exception
```

```vbnet
            MessageBox.Show("An error ocurred while trying to save. Make sure your entering all the required filds before
    attempting to save.", "Error found")
        End Try
    End Sub

    Private Sub Load_Load(sender As System.Object, e As System.EventArgs) Handles MyBase.Load

        'Cargar la forma con el valor selecionado en la variable strOrderId.
        Me.TblOrdersTableAdapter.Fill(Me.CustomersProductsOrdersDataSet.tblOrders)
        cboOrderId.SelectedValue = strOrderId.ToString()

    End Sub



    Private Sub CalculateTotalsOrderDetails()

        'Calcular el total de productos de una orden.
        Dim row As Integer
        row = dgvOrderDetails.RowCount - 1
        lblCustomerDiscount.Text = row.ToString

        'Calcular el cargo neto total.
        Dim TotalNetCharge As Double = 0
        For i = 0 To dgvOrderDetails.RowCount - 1
            TotalNetCharge += dgvOrderDetails.Rows(i).Cells("CalcNetProductsChargeDataGridViewTextBoxColumn8").Value()
        Next
        lblTotalNetCharge.Text = TotalNetCharge.ToString("c")

        'Calcular el precio total.
        Dim TotalProductsChargeAmount As Double = 0
        For i = 0 To dgvOrderDetails.RowCount - 1
            TotalProductsChargeAmount += (dgvOrderDetails.Rows(i).Cells("CalcTotalProductsChargeAmount").Value)
        Next
        lblTotalProductsChargeAmount.Text = TotalProductsChargeAmount.ToString("c")

    End Sub

    Private Sub Save_Click(sender As System.Object, e As System.EventArgs) Handles mnuSave.Click
```
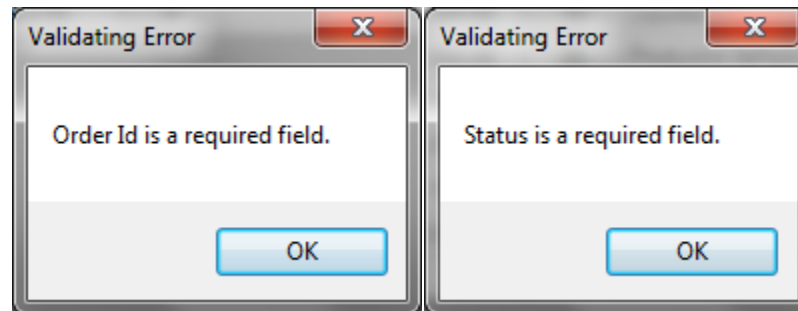
```vb
        'Almacenar aun sin haber perdido el foco, llenar nuevamente la tabla de Customers, recalcular totales y dar el foco a
        "txtLastNames".
        'De existir un error al almacenar o cargar se despliega un mensaje de error.
        Try
            dgvOrderDetails.EndEdit()
            TblOrderDetailsTableAdapter.Update(CustomersProductsOrdersDataSet.tblOrderDetails)
            Me.TblOrderDetailsTableAdapter.FillByOrderId(Me.CustomersProductsOrdersDataSet.tblOrderDetails,
cboOrderId.SelectedValue)
            CalculateTotalsOrderDetails()

        Catch ex As Exception
            MessageBox.Show("An error ocurred while trying to save. Make sure your entering all the required filds before
attempting to save.", "Error found")
        End Try

    End Sub
```

ORDER DETAILS CELL VALIDATING


```vb
    Private Sub CellValidating_CellValidating(sender As Object, e As
System.Windows.Forms.DataGridViewCellValidatingEventArgs) Handles dgvOrderDetails.CellValidating

        'Validar los datos entrados en los campos del DataGridView.
        Select Case e.ColumnIndex

            Case Is = 0
```

```vbnet
            If String.IsNullOrEmpty(e.FormattedValue.ToString()) Then
                MessageBox.Show("Order Id is a required field.", "Validating Error")
                e.Cancel = True
            End If

        Case Is = 1
            If String.IsNullOrEmpty(e.FormattedValue.ToString()) Then
                MessageBox.Show("Product Id is a required field.", "Validating Error")
                e.Cancel = True
            End If

        Case Is = 2

            If String.IsNullOrEmpty(e.FormattedValue.ToString()) Then
                MessageBox.Show("Quantity Sold is a required field.", "Validating Error")
                e.Cancel = True
            End If

        Case Is = 3
            If String.IsNullOrEmpty(e.FormattedValue.ToString()) Then
                MessageBox.Show("Price is a required field.", "Validating Error")
                e.Cancel = True
            End If

        Case Is = 4
            If String.IsNullOrEmpty(e.FormattedValue.ToString()) Then
                MessageBox.Show("Discount Percent is a required field.", "Validating Error")
                e.Cancel = True
            End If

        Case Is = 6
            If String.IsNullOrEmpty(e.FormattedValue.ToString()) Then
                MessageBox.Show("Status is a required field.", "Validating Error")
                e.Cancel = True
            End If

    End Select

End Sub
```

ORDER DETAIL DEFAULT VALUES NEEDED

Order Details

Save   Customer Order   Return

Search Order Details

Order Id: O111 ▼

Order Details

	Order Id	Product Id	Quantity Sold	Price	Discount Percent	Remarks	Status	Products Charge Amount	Net Products Charge
	O111	P111	10	$ 100.00	10	Liquidación	A	$ 1,000.00	$ 900.00
	O111	P222	5	$ 200.00	20	Productos dañados	A	$ 1,000.00	$ 800.00
	O111	P333	1	$ 100.00	10		A	$ 100.00	$ 90.00
	O111	P500	1	$ 20.00	50		A	$ 20.00	$ 10.00
	O111	P444	3	$ 10,000...	0	mbbmnn	A	$ 30,000.00	$ 30,000.00
▶*			1	$ 0.00	0		A		

Total Products: 5

Total Products Charge Amount: $ 32,120.00

Total Net Charges: $ 31,800.00

Customer Discount: |

*Status = A (Active) or I (Inactive)

Grand Total Amount: |

```vb
    Private Sub DefaultValuesNeeded_DefaultValuesNeeded(sender As Object, e As
System.Windows.Forms.DataGridViewRowEventArgs) Handles dgvOrderDetails.DefaultValuesNeeded

        'Asignar valores predetermiandos del DataGridView.
        With e.Row
            .Cells("fldQuantitySoldDataGridViewTextBoxColumn3").Value = "1"
            .Cells("fldPriceDataGridViewTextBoxColumn4").Value = "0"
            .Cells("fldDiscountPercentDataGridViewTextBoxColumn5").Value = "0"
            .Cells("fldStatusDataGridViewTextBoxColumn7").Value = "A"
        End With

    End Sub


    Private Sub GoToCustomerOrder_Click(sender As System.Object, e As System.EventArgs) Handles mnuCustomerOrder.Click

        'Abre ventana Customer Order.
        frmCustomerOrders.ShowDialog()

    End Sub

    Private Sub SelectedIndexChange_SelectedIndexChanged(sender As System.Object, e As System.EventArgs) Handles
cboOrderId.SelectedIndexChanged

        Me.TblOrderDetailsTableAdapter.FillByOrderId(CustomersProductsOrdersDataSet.tblOrderDetails,
cboOrderId.SelectedValue)
        CalculateTotalsOrderDetails()

    End Sub
```

```vbnet
    Private Sub UserAddedRow_UserAddedRow(sender As Object, e As System.Windows.Forms.DataGridViewRowEventArgs) Handles
dgvOrderDetails.UserAddedRow

        'Calcular justo cuando se va a agregar un nuevo record.
        CalculateTotalsOrderDetails()
```

```vbnet
    End Sub

    Private Sub UserDeletedRow_UserDeletedRow(sender As Object, e As System.Windows.Forms.DataGridViewRowEventArgs) Handles dgvOrderDetails.UserDeletedRow

        'Actualizar base de datos y recalcular luego de haber eliminado un record. De existir un error al almacenar o cargar
se despliega un mensaje de error.
        Try
            Me.TblOrderDetailsTableAdapter.Update(Me.CustomersProductsOrdersDataSet.tblOrderDetails)
            CalculateTotalsOrderDetails()
        Catch ex As Exception
            MessageBox.Show("An error ocurred while trying to save.", "Error found")
        End Try

    End Sub

    Private Sub UserDeletingRows_UserDeletingRow(sender As Object, e As System.Windows.Forms.DataGridViewRowCancelEventArgs) Handles dgvOrderDetails.UserDeletingRow

        Dim intSelectedRow As Integer
        intSelectedRow = dgvOrderDetails.CurrentRow.Index

    'Pedir eliminar un record, y salir de la forma. Si indica "NO" se cancela el evento.
        If MessageBox.Show("Are you sure you want permanently delete the product " + CStr(dgvOrderDetails.Item(1, intSelectedRow).Value) + " from order " + CStr(dgvOrderDetails.Item(0, intSelectedRow).Value) + "?", "Delete",
                        MessageBoxButtons.YesNo,
                        MessageBoxIcon.Warning,
                        MessageBoxDefaultButton.Button2) = Windows.Forms.DialogResult.No Then
            e.Cancel = True

        End If

    End Sub

End Class
```