



## CAPÍTULO VI

### PRUEBAS ORIENTADAS A OBJETOS

#### 6.1 PRUEBAS DE CAJA BLANCA Y CAJA NEGRA

##### 6.1.1 PRUEBAS DE CAJA NEGRA:

Esta prueba ayudará a encontrar casos en que el módulo no cumple con su funcionalidad. Probando los datos de entrada y estudiar la salida, sin preocuparse de lo que pueda estar haciendo el módulo por dentro.

Se toma los módulos por separado para analizar sus entradas y salidas, hemos iniciado con el módulo de ventas:

##### 6.1.1.1 MÓDULO VENTAS:

El módulo de Ventas registra la venta en cualquier restaurante del hotel, por eso es necesario comprobar que el momento de ingresar el valor del pago su resultado sea el correcto y manejarlo debidamente con el huésped.

Dentro de esta pantalla se debe ingresar: Nombre cliente, cédula, el número de mesa, el valor que el cliente paga, también se debe ingresar el tipo de menú que desea el cliente. (ver Figura No.93, 94)

##### 6.1.1.1.1 VALORES FÁCILES:



Iniciamos las pruebas ingresando valores sencillos:

- ✚ Clientes: En este campo solo se debe ingresar letras.
- ✚ Cédula: Este campo no se ha validado ya que se puede ingresar no solo la cédula también el número de pasaporte que en un hotel esto es muy común, en este caso se puede ingresar cualquier número.
- ✚ Número de Mesa: Aquí se debe ingresar solo número o letras.

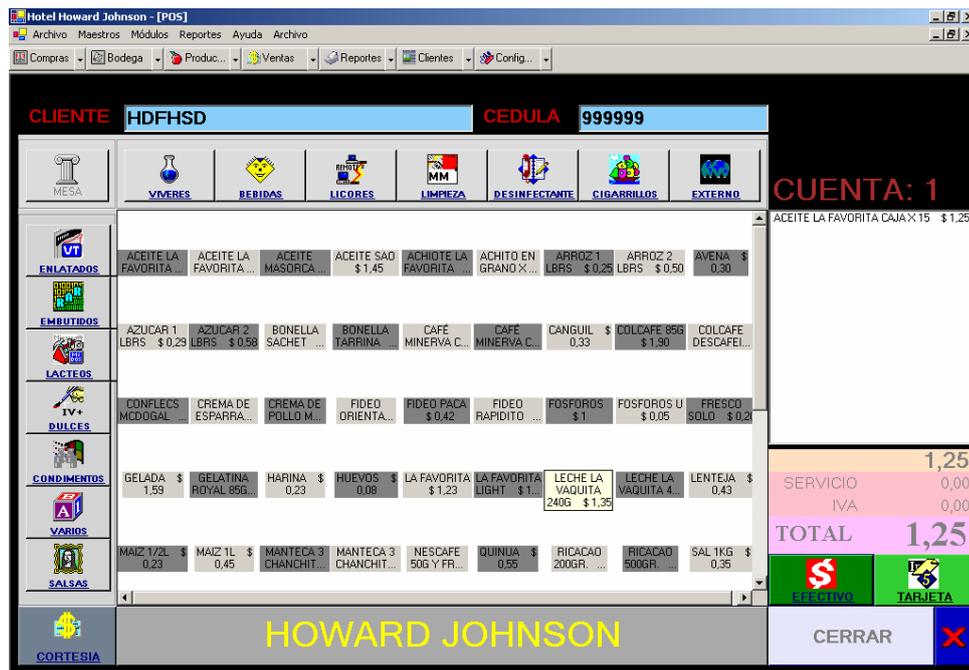


Figura No. 94 (Prueba 1 Pantalla de Ventas)



Figura No. 95 (Prueba 2 Pantalla de Ventas)

El total de la cuenta salió 1,25 se pagó con dos dólares el vuelto es 0,75 se realizó el cálculo correcto.

### 6.1.1.1.2 VALORES TÍPICOS REALISTAS:



Se ensayará un programa con datos seleccionados para que representen como se aplicará. Los Datos han de ser sencillos, de modo que los resultados sean verificables en forma manual.

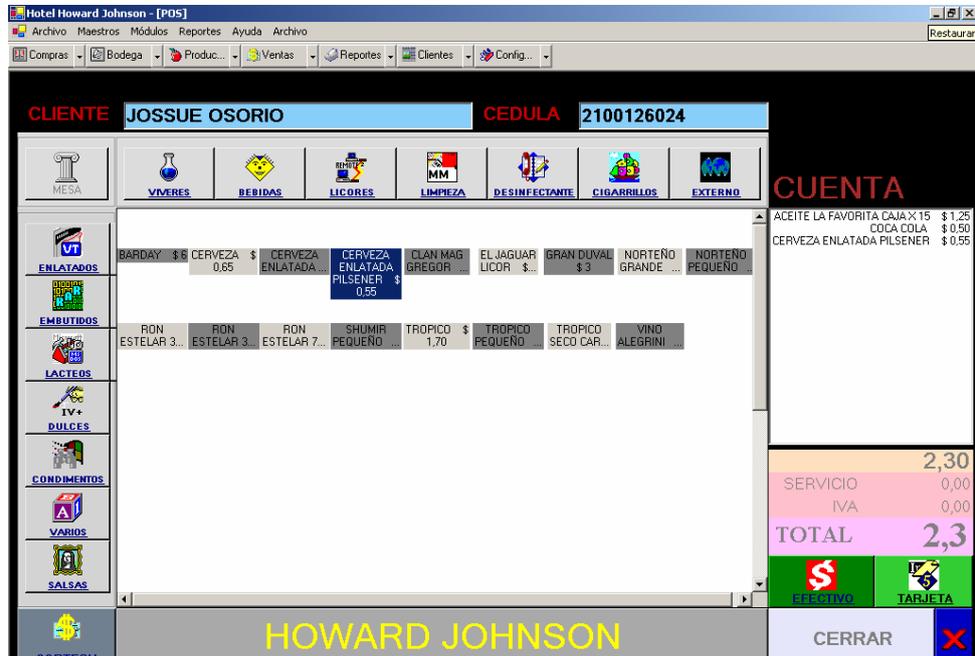


Figura No. 96 (Prueba 3 Pantalla de Ventas)

Se ha ingresado datos reales; no ha existido problema alguno hasta el momento.

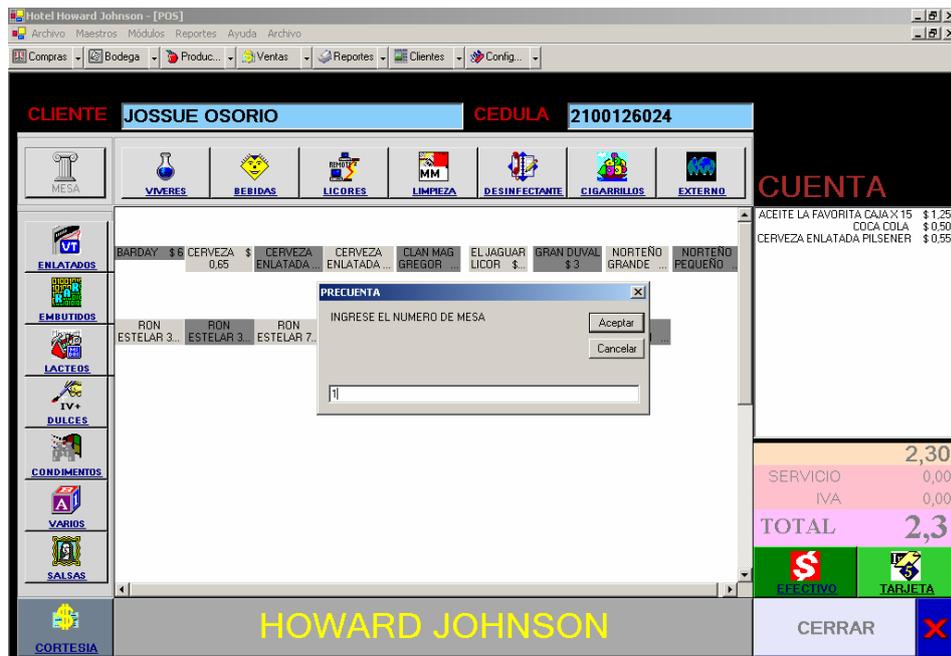


Figura No. 97 (Prueba 4 Pantalla de Ventas)

Se ingresó el número de mesa listo para realizar el pago de la factura.



En este caso se va hacer la prueba del pago con tarjeta de crédito, para esto tenemos que dar un clic en el icono tarjeta.

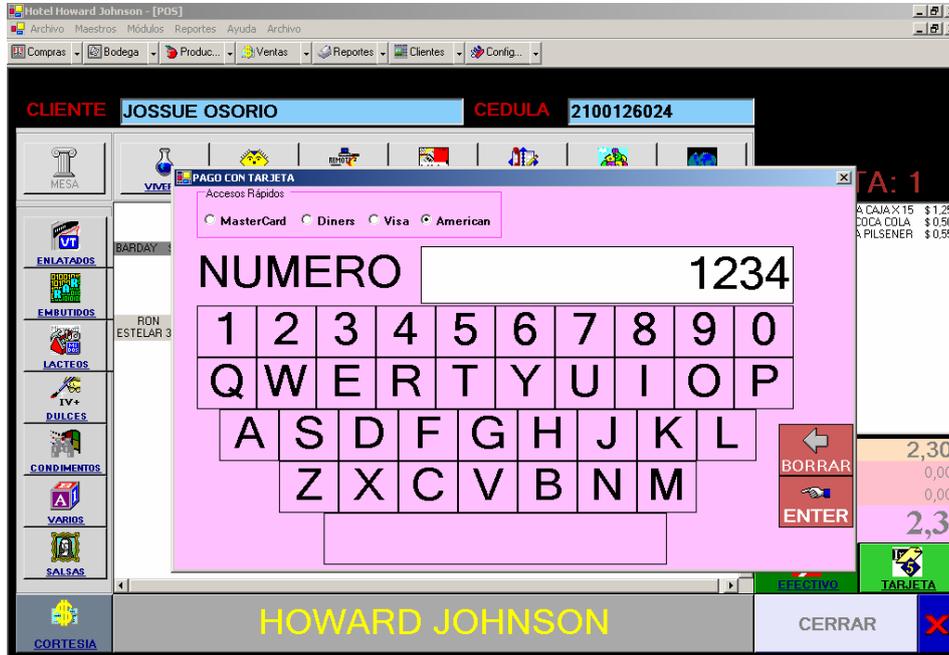


Figura No. 98 (Prueba 5 Pantalla de Ventas)

Debemos elegir que tipo de tarjeta, número de tarjeta y el número de aprobación que emite datafast, éste es un proceso normal y se lo maneja así mismo en el sistema.

Una vez terminado el pago cerramos la cuenta.

Al ingresar valores reales el sistema actúa de manera correcta sin ningún problema.

### 6.1.1.1.3 VALORES EXTREMOS



Para esta pantalla se tomó más énfasis en la parte del pago. Se tomó como valor extremo una cuenta de 2500\$.

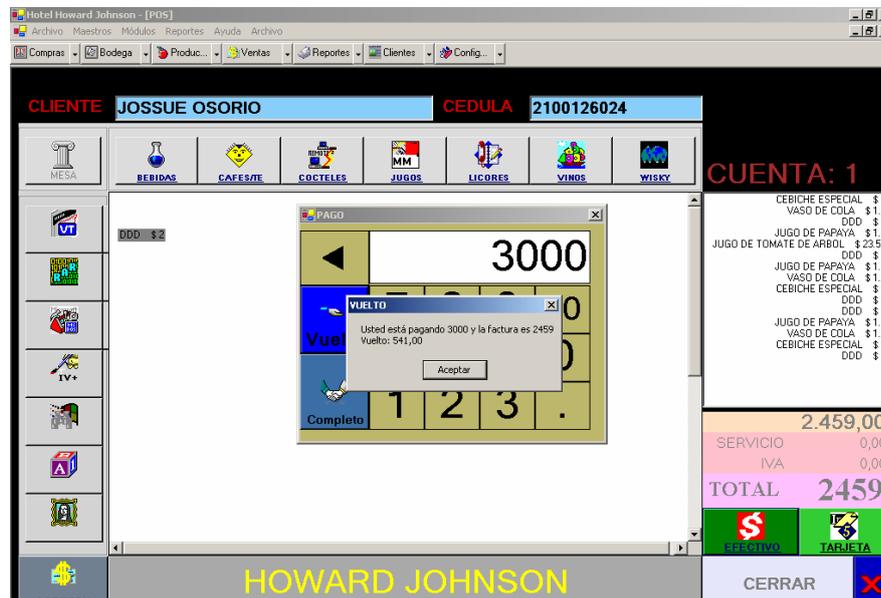


Figura No. 99 (Prueba 6 Pantalla de Ventas)

Se pagó en efectivo 3000\$ el cálculo de vuelto es correcto.

El pago con tarjeta por esa suma no representó ningún problema para el sistema.

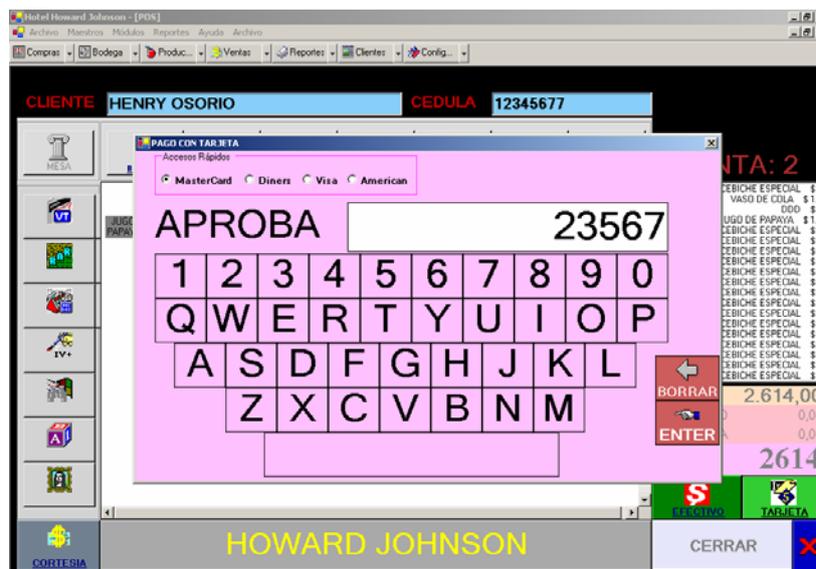


Figura No. 100 (Prueba 7 Pantalla de Ventas)

La cuenta se cerró con éxito.

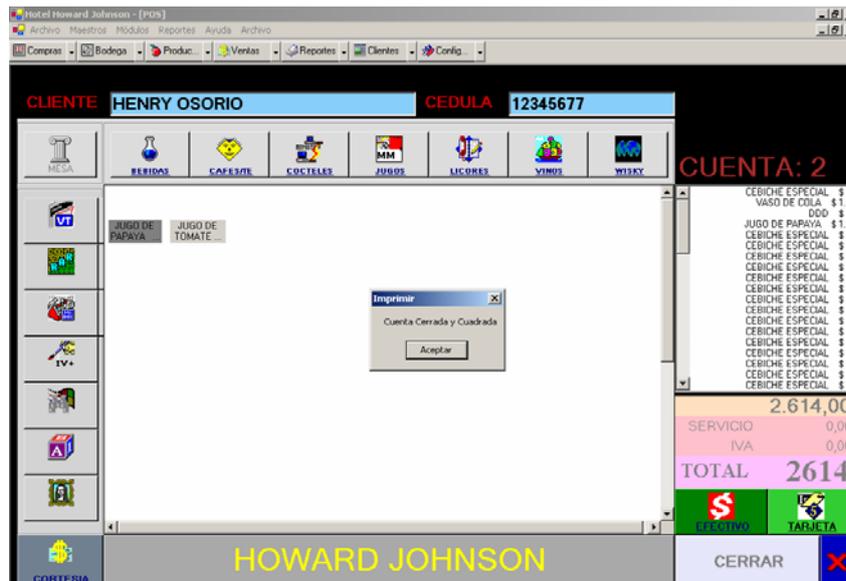


Figura No. 101 (Prueba 8 Pantalla de Ventas)

#### 6.1.1.1.4 VALORES ILEGALES:

Cuando en un programa entra basura, su salida habrá de ser un mensaje de error adecuado.

En la pantalla de ventas se ha validado que en la parte de nombre del cliente no permita que se ingrese números, solo debe ingresar letras; con esto se evita realizar facturas sin nombre.

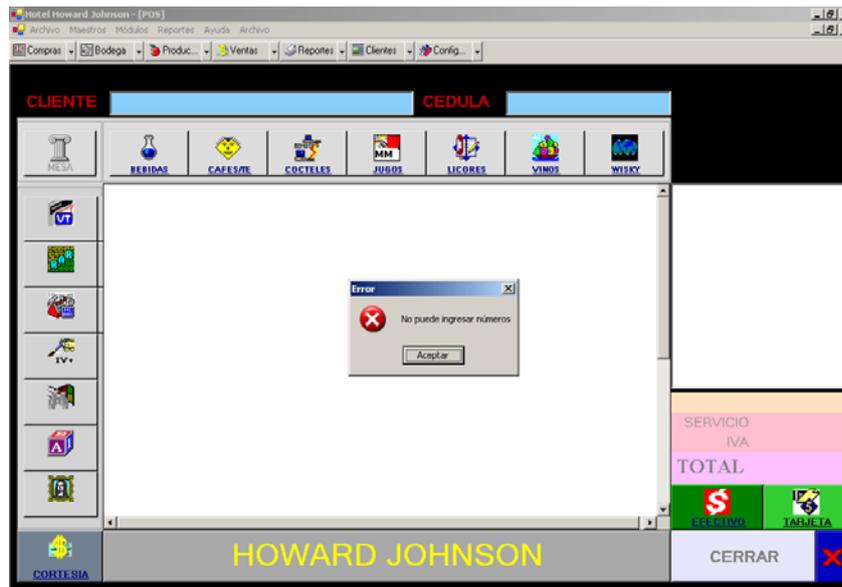


Figura No. 102 (Prueba 9 Pantalla de Ventas)

En la parte de cédula no se realizó ningún control ya que en el hotel se puede ingresar como cédula un número de pasaporte.

Al ingresar el número de mesa se ha validado para poder ingresar letras y números según la localización del hotel, pero no se permite ingresar números negativos.

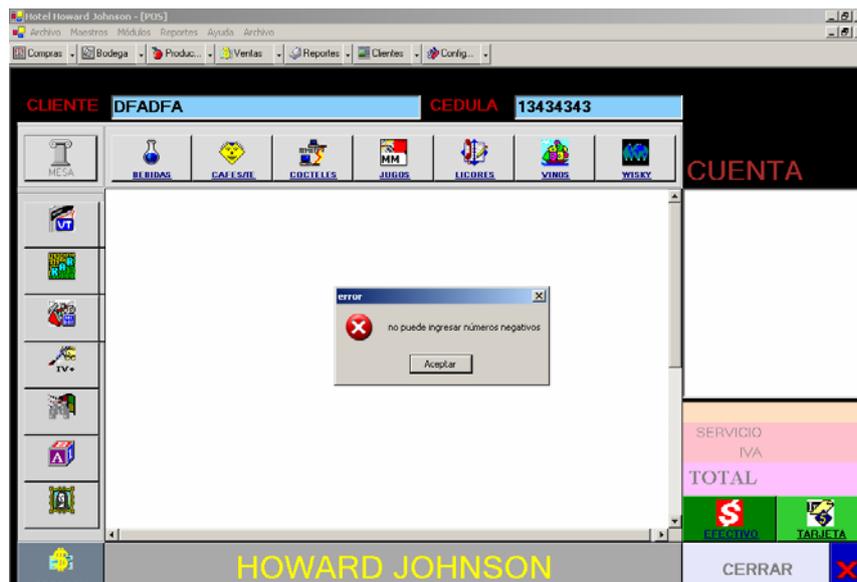


Figura No. 103 (Prueba 10 Pantalla de Ventas)



El momento de cuadrar la cuenta se controla que no se ingrese números menores que el total.

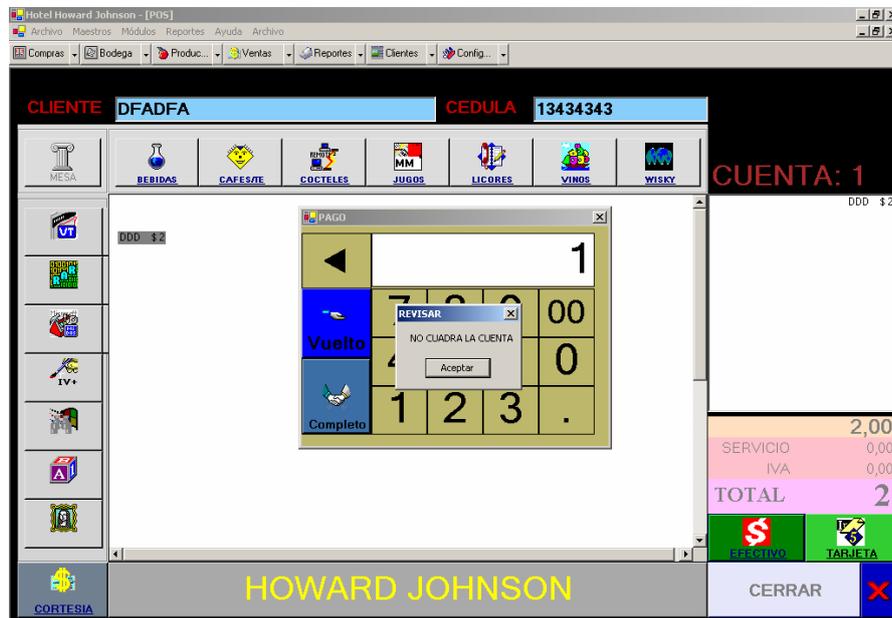


Figura No. 104 (Prueba 11 Pantalla de Ventas)

Si existe datos y el usuario sale de la pantalla de ventas sale el siguiente mensaje

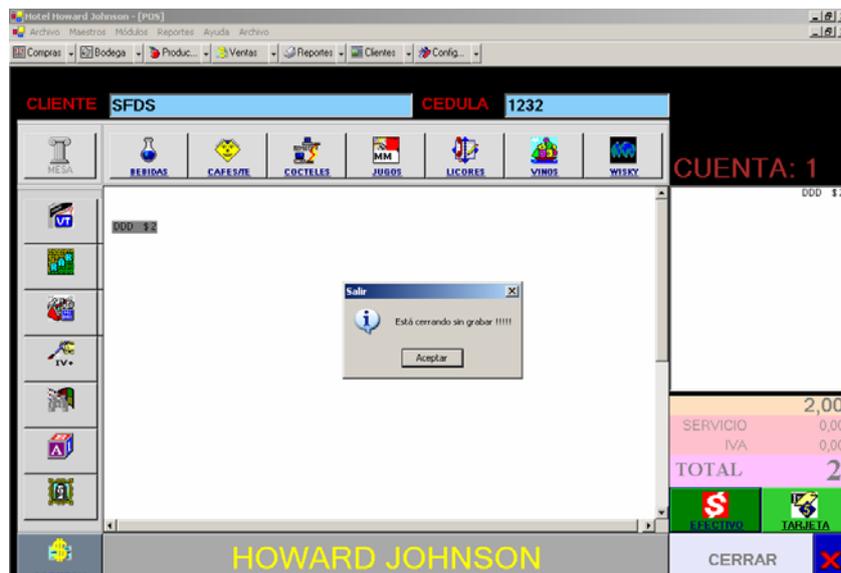


Figura No. 105 (Prueba 12 Pantalla de Ventas)



### 6.1.1.2 MÓDULO DE INVENTARIOS:

Se escogió la pantalla de productos la cual es una de las más importantes de este módulo, ya que de esta pantalla se extrae los valores a las demás pantallas:

#### 6.1.1.2.1 VALORES FÁCILES:

Iniciamos las pruebas ingresando valores sencillos, en esta pantalla se ingresan datos como:

- ✚ Código: Este campo se puede ingresar datos tanto letras como números.
- ✚ Descripción: En este campo solo se pueden ingresar datos alfabéticos
- ✚ Costo\_inicial: Aquí solo se deben ingresar datos numéricos.
- ✚ Costo\_promedio: Aquí solo se deben ingresar datos numéricos.
- ✚ Costo: Aquí solo se deben ingresar datos numéricos.
- ✚ Tipo conteo: Este dato es ingresado previamente.
- ✚ Tipo impuesto: Aquí solo se deben ingresar datos numéricos.
- ✚ Clase: Este dato es ingresado previamente.
- ✚ Proveedor: Este dato es ingresado previamente.
- ✚ Bodega: La bodega donde se ingresa el producto.
- ✚ Stock inicial: Aquí solo se deben ingresar datos numéricos.
- ✚ Stock: Aquí solo se deben ingresar datos numéricos.



- ✚ Máximos: Aquí solo se deben ingresar datos numéricos.
- ✚ Mínimos: Aquí solo se deben ingresar datos numéricos.
- ✚ Unidad compra: Aquí solo se deben ingresar datos numéricos.
- ✚ Unidad: Aquí solo se deben ingresar datos numéricos.

Para esta primera prueba se ha ingresado valores sencillos, aceptando el sistema sin ninguna problema.

Código	0001				
Descripción	PRUEBA				
Costo_inicial	0,05	Costo_compra	0,05	Clase	ALM
Costo_promedio	0,05	Tipo Conteo	MES	Proveedor	0016
Costo	0,10	Tipo impuesto	0	Bodega	
Stock Inicial	100	Máximos		U Compra	KG
Stock	100	Mínimos		Unidad	KG

Figura No. 106 (Prueba 1 Pantalla Productos)

#### 6.1.1.2 VALORES TÍPICOS REALISTAS:

Se ensayará un programa con datos seleccionados para que representen como se aplicará.



<b>Código</b>	0002		
<b>Descripción</b>	CAMARON		
<b>Costo_inicial</b>	5	<b>Costo_compra</b>	5
<b>Costo_promedio</b>	5	<b>Tipo Conteo</b>	MES
<b>Costo</b>	10	<b>Tipo impuesto</b>	0
		<b>Clase</b>	ALM
		<b>Proveedor</b>	0016
		<b>Bodega</b>	
<b>Stock Inicial</b>	100	<b>Máximos</b>	
<b>Stock</b>	100	<b>Mínimos</b>	
		<b>U Compra</b>	KG
		<b>Unidad</b>	KG

Figura No. 107 (Prueba 2 Pantalla Productos)

Hasta este momento se ha ingresado sin ninguna novedad.

### 6.1.1.2.3 VALORES EXTREMOS

Se ingresaran los costos más altos.

<b>Código</b>	0002		
<b>Descripción</b>	CAMARON		
<b>Costo_inicial</b>	350	<b>Costo_compra</b>	300
<b>Costo_promedio</b>	300	<b>Tipo Conteo</b>	MES
<b>Costo</b>	350	<b>Tipo impuesto</b>	0
		<b>Clase</b>	ALM
		<b>Proveedor</b>	0016
		<b>Bodega</b>	
<b>Stock Inicial</b>	100	<b>Máximos</b>	
<b>Stock</b>	100	<b>Mínimos</b>	
		<b>U Compra</b>	KG
		<b>Unidad</b>	KG

Figura No. 108 (Prueba 3 Pantalla Productos)

Hasta el momento el programa no ha dado problema en los ingresos



### 6.1.1.2.4 VALORES ILEGALES:

Cuando en un programa entra basura, su salida habrá de ser un mensaje de error adecuado.

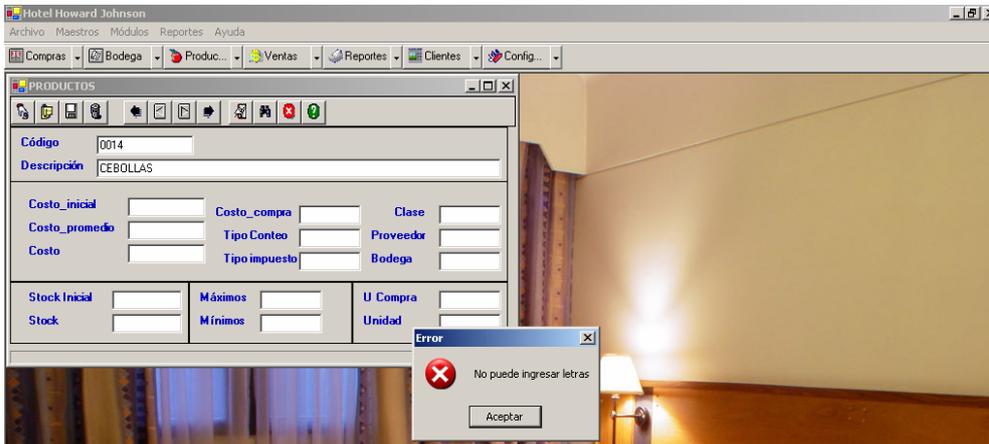


Figura No. 109 (Prueba 4 Pantalla Productos)

En la pantalla de proveedores se ha validado que en el campo de los costos solo ingrese valores numéricos.

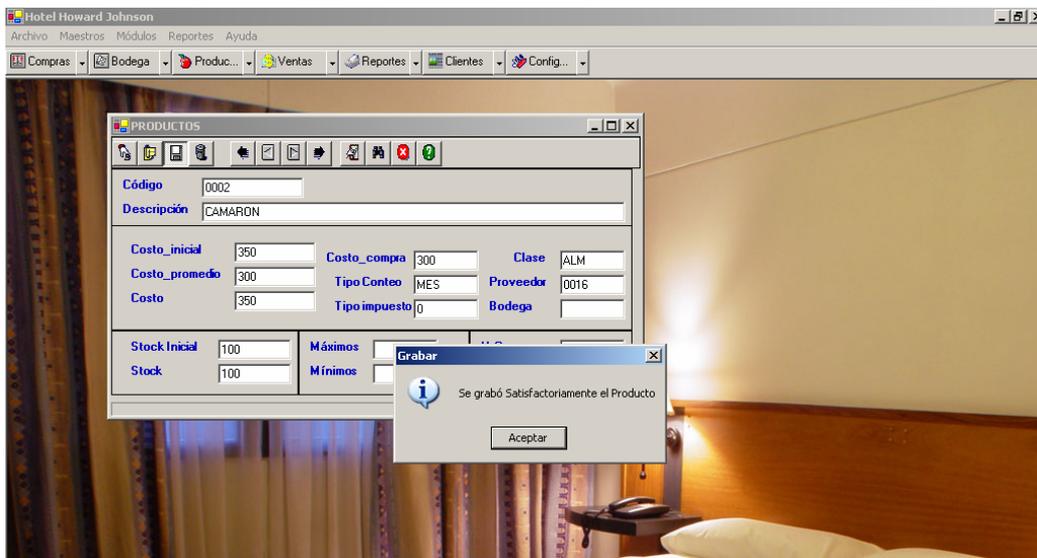


Figura No. 110 (Prueba 5 Pantalla Productos)

Al tener todos los datos correctos graba sin problema.



### 6.1.1.3 PANTALLA SUBRECETAS:

#### 6.1.1.3.1 VALORES FÁCILES:

Dentro inventarios tenemos varias pantallas la segunda pantalla que se realizará la prueba es subrecetas, ya que de la pantalla de productos llega a esta pantalla los datos de costo de cada producto y calcula el costo según la cantidad que se requiere.

En esta pantalla se debe ingresar:

- ✚ Nombre receta: Datos solo alfabéticos.
- ✚ Descripción: Datos solo alfabéticos
- ✚ Unidad de medida: Datos solo alfabéticos
- ✚ Productos: Es una lista de los productos que existe aquí solo se debe escoger el producto.
- ✚ Cantidad. Es un dato numérico.

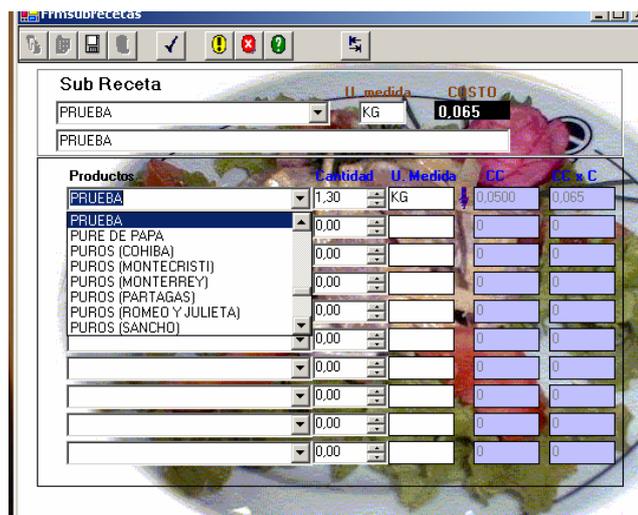


Figura No. 111 (Prueba 1 Pantalla Subreceta)



No presentó ningún problema y el resultado del cálculo es el correcto. Esto lo podemos ver en el campo costo.

### 6.1.1.3.2 VALORES TÍPICOS REALISTAS:

Se ensayará un programa con datos seleccionados para que representen como se aplicará.

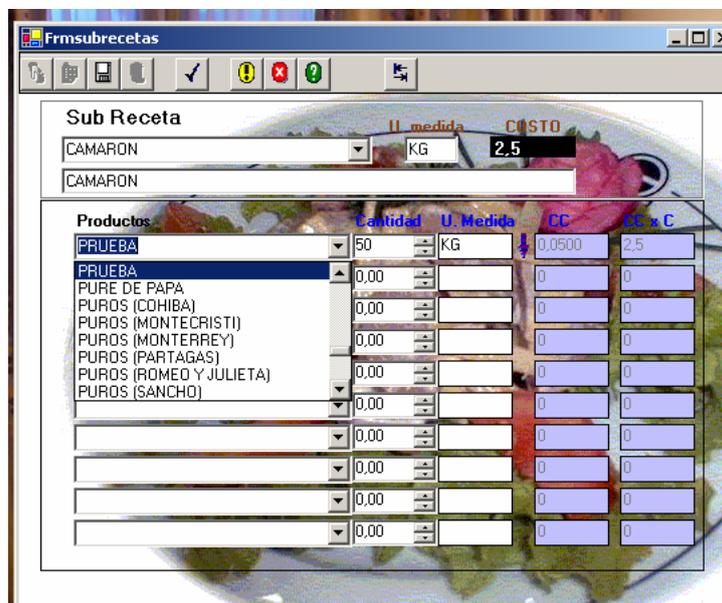


Figura No. 112 (Prueba 2 Pantalla Subreceta)

Al escoger un producto debe aparecer de inmediato los costos que se ingresaron anteriormente.

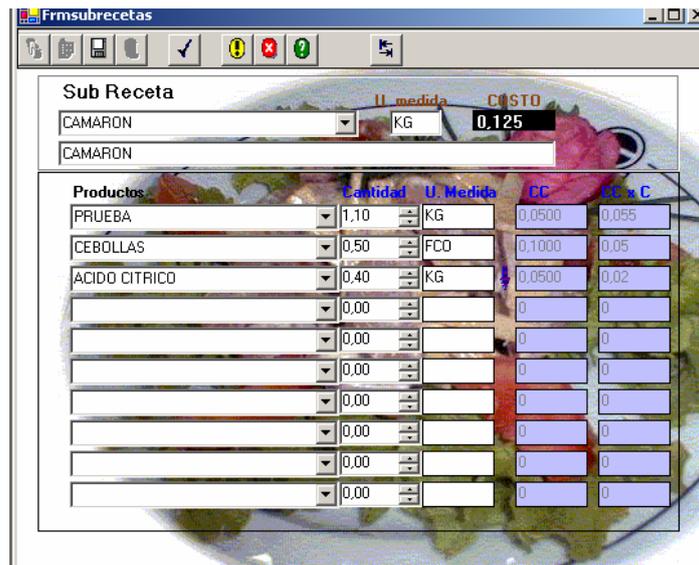


Figura No. 113 (Prueba 3 Pantalla Subreceta)

### 6.1.1.3.3 VALORES EXTREMOS

Se ingresaran cantidades altas

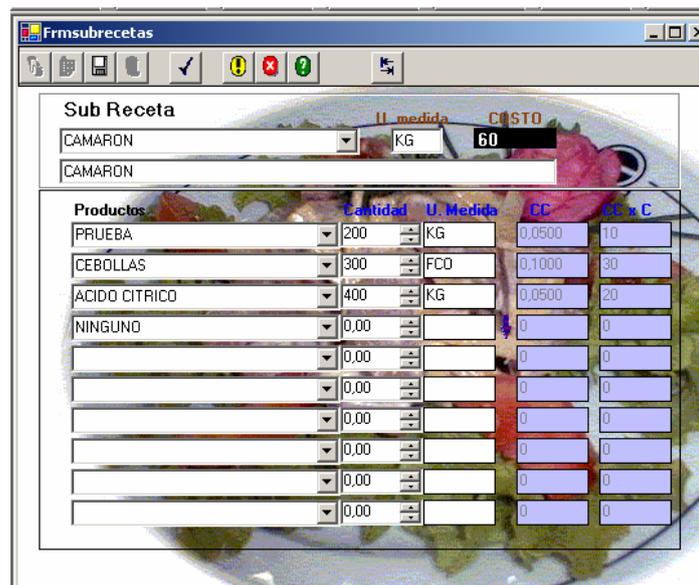


Figura No. 114 (Prueba 4 Pantalla Subreceta)

Al ingresar cantidades exageradas el cálculo del costo es el correcto.



### 6.1.1.3.4 VALORES ILEGALES:

Cuando en un programa entra basura, su salida habrá de ser un mensaje de error adecuado.

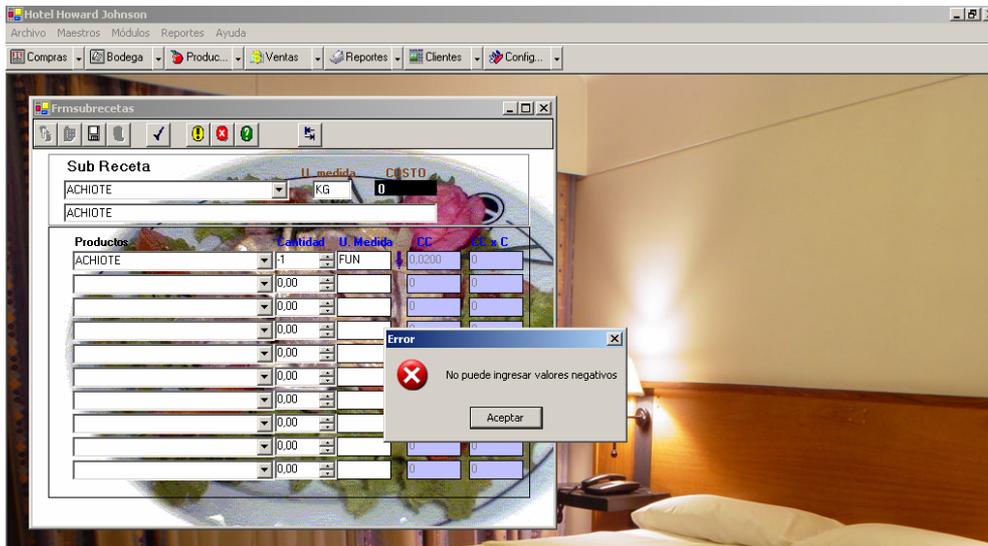


Figura No. 115 (Prueba 5 Pantalla Subreceta)

En cantidad no debe permitir ingresar valores negativos ya que en el cálculo del costo nos reflejaría un valor negativo. Por esto en el sistema emite el siguiente mensaje.

### 6.1.1.4 PANTALLA RESERVA DE SALONES VÍA WEB

#### 6.1.1.4.1 VALORES FÁCILES:

Iniciamos las pruebas ingresando valores sencillos

Lo primero que tenemos que hacer es escoger el hotel donde realizará el evento.

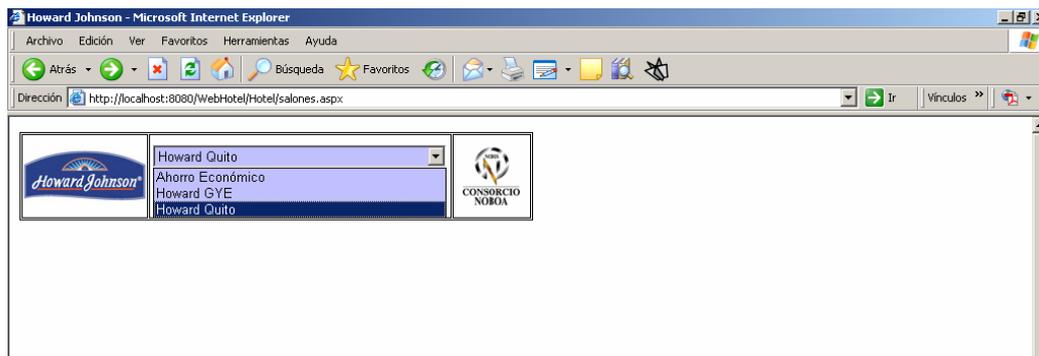


Figura No. 116 (Prueba 1 Pantalla Reserva Salones)

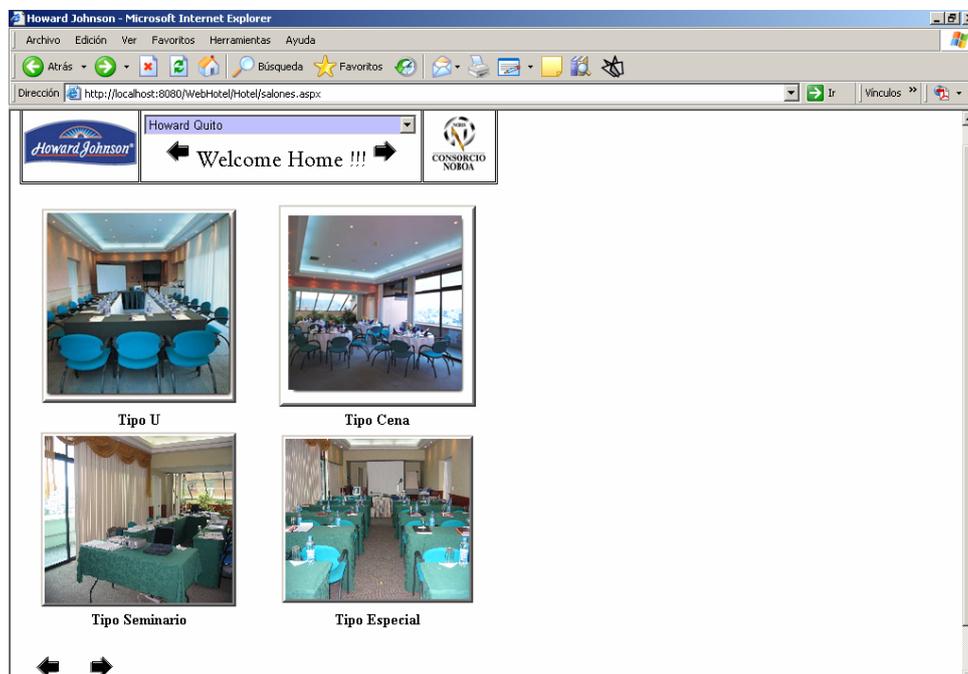


Figura No. 117 (Prueba 2 Pantalla Reserva Salones)

Aquí se debe seleccionar que tipo de ubicación desea, y avanzar con las flechas. Hasta el momento no presenta ningún inconveniente el sistema.



Plan novios (>45)	Cumpleaños (>40)	Seminario (>10)	Vegetariano (>15)	Gala
<b>Entrada:</b> Quiche de almejas en salsa de puerros	<b>Entrada:</b> Crepes de camarón y champiñones	<b>Entrada:</b> Ensalada Melissa	<b>Entrada:</b> Bollito de pan relleno de champiñón y zetas	<b>Entrada:</b> Pollo hojaldrado
<b>Fuerte:</b> Suprema de pollo al vino tinto/Papa croqueta/Legumbre de la temporada	<b>Fuerte:</b> Medallón de lomo gratinado/Arroz con hongos silvestres/Canasta de legumbres	<b>Fuerte:</b> Filete de pollo a la española	<b>Fuerte:</b> Terrina de vegetales	<b>Fuerte:</b> Corvina al ajillo/Legumbre/Arroz p
<b>Postre:</b> Bavarois al coiteau	<b>Postre:</b> Mousse multicolor	<b>Postre:</b> Peras flambeadas con helado de vainilla	<b>Postre:</b> Compota de frutilla	<b>Postre:</b> Flan de caramelo de frambuesa
<b>25 \$</b>	<b>21 \$</b>	<b>13 \$</b>	<b>14 \$</b>	<b>13 \$</b>
<i>Flores-Torta-Disco Movil- Descorche-Gaseosas- Traslado-Suit</i>	<i>Flores-Torta-Disco Movil- Descorche-Gaseosas</i>	<i>Una selección de un bocadito dulce y uno de sal - Café, Té, Aromáticas o Gaseosas</i>	<i>Solo para vegetarianos</i>	<i>Flores-Torta-Disco Movil- Descorche-Gaseosas</i>

Figura No. 118 (Prueba 3 Pantalla Reserva Salones)

Nos presenta una elección de los diferentes menús para elección. Al finalizar estas pantallas podemos llegar a la pantalla principal donde el huésped ingresa los datos para su reservación.

- ✚ Nombres: Dato alfabético
- ✚ Apellidos: Dato alfabético
- ✚ Ciudad: Dato alfabético
- ✚ País. Dato alfabético
- ✚ Cédula: Dato alfanumérico.
- ✚ Teléfono: Dato numérico
- ✚ Email: Dato alfanumérico
- ✚ Todos datos deben ser ingresados de manera obligatoria.
- ✚ Calendario: Donde se escoge los días que durara el evento.



- ✚ Tipo Evento: Se escoge la ubicación que anteriormente se pudo observar.
- ✚ Tipo Menú: Se escoge que tipo de menú se servirá ese día.
- ✚ Adicionales: El cliente ingresa si desea un extra al servicio.

Para este primer control se ingresaron datos sencillos. Hasta el momento no refleja problema alguno.

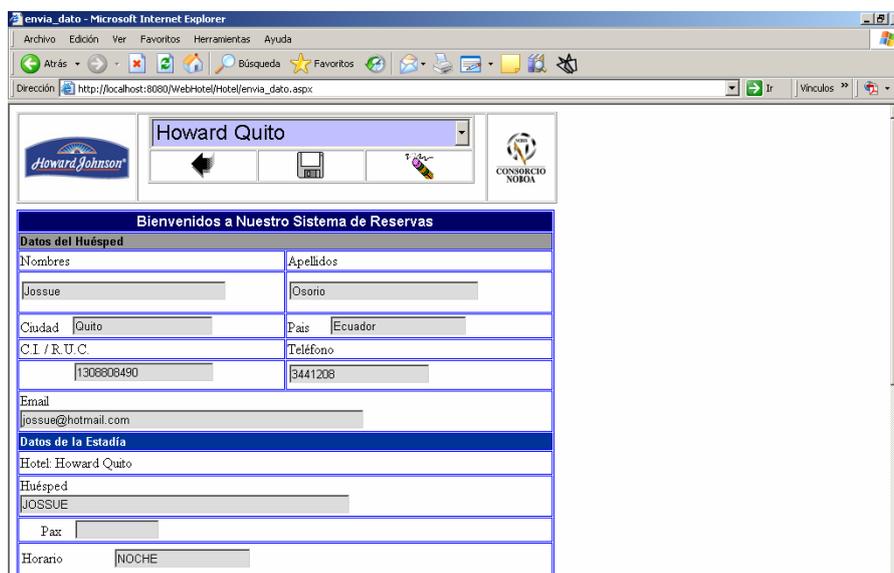


Figura No. 119 (Prueba 4 Pantalla Reserva Salones)

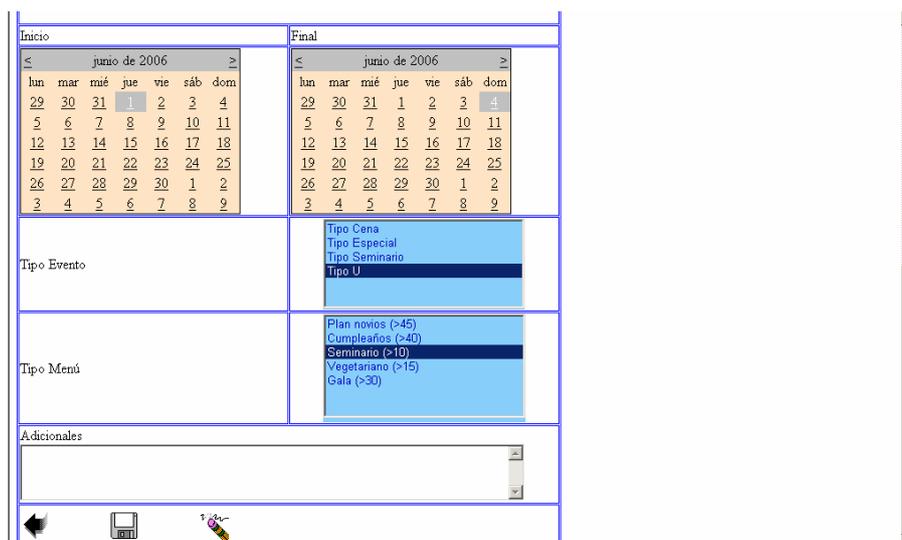


Figura No. 120 (Prueba 5 Pantalla Reserva Salones)



### 6.1.1.4.2 VALORES TÍPICOS REALISTAS:

Se ensayará un programa con datos seleccionados para que representen como se aplicará.

Se ingresaron valores reales, y el comportamiento del sistema fue normal sin encontrar errores en el ingreso y al guardar estos datos.

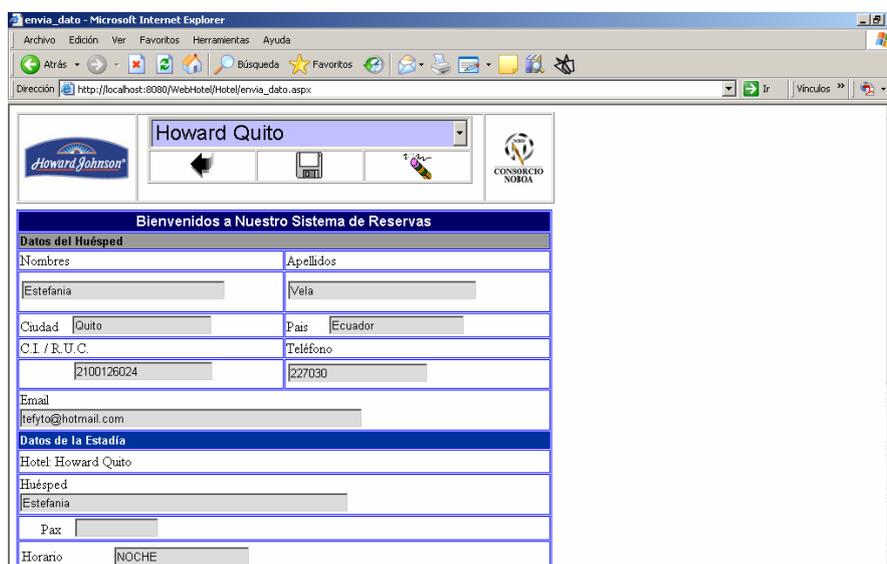


Figura No. 121 (Prueba 6 Pantalla Reserva Salones)

Se grabó con éxito.

### 6.1.1.4.3 VALORES EXTREMOS

Todos los valores se ingresaron de manera correcta. En el campo Pax se debe ingresar el número de personas que asistirán al evento. Este no debe ser mayor a 100.

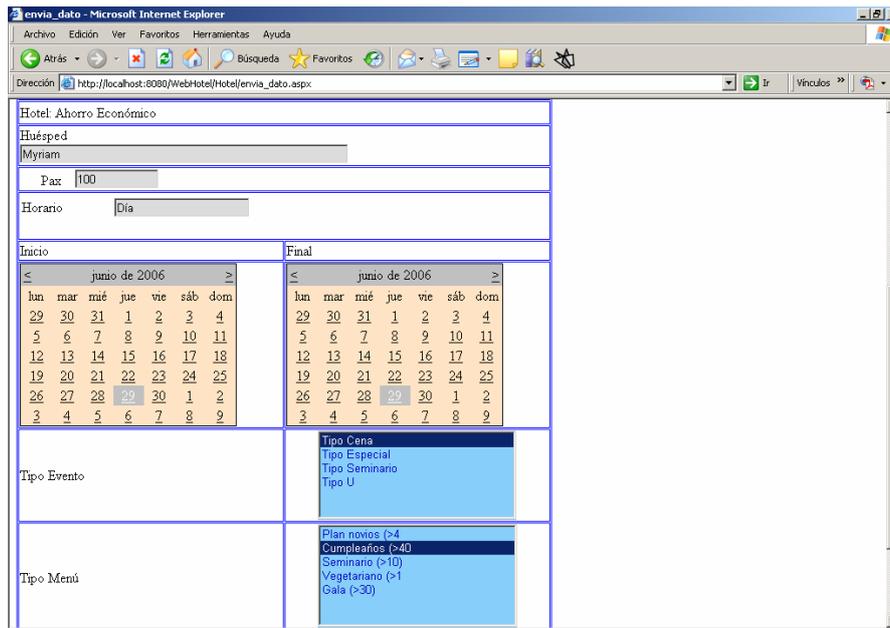


Figura No. 122 (Prueba 7 Pantalla Reserva Salones)

El sistema al ingresar un valor más de 100 en el campo Pax emite un mensaje de error. Hasta el momento no se ha encontrado problema alguno con esta pantalla.

#### 6.1.1.4.4 VALORES ILEGALES:

Cuando en un programa entra basura, su salida habrá de ser un mensaje de error adecuado.

Si no se ingresan valores en el nombre, teléfono el sistema no permite grabar.

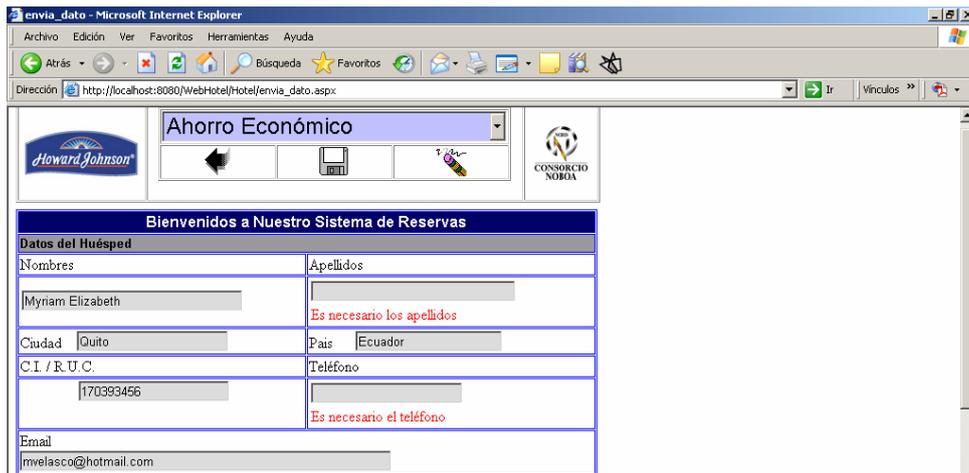


Figura No. 123 (Prueba 8 Pantalla Reserva Salones)

No permite ingresar valores mayores a 100 en el campo PAX y emite un mensaje de error. Si no ingresa todos los valores y se graba emite mensajes de error.

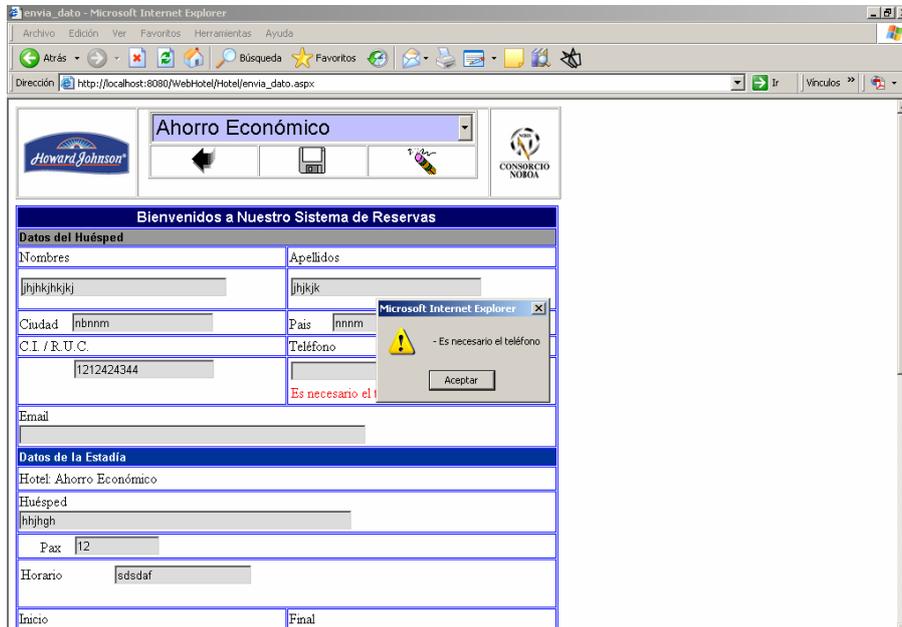


Figura No. 124 (Prueba 9 Pantalla Reserva Salones)



## 6.1.2 PRUEBAS DE CAJA BLANCA

En éstas pruebas se observa el código interno del sistema si sus condiciones, bucles cumplen con su función.

Para esto hemos escogido las principales pantallas del sistema

### 6.1.2.1 PANTALLA VENTAS

En esta pantalla hemos escogido el botón cerrar, la cual realiza el cálculo de la venta.

Primero revisa si no ingresó un valor negativo, que el número de mesa este bien ingresado, asigna un secuencial y entre al ciclo while para grabar los productos ingresados los cuales se almacenaran en el detalle de la factura. Seguido se graba la cabecera, se le incluye también su secuencial, si todos los datos son correctos nos emite el mensaje de cuenta cuadrada, encera datos y termina el proceso.

```
Private Sub btn_cerrar_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles btn_cerrar.Click
```

```
If Me.TXT_CEDULA.Text.Trim <> "" And Me.TXT_CLIENTE.Text.Trim <> "" Then
    If Val(Me.TXT_CEDULA.Text) < 0 Then
        MessageBox.Show("No puede ingresar números negativos", "Error", MessageBoxButtons.OK)
    Else
        If MESA <> 0 Then
            If CDbI(objeto.valor_pago_cliente) = CDbI(Me.LB_TOTAL.Text) Then
                arma_numero()
                While N < Me.LST_CUENTA.Items.Count
                    GRABAR_DET(N, CODIGO + N)
                    N = N + 1
                End While
                GRABAR_CAB()
                incluir_numero_secuencial()
                MessageBox.Show("Cuenta Cerrada y Cuadrada", "Imprimir", MessageBoxButtons.OK)
                encera_datos()
            Else
                MessageBox.Show("NO CUADRA LA CUENTA", "REVISAR", MessageBoxButtons.OK)
            End If
        End If
    End If
```



```

Else
    MessageBox.Show("Ingrese número de mesa", "# MESA", MessageBoxButtons.OK)
End If
End If
Else
    MessageBox.Show("Debe llenar los datos de Nombre y Cedula", MessageBoxButtons.OK)
End If
End Sub

```

Nombre	Valor
objeto.valor_pago_cliente	"9,89"
numero_secuencial	11
CODIGO	111
aux	"0,50"
aux1	"COCA COLA"
N	No se ha declarado el nombre 'N'.

Figura No. 125 (Prueba 1 Código Ventas)

Nombre	Valor
objeto.valor_pago_cliente	"9,89"
numero_secuencial	11
CODIGO	112
aux	"0,55"
aux1	"CERVEZA ENLATADA PILSENER"
N	No se ha declarado el nombre 'N'.

Figura No. 126 (Prueba 2 Código Ventas)

Esta función elimina un producto del listado. El producto seleccionado es removido de la lista, seguido toma su valor el cual restamos del total; el cálculo del porcentaje también se elimina, al igual que el IVA. Esto se hace para que el valor eliminado no afecte en el resultado de la factura.

```

Private Sub eliminar(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
    BTN_ELIMINAR.Click
    Me.LST_CUENTA.SelectedItem().substring(Me.LST_CUENTA.SelectedItem().indexOf("$") + 2)
    Me.LST_CUENTA.Items.RemoveAt(Me.LST_CUENTA.SelectedIndex())
    SUBTOTAL = SUBTOTAL - Format(CDb(m), "##,##0.00")
    Me.LB_SUBTOTAL.Text = Format(SUBTOTAL, "##,##0.00")
    Me.lb_porcentaje.Text = Format(((SUBTOTAL * porcentaje) / 100), "##,##0.00")
    Me.lb_iva.Text = Format(((SUBTOTAL * iva) / 100), "##,##0.00")
    Me.LB_TOTAL.Text = Format(SUBTOTAL + CDb(Me.lb_porcentaje.Text) +
    CDb(Me.lb_iva.Text), "##,##0.00")
End Sub

```



Nombre	Valor
SUBTOTAL	0.77
porcentaje	0

Figura No. 127 (Prueba 3 Código Ventas)

Esta función se la realiza cuando al facturar el cobro es exacto.

Private Sub btn\_completo\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn\_completo.Click

```

If Me.txt_pago.Text <> "" Then
    If Cdbl(objeto.valor_pago_sistema) = Cdbl(Me.txt_pago.Text) Then
        objeto.valor_pago_cliente = Me.txt_pago.Text.Trim
        Hide()
    Else
        MessageBox.Show("NO CUADRA LA CUENTA", MessageBoxButtons.OK)
    End If
Else
    objeto.valor_pago_cliente = objeto.valor_pago_sistema
    Hide()
End If
End Sub

```

Nombre	Valor
objeto.valor_pago_cliente	"9,96"
objeto.valor_pago_sistema	"9,96"

Figura No. 128 (Prueba 4 Código Ventas)

Nombre	Valor
objeto.valor_pago_sistema	"12,11"
txt_pago.Text	"15"
objeto.valor_vuelto	"2,89"
objeto.valor_pago_cliente	"12,11"

Figura No. 129 (Prueba 5 Código Ventas)



### 6.1.2.2 PANTALLA RECETAS:

Esta función llama a las subrecetas para escoger y poder armar la receta, a parte de tomar su nombre también se toma su unidad y costo.

```
Sub buscaproductos1(ByVal sql As String, ByVal tipo As String)
Try
If drcustomer.Read() Then
Me.txt_medida1.Text = drcustomer.Item("unidad").ToString
Me.txt_costo_1.Text = drcustomer.Item(tipo).ToString
End If
drcustomer.Close()
conn.Close()
cmd.Dispose()
conn.Dispose()
Catch err As SqlException
MessageBox.Show(err.message)
Catch err As Exception
MessageBox.Show(err.message)
End Try
End Sub
```

Nombre	Valor
g_conexion.conexion	"Server = HUGO_OSORIO;Database = Ahorro_economico;User ID = hosorio;Password="
Me.txt_costo_1.Text	"0,0200"
Me.txt_medida1.Text	"FUN"

Figura No. 130 (Prueba 6 Código Recetas)

### 6.1.2.3 PANTALLA RESERVA SALONES:

Está pantalla permite grabar todos los datos que se ingresan por la página Web, lo primero que revisa en a que instancia está conectada, si los campos ingresados son correctos y si todos los campos están grabados al término de esto finaliza encerando los datos y limpiando la pantalla.



```
public void graba()
{
if (this.drop_empresa.SelectedValue == "AHE")
    empresa = "Ahorro_Economico";
if (this.drop_empresa.SelectedValue == "GYE")
    empresa = "Hotel_GYE";
if (this.drop_empresa.SelectedValue == "UIO")
    empresa = "HOTEL";

string conexion1 = "Provider =SQLOLEDB; data source = Hugo_Osorio;initial catalog ="+
empresa +";User ID = hosorio;Password=";
OleDbConnection conn = new OleDbConnection();
OleDbCommand cmd= new OleDbCommand();

string cmdtext = "sp_eventos_web1";

try
{
    conn = new OleDbConnection(conexion1);
    conn.Open();
    cmd = new OleDbCommand(cmdtext, conn);
    cmd.CommandType = CommandType.StoredProcedure ;
    OleDbParameter pcodigo = new OleDbParameter("@cod",OleDbType.VarChar);
    pcodigo.Value = busca_numero(empresa);
    cmd.Parameters.Add(pcodigo);
    OleDbParameter pnombre = new OleDbParameter("@nombre",OleDbType.VarChar);
    pnombre.Value = this.txt_nombre.Text ;
    cmd.Parameters.Add(pnombre);
    OleDbParameter papellidos = new OleDbParameter("@apellido",OleDbType.VarChar);
    ;
    papellidos.Value = this.txt_apellidos.Text ;
    cmd.Parameters.Add(papellidos);
    OleDbParameter pciudad = new OleDbParameter("@ciudad",OleDbType.VarChar);
    pciudad.Value = this.txt_ciudad.Text ;
    cmd.Parameters.Add(pciudad);
    OleDbParameter ppais = new OleDbParameter("@pais",OleDbType.VarChar);
    ppais.Value = this.txt_pais.Text;
    cmd.Parameters.Add(ppais);
    OleDbParameter pc_i = new OleDbParameter("@ci",OleDbType.VarChar);
    pc_i.Value = this.txt_ruc.Text ;
    cmd.Parameters.Add(pc_i);
    OleDbParameter ptelefono = new OleDbParameter("@telefono",OleDbType.VarChar);
    ptelefono.Value = this.txt_telefono.Text;
    cmd.Parameters.Add(ptelefono);
    OleDbParameter pemail = new OleDbParameter("@email",OleDbType.VarChar);
    pemail.Value = this.txt_email.Text;
    cmd.Parameters.Add(pemail);
    OleDbParameter phuesped = new OleDbParameter("@huesped",OleDbType.VarChar);
    ;
    phuesped.Value = this.txt_huesped.Text ;
    cmd.Parameters.Add(phuesped);
    OleDbParameter phorario= new OleDbParameter("@horario",OleDbType.VarChar);
    phorario.Value = this.txt_horario.Text ;
    cmd.Parameters.Add(phorario);
    OleDbParameter padicional= new OleDbParameter("@adicion",OleDbType.VarChar);
    padicional.Value = this.txt_adicional.Text ;
    cmd.Parameters.Add(padicional);
    OleDbParameter pmenu= new OleDbParameter("@menu",OleDbType.Integer);
```



```

pmenu.Value = this.lst_menu.SelectedValue ;
cmd.Parameters.Add(pmenu);
OleDbParameter pevento= new OleDbParameter("@evento",OleDbType.Integer) ;
pevento.Value = this.lst_evento.SelectedValue ;
cmd.Parameters.Add(pevento);
OleDbParameter pfechaf= new OleDbParameter("@fecha_final",OleDbType.Date ) ;
pfechaf.Value = this.cal_final.SelectedDate ;
cmd.Parameters.Add(pfechaf);
OleDbParameter pfechai= new OleDbParameter("@fecha_inicio",OleDbType.Date ) ;
pfechai.Value = this.cal_inicio.SelectedDate ;
cmd.Parameters.Add(pfechai);
OleDbParameter ppax= new OleDbParameter("@pax",OleDbType.Integer) ;
ppax.Value = this.txt_pax.Text ;
cmd.Parameters.Add(ppax);
OleDbParameter pconversado= new
OleDbParameter("@conversado",OleDbType.VarChar) ;
pconversado.Value = "N" ;
cmd.Parameters.Add(pconversado);
OleDbParameter previsado= new OleDbParameter("@revisado",OleDbType.VarChar) ;
previsado.Value = "N" ;
cmd.Parameters.Add(previsado);
OleDbParameter pterminado= new
OleDbParameter("@terminado",OleDbType.VarChar) ;
pterminado.Value = "N" ;
cmd.Parameters.Add(pterminado);
OleDbParameter pdivision= new OleDbParameter("@division",OleDbType.VarChar) ;
pdivision.Value = "N" ;
cmd.Parameters.Add(pdivision);
OleDbParameter pcodigoreal= new
OleDbParameter("@codigoreal",OleDbType.VarChar) ;
pcodigoreal.Value = pcodigo.Value ;
cmd.Parameters.Add(pcodigoreal);
cmd.ExecuteNonQuery();
limpia_datos("");
    }
    finally
    {
        conn.Close();
    }
}
}
}

```

Nombre	Valor
conexion1	"Provider =SQLOLEDB; data source = Hugo_Osorio;initial catalog =Ahorro_Economico;User ID = hosorio;Password="
cmdtext	"sp_eventos_web1"
txt_nombre.Text	"ANDREA"
txt_apellidos.Text	"VELA"
txt_ciudad.Text	"QUITO"
txt_pais.Text	"ECUADOR"
txt_ruc.Text	"23737437866"
txt_horario.Text	"NOCHE"
cal_final.SelectedDate	{7/1/2006}
cal_inicio.SelectedDate	{7/1/2006}

Figura No. 131 (Prueba 7 Código Reserva Salones)



En este capítulo hemos puesto en práctica el uso de las pruebas de caja blanca y caja negra.

Lo hemos realizado en ciertos módulos que tienen la mayor importancia del sistema y constituyen las reglas del negocio.

No hemos puesto de todos los módulos por lo que se haría extenso la descripción de cada uno de ellos pero creemos suficiente con las pruebas realizadas para definir que el sistema cumple con las pruebas expuestas.