PA - Pruebas

Ingeniería del Software

José Ramón de Diego

Curso 2015-2016

Contenido

- Pruebas sobre código

Código a probar I

```
private void ActualizarTablaDetallePedido(int idPedido,String[] productos, int[] cantidades) {

        int iterador = 1;
        ResultSet rs = null;
        int IdProducto;
        try {
                Statement stmt = conexion.createStatement();
                while(productos[iterador] != null) {
                rs = stmt.executeQuery("SELECT IdProducto FROM Producto WHERE
Nombre='"+productos[iterador]+"'");
                rs.next();
                IdProducto = (int)rs.getObject("IdProducto");
                stmt.executeUpdate("INSERT INTO DetallePedido(IdPedidoFK,
IdProductoFK, NUnidades) VALUES('"+idPedido+"','"+IdProducto+"',‘
+cantidades[iterador]+"')");
                iterador++;
                                                        }
           }
        catch (SQLException e) {
             System.out.println("Error al insertar datos en la tabla DetallePedido: "+e);
                                        }
}
```

Perteneciente a HacerPedido.Java
(dentro de la capa de control)

Sentencias

```
private void ActualizarTablaDetallePedido(int idPedido,String[] productos, int[] cantidades) {

		int iterador = 1;
		ResultSet rs = null;
		int IdProducto;
		try {
			Statement stmt = conexion.createStatement();
			while(productos[iterador] != null) {
			rs = stmt.executeQuery("SELECT IdProducto FROM Producto WHERE
Nombre='"+productos[iterador]+"'");
			rs.next();
			IdProducto = (int)rs.getObject("IdProducto");
			stmt.executeUpdate("INSERT INTO DetallePedido(IdPedidoFK,
IdProductoFK, NUnidades) VALUES('"+idPedido+"','"+IdProducto+"','"
+cantidades[iterador]+"')");
			iterador++;
								}
		    }
		catch (SQLException e) {
		     System.out.println("Error al insertar datos en la tabla DetallePedido: "+e);
					}
}
```

Sentencias

```
private void ActualizarTablaDetallePedido(int idPedido,String[] productos, int[] cantidades) {

            int iterador = 1;
            ResultSet rs = null;
            int IdProducto;
            try {
                        Statement stmt = conexion.createStatement();
                        while(productos[iterador] != null) {
                        rs = stmt.executeQuery("SELECT IdProducto FROM Producto WHERE
Nombre="'+productos[iterador]+"'");
                        rs.next();
                        IdProducto = (int)rs.getObject("IdProducto");
                        stmt.executeUpdate("INSERT INTO DetallePedido(IdPedidoFK,
IdProductoFK, NUnidades) VALUES('"+idPedido+"','"+IdProducto+"','"
+cantidades[iterador]+"')");
                        iterador++;
                                                                }
               }
            catch (SQLException e) {
                System.out.println("Error al insertar datos en la tabla DetallePedido: "+e);
                                        }
}
```

Sentencias

```
private void ActualizarTablaDetallePedido(int idPedido,String[] productos, int[] cantidades) {

        int iterador = 1;
        ResultSet rs = null;
        int IdProducto;
        try {
                Statement stmt = conexion.createStatement();
                while(productos[iterador] != null) {
                rs = stmt.executeQuery("SELECT IdProducto FROM Producto WHERE
Nombre='"+productos[iterador]+"'");
                rs.next();
                IdProducto = (int)rs.getObject("IdProducto");
                stmt.executeUpdate("INSERT INTO DetallePedido(IdPedidoFK,
IdProductoFK, NUnidades) VALUES('"+idPedido+"','"+IdProducto+"','"
+cantidades[iterador]+"')");
                iterador++;
                                                        }
           }
        catch (SQLException e) {
            System.out.println("Error al insertar datos en la tabla DetallePedido: "+e);
                                        }
}
```

Se ejecuta siempre independientemente del caso de prueba (A)

Se ejecuta siempre (B)

Necesitamos un caso con al menos 1 producto (C)

Necesitamos un caso donde la BD no este disponible (D)

Sentencias

Caso de prueba 1:
Condiciones de la prueba:

Entradas:

Salida esperada:

Cubrimos las sentencias:

Caso de prueba 2:
Condiciones de la prueba:

Entradas:

Salida esperada:

Cubrimos las sentencias:

Decisiones

```
private void ActualizarTablaDetallePedido(int idPedido,String[] productos, int[] cantidades) {

        int iterador = 1;
        ResultSet rs = null;
        int IdProducto;
        try {
                Statement stmt = conexion.createStatement();
                while(productos[iterador] != null) {
                rs = stmt.executeQuery("SELECT IdProducto FROM Producto WHERE
Nombre='"+productos[iterador]+"'");
                rs.next();
                IdProducto = (int)rs.getObject("IdProducto");
                stmt.executeUpdate("INSERT INTO DetallePedido(IdPedidoFK,
IdProductoFK, NUnidades) VALUES('"+idPedido+"','"+IdProducto+"',‘“
+cantidades[iterador]+"')");
                iterador++;
                                                        }
           }
        catch (SQLException e) {
             System.out.println("Error al insertar datos en la tabla DetallePedido: "+e);
                                        }
}
```

Decisiones

```
private void ActualizarTablaDetallePedido(int idPedido,String[] productos, int[] cantidades) {

            int iterador = 1;
            ResultSet rs = null;
            int IdProducto;
            try {
                        Statement stmt = conexion.createStatement();
                        while(productos[iterador] != null) {
                        rs = stmt.executeQuery("SELECT IdProducto FROM Producto WHERE
Nombre='"+productos[iterador]+"'");
                        rs.next();
                        IdProducto = (int)rs.getObject("IdProducto");
                        stmt.executeUpdate("INSERT INTO DetallePedido(IdPedidoFK,
IdProductoFK, NUnidades) VALUES('"+idPedido+"','"+IdProducto+"','"
+cantidades[iterador]+"')");
                        iterador++;
                                                                }
               }
            catch (SQLException e) {
                System.out.println("Error al insertar datos en la tabla DetallePedido: "+e);
                                        }
}
```

(D1)

(D2)

Decisiones

Disponibilidad BD	Productos	D1	D2

Decisiones

Caso de prueba 3:
Condiciones de la prueba:

Entradas:

Salida esperada:

Código a probar II

```
public int RealizarPedido(JTable table, JTextPane textPane, JFormattedTextField formattedTextField)
{    int iterador = 1;
     Estado = "Pendiente";
     int opcion = JOptionPane.showConfirmDialog(null,"¿Está usted seguro de querer procesar este
pedido?","Confirmación",JOptionPane.YES_NO_OPTION);
     if (opcion == 1) // 1 = NO
                return -1;
     else { // !=1 = NO
               IdPedido = ActualizarTablaPedido(formattedTextField.getText(),Estado, textPane.getText());
               Productos = new String[table.getRowCount()];
               Cantidades = new int[table.getRowCount()];

               while(iterador < (table.getRowCount()-1) && table.getValueAt(iterador,0) != null)
                    { Productos[iterador] = (String) table.getValueAt(iterador,0) ;
                      Cantidades[iterador] =(int)table.getValueAt(iterador,1);
                      iterador++;
                     }
                ActualizarTablaDetallePedido(IdPedido, Productos, Cantidades);
          }
   return IdPedido;
}
```

Perteneciente a HacerPedido.Java
(dentro de la capa de control)

Sentencias

```
public int RealizarPedido(JTable table, JTextPane textPane, JFormattedTextField formattedTextField)
{    int iterador = 1;
     Estado = "Pendiente";
     int opcion = JOptionPane.showConfirmDialog(null,"¿Está usted seguro de querer procesar este
pedido?","Confirmación",JOptionPane.YES_NO_OPTION);
     if (opcion == 1) // 1 = NO
               return -1;
     else { // !=1 = NO
               IdPedido = ActualizarTablaPedido(formattedTextField.getText(),Estado, textPane.getText());
               Productos = new String[table.getRowCount()];
               Cantidades = new int[table.getRowCount()];

               while(iterador < (table.getRowCount()-1) && table.getValueAt(iterador,0) != null)
                    { Productos[iterador] = (String) table.getValueAt(iterador,0) ;
                      Cantidades[iterador] =(int)table.getValueAt(iterador,1);
                      iterador++;
                     }
               ActualizarTablaDetallePedido(IdPedido, Productos, Cantidades);
          }
   return IdPedido;
}
```

Se ejecuta siempre independientemente del caso de prueba (A)

Necesitamos un caso donde diga que no. (B)

Necesitamos un caso donde diga que si. (C)

Necesitamos un caso donde haya productos a pedir. (D)

(E)

Sentencias

Caso de prueba 4:
Condiciones de la prueba:

Entradas:

Salida esperada:

Cubrimos las sentencias:

Sentencias

Caso de prueba 5:
Condiciones de la prueba:

Entradas:

Salida esperada:

Cubrimos las sentencias:

Decisiones

```
public int RealizarPedido(JTable table, JTextPane textPane, JFormattedTextField formattedTextField)
{    int iterador = 1;
     Estado = "Pendiente";
     int opcion = JOptionPane.showConfirmDialog(null,"¿Está usted seguro de querer procesar este
pedido?","Confirmación",JOptionPane.YES_NO_OPTION);
     if (opcion == 1) // 1 = NO
                return -1;
     else { // !=1 = NO
               IdPedido = ActualizarTablaPedido(formattedTextField.getText(),Estado, textPane.getText());
               Productos = new String[table.getRowCount()];
               Cantidades = new int[table.getRowCount()];

               while(iterador < (table.getRowCount()-1) && table.getValueAt(iterador,0) != null)
                    { Productos[iterador] = (String) table.getValueAt(iterador,0) ;
                      Cantidades[iterador] =(int)table.getValueAt(iterador,1);
                      iterador++;
                     }
               ActualizarTablaDetallePedido(IdPedido, Productos, Cantidades);
          }
   return IdPedido;
}
```

Decisiones

```
public int RealizarPedido(JTable table, JTextPane textPane, JFormattedTextField formattedTextField)
{    int iterador = 1;
     Estado = "Pendiente";
     int opcion = JOptionPane.showConfirmDialog(null,"¿Está usted seguro de querer procesar este
pedido?","Confirmación",JOptionPane.YES_NO_OPTION);
     if (opcion == 1) // 1 = NO
                return -1;
     else { // !=1 = NO
               IdPedido = ActualizarTabla                                          textPane.getText());
               Productos = new String[table.getRowCount()];
               Cantidades = new int[table.getRowCount()];

               while(iterador < (table.getRowCount()-1) && table.getValueAt(iterador,0) != null)
                    { Productos[iterador] = (String) table.getValueAt(iterador,0) ;
                      Cantidades[iterador] =(int)table.getValueA
                      iterador++;
                     }
               ActualizarTablaDetallePedido(IdPedido, Produ
          }
   return IdPedido;
}
```

Necesitamos un caso
donde conteste que no -> C
y otro donde conteste que si-> F

Necesitamos un caso
donde haya productos a pedir -> C
y otro donde no haya productos en la tabla -> F

Decisiones

opcion	table	opcion == 1	iterador < (table.getRowCount()-1) && table.getValueAt(iterador,0) != null

Decisiones

Caso de prueba 6:
Condiciones de la prueba:

Entradas:

Salida esperada:

Múltiple condición

```
public int RealizarPedido(JTable table, JTextPane textPane, JFormattedTextField formattedTextField)
{    int iterador = 1;
     Estado = "Pendiente";
     int opcion = JOptionPane.showConfirmDialog(null,"¿Está usted seguro de querer procesar este
pedido?","Confirmación",JOptionPane.YES_NO_OPTION);
     if (opcion == 1) // 1 = NO
                return -1;
     else { // !=1 = NO
               IdPedido = ActualizarTablaPedido(formattedTextField.getText(),Estado, textPane.getText());
               Productos = new String[table.getRowCount()];
               Cantidades = new int[table.getRowCount()];

               while(iterador < (table.getRowCount()-1) && table.getValueAt(iterador,0) != null)
                    { Productos[iterador] = (String) table.getValueAt(iterador,0) ;
                      Cantidades[iterador] =(int)table.getValueAt(iterador,1);
                      iterador++;
                     }
                ActualizarTablaDetallePedido(IdPedido, Productos, Cantidades);
           }
   return IdPedido;
}
```

C
F

C C
F C
C F
F F

Múltiple condición

opcion	table	opcion == 1	it.. < Count()-1) && tab..getValueAt!= null	

Múltiple condición

Caso de prueba 7:
Condiciones de la prueba:

Entradas:

Salida esperada:

Múltiple condición

Caso de prueba 8:
Condiciones de la prueba:

Entradas:

Salida esperada: