**Reporte de Pruebas**

|  |  |  |  |
| --- | --- | --- | --- |
| **Estudiante** | Edwin Gamboa | **Fecha** | 12/May/2015 |
| **Proyecto/Tarea** | /NoProyecto/Entrega1BL | | |

|  |  |
| --- | --- |
| **Nombre o # de Prueba** | Test 1 |
| **Objetivo de la Prueba** | Verificar que el programa cree un pedido y lo almacene en la BD. |
| **Descripción de la Prueba** | Se debe construir un objeto de tipo PedidoEntidad y hacer un llamado al método CrearPedido. Realizar este proceso dos veces para dos negocios. |
| **Condiciones de la Prueba** | La base de datos debe estar creada en el computador donde se ejecuten las pruebas.  En la BD deben estar registrados 2 productos (1 por categoría), 2 negocios, 1 usuario (con perfil vendedor) , 1 perfil de vendedor y 2 categorías. |
| **Resultados Esperados** | Cuatro registros nuevos en la base de datos con los pedidos que se crearon (2 por negocio). |
| **Resultados Obtenidos** | **Excepción**: System.NullReferenceException |

|  |  |
| --- | --- |
| **Nombre o # de Prueba** | Test 2 |
| **Objetivo de la Prueba** | Verificar que el programa consulte los pedidos realizados por un negocio. |
| **Descripción de la Prueba** | Se debe construir un objeto de tipo NegocioEntidad (para uno de los dos negocios) y hacer un llamado al método ConsultarPedidoPorNegocio, pasando el NegocioEntidad como parametro. |
| **Condiciones de la Prueba** | La base de datos debe estar creada en el computador donde se ejecuten las pruebas.  En la BD deben estar registrados 2 productos (1 por categoría), 2 negocios, 1 usuario (con perfil vendedor) , 1 perfil de vendedor, 2 pedidos por cada negocio y 2 categorías: |
| **Resultados Esperados** | Un ObservableCollection con **dos** objetos de tipo PedidoEntidad, que corresponde con los pedidos hechos para uno de los dos negocios *(Código de negocio = 2)*. |
| **Resultados Obtenidos** |  |

|  |  |
| --- | --- |
| **Nombre o # de Prueba** | Test 3 |
| **Objetivo de la Prueba** | Verificar que el programa consulte un negocio con su nombre. |
| **Descripción de la Prueba** | Se debe construir un objeto de tipo string (con el nombre de uno de los negocios) y hacer un llamado al método ConsultarNegocioPorNombre pasando el string como parametro. |
| **Condiciones de la Prueba** | La base de datos debe estar creada en el computador donde se ejecuten las pruebas.  En la BD deben estar registrados 2 productos (1 por categoría), 2 negocios, 1 usuario (con perfil vendedor) , 1 perfil de vendedor, 2 pedidos por cada negocio y 2 categorías |
| **Resultados Esperados** | Un objeto de tipo NegocioEntidad que corresponde al negocio con el nombre esperado (*surtimedellin*). |
| **Resultados Obtenidos** |  |

|  |  |
| --- | --- |
| **Nombre o # de Prueba** | Test 4 |
| **Objetivo de la Prueba** | Verificar que el programa consulte el listado de productos que coinciden con un nombre. |
| **Descripción de la Prueba** | Se debe construir un objeto de tipo string (con el nombre de uno de los productos) y hacer un llamado al método ConsultarProductoPorNombre pasando el string como parametro. |
| **Condiciones de la Prueba** | La base de datos debe estar creada en el computador donde se ejecuten las pruebas.  En la BD deben estar registrados 2 productos (1 por categoría), 2 negocios, 1 usuario (con perfil vendedor) , 1 perfil de vendedor, 2 pedidos por cada negocio y 2 categorías |
| **Resultados Esperados** | Un ObservableCollection con dos objetos de tipo ProductoEntidad, que corresponde con el producto que coincide con ese nombre (*colgate*). |
| **Resultados Obtenidos** |  |

|  |  |
| --- | --- |
| **Nombre o # de Prueba** | Test 5 |
| **Objetivo de la Prueba** | Verificar que el programa consulte el listado de productos que pertenecen a una Categoria\_Producto. |
| **Descripción de la Prueba** | Se debe construir un objeto de tipo Categoria\_Producto(con el nombre de uno de los productos) y hacer un llamado al método ConsultarProductoPorNombre pasando este como parametro. |
| **Condiciones de la Prueba** | La base de datos debe estar creada en el computador donde se ejecuten las pruebas.  En la BD deben estar registrados 2 productos (1 por categoría), 2 negocios, 1 usuario (con perfil vendedor) , 1 perfil de vendedor, 2 pedidos por cada negocio y 2 categorías. |
| **Resultados Esperados** | Un ObservableCollection con 1 objeto de tipo ProductoEntidad, que corresponde con el producto que pertenece a esa categoría (*aseo*). |
| **Resultados Obtenidos** |  |

|  |  |
| --- | --- |
| **Nombre o # de Prueba** | Test 6 |
| **Objetivo de la Prueba** | Verificar que el programa consulte el listado de productos que pertenecen a una Categoria\_Producto y coinciden con un nombre. |
| **Descripción de la Prueba** | Se debe construir un objeto de tipo Categoria\_Producto(con el nombre de uno de los productos) y un string con el nombre de un producto, y hacer un llamado al método ConsultarProductoPorNombreYCategoria pasando estos como parametros. |
| **Condiciones de la Prueba** | La base de datos debe estar creada en el computador donde se ejecuten las pruebas.  En la BD deben estar registrados 2 productos (1 por categoría), 2 negocios, 1 usuario (con perfil vendedor) , 1 perfil de vendedor, 2 pedidos por cada negocio y 2 categorías. |
| **Resultados Esperados** | Un ObservableCollection con 1 objeto de tipo ProductoEntidad, que corresponde con el producto que pertenece a esa categoría (*aseo*) y tiene el nombre ingresado (*ariel*). |
| **Resultados Obtenidos** |  |