|  |  |
| --- | --- |
| NOMBRE | R1: Ordenar productos según criterios |
| RESUMEN | Permite ordenar productos según el criterio deseado |
| ENTRADA | |
| El criterio, puede ser:   * Precio * Nombre * Categoría * Código | |
| RESULTADO | |
| Se ordena adecuadamente los productos según el criterio deseado | |

|  |  |
| --- | --- |
| NOMBRE | R2: Buscar productos según criterio |
| RESUMEN | Permite Buscar productos según el criterio deseado |
| ENTRADA | |
| El criterio, puede ser:   * Precio * Nombre * Categoría * Código | |
| RESULTADO | |
| Se filtra adecuadamente los productos según el criterio deseado | |

|  |  |
| --- | --- |
| NOMBRE | R3: Pedir nuevos productos |
| RESUMEN | Permite agregar nuevos productos al inventario del mercado |
|  |  |
|  |  |
| ENTRADA | |
| * Distribuidor * Código del producto * Cantidad a pedir | |
| RESULTADO | |
| Se registra la llegada del producto solicitado con la información ingresada por parámetro. | |

|  |  |
| --- | --- |
| NOMBRE | R4: Vender productos |
| RESUMEN | Permite vender productos a un cliente |
| ENTRADA | |
| El criterio, puede ser:   * Precio * Nombre * Categoría * Código | |
| RESULTADO | |
| Se filtra adecuadamente los productos según el criterio deseado | |

Escenarios de pruebas

|  |  |  |
| --- | --- | --- |
| Nombre Escenario | Clase | Escenario |
| setupScenary1 | MarketTest |  |
| setupScenary2 | DistributorTest |  |
|  |  |  |
|  |  |  |

Casos de prueba

|  |
| --- |
| Objetivo de la prueba: Comprobar que los productos en atibuto productsToShow se ordenan de manera correcta en la clase Distributor. |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | Clase | Método | Escenario | Valores de entrada | Resultado | | Distributor | sortProductByname | setpUpScenary2 | ninguno | El ArrayList de productos queda ordenado por nombre | | Distributor | sortProductByPrice | setUpScenary2 | Ninguno | El ArrayList de productos queda ordenado por precio | | Distributor | sortProductByQuantity | setUpScenary2 | Ninguno | El ArrayList de productos queda ordenado por  cantidad | | Distributor | sortProductByCode | setUpScenary2 | Ninguno | El ArrayList de productos queda ordenado por  Codigo | |

|  |
| --- |
| Objetivo de la prueba: Comprobar que los productos del atributo inventory se ordenan de manera correcta en la clase Market. |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | Clase | Método | Escenario | Valores de entrada | Resultado | | Market | sortProductByName | setUpScenary1 | Ninguno | El ArrayList de productos de la clase Market queda ordenado por nombre | | Market | sortProductByPrice | setUpScenary1 | Ninguno | El ArrayList de productos de la clase Market queda ordenado por precio | | Market | sortProductByQuantity | setUpScenary1 | Ninguno | El ArrayList de productos de la clase Market queda ordenado por cantidad | | Market | sortProductByCode | setUpScenary1 | Ninguno | El ArrayList de productos de la clase Market queda ordenado por codigo | | |

|  |
| --- |
| Objetivo de la prueba: Asegurar que el método de busqueda funcionan en la clase Distributor |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | Clase | Método | Escenario | Valores de entrada | Resultado | | Distributor | searchingByName | setUpScenary2 | name = “D” | Método encuentra el objeto deseado | | Distributor | searchingByCode | setUpScenary2 | name = “A” | Método encuentra el objeto deseado | |

|  |
| --- |
| Objetivo de la prueba: Asegurar que el método de busqueda funcionan en la clase Market |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | Clase | Método | Escenario | Valores de entrada | Resultado | | Market | searchingByName | setUpScenary1 | name = “D” | Método encuentra el objeto deseado | | Market | searchingByCode | setUpScenary1 | name = “A” | Método encuentra el objeto deseado | |

|  |
| --- |
| Objetivo de la prueba: Asegurar que el método de busqueda NO funciona pero lanza la exception en la clase Distributor |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | Clase | Método | Escenario | Valores de entrada | Resultado | | Distributor | searchingByName | setUpScenary1 | name = “F” | Método lanza una NotFoundException | | Distributor | searchingByCode | setUpScenary1 | name = “F” | Método lanza una  NotFoundException | |

|  |
| --- |
| Objetivo de la prueba: Asegurar que el método de busqueda NO funciona pero lanza la exception en la clase Market |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | Clase | Método | Escenario | Valores de entrada | Resultado | | Market | searchingByName | setUpScenary1 | name = “F” | Método lanza una  NotFoundException | | Market | searchingByCode | setUpScenary1 | name = “F” | Método lanza una  NotFoundException | |

|  |
| --- |
| Objetivo de la prueba: Comprobar que se despues de ordenar más productos a un proveedor estos se agreguen al inventario y se registren en el distribuidor. |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | Clase | Método | Escenario | Valores de entrada | Resultado | | Market | getMoreProducts | setUpScenary2 | distributor ,  ,code = 5, quantity = 2; | Se deben sumar al atributo de la clase distributor dos productos más y en el inventario | |

|  |
| --- |
| Objetivo de la prueba: Comprobar que se vendan productos y estos mismos se resten del inventario |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | Clase | Método | Escenario | Valores de entrada | Resultado | | Market | refreshInventorySale | setupScenary2 | Bill = (date,”2”,”30”,array) | se deben restar del inventario los productos comprados y imprimir una factura | |