**Gonzalo De Varona – A00358687**

**Diseño de casos de prueba**

**Casos de pruebas unitarias / Mocks**

**Set up**

|  |  |  |
| --- | --- | --- |
| Nombre | Clase a probar | Escenario |
| setupProductTest() | ProductServiceImp | Se crea un objeto de tipo Product, uno de tipo Productcategory y otro de tipo Productsubcategory, los cuales están disponibles en ProductRepository, ProductcategoryRepository y ProductsubcategoryRepository respectivamente |

|  |  |  |
| --- | --- | --- |
| Nombre | Clase a probar | Escenario |
| setupLocationTest() | LocationServiceImp | Se crea un objeto de tipo Location el cual está disponible en LocationRepository |

|  |  |  |
| --- | --- | --- |
| Nombre | Clase a probar | Escenario |
| setupProductcosthistoryTest() | ProductcosthistoryServiceImp | Se crea un objeto de tipo Productcosthistory el cual está disponible en ProductCostHistoryRepository |

|  |  |  |
| --- | --- | --- |
| Nombre | Clase a probar | Escenario |
| setupProductinventory() | ProductinventoryServiceImp | Se crea un objeto de tipo Productinventory el cual está disponible en ProductInventoryRepository |

**Casos de prueba**

Clase: Taller1ProductApplicationTests

Test: testSave()

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Objetivo de la prueba: Verificar que ProductServiceImp guarde correctamente un producto con un product category y product subcategory asociado. | | | | |
| Clase a probar | Método | Escenario | Input | Output |
| ProductServiceImp | save() | setupProductTest() | ProductcategoryId= 1  ProductsubcategoryId=1  Productnumber =”1010”  sellStartDate=”24/10/2021”  sellEndDate = “24/11/2021”  weight=1  size=2 | El productcategory y productsubcategory son encontrados. El número de producto no está vacio, la fecha de venta inicial es antes de la fecha de venta final, el peso y el tamaño son mayores a cero.  Los atributos del objeto al buscarlo coinciden con los parámetros pasados inicialmente |

Test: testEdit()

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Objetivo de la prueba: Verificar que ProductServiceImp edite correctamente un producto con un product category y product subcategory asociado. | | | | |
| Clase a probar | Método | Escenario | Input | Output |
| ProductServiceImp | edit() | setupProductTest() | Productnumber =””  sellStartDate=”24/09/2021”  sellEndDate = “24/10/2021”  weight=1  size=2 | Todos los parámetros son validados excepto por el número de producto que está vacío, por ende no se puede editar el producto con éxito. |
| ProductServiceImp | edit() | setupProductTest() | Productnumber =”1010”  sellStartDate=”24/09/2021”  sellEndDate = “24/10/2021”  weight=1  size=2 | El número de producto no está vacío, la fecha de venta inicial es antes de la fecha final de venta, el peso y el tamaño son mayores a cero, por ende el producto es editado con éxito. |

Clase: Taller1LocationApplicationTests

Test: testSave()

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Objetivo de la prueba: Verificar que LocationServiceImp guarde correctamente una ubicación. | | | | |
| Clase a probar | Método | Escenario | Input | Output |
| LocationServiceimp | save() | setupLocationTest() | LocationId= 1  name=”Ibiza”  availability =2  Costrate=0.5  modifiedDate = “24/10/2021” | El nombre de la ubicación no tiene menos de 5 letras, la disponibilidad se encuentra entre 1 y 10 y la tasa de costo se encuentra entre 0 y 1.  Los atributos del objeto al buscarlo coinciden con los parámetros pasados inicialmente |

Test: testEdit()

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Objetivo de la prueba: Verificar que LocationServiceimp edite correctamente una ubicación. | | | | |
| Clase a probar | Método | Escenario | Input | Output |
| LocationServiceimp | edit() | setupLocationTest() | name=”Cali”  availability =3  Costrate=0.2 | Todos los parámetros son validados excepto por el nombre de la ubicación que tiene menos de 5 letras, por ende no se puede editar la ubicación con éxito. |
| LocationServiceimp | edit() | setupLocationTest() | LocationId= 1  name=”Ibiza”  availability =4  Costrate=0.5 | El nombre de la ubicación no tiene menos de 5 letras, la disponibilidad está entre 1 y 10 y la tasa de costo está entre 0 y 1, por ende la ubicación es editada con éxito. |

Clase: Taller1ProductcosthistoryApplicationTests

Test: testSave()

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Objetivo de la prueba: Verificar que ProductcosthistoryServiceImp guarde correctamente un Productcosthistory con un producto asociado. | | | | |
| Clase a probar | Método | Escenario | Input | Output |
| ProductcosthistoryServiceImp | save() | setupProductcosthistoryTest() | standardCost= 3  ProductId=1  endDate = “24/10/2021” | El product es encontrado. El costo estándar no es negativo y la fecha de finalización no es mayor a la actual.  Los atributos del objeto al buscarlo coinciden con los parámetros pasados inicialmente |

Test: testEdit()

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Objetivo de la prueba: Verificar que ProductcosthistoryServiceImp edite correctamente un Productcosthistory con un producto asociado. | | | | |
| Clase a probar | Método | Escenario | Input | Output |
| ProductcosthistoryServiceImp | edit() | setupProductcosthistoryTest() | standardCost= 2  endDate = “24/09/2025” | Todos los parámetros son validados excepto por la fecha de finalización que es después de la fecha actual, por ende no se puede editar la costo histórico del producto con éxito. |
| ProductcosthistoryServiceImp | edit() | setupProductcosthistoryTest() | Id=1  standardCost= 2  endDate = “24/09/2020” | El costo estándar no es negativo y la fecha de finalización es antes de la actual, por ende el Productcosthistory es editado con éxito |

Clase: Taller1ProductinventoryApplicationTests

Test: testSave()

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Objetivo de la prueba: Verificar que ProductinventoryServiceImp guarde correctamente un Productinventory con un product y un location asociado. | | | | |
| Clase a probar | Método | Escenario | Input | Output |
| ProductinventoryServiceImp | save() | setupProductinventoryTest() | Quantity=5  ProductId=1  LocationId=1 | El product y el location es encontrado. La cantidad no es inferior a cero.  Los atributos del objeto al buscarlo coinciden con los parámetros pasados inicialmente. |

Test: testEdit()

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Objetivo de la prueba: Verificar que ProductinventoryServiceImp edite correctamente un Productinventory con un product y un location asociado. | | | | |
| Clase a probar | Método | Escenario | Input | Output |
| ProductinventoryServiceImp | edit() | setupProductinventoryTest() | Quantity=-2 | La cantidad es inferior a cero, por ende no se puede editar el inventario del producto con éxito. |
| ProductinventoryServiceImp | edit() | setupProductinventoryTest() | Quantity=2 | La cantidad no es inferior a cero, por ende el Productinventory es editado con éxito |

**Pruebas de integración**

**Set up**

|  |  |  |
| --- | --- | --- |
| Nombre | Clase a probar | Escenario |
| setupProductService() | ProductServiceImp | Se crea un objeto de tipo Product, uno de tipo Productcategory y otro de tipo Productsubcategory, los cuales están disponibles en ProductRepository, ProductcategoryRepository y ProductsubcategoryRepository respectivamente |