# Modulo: Soft.Services.Test.Catalog

## Clase: PriceCalculationServiceTests

## Resumen

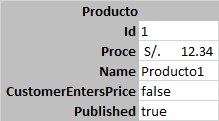
Se busca comprobar el correcto funcionamiento del servicio **PriceCalculationService** el cual se encarga de obtener el precio final de un producto

|  |  |
| --- | --- |
| Clase | PriceCalculationServiceTests |
| Objetos | ICacheManager \_cacheManager;  IWorkContext \_workContext;  IStoreContext \_storeContext;  IDiscountService \_discountService;  ICategoryService \_categoryService;  IProductService \_productService;  IProductAttributeParser \_productAttributeParser;  ShoppingCartSettings \_shoppingCartSettings;  CatalogSettings \_catalogSettings;  IPriceCalculationService \_priceCalcService;  Store \_store; |
| Creaciones | * \_workContext se asigna un valor nulo ya que no hay servicio web * Se crea un \_store con Id =1 * Se asigna dicho \_store al \_storeContext bajo el metodo **CurrentStore** * \_cacheManager asignado a un contexto nulo * Se crean las instancias para \_discountService, \_categoryService, \_productService, \_productAttributeParser, \_shoppingCartSettings, \_catalogSettings * Se crea el servicio para \_priceCalcService con los siguientes parametros:   \_workContext, \_storeContext, \_discountService, \_categoryService,  \_productAttributeParser, \_productService, \_cacheManager,  \_shoppingCartSettings, \_catalogSettings, \_catalogSettings |
| Post Condición | La instancia solo debe contener los valores por defectos creados **PriceCalculationService** y con una instancia de un producto de ejemplo creada para su prueba |

# Caso de prueba: Obtener el precio final de un producto

**Escenario**:

* 1 Producto
* 1 Cliente



**Prueba**:

\_priceCalcService.GetFinalPrice

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Producto | Cliente | Cargo Adiciona | Incluye descuento? | Cantida | Valor esperado |
| Producto1 | Customer | 0 | False | 1 | 12.34 |
| Producto1 | Customer | 0 | False | 2 | 12.34 |

# Caso de prueba: Obtener el precio final de un producto con un precio de nivel

**Escenario**:

* 1 Producto
* 2 Precios de nivel
* 1 Cliente

|  |  |
| --- | --- |
|  |  |

**Prueba**:

\_priceCalcService.GetFinalPrice

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Producto | Cliente | Cargo Adiciona | Incluye descuento? | Cantida | Valor esperado |
| Producto1 | Customer | 0 | False | 1 | 12.34 |
| Producto1 | Customer | 0 | False | 2 | 10 |
| Producto1 | Customer | 0 | False | 3 | 10 |
| Producto1 | Customer | 0 | False | 5 | 8 |

# Caso de prueba: Obtener el precio final de un producto con precios de nivel y roles de clientes

**Escenario**:

* 1 Producto
* 2 Roles de clientes
* 4 Precios de nivel
* 1 Cliente con el Rol1 Asignado

|  |  |
| --- | --- |
|  |  |

**Prueba**:

\_priceCalcService.GetFinalPrice

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Producto | Cliente | Cargo Adiciona | Incluye descuento? | Cantida | Valor esperado |
| Producto1 | Customer | 0 | False | 1 | 12.34 |
| Producto1 | Customer | 0 | False | 2 | 10 |
| Producto1 | Customer | 0 | False | 3 | 10 |
| Producto1 | Customer | 0 | False | 5 | 8 |
| Producto1 | Customer | 0 | False | 10 | 8 |

# Caso de prueba: Obtener el precio final de un producto con adicionales

**Escenario**:

* 1 Producto
* 1 Cliente
* 1 Descuento
* Al \_discountService se le dice que dicho descuento esta asignado al cliente
* Admas el \_discountService tiene una lista de descuentos a la categorias y no al producto

|  |  |
| --- | --- |
|  |  |

**Prueba**:

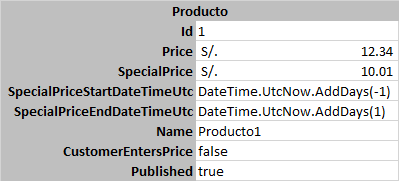
\_priceCalcService.GetFinalPrice

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Producto | Cliente | Cargo Adiciona | Incluye descuento? | Cantida | Valor esperado |
| Producto1 | Customer | 0 | True | 1 | 9.34 |

# Caso de prueba: Obtener el precio final de un producto un precio especial

**Escenario**:

* 1 Producto
* 1 Cliente



**Prueba**:

\_priceCalcService.GetFinalPrice

**Fechas Validas**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Producto | Cliente | Cargo Adiciona | Incluye descuento? | Cantida | Valor esperado |
| Producto1 | Customer | 0 | True | 1 | 10.01 |

**Fecha Invalida**

product.SpecialPriceStartDateTimeUtc = DateTime.UtcNow.AddDays(1);

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Producto | Cliente | Cargo Adiciona | Incluye descuento? | Cantida | Valor esperado |
| Producto1 | Customer | 0 | True | 1 | 12.34 |

**Sin fecha**

product.SpecialPriceStartDateTimeUtc = null;

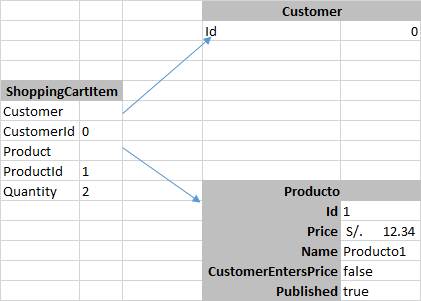
product.SpecialPriceEndDateTimeUtc = null;

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Producto | Cliente | Cargo Adiciona | Incluye descuento? | Cantida | Valor esperado |
| Producto1 | Customer | 0 | True | 1 | 10.01 |

# Caso de prueba: Obtener el precio unitario de un item del carrito de compras

**Escenario**:

* 1 Cliente
* 1 Producto
* 1 Carrito de compras



**Prueba**:

\_priceCalcService.GetUnitPrice

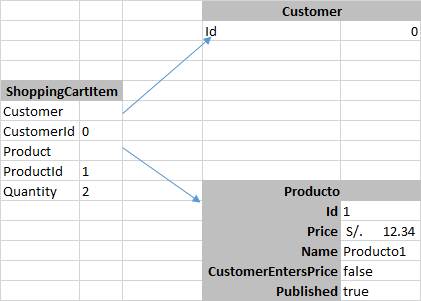
|  |  |
| --- | --- |
| ShoppingCartItem | Valor esperado |
| ShoppingCartItem | 12.34 |

\*Nota esta funcion sumariza el costo adicional de los atributos que se pueden colocar en un producto

# Caso de prueba: Obtener el subtotal del carrito de compras

**Escenario**:

* 1 Cliente
* 1 Producto
* 1 Carrito de compras



**Prueba**:

\_priceCalcService. GetSubTotal

|  |  |
| --- | --- |
| ShoppingCartItem | Valor esperado |
| ShoppingCartItem | 24.68 |