

# DETALHES PARA RESOLUÇÃO DE TESTE EVERYMIND COM IMPLEMENTAÇÃO DE INTERFACE "PricingService".

## Pré-requisitos:

- Remote Site Setting: MockyEndpoint -> https://686526645b5d8d03397fef5b.mockapi.io
- Platform Cache: PriceCache (org-cache = 3 MB)

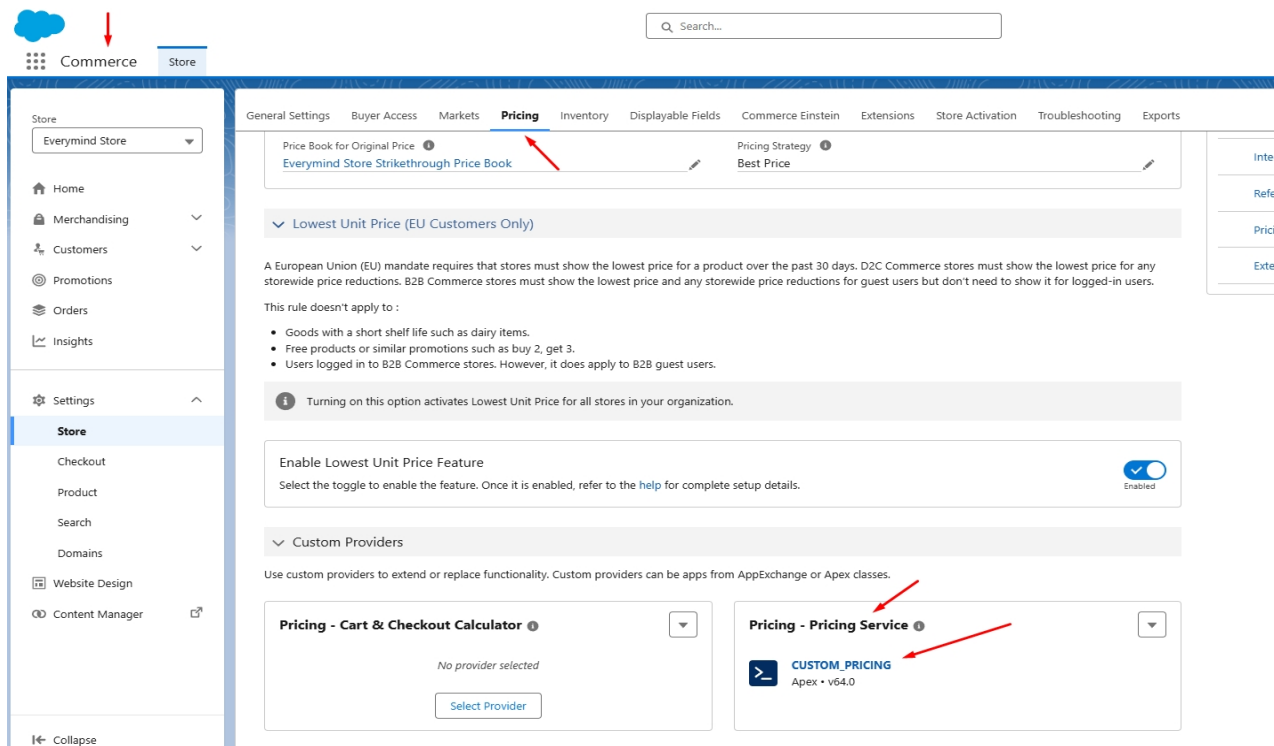
## Passos para realizar os testes:

A) identificar a classe e registrar como EXTENSION provider:

```
ApexClass cls = [
  SELECT Id
  FROM ApexClass
  WHERE Name = 'CustomPricingExtension'
  LIMIT 1
];
```

```
RegisteredExternalService reg = new RegisteredExternalService(
  DeveloperName      = 'CUSTOM_PRICING',
  MasterLabel        = 'Custom Pricing Extension',
  ExtensionPointName = 'Commerce Domain Pricing Service',
  ExternalServiceProviderType = 'Extension',
  ExternalServiceProviderId = cls.Id
);
insert reg;
System.debug('Provider criado: ' + reg.Id);
```

B) Em "**Pricing - Pricing Service**" na Store "**Everymind Store**" foi adicionado a classe apex customizada pelo developername registrado com "**CUSTOM\_PRICING**", como a seguir:



**C)** Não foi possível associar um usuário as contas criadas na Org Developer, então eu realizei manualmente a chamada apresentando o resultado nos logs para os seguintes filtros abaixo nos scripts de teste abaixo para retornar os dados para um produto do user de External\_Id passado no filtro da query abaixo:

```

/*****

```

```

* 1) Localiza registros já existentes

```

```

*****/

```

```

Account acc = [
  SELECT Id
  FROM Account
  WHERE External_Id__c = 'CLC-001'
  LIMIT 1
];

```

```

Product2 prod = [
  SELECT Id
  FROM Product2
  WHERE SKU__c = 'SKU-001'
  LIMIT 1
];

```

```

/*****

```

```

* 2) Cria o item de preço

```

```

* (único construtor: new PricingRequestItem(productId))
*****/
commercestorepricing.PricingRequestItem item =
    new commercestorepricing.PricingRequestItem(prod.Id);

/*****

* 3) Monta o PricingRequest
*****/
commercestorepricing.PricingRequest req =
    new commercestorepricing.PricingRequest();
req.setCurrencyIsoCode('BRL');
req.setEffectiveAccountId(acc.Id);
req.addPricingRequestItem(item);

/*****

* 4) Executa a CustomPricingExtension
*****/
CustomPricingExtension svc = new CustomPricingExtension();
commercestorepricing.PricingResponse resp = svc.processPrice(req);

/*****

* 5) Exibir o resultado no log
*****/
commercestorepricing.PricingResponseItemCollection resItems =
    resp.getPricingResponseItems();

for (Integer i = 0; i < resItems.size(); i++) {
    commercestorepricing.PricingResponseItem r = resItems.get(i);

    System.debug('==== ITEM ' + (i+1) + ' =====');
    System.debug('ProductId.....: ' + r.getProductId());
    System.debug('List Price (BRL)....: ' + r.getListPrice());
    System.debug('Sales Price (BRL)....: ' + r.getNegotiatedPrice());

    String err = r.getErrorMessage();
    if (!String.isBlank(err)) {
        System.debug('Erro.....: ' + err);
    }
}
}

```

Resultado no log:

File Edit Debug Test Workspace Help < >		
CustomPricingExtension.apxc		Log executeAnonymous @04/07/2025, 02:42:51
Execution Log		
Timestamp	Event	Details
02:42:51:127	USER_DEBUG	[54] DEBUG ===== ITEM 1 =====
02:42:51:127	USER_DEBUG	[55] DEBUG ProductId.....: 01tgL000003T9pNQAS
02:42:51:127	USER_DEBUG	[56] DEBUG List Price (BRL)....: 100
02:42:51:127	USER_DEBUG	[57] DEBUG Sales Price (BRL)....: 95

5) O seguinte script de teste retorna todos os dados de produtos disponíveis na Org para o user de External\_Id passado no filtro da query abaixo:

```

/*****
* 1) Localiza a conta
*****/
Account acc = [
    SELECT Id
    FROM Account
    WHERE External_Id__c = 'CLC-001'
    LIMIT 1
];

/*****
* 2) Consulta todos os produtos ativos que queira testar
*****/
List<Product2> products = [
    SELECT Id, SKU__c
    FROM Product2
    WHERE IsActive = true
    AND SKU__c IN ('SKU-001','SKU-002','SKU-003','SKU-004','SKU-005','SKU-
006','SKU-007','SKU-008','SKU-009','SKU-010')
];

/*****
* 3) Monta um PricingRequestItem para cada produto
*****/
commercestorepricing.PricingRequest req = new
commercestorepricing.PricingRequest();
req.setCurrencyIsoCode('BRL');
req.setEffectiveAccountId(acc.Id);

for (Product2 p : products) {

```

```
    commercestorepricing.PricingRequestItem item = new  
    commercestorepricing.PricingRequestItem(p.Id);  
    req.addPricingRequestItem(item);  
}
```

```
/******
```

```
* 4) Executa a extensão e mostra o resultado
```

```
*****/
```

```
CustomPricingExtension svc = new CustomPricingExtension();  
commercestorepricing.PricingResponse resp = svc.processPrice(req);
```

```
commercestorepricing.PricingResponseItemCollection resItems =  
resp.getPricingResponseItems();
```

```
for (Integer i = 0; i < resItems.size(); i++) {  
    commercestorepricing.PricingResponseItem r = resItems.get(i);  
    System.debug('==== ITEM ' + (i+1) + ' ====');  
    System.debug('ProductId.....: ' + r.getProductId());  
    System.debug('List Price (BRL)....: ' + r.getListPrice());  
    System.debug('Sales Price (BRL)....: ' + r.getNegotiatedPrice());
```

```
    String err = r.getErrorMessage();  
    if (!String.isBlank(err)) {  
        System.debug('Erro.....: ' + err);  
    }  
}
```

## Resultado no log:

File Edit Debug Test Workspace Help < >			
CustomPricingExtension.apxc		Log executeAnonymous @04/07/2025, 02:42:51	Log executeAnonymous @04/07/2025, 02:57:18
Execution Log			
Timestamp	Event	Details	
02:57:18:361	USER_DEBUG	[43] DEBUG ===== ITEM 1 =====	
02:57:18:361	USER_DEBUG	[44] DEBUG ProductId.....: 01tgL000003T6hrQAC	
02:57:18:361	USER_DEBUG	[45] DEBUG List Price (BRL)....: 90	
02:57:18:361	USER_DEBUG	[46] DEBUG Sales Price (BRL)....: 85	
02:57:18:362	USER_DEBUG	[43] DEBUG ===== ITEM 2 =====	
02:57:18:362	USER_DEBUG	[44] DEBUG ProductId.....: 01tgL000003T91OQAS	
02:57:18:362	USER_DEBUG	[45] DEBUG List Price (BRL)....: 150	
02:57:18:362	USER_DEBUG	[46] DEBUG Sales Price (BRL)....: 140	
02:57:18:362	USER_DEBUG	[43] DEBUG ===== ITEM 3 =====	
02:57:18:362	USER_DEBUG	[44] DEBUG ProductId.....: 01tgL000003T9pNQAS	
02:57:18:362	USER_DEBUG	[45] DEBUG List Price (BRL)....: 100	
02:57:18:362	USER_DEBUG	[46] DEBUG Sales Price (BRL)....: 95	
02:57:18:362	USER_DEBUG	[43] DEBUG ===== ITEM 4 =====	
02:57:18:362	USER_DEBUG	[44] DEBUG ProductId.....: 01tgL000003TB9dQAG	
02:57:18:362	USER_DEBUG	[45] DEBUG List Price (BRL)....: 120	
02:57:18:362	USER_DEBUG	[46] DEBUG Sales Price (BRL)....: 110	
02:57:18:363	USER_DEBUG	[43] DEBUG ===== ITEM 5 =====	
02:57:18:363	USER_DEBUG	[44] DEBUG ProductId.....: 01tgL000003TBETQA4	
02:57:18:363	USER_DEBUG	[45] DEBUG List Price (BRL)....: 250	
02:57:18:363	USER_DEBUG	[46] DEBUG Sales Price (BRL)....: 230	
02:57:18:363	USER_DEBUG	[43] DEBUG ===== ITEM 6 =====	
02:57:18:363	USER_DEBUG	[44] DEBUG ProductId.....: 01tgL000003TBG5QAO	
02:57:18:363	USER_DEBUG	[45] DEBUG List Price (BRL)....: 180	
02:57:18:363	USER_DEBUG	[46] DEBUG Sales Price (BRL)....: 170	
02:57:18:363	USER_DEBUG	[43] DEBUG ===== ITEM 7 =====	
02:57:18:363	USER_DEBUG	[44] DEBUG ProductId.....: 01tgL000003TBKvQAO	
02:57:18:363	USER_DEBUG	[45] DEBUG List Price (BRL)....: 75	
02:57:18:363	USER_DEBUG	[46] DEBUG Sales Price (BRL)....: 70	
02:57:18:364	USER_DEBUG	[43] DEBUG ===== ITEM 8 =====	
02:57:18:364	USER_DEBUG	[44] DEBUG ProductId.....: 01tgL000003TBMXQA4	
02:57:18:364	USER_DEBUG	[45] DEBUG List Price (BRL)....: 300	
02:57:18:364	USER_DEBUG	[46] DEBUG Sales Price (BRL)....: 280	
02:57:18:364	USER_DEBUG	[43] DEBUG ===== ITEM 9 =====	
02:57:18:364	USER_DEBUG	[44] DEBUG ProductId.....: 01tgL000003TBRNQA4	
02:57:18:364	USER_DEBUG	[45] DEBUG List Price (BRL)....: 400	
02:57:18:364	USER_DEBUG	[46] DEBUG Sales Price (BRL)....: 380	
02:57:18:364	USER_DEBUG	[43] DEBUG ===== ITEM 10 =====	
02:57:18:364	USER_DEBUG	[44] DEBUG ProductId.....: 01tgL000003TDBRQA4	
02:57:18:364	USER_DEBUG	[45] DEBUG List Price (BRL)....: 220	
02:57:18:364	USER_DEBUG	[46] DEBUG Sales Price (BRL)....: 210	