

Ejercicio 7. Creación de la Factura



Desarrollaremos el siguiente requisito:

4. Al crear la factura de compra mediante la funcionalidad de "Traer Albaranes":
 - Satisfactorio: se crea la línea de factura de forma normal
 - No Satisfactorio: se crea la línea de factura con un 100% de descuento

En este ejercicio aprenderemos:

- A utilizar Enums
- A desarrollar y ejecutar Tests para nuestros desarrollos
- A utilizar el nuevo comando **AL: Find event**

Indicaciones

Para completar el ejercicio realiza las siguientes acciones:

1. Crea una nueva rama de Git
2. Haz los tests necesarios para asegurar que se cumplen los requisitos
3. Haz el desarrollo necesario para asignar un 100% de descuento a una línea de factura de un producto recepcionado como No Satisfactorio
4. Haz un merge de tu rama de desarrollo con la rama principal
5. Elimina la rama del desarrollo

Indicaciones paso a paso

Para completar el ejercicio sigue los siguientes pasos en Visual Studio Code:

1. Crea una nueva rama de Git llamada *Feature_InvoiceCreation*
2. Desarrollo de tests
 1. En la extensión de test, añade el siguiente procedimiento a la codeunit de test

```
[Test]
[HandlerFunctions('GetReceiptLinesPage')]
procedure InvoiceCreationP008();
var
    PurchaseOrderHeader: Record "Purchase Header";
    PurchaseInvoiceHeader: Record "Purchase Header";
    PurchaseLine: Record "Purchase Line";
    InvoiceLine: Record "Purchase Line";
    Item: Record Item;
    QualityControlLibrary: Codeunit "Clip Quality Control - Library";
    PurchaseLibrary: Codeunit "Library - Purchase";
    PurchaseDocumentType: Enum "Purchase Document Type";
    PostedDocumentNo: Code[20];
begin
    // [Scenario] When an invoice is created with the "Get Receptions"
    functionality, items received as non-satisfactory are created with a
```

```

100% discount

    // [Given] Setup: An item that requires Quality Control
    //                      A Purchase Order that uses the items
    //                      Non-Satisfactory Quality Control information set
on the order
    //                      A Posted Receipt for the order
    //                      A Purchase Invoice header
    QualityControlLibrary.CreateItemWithQualityMeasures(Item);

    PurchaseLibrary.CreatePurchHeader(PurchaseOrderHeader,
PurchaseDocumentType::Order.AsInteger(), '');
    PurchaseLibrary.CreatePurchaseLine(PurchaseLine,
PurchaseOrderHeader, PurchaseLine.Type::Item.AsInteger(), Item."No.",
10);

QualityControlLibrary.SetQualityControlInformationOnPurchaseLine(Purcha
seLine, PurchaseLine."Clip Quality Control Result"::"Non-
Satisfactory");

    PostedDocumentNo :=
PurchaseLibrary.PostPurchaseDocument(PurchaseOrderHeader, true, false);

    PurchaseLibrary.CreatePurchHeader(PurchaseInvoiceHeader,
PurchaseDocumentType::Invoice.AsInteger(), PurchaseOrderHeader."Buy-
from Vendor No.");

    // [When] Exercise: Use the "Get Receipts" functionality to
create the Invoice lines
    InvoiceLine."Document Type" := PurchaseInvoiceHeader."Document
Type";
    InvoiceLine."Document No." := PurchaseInvoiceHeader."No.";
    PurchaseLibrary.GetPurchaseReceiptLine(InvoiceLine);

    // [Then] Verify: The non-satisfactory item line has a 100%
discount
    InvoiceLine.Get(PurchaseInvoiceHeader."Document Type",
PurchaseInvoiceHeader."No.", 20000);
    AssertThat.AreEqual(100, InvoiceLine."Line Discount %",
'C01.P008.A001 InvoiceLine."Line Discount %"');
end;

[ModalPageHandler]
procedure GetReceiptLinesPage(var GetReceiptLines: TestPage "Get
Receipt Lines");
begin
    GetReceiptLines.OK().Invoke();
end;

```

2. Publica y ejecuta los tests para comprobar que este último falla

3. Creación de líneas de factura con 100% de descuento

1. Crea una nueva codeunit 50103 "Clip Purchase Invoice Creation", utiliza el comando **AL: Find Event** para suscribirte al evento *OnBeforeInsertInvLineFromRcptLine* y escribe el siguiente código:

```
codeunit 50103 "Clip Purchase Invoice Creation"
{
    [EventSubscriber(ObjectType::Table, Database::"Purch. Rcpt. Line",
    'OnBeforeInsertInvLineFromRcptLine', '', false, false)]
    local procedure OnBeforeInsertInvLineFromRcptLine(var
    PurchRcptLine: Record "Purch. Rcpt. Line"; var PurchLine: Record
    "Purchase Line"; PurchOrderLine: Record "Purchase Line");
    begin
        case PurchOrderLine."Clip Quality Control Result" of
            PurchOrderLine."Clip Quality Control Result"::" ",
            PurchOrderLine."Clip Quality Control Result"::Satisfactory:
                ;// Do nothing
            PurchOrderLine."Clip Quality Control Result"::"Non-
            Satisfactory":
                PurchLine.Validate("Line Discount %", 100);
        end;
    end;
}
```

2. Publica el desarrollo
3. Ejecuta los tests para comprobar que el desarrollo realizado cumple con los tests planteados
4. Haz un merge de tu rama de desarrollo con la rama principal
5. Elimina la rama de desarrollo