

## Ejercicio 9. Vistas

Desarrollaremos el siguiente requisito:

1. La información relativa al control de calidad debe ser visible desde los Movimientos de Productos

En este ejercicio aprenderemos:

- A traspasar información desde un documento de compra hasta los movimientos de productos
- A programar vistas

### 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 cumple el requisito
3. Haz el desarrollo necesario para traspasar información desde el documento de compra hasta los movimientos de producto
4. Desarrolla algunas vistas en los movimientos de producto para una mejor experiencia de usuario
5. Haz un merge de tu rama de desarrollo con la rama principal
6. 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\_QConILE*
2. Desarrollo de tests
  1. Amplia los procedimientos de test `PurchaseOrderPostingP007` y `PurchaseOrderPostingP010` para comprobar que al registrar, la información relativa al control de calidad es traspasada a los movimientos de producto
    - `PurchaseOrderPostingP007`

```
ItemLedgerEntry.FindFirst();
AssertThat.AreEqual(PurchaseLine1."Clip Quality Control Result",
ItemLedgerEntry."Clip Quality Control Result", 'C01.P007.A002
First ItemLedgerEntry."Clip Quality Control Result');
```

```
AssertThat.AreEqual(PurchaseLine2."Clip Quality Control Result",
ItemLedgerEntry."Clip Quality Control Result", 'C01.P007.A012
Second ItemLedgerEntry."Clip Quality Control Result');
```

```
AssertThat.AreEqual(PurchaseLine2."Clip Quality Control Result",
ItemLedgerEntry."Clip Quality Control Result", 'C01.P007.A023
Second ItemLedgerEntry."Clip Quality Control Result');
```

### ■ PurchaseOrderPostingP010

```

ItemLedgerEntry.FindFirst();
AssertThat.AreEqual(PurchaseLine1."Clip Quality Control Result",
ItemLedgerEntry."Clip Quality Control Result", 'C01.P010.A002
First ItemLedgerEntry."Clip Quality Control Result"');

AssertThat.AreEqual(PurchaseLine2."Clip Quality Control Result",
ItemLedgerEntry."Clip Quality Control Result", 'C01.P010.A013
Second ItemLedgerEntry."Clip Quality Control Result"');

AssertThat.AreEqual(PurchaseLine2."Clip Quality Control Result",
ItemLedgerEntry."Clip Quality Control Result", 'C01.P010.A024
Second ItemLedgerEntry."Clip Quality Control Result"');

AssertThat.AreEqual(PurchaseLine2."Clip Quality Control Result",
ItemLedgerEntry."Clip Quality Control Result", 'C01.P010.A034
Second ItemLedgerEntry."Clip Quality Control Result"');

```

### 3. Traspaso del Resultado del Control de Calidad del documento de compra a los movimientos de producto

#### 1. Escribe el siguiente código para crear y mostrar un campo en los movimientos de producto

```

tableextension 50102 "Clip Item Ledger Entry" extends "Item Ledger
Entry"
{
    fields
    {
        field(50100; "Clip Quality Control Result"; enum "Clip Quality
Control Result")
        {
            Caption = 'Quality Control Result', comment =
'ESP="Resultado Control de Calidad"';
            DataClassification = CustomerContent;
        }
    }
}

pageextension 50103 "Clip Item Ledger Entries" extends "Item Ledger
Entries"
{
    layout
    {
        addafter("Location Code")
        {
            field("Clip Quality Control Result"; "Clip Quality Control

```

```

Result")
    {
        ApplicationArea = Basic, Suite;
        Tooltip = 'Indicates the global result of the Quality
Control process', comment = 'ESP="Indica el resultado general del
control de calidad"';
    }
}
}
}

```

2. Escribe el siguiente código para traspasar la información del documento de compra al movimiento de producto

```

tableextension 50103 "Clip Item Journal Line" extends "Item Journal
Line"
{
    fields
    {
        field(50100; "Clip Quality Control Result"; enum "Clip Quality
Control Result")
        {
            Caption = 'Quality Control Result', comment =
'ESP="Resultado Control de Calidad"';
            DataClassification = CustomerContent;
        }
    }
}

[EventSubscriber(ObjectType::Table, Database::"Item Journal Line",
'OnAfterCopyItemJnlLineFromPurchLine', '', false, false)]
local procedure OnAfterCopyItemJnlLineFromPurchLine(var ItemJnlLine:
Record "Item Journal Line"; PurchLine: Record "Purchase Line");
begin
    ItemJnlLine."Clip Quality Control Result" := PurchLine."Clip
Quality Control Result";
end;

[EventSubscriber(ObjectType::Codeunit, Codeunit::"Item Jnl.-Post Line",
'OnAfterInitItemLedgEntry', '', false, false)]
local procedure OnAfterInitItemLedgEntry(varNewItemLedgEntry: Record
"Item Ledger Entry"; ItemJournalLine: Record "Item Journal Line"; var
ItemLedgEntryNo: Integer);
begin
   NewItemLedgEntry."Clip Quality Control Result" :=
ItemJournalLine."Clip Quality Control Result";
end;

```

3. Publica las modificaciones y ejecuta los tests para comprobar que todos ellos son satisfactorios

4. Haz un commit con las modificaciones realizadas hasta el momento

#### 4. Programación de vistas

1. Escribe el siguiente código en la pageextension 50103 "Clip Item Ledger Entries" para crear una vista que establezca un filtro para ver solo movimientos satisfactorios

```
views
{
    addfirst
    {
        view(Satisfactory)
        {
            Caption = 'Satisfactory', comment = 'ESP="Satisfactorio"';
            Filters = where("Clip Quality Control Result" =
const(Satisfactory));
        }
    }
}
```

2. Crea una nueva vista en la misma página que además modifique el layout de la página cuando el usuario seleccione la vista

```
view(ClipQCPending)
{
    Caption = 'QC Pending', comment = 'ESP="CC Pendiente"';
    Filters = where("Clip Quality Control Result" = const(Pending));
    SharedLayout = false;

    layout
    {
        modify("Invoiced Quantity") { Visible = false; }
        modify(Description) { Visible = false; }
        movefirst(Control1; "Entry No.")
    }
}
```

3. Publica las modificaciones y haz algunas pruebas

4. Haz un commit con las modificaciones realizadas hasta el momento

5. Haz un merge de tu rama de desarrollo con la rama principal

6. Elimina la rama de desarrollo