Desarrollo desde 0

Dynamics 365 Business Central

- La sesión empezará en breve -

ClipPlatform

Ayudas al desarrollo

ClipPlatform

- Instala la extensión de Visual Studio CRS AL Language Extension
- Configura la extensión para nombrar los archivos .al según los siguientes patrones

```
{
    "CRS.OnSaveAlFileAction": "Rename",
    "CRS.FileNamePattern": "<ObjectNameShort>.<ObjectTypeShortPascalCase>.al",
    "CRS.FileNamePatternExtensions": "<ObjectNameShort>.<ObjectTypeShortPascalCase>.al",
    "CRS.FileNamePatternPageCustomizations": "<ObjectNameShort>.<ObjectTypeShortPascalCase>.al",
    "CRS.ExtensionObjectNamePattern": "<Prefix><BaseName>",
    "CRS.ObjectNamePrefix": "CLIP ",
    "CRS.RemovePrefixFromFilename": true
}
```

Ayudas al desarrollo

ClipPlatform

- Renombra todos los archivos
- Publica la extensión con la opción
  `"schemaUpdateMode": "ForceSync“`
- Haz un commit con los cambios

Desarrollo – Datos Maestros

Se creará también una tabla para almacenar datos relativos a las Ediciones de los Cursos. La tabla de Ediciones de Cursos tendrá la siguiente estructura:

Campo	Tipo de Datos	Comentario
Course No.	Code 20	Clave Primaria. Debe tener relación con la tabla de cursos
Edition	Code 20	Clave Primaria
Start Date	Date	
Max. Students	Integer	

La tabla de Ediciones de Curso contará con una lista que será visible en la ficha del curso como subpágina, y en la lista de cursos como FactBox.

Desarrollo – Datos Maestros

```
table 50102 "CLIP Course Edition"
{
    CaptionML = ENU = 'Course Edition', ESP = 'Edición Curso';

    fields
    {
        4 references
        field(1; "Course No."; Code[20])
        {
            CaptionML = ENU = 'Course No.', ESP = 'Nº Curso';
        }
        2 references
        field(2; Edition; Code[20])
        {
            CaptionML = ENU = 'Edition', ESP = 'Edición';
        }
        1 reference
        field(10; "Start Date"; Date)
        {
            CaptionML = ENU = 'Start Date', ESP = 'Fecha Inicio';
        }
        1 reference
        field(20; "Max. Students"; Integer)
        {
            CaptionML = ENU = 'Max. Students', ESP = 'Máx. Alumnos';
        }
    }

    keys
    {
        - reference
        key(Key1; "Course No.", Edition)
        {
            Clustered = true;
        }
    }
}
```

```
page 50103 "CLIP Course Editions"
{
    CaptionML = ENU = 'Editions', ESP = 'Ediciones';
    DataCaptionFields = "Course No.";
    PageType = ListPart;
    SourceTable = "CLIP Course Edition";

    layout
    {
        0 references
        area(content)
        {
            0 references
            repeater(General)
            {
                0 references
                field(Edition; Rec.Edition)
                {
                    ApplicationArea = All;
                }
                0 references
                field("Start Date"; Rec."Start Date")
                {
                    ApplicationArea = All;
                }
                0 references
                field("Max. Students"; Rec."Max. Students")
                {
                    ApplicationArea = All;
                }
            }
        }
    }
}
```

Desarrollo – Datos Maestros

```
page 50101 "CLIP Course Card"
{
    CaptionML = ENU = 'Course', ESP = 'Curso';
    PageType = Card;
    UsageCategory = None;
    SourceTable = "CLIP Course";

    layout
    {
        area(Content)
        {
            group(General)
            {
                CaptionML = ENU = 'General', ESP = 'General';
                field("No."; Rec."No.") { ApplicationArea = All; }
                field(Name; Rec.Name) { ApplicationArea = All; }
            }
            group(Training)
            {
                CaptionML = ENU = 'Training', ESP = 'Formación';
                field(Type; Rec.Type) { ApplicationArea = All; }
                field("Duration (hours)"; Rec."Duration (hours)") { ApplicationArea = All; }
                field("Content Description"; Rec."Content Description") { ApplicationArea = All; }
                part(Editions; "CLIP Course Editions")
                {
                    ApplicationArea = All;
                    SubPageLink = "Course No." = field("No.");
                }
            }
            group(Invoicing)
            {
                CaptionML = ENU = 'Invoicing', ESP = 'Facturación';
                field(Price; Rec.Price) { ApplicationArea = All; }
            }
        }
    }
}
```

```
page 50100 "CLIP Course List"
{
    CaptionML = ENU = 'Courses', ESP = 'Cursos';
    PageType = List;
    ApplicationArea = All;
    UsageCategory = Lists;
    SourceTable = "CLIP Course";
    Editable = false;
    CardPageId = "CLIP Course Card";

    layout
    {
        area(Content)
        {
            repeater(RepeaterControl)
            {
                field("No."; Rec."No.") { ApplicationArea = All; }
                field(Name; Rec.Name) { ApplicationArea = All; }
                field("Content Description"; Rec."Content Description") { ApplicationArea = All; }
                field("Duration (hours)"; Rec."Duration (hours)") { ApplicationArea = All; }
                field(Price; Rec.Price) { ApplicationArea = All; }
                field(Type; Rec.Type) { ApplicationArea = All; }
            }
        }
        area(FactBoxes)
        {
            part(Editions; "CLIP Course Editions")
            {
                ApplicationArea = All;
                SubPageLink = "Course No." = field("No.");
            }
        }
    }
}
```

Desarrollo – Datos Maestros

ClipPlatform

- Haz un commit con los cambios

Ayudas al desarrollo

- Configura los analizadores de código en el archivo `settings.json`

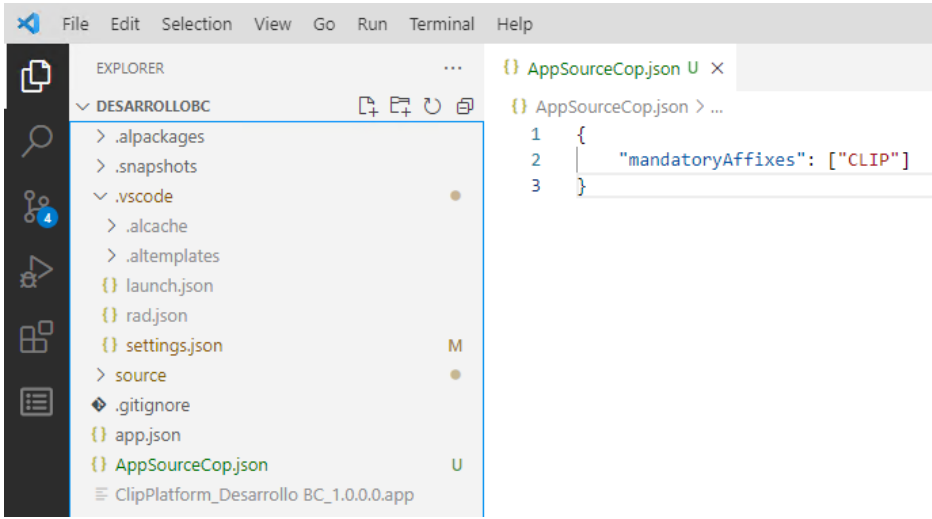
```
"al.enableCodeAnalysis": true,
"al.codeAnalyzers": [
    "${CodeCop}",
    "${UICop}",
    "${PerTenantExtensionCop}",
    "${AppSourceCop}"
]
```

- Resuelve todos los problemas que los analizadores de código detectan

https://docs.microsoft.com/en-us/dynamics365/business-central/dev-itpro/developer/devenv-using-code-analysis-tool

Resolución de problemas

- Crea el archivo `AppSourceCop.json` en la carpeta raíz de la extensión
- Indica CLIP como afijo



https://docs.microsoft.com/en-us/dynamics365/business-central/dev-itpro/developer/devenv-using-code-analysis-tool

Resolución de problemas

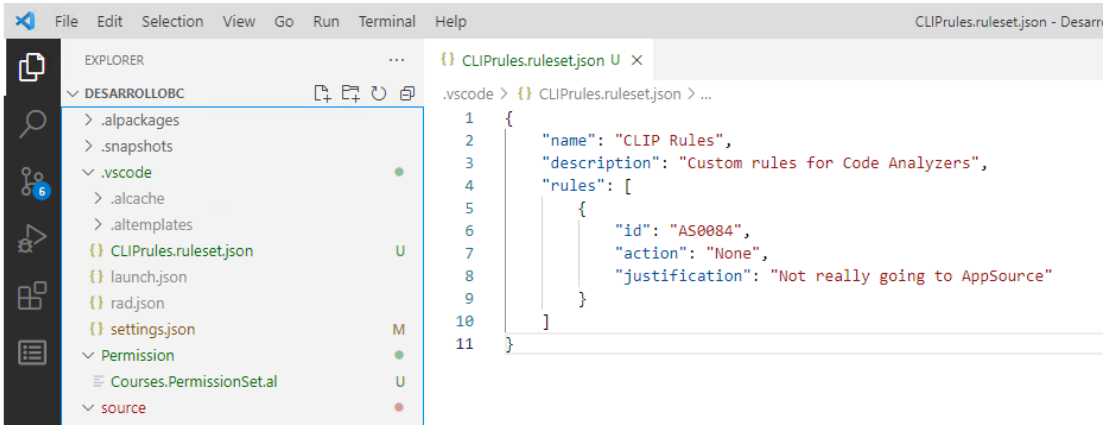
- Crea un objeto de permisos que de permisos totales sobre todos los objetos creados hasta el momento

```
permissionset 50100 "CLIP Courses"
{
    CaptionML = ENU = 'Courses - Edit', ESP = 'Cursos - Edición';
    Assignable = true;
    Permissions =
        tabledata "CLIP Courses Setup" = RIMD,
        tabledata "CLIP Course" = RIMD,
        tabledata "CLIP Course Edition" = RIMD,
        table "CLIP Courses Setup" = X,
        table "CLIP Course" = X,
        table "CLIP Course Edition" = X,
        page "CLIP Courses Setup" = X,
        page "CLIP Course List" = X,
        page "CLIP Course Card" = X,
        page "CLIP Course Editions" = X;
}
```

https://docs.microsoft.com/en-us/dynamics365/business-central/dev-itpro/developer/devenv-permissionset-object

Resolución de problemas

- Desactiva algunas de las normas de los analizadores de código
  - Crea un archivo CLIPrules.ruleset.json



  - Indica la ruta al archivo en el settings.json

```
"al.ruleSetPath": "./.vscode/CLIPrules.ruleset.json"
```

https://docs.microsoft.com/en-us/dynamics365/business-central/dev-itpro/developer/devenv-using-code-analysis-tool-with-rule-set

Resolución de problemas

- Activa la nueva sintaxis de traducciones en el archivo app.json

```
"features": ["TranslationFile"]
```

- Instala la extension AL Language Tools
- Utiliza la siguiente expresión regular para reemplazar la aparición de las propiedades ML por su equivalente con el nuevo modelo de traducciones
  - Buscar: (.*)ML = ENU = '(.*)', ESP = '(.*)';
  - Reemplazar: $1 = '$2', comment = 'ESP="$3"';

https://docs.microsoft.com/en-us/dynamics365/business-central/dev-itpro/developer/devenv-work-with-translation-files

Resolución de problemas

- Haz un commit con los cambios

Desarrollo – Venta de Cursos

ClipPlatform

DES02 Venta de Cursos

En un Documento de Venta de Business Central, un usuario puede seleccionar vender un Producto, un Recurso, un Activo Fijo, etc. Dicha selección se ampliará para permitir seleccionar vender un Curso.

Al seleccionar un curso en un documento de venta, el sistema traerá al documento la siguiente información adicional:

- El Nombre del curso
- El Precio Unitario del curso
- El Grupo Registro IVA Producto[*]
- El Grupo Registro Producto[*]

[*] Se deberá crear el campo en la tabla de Cursos

Después de seleccionar un Curso, el usuario deberá también seleccionar también la Edición. Para ello, se deberá crear un campo adicional en las Líneas de Venta.

Campo	Tipo de Datos	Comentario
Course Edition	Code 20	Debe tener relación con la tabla de ediciones de cursos, permitiendo seleccionar solo ediciones del curso en cuestión

Este mismo campo debe crearse también en las distintas tablas de documentos históricos de venta: factura, abono, albarán, albarán de devolución, archivo.

Al seleccionar un curso y edición, el sistema deberá comprobar si con las ventas previas más la venta actual, se ha llegado al número máximo de alumnos permitidos. En caso de haber superado el número máximo de alumnos, el sistema deberá notificarlo al usuario, pero permitirle efectuar la venta igualmente.

Desarrollo – Venta de Cursos

```
enumextension 50100 "CLIP Sales Line Type" extends "Sales Line Type"
{
    0 references
    value(50100; "CLIP Course")
    {
        Caption = 'Course', comment = 'ESP="Curso"';
    }
}
```

```
tableextension 50100 "CLIP Sales Line" extends "Sales Line"
{
    fields
    {
        modify("No.")
        {
            TableRelation = if (Type = const("CLIP Course")) "CLIP Course";
        }
    }
}
```

Codeunits y Eventos

ClipPlatform

- Utilizar el snippet `tcodeunit` para crear una codeunit
- Suscríbete a eventos lanzados en la tabla Sales Line

https://docs.microsoft.com/en-us/dynamics365/business-central/dev-itpro/developer/devenv-codeunit-object
https://docs.microsoft.com/en-us/dynamics365/business-central/dev-itpro/developer/devenv-events-in-al

Desarrollo – Venta de Cursos

```
codeunit 50100 "CLIP Course - Sales Management"
{
    [EventSubscriber(ObjectType::Table, Database::"Sales Line", 'OnAfterAssignFieldsForNo', '', false, false)]
    0 references
    local procedure CopyFromCourse(var SalesLine: Record "Sales Line")
    var
        Course: Record "CLIP Course";
    begin
        if SalesLine.Type <> SalesLine.Type::"CLIP Course" then
            exit;

        Course.Get(SalesLine."No.");
        Course.TestField("Gen. Prod. Posting Group");
        SalesLine.Description := Course.Name;
        SalesLine."Gen. Prod. Posting Group" := Course."Gen. Prod. Posting Group";
        SalesLine."VAT Prod. Posting Group" := Course."VAT Prod. Posting Group";
        SalesLine."Unit Price" := Course.Price;
    end;
}
```

Desarrollo – Venta de Cursos

ClipPlatform

```
1  table 50100 "CLIP Course"
53+        field(70; "Gen. Prod. Posting Group"; Code[20])
54+        {
55+            Caption = 'Gen. Prod. Posting Group', Comment = 'ESP="Grupo contable prod. gen."';
56+            TableRelation = "Gen. Product Posting Group";
57+            DataClassification = CustomerContent;
58+
59+            trigger OnValidate()
60+            var
61+                GenProdPostingGrp: Record "Gen. Product Posting Group";
62+            begin
63+                if xRec."Gen. Prod. Posting Group" <> "Gen. Prod. Posting Group" then
64+                    if GenProdPostingGrp.ValidateVatProdPostingGroup(GenProdPostingGrp, "Gen. Prod. Posting Group") then
65+                        Validate("VAT Prod. Posting Group", GenProdPostingGrp."Def. VAT Prod. Posting Group");
66+            end;
67+        }
68+        field(80; "VAT Prod. Posting Group"; Code[20])
69+        {
70+            Caption = 'VAT Prod. Posting Group', Comment = 'ESP="Grupo contable IVA Prod."';
71+            TableRelation = "VAT Product Posting Group";
72+            DataClassification = CustomerContent;
73+        }
```

Desarrollo – Venta de Cursos

ClipPlatform

```
1  page 50101 "CLIP Course Card"
```

```
58+             field("Gen. Prod. Posting Group"; Rec."Gen. Prod. Posting Group")
59+             {
60+                 ApplicationArea = Jobs;
61+                 Importance = Promoted;
62+                 ToolTip = 'Specifies the item''s product type to link transactions made for this item with the appropriate general
  +                 ledger account according to the general posting setup.', Comment = 'ESP="Especifica el tipo de producto del curso
  +                 para vincular las transacciones realizadas para este curso con la cuenta de contabilidad general correspondiente
  +                 según la configuración de registro general."';
63+             }
64+             field("VAT Prod. Posting Group"; Rec."VAT Prod. Posting Group")
65+             {
66+                 ApplicationArea = Basic, Suite;
67+                 Importance = Promoted;
68+                 ToolTip = 'Specifies the VAT specification of the involved item or resource to link transactions made for this
  +                 record with the appropriate general ledger account according to the VAT posting setup.', Comment = 'ESP="Indica la
  +                 especificación de IVA del elemento relacionado para vincular las transacciones realizadas para este registro con la
  +                 cuenta de contabilidad general correspondiente de acuerdo con la configuración de registro de IVA."';
69+             }
```

Desarrollo – Venta de Cursos

ClipPlatform

- Haz un commit con los cambios

Desarrollo – Venta de Cursos

Después de seleccionar un Curso, el usuario deberá también seleccionar también la Edición. Para ello, se deberá crear un campo adicional en las Líneas de Venta.

Campo	Tipo de Datos	Comentario
Course Edition	Code 20	Debe tener relación con la tabla de ediciones de cursos, permitiendo seleccionar solo ediciones del curso en cuestión

Este mismo campo debe crearse también en las distintas tablas de documentos históricos de venta: factura, abono, albarán, albarán de devolución, archivo.

- Mostrar el campo en las siguientes páginas:
  - Subformulario de pedidos de venta
  - Subformulario de facturas de venta
  - Subformulario de ofertas de venta
  - Subformulario de abonos de venta
  - Subformulario de pedidos de devolución
  - Todos los subformularios de documentos históricos (factura, abono, albarán, albarán de devolución, archivo)

Desarrollo – Venta de Cursos

ClipPlatform

```
field(50100; "CLIP Course Edition"; Code[20])
{
    Caption = 'Course Edition', comment = 'ESP="Edición Curso"';
    DataClassification = CustomerContent;
    TableRelation = "CLIP Course Edition".Edition where("Course No." = field("No."));
}
```

```
pageextension 50100 "CLIP Sales Order Subform" extends "Sales Order Subform"
{
    ...
    layout
    {
        addafter("No.")
        {
            field("CLIP Course Edition"; Rec."CLIP Course Edition")
            {
                ApplicationArea = All;
                ToolTip = 'Identifies the Edition of a Course when the Type is Course',
                    comment = 'ESP="Identifica la Edición de un curso cuando el Tipo es Curso"';
            }
        }
    }
}
```

Desarrollo – Venta de Cursos

ClipPlatform

- Haz un commit con los cambios

Desarrollo – Venta de Cursos

Al seleccionar un curso y edición, el sistema deberá comprobar si con las ventas previas más la venta actual, se ha llegado al número máximo de alumnos permitidos. En caso de haber superado el número máximo de alumnos, el sistema deberá notificarlo al usuario, pero permitirle efectuar la venta igualmente.

- Haremos este desarrollo más adelante, después de DES03 Registro de Cursos

Testing – Venta de Cursos

ClipPlatform

- Durante el Desarrollo hemos probado:
  - Que se puede seleccionar un curso en una línea de venta
  - Que el sistema rellena la descripción, los grupos contables y el precio de la línea de venta
  - Que se puede registrar un documento de venta con una línea de tipo Curso
  - Que el sistema ha realizado los asientos contables adecuados
  - Que en los documentos registrados, el campo Edición se ha rellenado
- Vamos a realizar las mismas pruebas pero de forma automatizada

Testing – Preparación

ClipPlatform

- Creación de una extensión de Test
  - Crea una subcarpeta MainApp y mueve el actual proyecto AL a la nueva carpeta
  - Crea una nueva extension TestApp con el comando AL: Go!
    - Modifica el archivo app.json para utilizar el rango 50140..50149
    - Copia los archivos launch.json, settings.json, CLIPrules.ruleset.json y AppSourceCop.json de la MainApp
  - Descarga los símbolos de la BaseApp
  - Crea y guarda un workspace con ambas extensiones

Testing - Ejemplo

CoursesTest.Codeunit.al U

AL Page Designer

TestApp > source > CoursesTest.Codeunit.al > Codeunit 50140 "CLIP Courses Test" > Test001()

```
0 references
1  codeunit 50140 "CLIP Courses Test"
2  {
3      Subtype = Test;
4
5      [Test]
       0 references
6      procedure Test001()
7      var
8          CLIPMin: Codeunit "CLIP Min";
9          Value1, Value2 : Decimal;
10         Result: Decimal;
11     begin
12         // [Scenario] Una función llamada GetMin devuelve el mínimo de 2 valores numéricos
13
14         // [Given] Setup: 2 valores numéricos
15         Value1 := 1;
16         Value2 := 2;
17
18         // [When] Exercise: Llamada a la función GetMin
19         Result := CLIPMin.GetMin(Value1, Value2);
20
21         // [Then] Verify: El resultado es Value1
22         if Result <> value1 then
23             Error('El resultado no es correcto');
24     end;
25 }
```

Min.Codeunit.al U

TestApp > source > Min.Codeunit.al > Codeunit 50141 "CLIP Min" > GetMin(v1: Decimal, v2: Decimal): D

```
2 references
1  codeunit 50141 "CLIP Min"
2  {
       2 references
3      procedure GetMin(v1: Decimal; v2: Decimal): Decimal
4      begin
5          if v1 <= v2 then
6              exit(v1);
7
8          exit(v2)
9      end;
10 }
```

https://docs.microsoft.com/en-us/dynamics365/business-central/dev-itpro/developer/devenv-testing-application

Testing – Preparación

ClipPlatform

- En el app.json, pon dependencias a las extensiones:
  - Library Assert
  - Tests-TestLibraries
  - Desarrollo BC

Testing – Venta de Cursos

- Realiza un test en el que se seleccione un curso en una línea de venta y se compruebe que el Sistema ha rellenado la descripción, los grupos contables y el precio

```
codeunit 50140 "CLIP Course Test"
{
    Subtype = Test;

    [Test]
    0 references
    procedure SelectingACourseOnASalesLineP001();
    begin
        // [Scenario] When selecting a course on a Sales Line document, the system fills in related data

        // [Given] Setup: A course and a Sales document with a Sales Line

        // [When] Exercise: Select the course on the sales line

        // [Then] Verify: The Sales Line has the correct Description, Posting Groups and Price
    end;

    var
        0 references
        LibraryAssert: Codeunit "Library Assert";    Variable 'LibraryAssert' is unused in '"CLIP Course T
}
```

Testing – Venta de Cursos

```al
codeunit 50140 "CLIP Course Test"
{
    Subtype = Test;

    [Test]
    0 references
    procedure SelectingACourseOnASalesLineP001();
    var
        Course: Record "CLIP Course";
        SalesHeader: Record "Sales Header";
        SalesLine: Record "Sales Line";
        LibraryCourse: Codeunit "CLIP Library - Course";
        LibrarySales: Codeunit "Library - Sales";
    begin
        // [Scenario] When selecting a course on a Sales Line document, the system fills in related data

        // [Given] Setup: A course and a Sales document with a Sales Line
        Course := LibraryCourse.CreateCourse();

        LibrarySales.CreateSalesHeader(SalesHeader, "Sales Document Type"::Order, '');
        LibrarySales.CreateSalesLineSimple(SalesLine, SalesHeader);

        // [When] Exercise: Select the course on the sales line
        SalesLine.Validate(Type, "Sales Line Type"::"CLIP Course");
        SalesLine.Validate("No.", Course."No.");

        // [Then] Verify: The Sales Line has the correct Description, Posting Groups and Price
        LibraryAssert.AreEqual(Course.Name, SalesLine.Description, 'Incorrect Description');
        LibraryAssert.AreEqual(Course.Price, SalesLine."Unit Price", 'Incorrect Price');
        LibraryAssert.AreEqual(Course."Gen. Prod. Posting Group", SalesLine."Gen. Prod. Posting Group", 'Incorrect Posting Group');
        LibraryAssert.AreEqual(Course."VAT Prod. Posting Group", SalesLine."VAT Prod. Posting Group", 'Incorrect Posting Group');
    end;

    var
        4 references
        LibraryAssert: Codeunit "Library Assert";
}
```

Testing – Venta de Cursos

- Realiza un test en el que se compruebe que la Edición del Curso seleccionada en un documento de venta se ha guardado correctamente en los documentos registrados

```
[Test]
0 references
procedure CourseSalesPostingP002()
var
    Course: Record "CLIP Course";
    CourseEdition: Record "CLIP Course Edition";
    SalesHeader: Record "Sales Header";
    SalesLine: Record "Sales Line";
    SalesInvoiceLine: Record "Sales Invoice Line";
    LibraryCourse: Codeunit "CLIP Library - Course";
    LibrarySales: Codeunit "Library - Sales";
    PostedDocumentNo: Code[20];
begin
    // [Scenario] when posting a the sale of a course and edition, the edition is saved on the posted documents

    // [Given] Setup: A course with Edition
    Course := LibraryCourse.CreateCourse();
    CourseEdition := LibraryCourse.CreateEdition(Course);

    // [Given] Setup: A Sales Document for the Course and Edition
    LibrarySales.CreateSalesHeader(SalesHeader, "Sales Document Type"::Order, '');
    LibrarySales.CreateSalesLine(SalesLine, SalesHeader, "Sales Line Type"::"CLIP Course", Course."No.", 1);
    SalesLine.Validate("CLIP Course Edition", CourseEdition.Edition);
    SalesLine.Modify(true);

    // [When] Exercise: Post the Sales Document
    PostedDocumentNo := LibrarySales.PostSalesDocument(SalesHeader, true, true);

    // [Then] Verify: The Edition is filled in on the posted document
    SalesInvoiceLine.SetRange("Document No.", PostedDocumentNo);
    SalesInvoiceLine.FindFirst();
    LibraryAssert.AreEqual(CourseEdition.Edition, SalesInvoiceLine."CLIP Course Edition", 'Edition is not correct on Sales Invoice Line');
end;
```

Testing – Venta de Cursos

```
codeunit 50141 "CLIP Library - Course"
{
    var
        LibraryERM: Codeunit "Library - ERM";
        LibraryRandom: Codeunit "Library - Random";
        LibraryUtility: Codeunit "Library - Utility";

   local procedure CourseNoSeriesSetup()
    var
        CoursesSetup: Record "CLIP Courses Setup";
        NoSeriesCode: Code[20];
    begin
        if not CoursesSetup.Get() then
            CoursesSetup.Insert();
        NoSeriesCode := LibraryUtility.GetGlobalNoSeriesCode();
        if NoSeriesCode <> CoursesSetup."Course No." then begin
            CoursesSetup.Validate("Course No.", LibraryUtility.GetGlobalNoSeriesCode());
            CoursesSetup.Modify(true);
        end;
    end;



}
```

```
procedure CreateCourse() Course: Record "CLIP Course";
var
    GeneralPostingSetup: Record "General Posting Setup";
    VATPostingSetup: Record "VAT Posting Setup";
begin
    CourseNoSeriesSetup();
    LibraryERM.FindGeneralPostingSetupInvtFull(GeneralPostingSetup);
    LibraryERM.FindVATPostingSetupInvt(VATPostingSetup);

    Course.Insert(true);
    Course.Validate(Name, LibraryRandom.RandText(MaxStrLen(Course.Name)));
    Course.Validate(Price, LibraryRandom.RandDecInRange(1, 1000, 2));

    Course.Validate("Gen. Prod. Posting Group", GeneralPostingSetup."Gen. Prod. Posting Group");
    Course.Validate("VAT Prod. Posting Group", VATPostingSetup."VAT Prod. Posting Group");
    Course.Modify(true);
end;

procedure CreateEdition(Course: Record "CLIP Course") CourseEdition: Record "CLIP Course Edition"
begin
    CourseEdition.Init();
    CourseEdition.Validate("Course No.", Course."No.");
    CourseEdition.Validate(Edition, LibraryRandom.RandText(MaxStrLen(CourseEdition.Edition)));
    CourseEdition.Validate("Start Date", LibraryRandom.RandDateFrom(Today(), 90));
    CourseEdition.Validate("Max. Students", LibraryRandom.RandIntInRange(1, 10));
    CourseEdition.Insert(true);
end;
```