

# **Desarrollo desde 0**

**Dynamics 365 Business Central**

**- La sesión empezará en breve -**

*ClipPlatform*

# 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

# Campos especiales

- FlowFields
- FlowFilters
- SumIndexFields

# FlowFields

- Crea un campo FlowField en la tabla de Ediciones de curso con la siguiente fórmula de calculo:
  - Suma del campo Quantity de la tabla “CLIP Course Ledger Entry”
  - Cuando el campo “Course No.” es igual que el valor del campo “Course No.” y “Course Edition” es igual que el valor del campo Edition

```
field(30; "Sales (Qty.)"; Decimal)
{
    Caption = 'Sales (Qty.)', comment = 'ESP="Ventas (cdad.)"';
    FieldClass = FlowField;
    CalcFormula = sum("CLIP Course Ledger Entry".Quantity where(
        "Course No." = field("Course No."),
        "Course Edition" = field(Edition)
    ));
}
```

- Muestra el campo en la lista de Ediciones de curso

# FlowFilters

- Crea un campo FlowFilter de tipo Date en la tabla de Ediciones de curso
- Añádelo a la fórmula de calculo del FlowField

# 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.

```
[EventSubscriber(ObjectType::Table, Database::"Sales Line", 'OnAfterValidateEvent', 'Quantity', false, false)]
local procedure OnAfterValidateQuantity_CheckSalesForCourseEdition(var Rec: Record "Sales Line")
begin
    CheckSalesForCourseEdition(Rec);
end;

[EventSubscriber(ObjectType::Table, Database::"Sales Line", 'OnAfterValidateEvent', 'CLIP Course Edition', false, false)]
local procedure OnAfterValidateEdition_CheckSalesForCourseEdition(var Rec: Record "Sales Line")
begin
    CheckSalesForCourseEdition(Rec);
end;

local procedure CheckSalesForCourseEdition(var SalesLine: Record "Sales Line")
var
    CourseEdition: Record "CLIP Course Edition";
    MaxStudentsExceededMsg: Label 'The current sale for course %1 edition %2 will exceed the maximum number of students %3',
        comment = 'ESP="La venta actual para el curso %1 edición %2 superará el número máximo de alumnos %3"';
begin
    if SalesLine.Type <> Enum::"Sales Line Type"::"CLIP Course" then
        exit;
    if (SalesLine."No." = '') or (SalesLine."CLIP Course Edition" = '') then
        exit;

    CourseEdition.Get(SalesLine."No.", SalesLine."CLIP Course Edition");
    CourseEdition.CalcFields("Sales (Qty.)");
    if (CourseEdition."Sales (Qty.)" + SalesLine.Quantity) > CourseEdition."Max. Students" then
        Message(MaxStudentsExceededMsg, SalesLine."No.", SalesLine."CLIP Course Edition", CourseEdition."Max. Students");
end;
```