

Proyecto Final

Creación de una aplicación, donde en la pantalla inicial se muestran todos los proveedores disponibles:

- En esta primera pantalla Agregar al menos un filtro de proveedor, que permita filtrar tanto por ID de proveedor como por nombre. (Si encuentra más filtros por implementar es un extra)
- Usar ya sea tabla **sap.m.table** o **sap.ui.table**
- Estas dos tipos de tabla permite ordenar por cada columna, investigar sobre como implementar el ordenamiento en la tabla que vayan usar (Tip: es más fácil la implementación del ordenamiento en tablas sap.ui.table)
- Respetar los patrones de diseño de Fiori. Filtros en la cabecera y tabla en la parte inferior.
- Generar la navegación a una vista de detalle.

En la vista de detalle generar un form donde se muestre más información del proveedor. Si existen campos que estén relacionados con otras entidades, por ejemplo si el proveedor tiene un campo país, este va a contener el código del país, la idea es tener el nombre del país para visualizar en el detalle (Esto es solo un ejemplo, pero si existe algún atributo del proveedor donde esto se pueda aplicar)

En esta vista detalle se deben mostrar los materiales que sean de este proveedor en una tabla. Acá se puede usar smarttable o cualquiera de las otras tablas que hemos usado.

Al dar Click en alguno de los materiales se debe abrir un Dialog(**sap.m.Dialog**), que va incluido dentro de una definición de un Fragment y mostrar en un form algunos datos del material seleccionado.

Ejemplo de definición del Fragment:

```
JavaScript
<core:FragmentDefinition
    xmlns="sap.m"
    xmlns:l="sap.ui.layout"
    xmlns:f="sap.ui.layout.form"
    xmlns:core="sap.ui.core">
    <Dialog
        id="NewItemDialog"
        title="{i18n>newItemDialogTitle}"
        icon="sap-icon://activity-items"
        busy="false"
        busyIndicatorDelay="0"
        class="sapUiSizeCompact">
        <content>
            <f:SimpleForm editable="true" id="ListForm" layout="ResponsiveGridLayout"
```

```

        labelSpanXL="4" labelSpanL="4" labelSpanM="4" labelSpanS="4"
adjustLabelSpan="false"
        columnsXL="1" columnsL="1" columnsM="1"
        singleContainerFullSize="false" width="30rem">
<f:content>
    <Label labelFor="ValidFrom" text="{i18n>ValidFrom}" required="true"/>
    <DatePicker
        id="ValidFrom"
        dateValue="{oModel>/ValidFrom}"
        displayFormat="long"
        enabled="true"
        class="sapUiSmallMarginBottom"/>
</f:content>
</f:SimpleForm>
</content>
</Dialog>
</core:FragmentDefinition>

```

Investigar cómo agregar botones en el Fragment para para que desde un Click se ejecute una función para cerrar el Dialogo.

Agregar un botón que permita la simulación de creación de un nuevo material(Podrían reutilizar el Dialog anterior, validando un estado de Visualización y otro de Creación para habilitar o no la modificación de los campos) y abra un Dialog([sap.m.Dialog](#)), que va incluido dentro de una definición de un Fragment y mostrar en un form los principales campos para la creación del material(Incluir Combo Box para los campos que apliquen), se deben tener 2 botones que permitan la opción de creación(En realidad no van a crear pero simulan la creación, pero lo que si deben es validar que los campos que definan como obligatorios estén diligenciados sino mostrar un mensaje que le informe el error al usuario) y un botón para cancelar la creación que debe cerrar el Dialog:

JavaScript

```

<core:FragmentDefinition
    xmlns="sap.m"
    xmlns:l="sap.ui.layout"
    xmlns:f="sap.ui.layout.form"
    xmlns:core="sap.ui.core">
<Dialog
    id="NewItemDialog"
    title="{i18n>newItemDialogTitle}"
    icon="sap-icon://activity-items"
    busy="false"
    busyIndicatorDelay="0"
    class="sapUiSizeCompact">
<content>
    <f:SimpleForm editable="true" id="ListForm"
layout="ResponsiveGridLayout"

```

```
labelSpanM="4" labelSpanS="4" adjustLabelSpan="false"
columnsM="1"
width="30rem">
    <f:content>
        </f:content>
    </f:SimpleForm>
</content>
<beginButton>
    <Button visible="true" text="{i18n>newItemDialogCreateBtn}"
type="Emphasized" press=".onPressSaveData" />
</beginButton>

<endButton>
    <Button text="{i18n>newItemDialogCloseButton}"
press=".onCloseAddItemDialog"/>
</endButton>
</Dialog>
</core:FragmentDefinition>
```

Para este ejercicio puede usar cualquiera de los dos servicios que se han usado: Northwind o ES5.

Cualquier control adicional que quieran implementar es bienvenido.