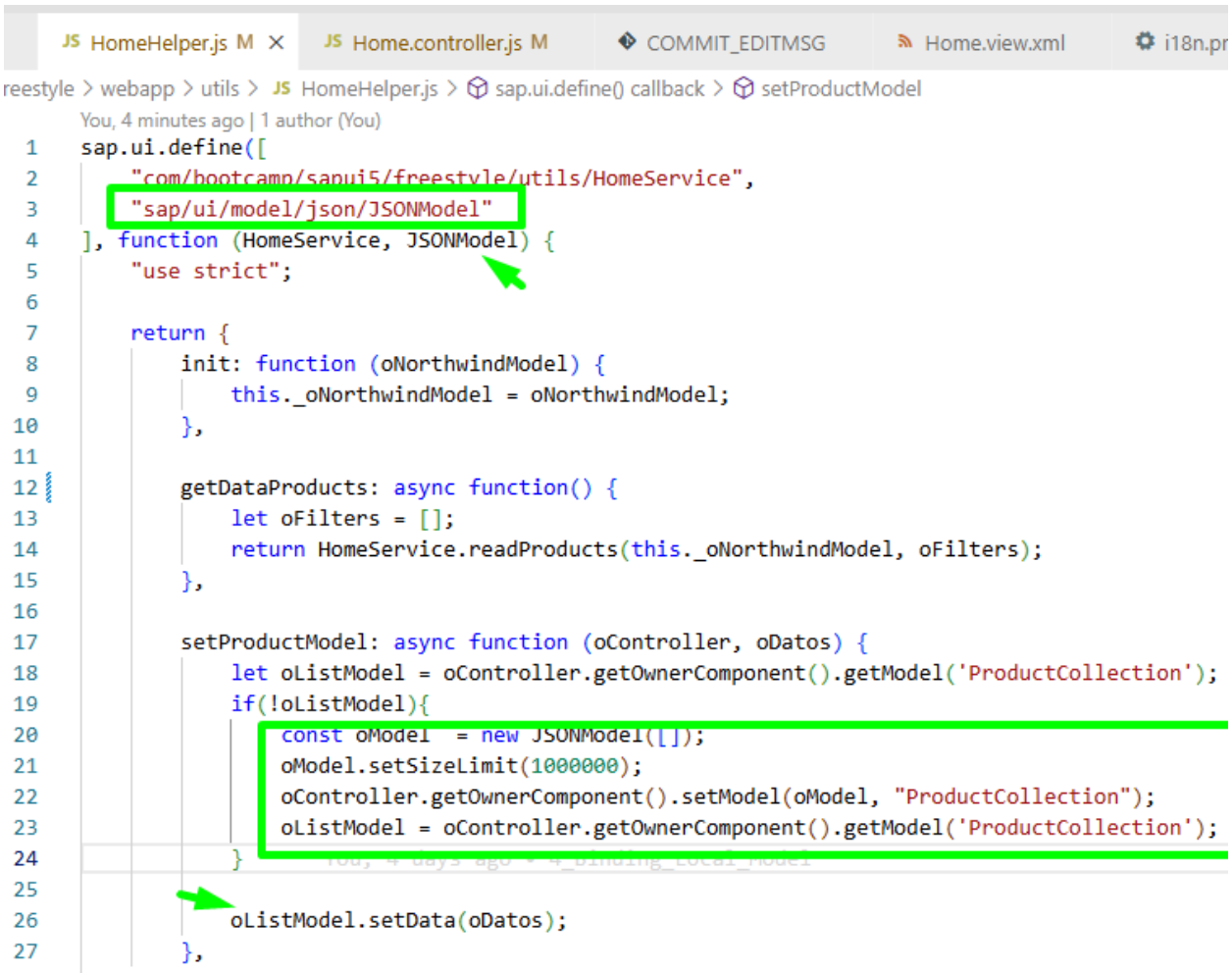
Solicitud de Productos (Parte 2)

Ahora vamos a crear un modelo local para asignar la lista de productos previamente consultados desde el botón Buscar.

Agregamos la función **setProductModel** en el Helper **HomeHelper.js:**



En este punto usamos la librería **JSONModel("sap/ui/model/json/JSONModel")**

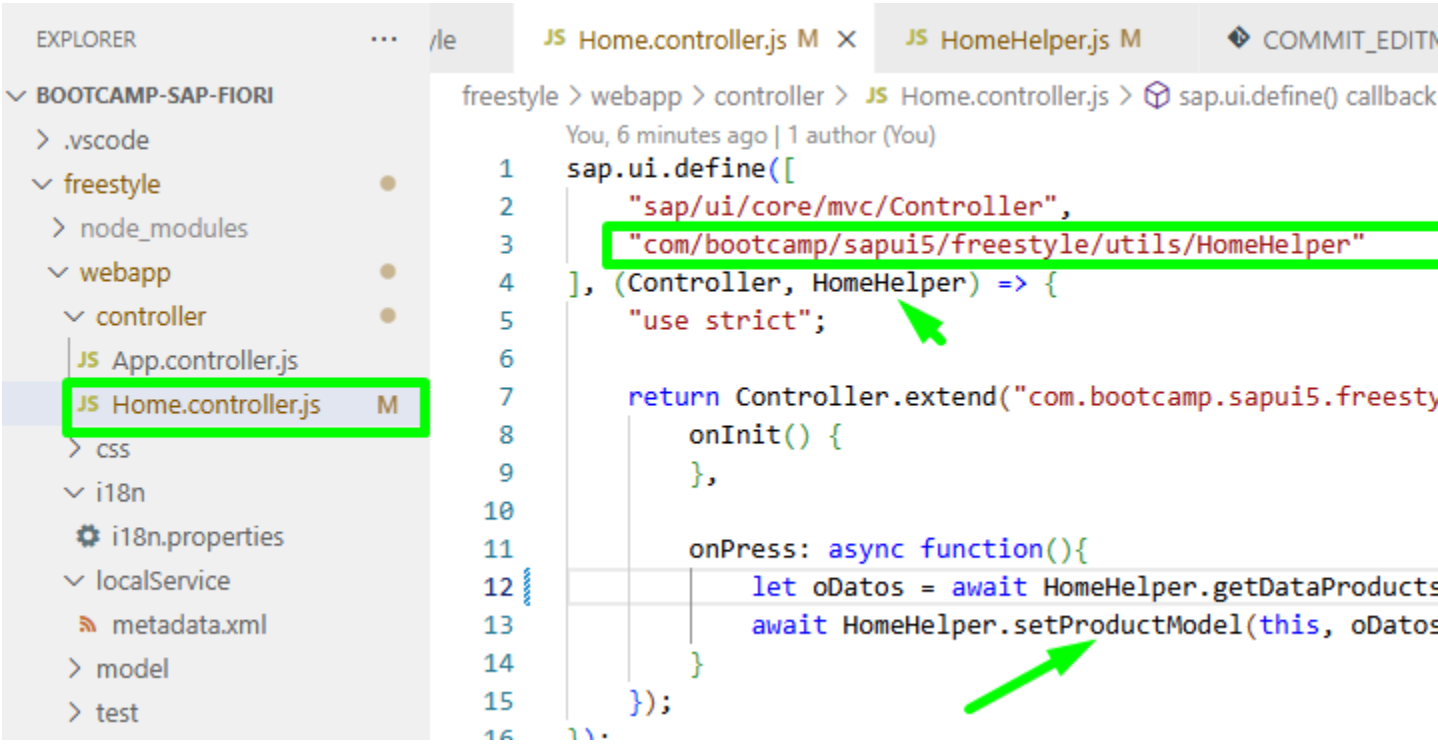
JavaScript

```
        setProductModel: async function (oController, oDatos) {
            let oListModel =
oController.getOwnerComponent().getModel('ProductCollection');
```

```
            if(!oListModel){
                const oModel  = new JSONModel([]);
                oModel.setSizeLimit(1000000);
                oController.getOwnerComponent().setModel(oModel, "ProductCollection");
                oListModel =
oController.getOwnerComponent().getModel('ProductCollection');
            }

            oListModel.setData(oDatos);
        },
```

Ahora Invocamos la función desde el controlador de la vista **Home.controller.js:**



Ahora agregamos en la vista Home una tabla sap.m.table:

A esta tabla le asignamos en la propiedad items el modelo local creado previamente (ProductCollection)

Home.view.xml M | HomeHelper.js M | i18n.properties M | metadata.xml | App.vi

freestyle > webapp > view > Home.view.xml

```
1   <mvc:View controllerName="com.bootcamp.sapui5.freestyle.controller.Home"
4       <Page id="page" title="{i18n>title}">
11          <VBox id="vTest2">
12
13              <Table id="idProductsTable"
14                  inset="false"
15                  items="{
16                      path: 'ProductCollection>/'
17                  }">
18                  <columns>
19                      <Column id="ID_01">
20                          <Text id="ID_T01" text="{i18n>Product}" />
21                      </Column>
22                      <Column id="ID_02">
23                          <Text id="ID_T05" text="{i18n>UnitPrice}" />        You, 3
24                      </Column>
25                      <Column id="ID_05">
26                          <Text id="ID_T04" text="{i18n>UnitsInStock}" />
27                      </Column>
28                      <Column id="ID_03">
29                          <Text  id="ID_T02" text="{i18n>QuantityPerUnit}" />
30                      </Column>
```

Se hace el binding con las diferentes columnas configuradas así:

Home.view.xml M | HomeHelper.js M | i18n.properties M | metadata.xml | App.view.x

freestyle > webapp > view > Home.view.xml

```
1   <mvc:View controllerName="com.bootcamp.sapui5.freestyle.controller.Home"
4       <Page id="page" title="{i18n>title}">
11          <VBox id="vTest2">
13              <Table id="idProductsTable"
32                  <items>
33                      <ColumnListItem id="ID_06" vAlign="Middle">
34                          <cells>
35                              <ObjectIdentifier id="ID_07"
36                                  title="{ProductCollection>ProductID}"
37                                  text="{ProductCollection>ProductName}"/>
38                              <ObjectNumber id="ID_11"
39                                  number="{
40                                      parts:[{path:'ProductCollection>UnitPrice'},{pat
41                                      type: 'sap.ui.model.type.Currency',
42                                      formatOptions: {showMeasure: true}
43                                  }"
44                                  unit="USD" />
45                              <Text id="ID_09"
46                                  text="{ProductCollection>UnitsInStock}" />
47                              <Text id="ID_08"
48                                  text="{ProductCollection>QuantityPerUnit}" />
49                          </cells>
50                      </ColumnListItem>
51                  </items>
52              </Table>
53          </VBox>
54      </Page>
55  </mvc:View>
```

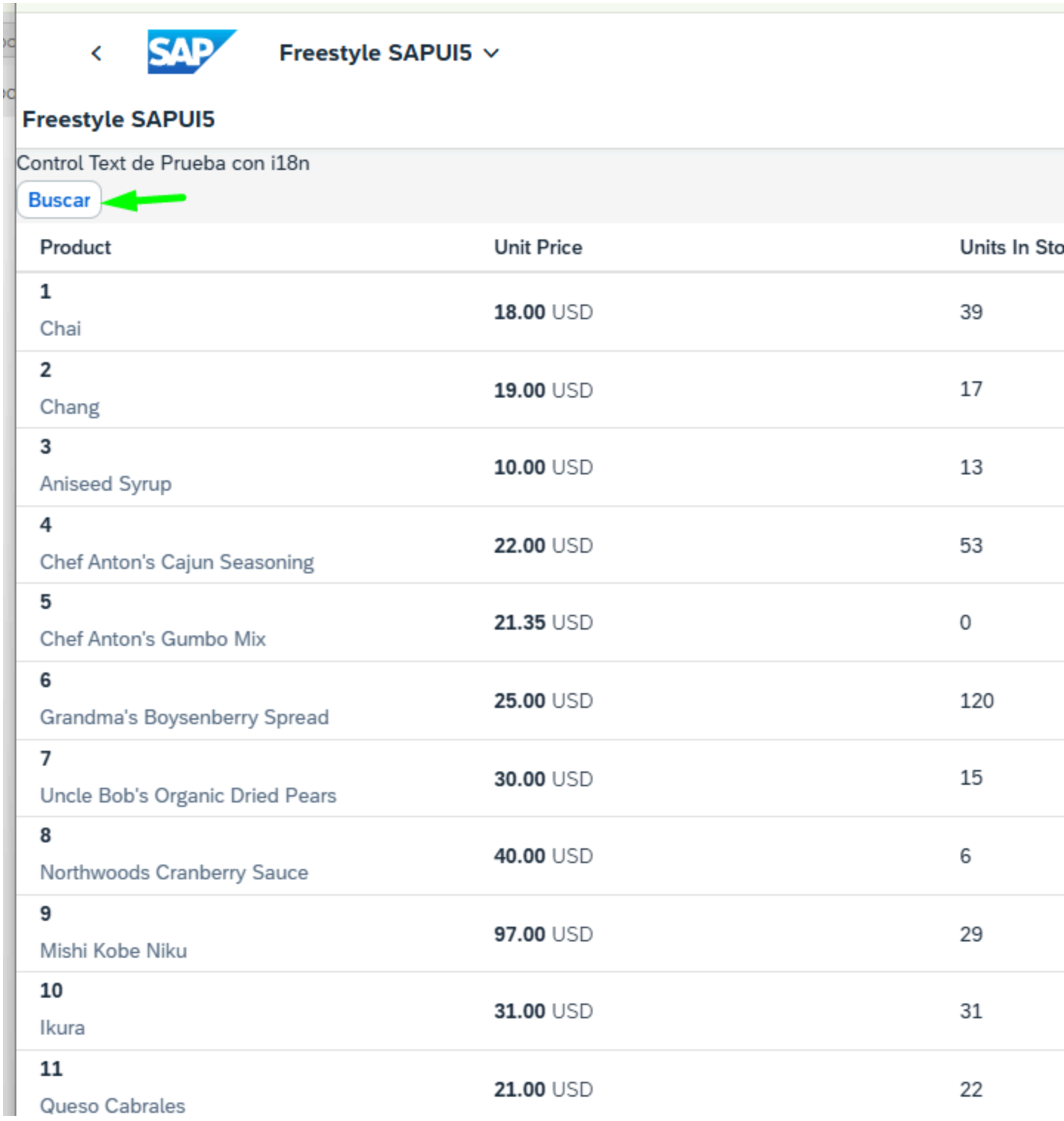
Java

```
<Table id="idProductsTable"
    inset="false"
    items="{
        path: 'ProductCollection>/'
    }">
    <columns>
        <Column id="ID_01">
            <Text id="ID_T01" text="{i18n>Product}" />
```

```
                    </Column>
                    <Column id="ID_02">
                        <Text id="ID_T05" text="{i18n>UnitPrice}" />
                    </Column>
                    <Column id="ID_05">
                        <Text id="ID_T04" text="{i18n>UnitsInStock}" />
                    </Column>
                    <Column id="ID_03">
                        <Text  id="ID_T02" text="{i18n>QuantityPerUnit}" />
                    </Column>
                </columns>
                <items>
                    <ColumnListItem id="ID_06" vAlign="Middle">
                        <cells>
                            <ObjectIdentifier id="ID_07"
                                title="{ProductCollection>ProductID}"
                                text="{ProductCollection>ProductName}"/>
                            <ObjectNumber id="ID_11"
                                number="{

parts:[{path:'ProductCollection>UnitPrice'},{path:''}],
                                    type: 'sap.ui.model.type.Currency',
                                    formatOptions: {showMeasure: true}
                                }"
                                unit="USD" />
                            <Text id="ID_09"
                                text="{ProductCollection>UnitsInStock}" />
                            <Text id="ID_08"
                                text="{ProductCollection>QuantityPerUnit}" />
                        </cells>
                    </ColumnListItem>
                </items>
            </Table>
        </VBox>
```

Ejecutamos la aplicación y damos click en el Botón Buscar:


Freestyle SAPUI5
Freestyle SAPUI5
Control Text de Prueba con i18n
Buscar
Product
Unit Price
Units In Sto
1
Chai
18.00 USD
39
2
Chang
19.00 USD
17
3
Aniseed Syrup
10.00 USD
13
4
Chef Anton's Cajun Seasoning
22.00 USD
53
5
Chef Anton's Gumbo Mix
21.35 USD
0
6
Grandma's Boysenberry Spread
25.00 USD
120
7
Uncle Bob's Organic Dried Pears
30.00 USD
15
8
Northwoods Cranberry Sauce
40.00 USD
6
9
Mishi Kobe Niku
97.00 USD
29
10
Ikura
31.00 USD
31
11
Queso Cabrales
21.00 USD
22

Fragment

Un Fragment en SAPUI5 es una unidad reutilizable de interfaz de usuario que se utiliza para crear componentes de UI de manera modular. Los **fragments** permiten definir partes de una vista (view) que pueden ser reutilizadas en diferentes vistas o en el mismo contexto sin necesidad de duplicar código.

Características de los Fragments:

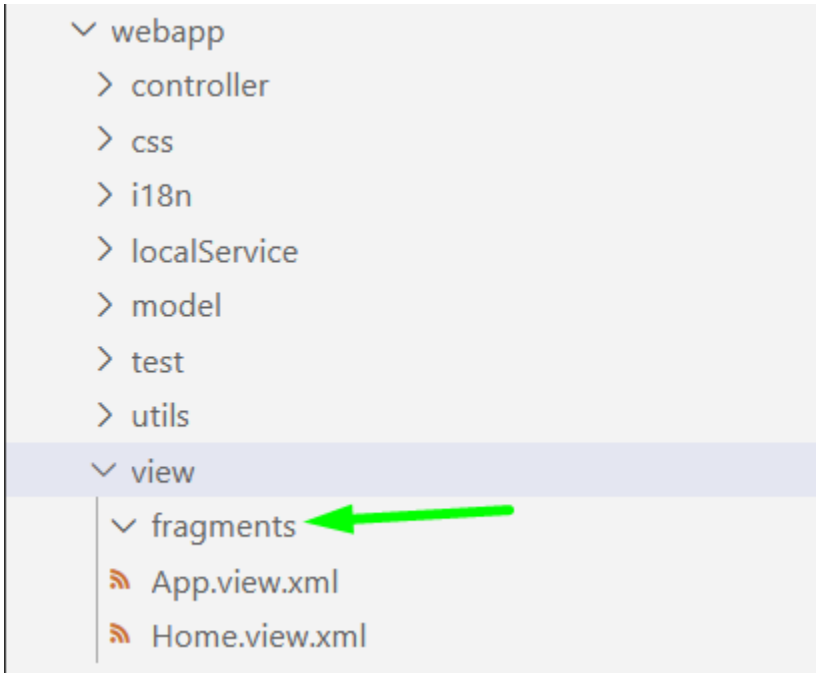
- **Reutilización**: Los fragments pueden ser utilizados en múltiples vistas, lo que ayuda a mantener el código limpio y DRY (Don't Repeat Yourself).

- **No Tienen Controlador Propio:** A diferencia de las vistas, los **fragments** no tienen su propio controlador. En su lugar, utilizan el controlador de la vista que los incluye.

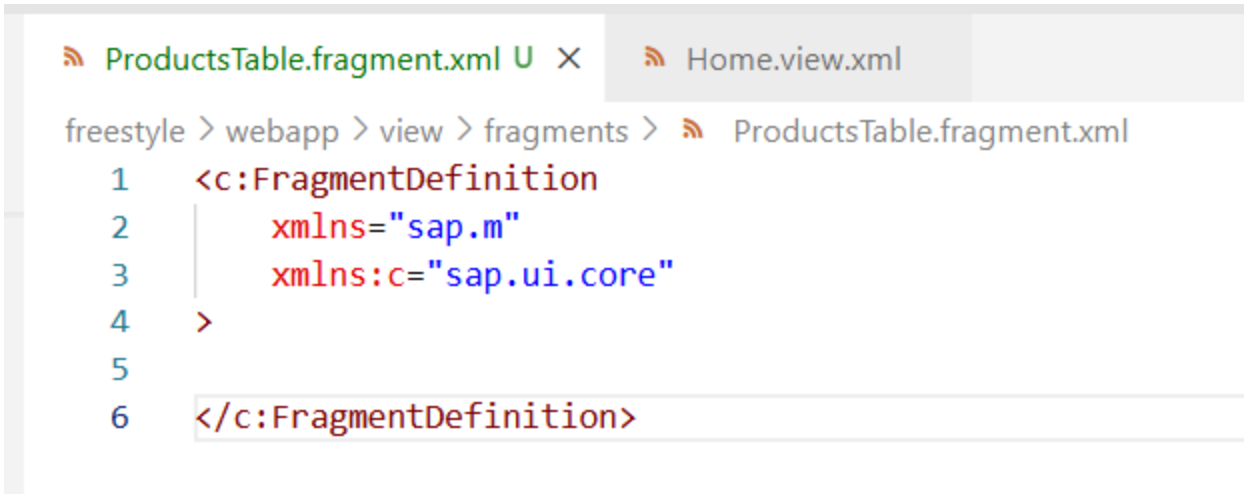
- **Definición en XML o HTML:** Pueden ser definidos en archivos XML o HTML, lo que permite una fácil integración con el resto de la aplicación.

Apliquemos a nuestro proyecto:

Creamos una carpeta llamada **fragments** en la ruta **webapp/view/fragments**



- Creamos un nuevo archivo **ProductsTable.fragment.xml** y agregamos la definición inicial del fragment así:



JavaScript

```
<c:FragmentDefinition
    xmlns="sap.m"
    xmlns:c="sap.ui.core"
>

</c:FragmentDefinition>
```

- Ahora copiamos todo el contenido de la definición de la tabla de la vista Home al **fragment** que acabamos de definir:

JavaScript

```
<c:FragmentDefinition
    xmlns="sap.m"
    xmlns:c="sap.ui.core"
>
    <Table id="idProductsTable"
        inset="false"
        items="{
            path: 'ProductCollection>/'
        }">
        <columns>
            <Column id="ID_01">
                <Text id="ID_T01" text="{i18n>Product}" />
            </Column>
```

```
            <Column id="ID_02">
                <Text id="ID_T05" text="{i18n>UnitPrice}" />
            </Column>
            <Column id="ID_05">
                <Text id="ID_T04" text="{i18n>UnitsInStock}" />
            </Column>
            <Column id="ID_03">
                <Text  id="ID_T02" text="{i18n>QuantityPerUnit}" />
            </Column>
        </columns>
        <items>
            <ColumnListItem id="ID_06" vAlign="Middle">
                <cells>
                    <ObjectIdentifier id="ID_07"
                        title="{ProductCollection>ProductID}"
                        text="{ProductCollection>ProductName}"/>
                    <ObjectNumber id="ID_11"
                        number="{

parts:[{path:'ProductCollection>UnitPrice'},{path:''}],
                            type: 'sap.ui.model.type.Currency',
                            formatOptions: {showMeasure: true}
                        }"
                        unit="USD" />
                    <Text id="ID_09"
                        text="{ProductCollection>UnitsInStock}" />
                    <Text id="ID_08"
                        text="{ProductCollection>QuantityPerUnit}" />
                </cells>
            </ColumnListItem>
        </items>
    </Table>
</c:FragmentDefinition>
```

La definición del Fragment queda ya con toda la adición de la tabla [sap.ui.table]

Ahora vamos modificar nuestra vista **Home** para que se cargue el fragment que definimos así:

JavaScript

```
<mvc:View controllerName="com.bootcamp.sapui5.freestyle.controller.Home"
    xmlns:mvc="sap.ui.core.mvc"
    xmlns:c="sap.ui.core"
    xmlns="sap.m">
    <Page id="page" title="{i18n>title}">

        <VBox id="vTest">
            <Text id="textPrueba" text="{i18n>textPruebai18n}"></Text>
            <Button id="btnBuscar" text="{i18n>textBuscar}"
press="onPress" />
        </VBox>

        <VBox id="vTest2">
```

```
            <c:Fragment
        fragmentName="com.bootcamp.sapui5.freestyle.view.fragments.ProductsTable"
                    type="XML"
                />
            </VBox>
        </Page>
    </mvc:View>
```

Al ejecutar la aplicación se presenta de la misma manera al usuario: