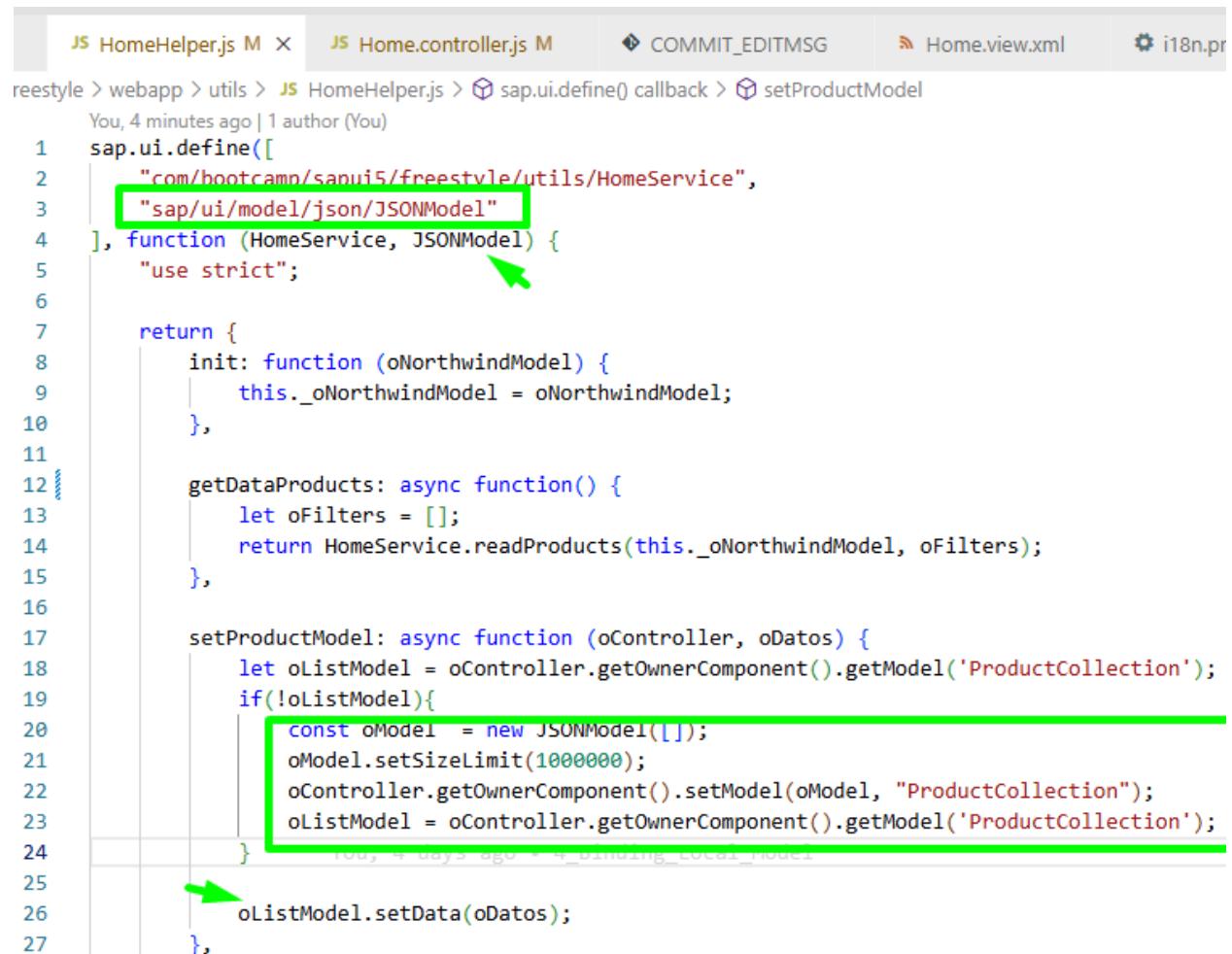


Solicitud de Productos (Parte 2)

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**:



```

JS HomeHelper.js M X JS Home.controller.js M COMMIT_EDITMSG Home.view.xml i18n.pr
reestyle > webapp > utils > JS HomeHelper.js > sap.ui.define() callback > setProductModel
You, 4 minutes ago | 1 author (You)
1 sap.ui.define([
2   "com/hootcamp/sanui5/freestyle/utils/HomeService",
3   "sap/ui/model/json/JSONModel" // Annotation
4 ], function (HomeService, JSONModel) {
5   "use strict";
6
7   return {
8     init: function (oNorthwindModel) {
9       this._oNorthwindModel = oNorthwindModel;
10      },
11
12      getDataProducts: async function() {
13        let oFilters = [];
14        return HomeService.readProducts(this._oNorthwindModel, oFilters);
15      },
16
17      setProductModel: async function (oController, oDatos) {
18        let oListModel = oController.getOwnerComponent().getModel('ProductCollection');
19        if(!oListModel){
20          const oModel = new JSONModel([ ]);
21          oModel.setSizeLimit(1000000);
22          oController.getOwnerComponent().setModel(oModel, "ProductCollection");
23          oListModel = oController.getOwnerComponent().getModel('ProductCollection');
24        }
25        oListModel.setData(oDatos);
26      },
27    }
28  }

```

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

JavaScript

```

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

```

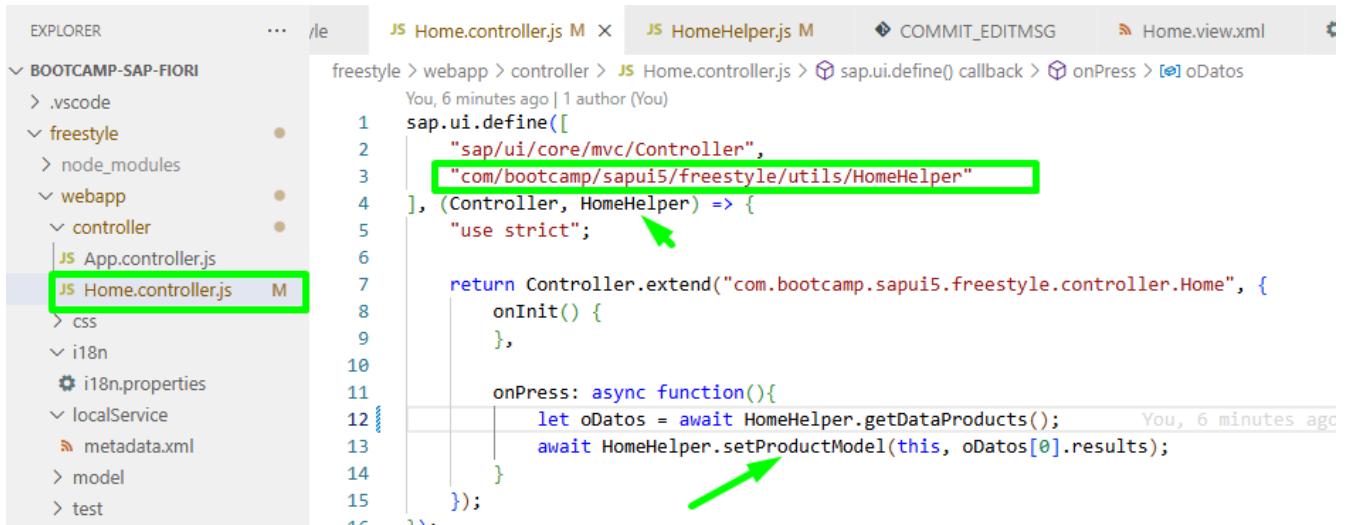
```

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**:



```

sap.ui.define([
    "sap/ui/core/mvc/Controller",
    "com/bootcamp/sapui5/freestyle/utils/HomeHelper"
], (Controller, HomeHelper) => {
    "use strict";

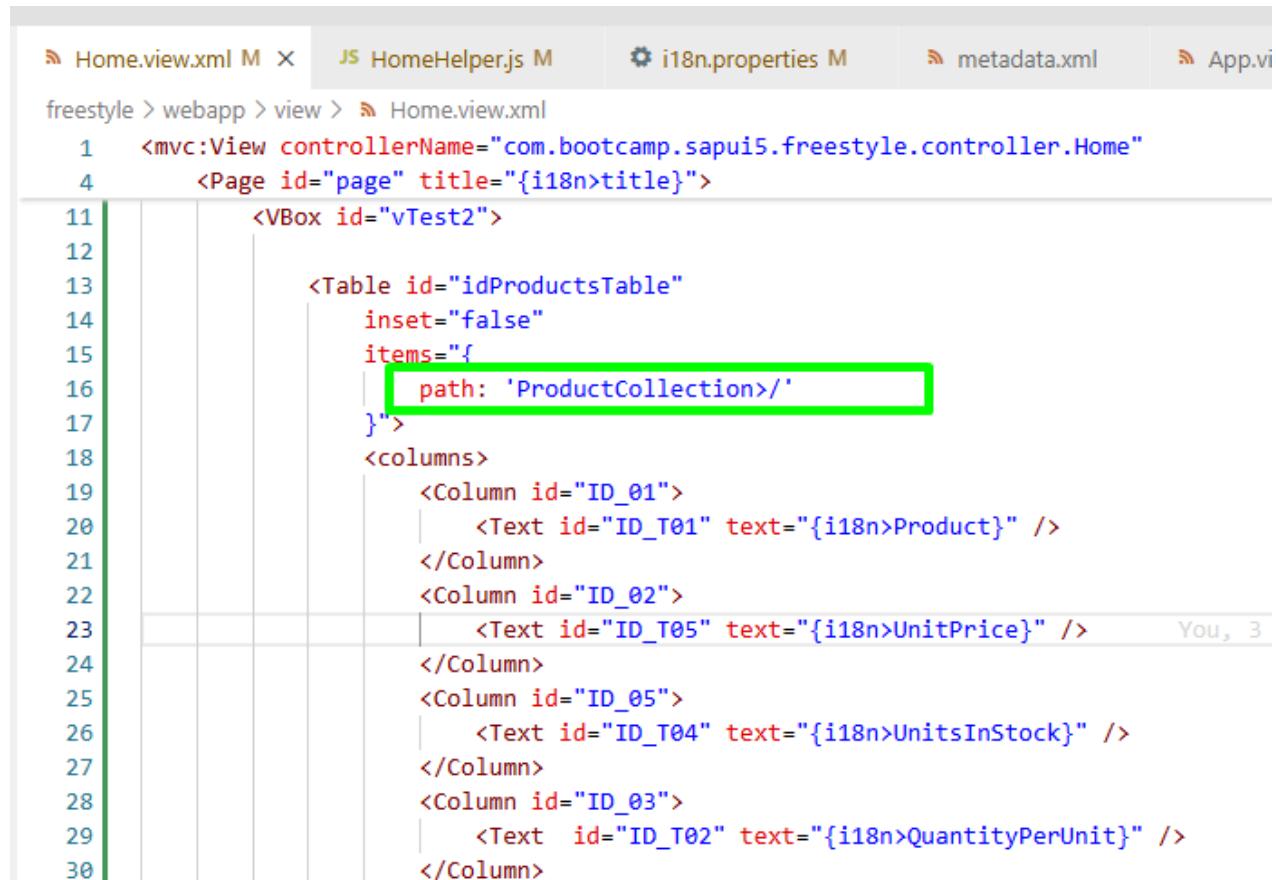
    return Controller.extend("com.bootcamp.sapui5.freestyle.controller.Home", {
        OnInit() {
        },

        onPress: async function(){
            let oDatos = await HomeHelper.getDataProducts();
            await HomeHelper.setProductModel(this, oDatos[0].results);
        }
    });
}

```

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](#))



The screenshot shows a code editor interface for a SAP Fiori application. The top navigation bar includes links for Home.view.xml, HomeHelper.js, i18n.properties, metadata.xml, and App.vi. Below the navigation, a breadcrumb trail indicates the current file path: freestyle > webapp > view > Home.view.xml. The main content area displays the XML code for the view. A specific line of code, 'path: 'ProductCollection'/' (line 16), is highlighted with a green rectangular box.

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

```

Se hace el binding con las diferentes columnas configuradas así:

Home.view.xml M JS HomeHelper.js M i18n.properties M metadata.xml App.view.xml JS HomeSer

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"
51          </columns>
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'},{path:''}],
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>
```

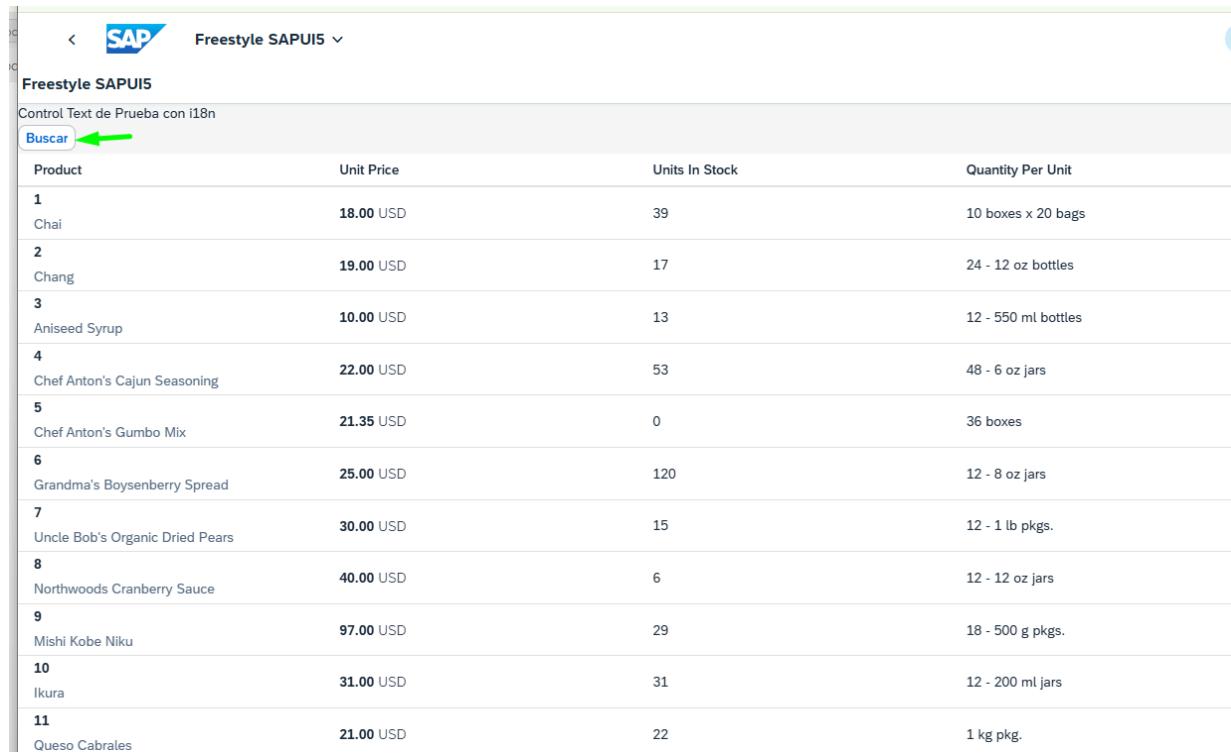
Java

```

<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:



SAP Freestyle SAPUIS

Control Text de Prueba con i18n

Freestyle SAPUIS

Buscar

Product	Unit Price	Units In Stock	Quantity Per Unit
1 Chai	18.00 USD	39	10 boxes x 20 bags
2 Chang	19.00 USD	17	24 - 12 oz bottles
3 Aniseed Syrup	10.00 USD	13	12 - 550 ml bottles
4 Chef Anton's Cajun Seasoning	22.00 USD	53	48 - 6 oz jars
5 Chef Anton's Gumbo Mix	21.35 USD	0	36 boxes
6 Grandma's Boysenberry Spread	25.00 USD	120	12 - 8 oz jars
7 Uncle Bob's Organic Dried Pears	30.00 USD	15	12 - 1 lb pkgs.
8 Northwoods Cranberry Sauce	40.00 USD	6	12 - 12 oz jars
9 Mishi Kobe Niku	97.00 USD	29	18 - 500 g pkgs.
10 Ikura	31.00 USD	31	12 - 200 ml jars
11 Queso Cabrales	21.00 USD	22	1 kg pkg.