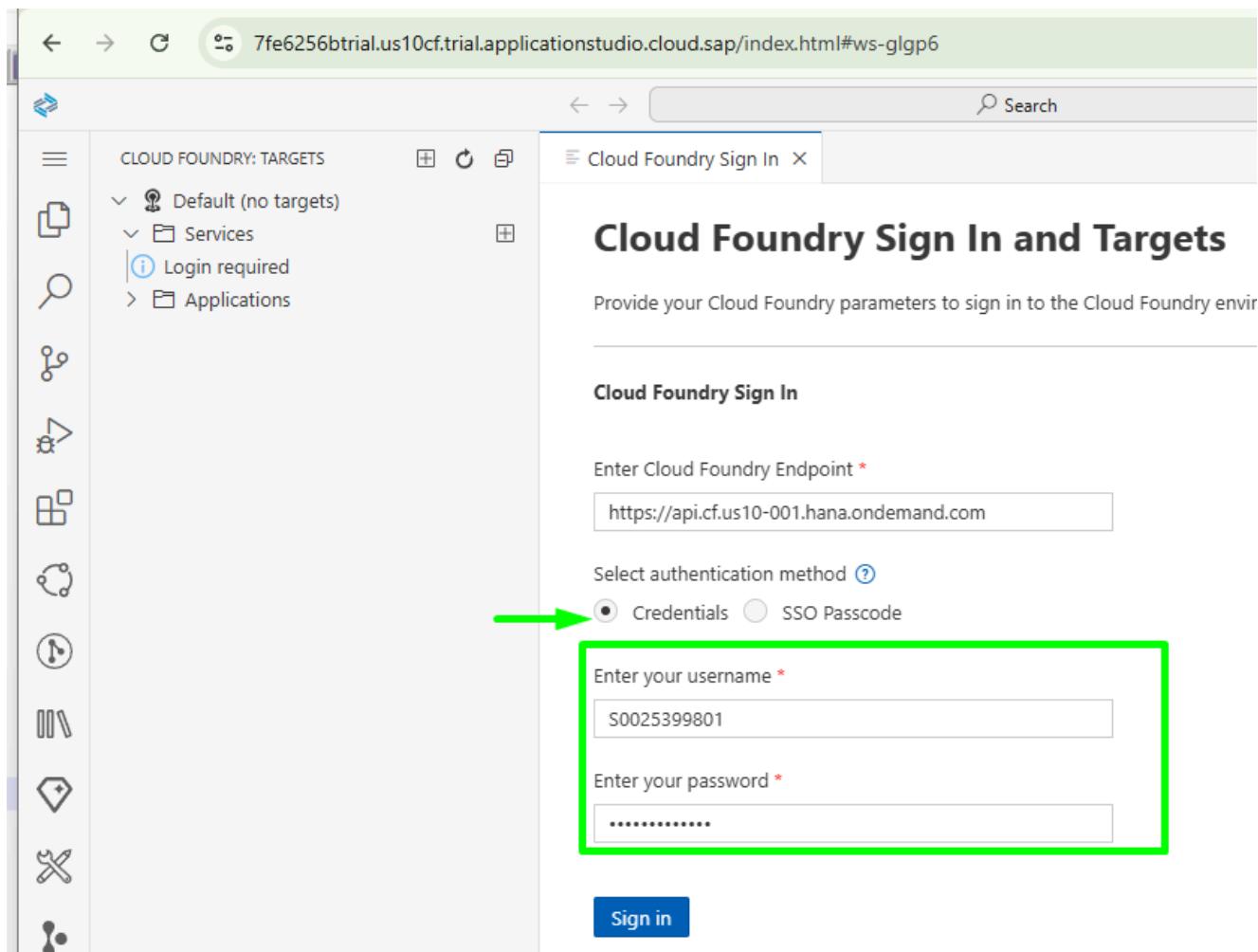


# Creación de la primera app (Hola Mundo)

Previo a este ejercicio se debió [crear en el Destination](#) en el Space generado en su cuenta SAP BTP Trial:

The screenshot shows the SAP Application Studio interface. On the left, there's a sidebar with various icons. One icon, a user profile, is highlighted with a green box and has a green arrow pointing to it from below. Another icon, a gear, is also highlighted with a green box. In the main area, under 'CLOUD FOUNDRY: TARGETS', a 'Default (no targets)' section is expanded, showing a 'Login required' message. A green box highlights this message. To the right, the 'Get Started' screen is visible, featuring a blue cube icon, the text 'Get Started w', 'dev', and 'SAP Fiori'. Below this are buttons for '+ New Project' and 'New Project from'. Under 'Projects (0)', it says 'There are currently no projects in your dev spa'. Under 'Sample Apps (2)', there are two cards: 'SAP Fiori elements List Report Application' (described as an app to manage OData V2 floorplan) and 'Custom App built using Basic template' (described as a custom app to ...).

**Datos Login:**



Luego selecciona la Subcuenta y el Space, que para nuestro caso es **dev**.

Ahora creamos nuestra primera App:

Screenshot of SAP Application Studio interface showing the creation of a new project from a template.

The top navigation bar shows the URL: `7fe6256btrial.us10cf.trial.applicationstudio.cloud.sap/index.html#ws-85rrf`.

The left sidebar contains icons for Back, Forward, Stop, Refresh, and Search, along with a "File" menu icon.

The "File" menu is open, showing the following options:

- New Text File (Ctrl+K N)
- New File... (Ctrl+Alt+Windows+N)
- New Project from Template** (highlighted)
- New Window (Ctrl+K Shift+N)
- New Window with Profile >
- Import Project
- Open File... (Ctrl+O)
- Open Folder... (Ctrl+K Ctrl+O)
- Open Workspace from File...
- Open Recent >

A "New Project from" button is visible at the bottom of the menu.

The main content area displays the "New Project From Template" wizard:

- Get Started** tab is selected.
- Project From Template** tab is visible.
- Select Template and Target Location** section:
  - SAP Fiori generator** is selected.
  - Text: "Select the template that best fits the type of application you want to develop. The template will be generated in the folder specified below and it will contain the relevant files and configurations that will help you create a project quickly and easily."
  - Input field: "Specify a target folder path" (value: "/home/user/projects")
  - Text: "Where do you want to open the project?" (with "Preferences" link)
  - Text: "Open the project in a multi-root workspace"
  - Text: "Templates" (with "More Information" link)
- Basic Multitarget Appli...** template card:
  - Description: "Create a project for developing a multitarget application (MTA). MTAs are comprised of multiple software modules that are created with different technologies, yet share the same development lifecycle."
  - [More Information](#)
  - Icon: Database and gear.
- SAPUI5 Adaptation Pro...** template card:
  - Description: "Adaptation project allows you to create an app variant for an existing SAP Fiori elements-based or SAPUI5 freestyle application, without changing the original application."
  - [More Information](#)
  - Icon: Three overlapping documents.
- SAP Fiori generator** template card (highlighted with a green border and a large green arrow pointing to it):
  - Description: "Create an SAPUI5 application using SAP Fiori elements or a freestyle approach"
  - [More Information](#)
  - Icon: A stylized "F" shape with a gear.

**SAP Fiori generator**

**Template Selection**  
Choose your application template.

Which template do you want to use? \*

- Basic** Create a freestyle application, starting with an empty page. [More Information](#)
- Custom Page** Create an SAP Fiori elements application containing a custom page based on the flexible programming model. [More Information](#)
- List Report Page** Create an SAP Fiori elements application containing a list and an object page. [More Information](#)

**CLOUD FOUNDRY: TARGETS**

- Default (no targets)
- Services
- Applications

**SAP Fiori generator**

**Select Template and Target Location**  
Configure the data source and select a service.

**Template Selection**  
Template: Basic

**Data Source and Service Selection**

Data source \* Connect to a System

System ? \* Northwind

Service path ? \* /northwind/northwind.svc/

Service \* https://services.odata.org/northwind/northwind....

**Entity Selection**

**Project Attributes**

SAP Fiori generator ⓘ

**Select Template and Target Location**

SAP Fiori generator

**Template Selection**

Template: Basic

**Data Source and Service Selection**

Data Source: Connect to a System  
System: Northwind  
Service Path: /northwind/northwind.svc/  
[More...](#)

**Entity Selection**

Configure the selected service.

View name  Home

**Entity Selection**

View Name: Home

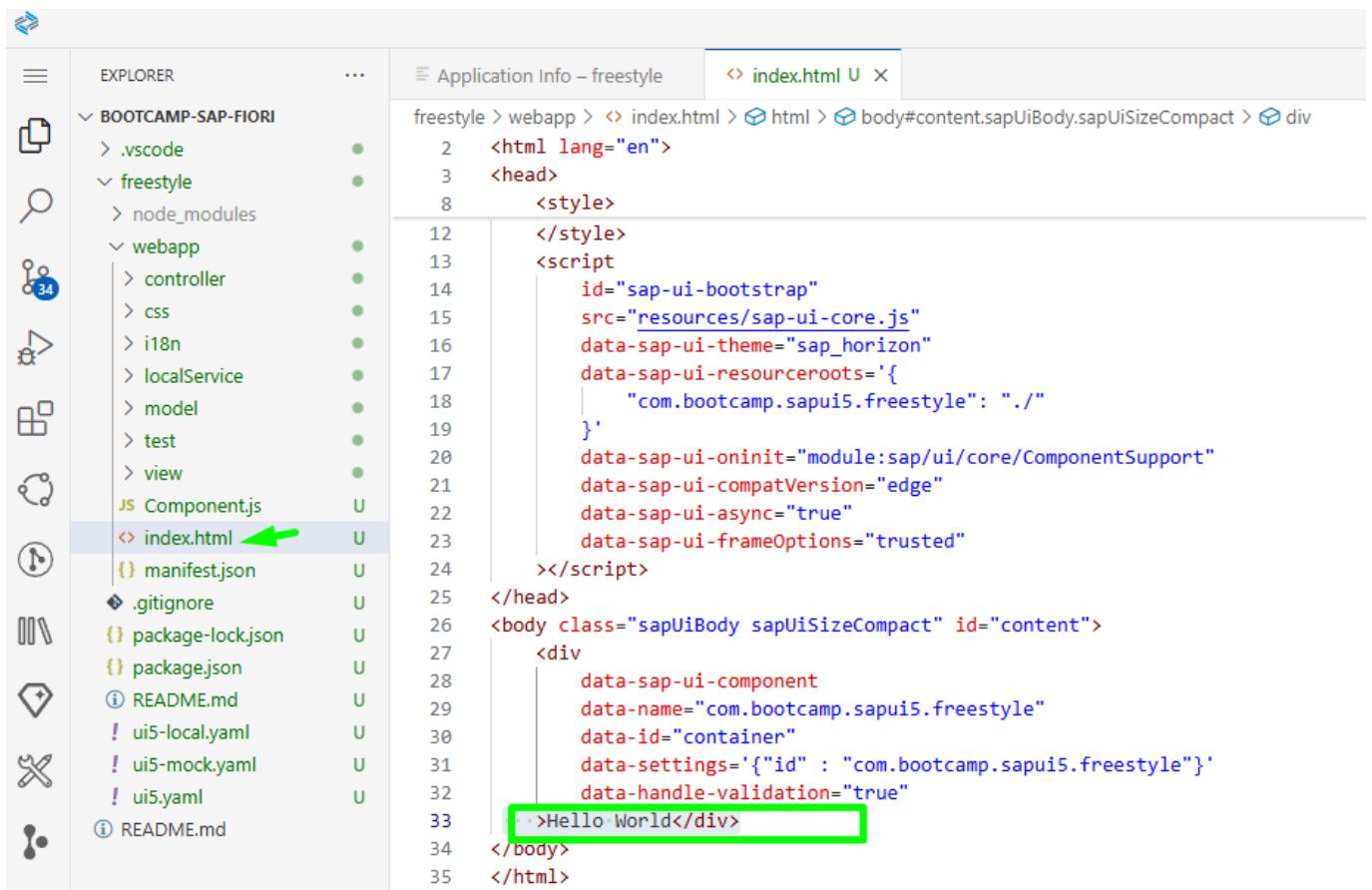
**Project Attributes**

Configure the main project attributes.

**Project Attributes**

Module name  \*  
Application title   
Application namespace   
Description   
Project folder path   
Minimum SAPUI5 version  \*  
Enable TypeScript

Ahora agregamos “Hola Mundo” en el archivo Index.html

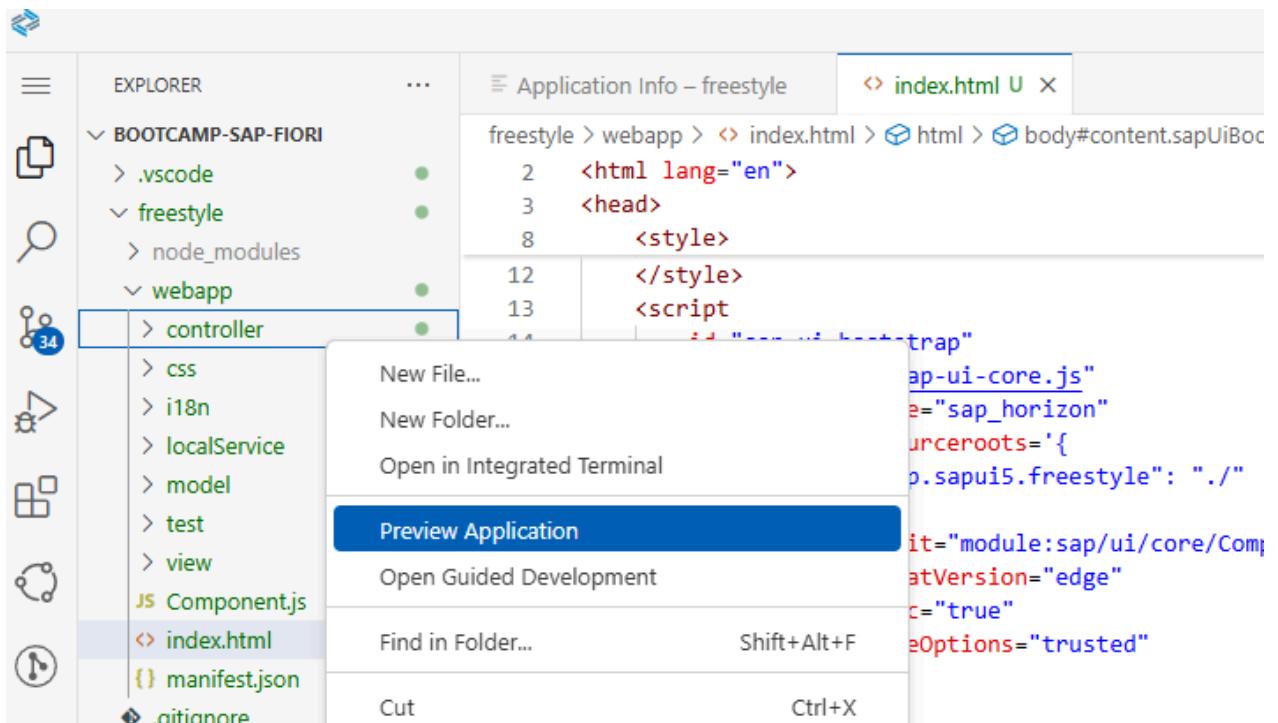


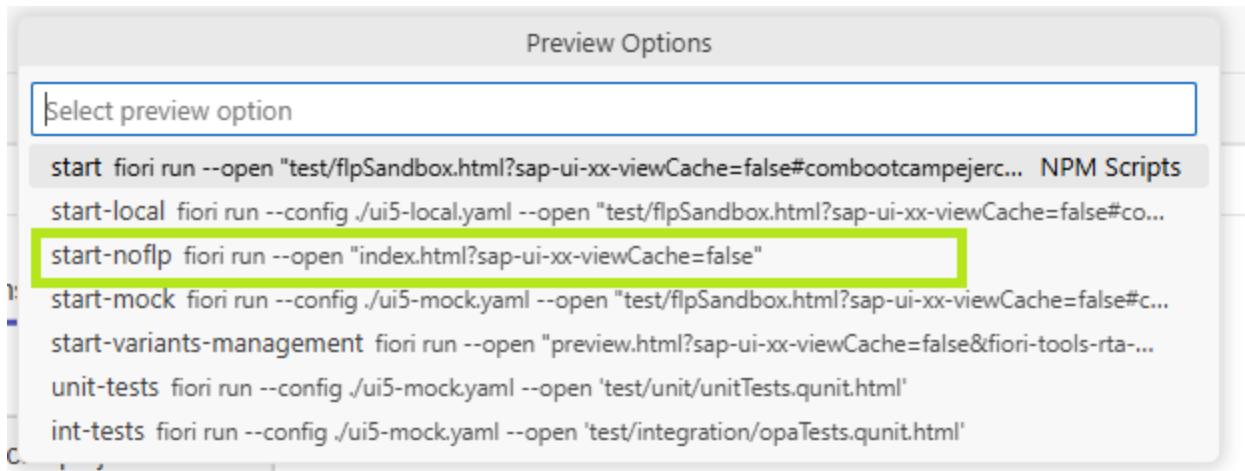
Application Info – freestyle

```

freestyle > webapp > index.html > html > body#content.sapUiSizeCompact > div
  2   <html lang="en">
  3     <head>
  8       <style>
12     </style>
13   <script
14     id="sap-ui-bootstrap"
15     src="resources/sap-ui-core.js"
16     data-sap-ui-theme="sap_horizon"
17     data-sap-ui-resourceroots='{
18       "com.bootcamp.sapui5.freestyle": "./"
19     }'
20     data-sap-ui-oninit="module:sap/ui/core/ComponentSupport"
21     data-sap-ui-compatVersion="edge"
22     data-sap-ui-async="true"
23     data-sap-ui-frameOptions="trusted"
24   ></script>
25 </head>
26 <body class="sapUiBody sapUiSizeCompact" id="content">
27   <div
28     data-sap-ui-component
29     data-name="com.bootcamp.sapui5.freestyle"
30     data-id="container"
31     data-settings='{ "id" : "com.bootcamp.sapui5.freestyle" }'
32     data-handle-validation="true"
33   >Hello World</div>
34 </body>
35 </html>

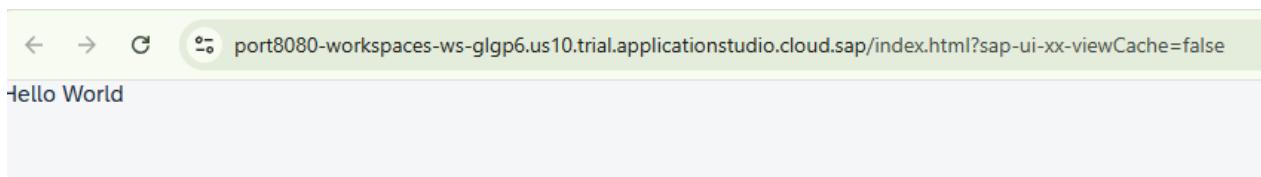
```





### Start-noflp

Esto inicia la aplicación a través de la página HTML index.html. La aplicación no está incrustada en un sandbox de SAP Fiori launchpad.



## Agregar App a Repo Público Github

Como requisito debemos tener una cuenta creada en [GitHub](#) y un Repositorio público donde deben ir actualizando el avance de cada una de las Actividades:

Para nuestro caso creamos el repo <https://github.com/jegiraldog/Bootcamp-SAP-FIORI-P> donde iremos actualizando el avance de cada una de las sesiones.

Para iniciar el repo en BAS nos ubicamos en la raíz del proyecto en la línea de comandos y seguimos las instrucciones que github nos plantea al crear el repo:

```
JavaScript
echo "# Bootcamp-SAP-FIORI-P" >> README.md
```

```
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/jegiraldog/Bootcamp-SAP-FIORI-P.git
git push -u origin main
```

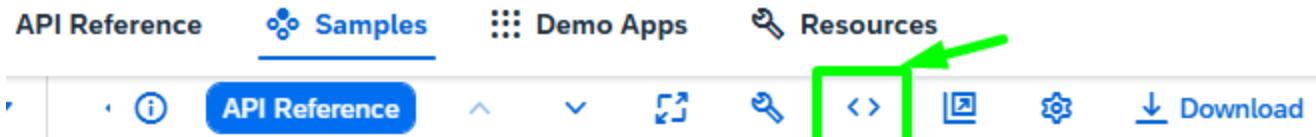
## Añadir elementos de texto a la vista. Binding I18n.

Ahora Vamos a la Vista Home y agregamos un control [Text](#)

API Reference: [sap.m.Text](#)

The screenshot shows the SAP UI5 Demo Kit interface. On the left, there's a sidebar with categories like Text, Layout, and List. Under Text, 'Text' is selected and highlighted with a green border. In the main content area, the title 'sap.m.Text' is displayed above a detailed description. The 'Samples' tab is active, showing a table with two rows: 'Text' and 'Text - Max Lines'. Both rows have their names and descriptions highlighted with green borders. The 'About' tab is also visible. At the top, there are navigation links for Home, Documentation, API Reference, Samples, Demo Apps, and Resources. A search bar is located at the top right.

Navegamos al primer ejemplo y visualizamos su implementación así:



Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat

```

Run Auto Run Reset all Changes Download
V.view.xml manifest.json

1 <mvc:View>
2   xmlns="sap.m"
3   xmlns:mvc="sap.ui.core.mvc"
4   <VBox class="sapUiSmallMargin">
5     <Text text="Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy
eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam
voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita
kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem
ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor
invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. Lorem ipsum
dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor
invidunt ut labore et dolore magna aliquyam erat" />
6   </VBox>
7 </mvc:View>
8

```

Ahora agregamos un control Text a nuestro proyecto en la vista Home:

```
<Text id="textPrueba" text="Control Text de Prueba"></Text>
```

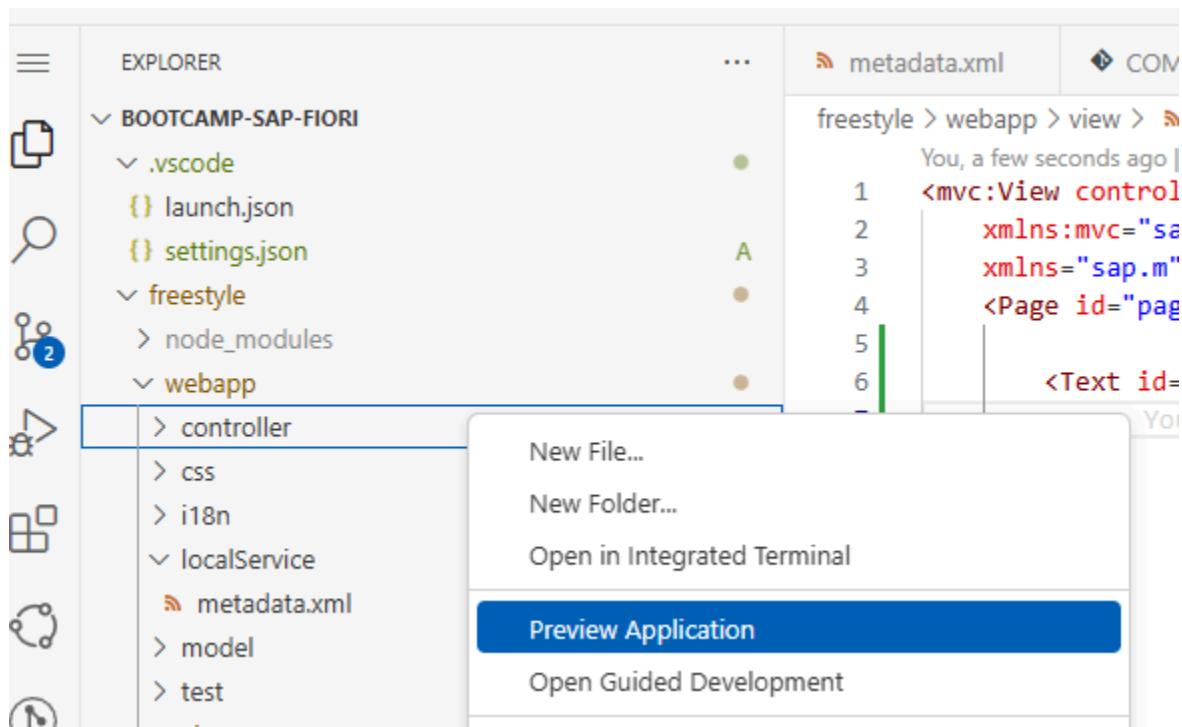


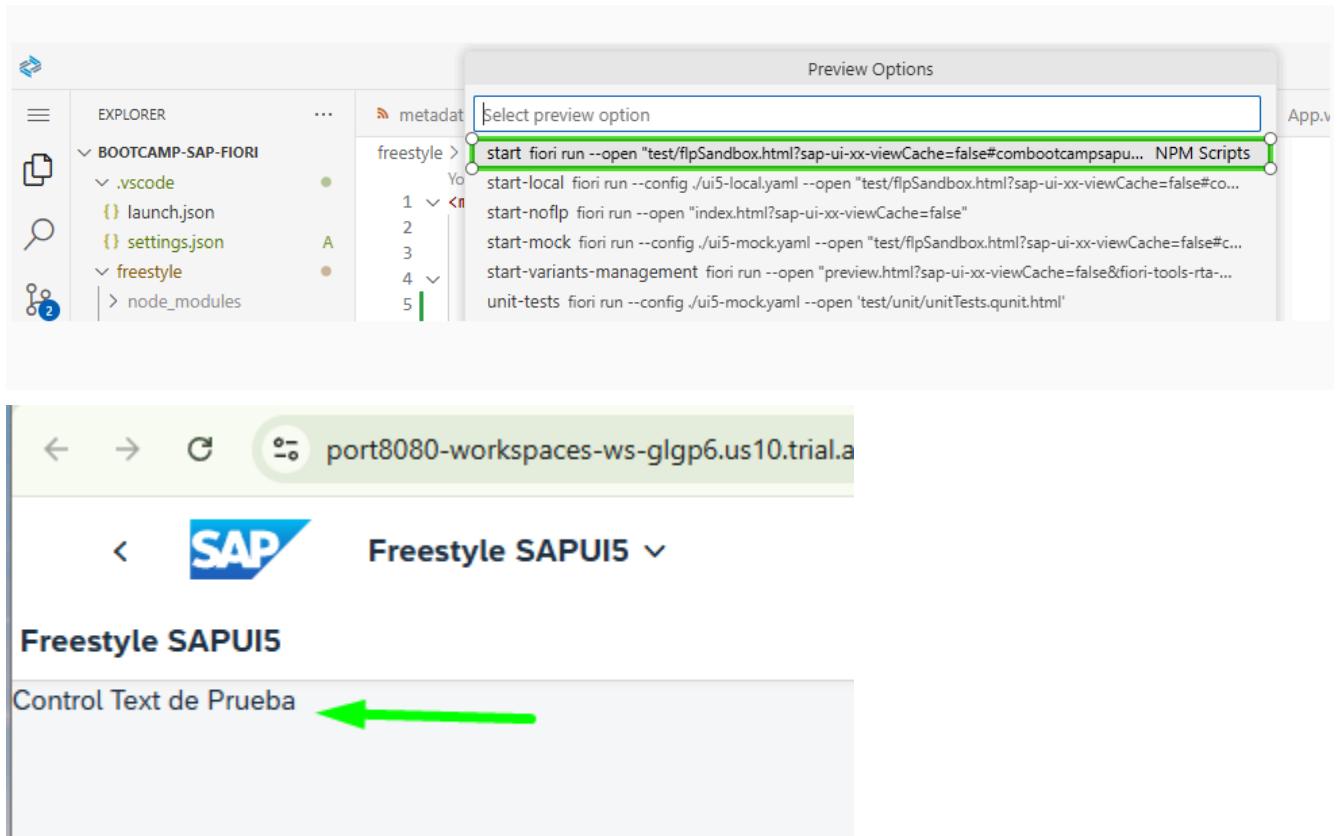
```

1 <mvc:View controllerName="com.bootcamp.sapui5.freestyle.controller.Home"
2   xmlns:mvc="sap.ui.core.mvc"
3   xmlns="sap.m">
4   <Page id="page" title="{i18n>title}">
5     <Text id="textPrueba" text="Control Text de Prueba"></Text>
6   </Page>
7 </mvc:View>

```

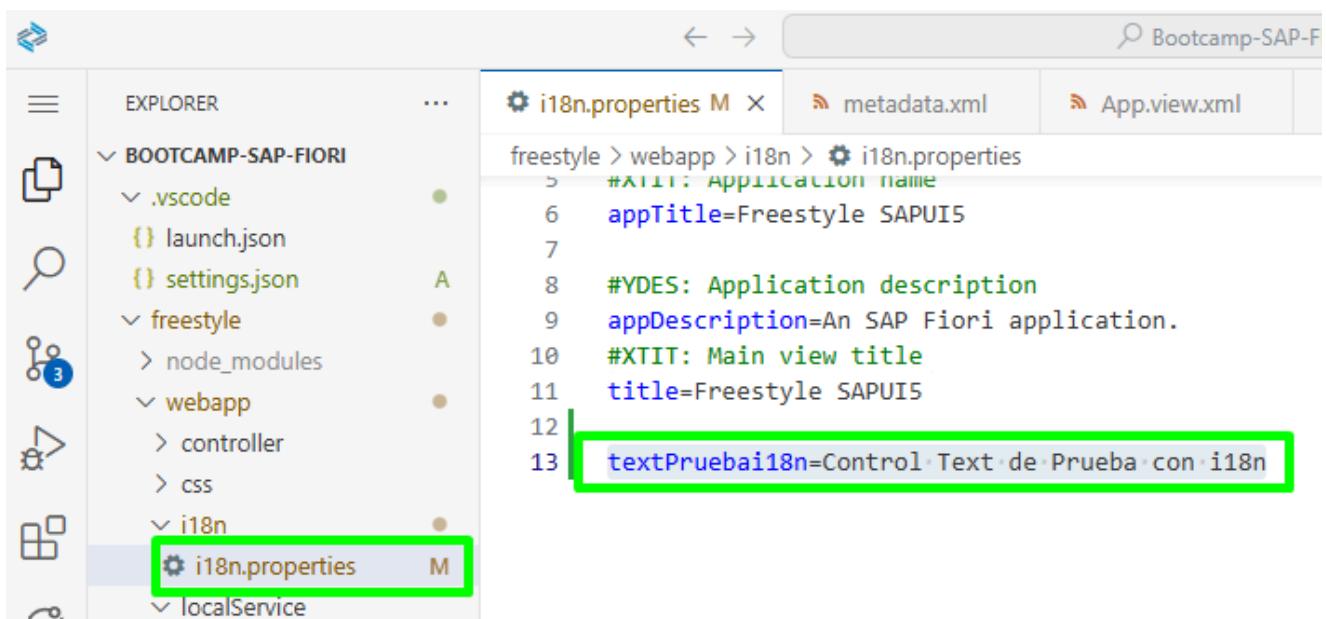
Ejecutamos la App:



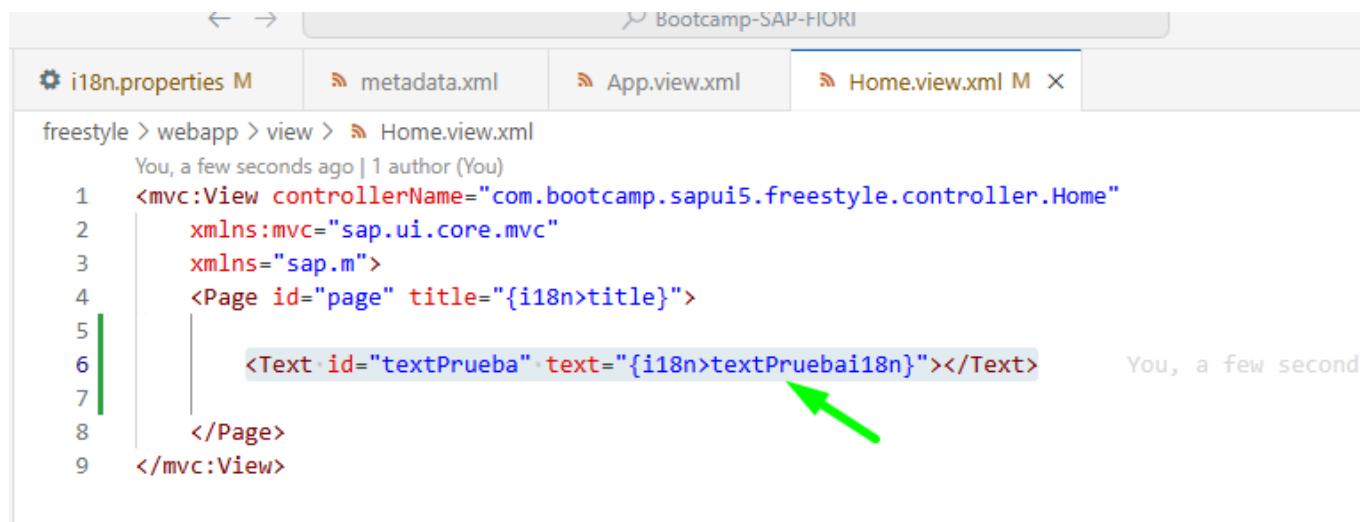


Ahora vamos a agregar el texto “**Control Text de Prueba**” en el archivo i18n

`textPruebab18n=Control Text de Prueba con i18n`



```
<Text id="textPrueba" text="{i18n>textPruebai18n}"></Text>
```



The screenshot shows the SAP Studio interface with the 'Home.view.xml' tab selected. The XML code defines a page with a text control that uses an i18n placeholder:

```
1 <mvc:View controllerName="com.bootcamp.sapui5.freestyle.controller.Home"
2   xmlns:mvc="sap.ui.core.mvc"
3   xmlns="sap.m">
4   <Page id="page" title="{i18n>title}">
5     <Text id="textPrueba" text="{i18n>textPruebai18n}"></Text>
6   </Page>
7 </mvc:View>
```

A green arrow points to the text node containing the placeholder '{i18n>textPruebai18n}'.

Si mantenemos la ejecución de la aplicación se podrán visualizar los cambios:

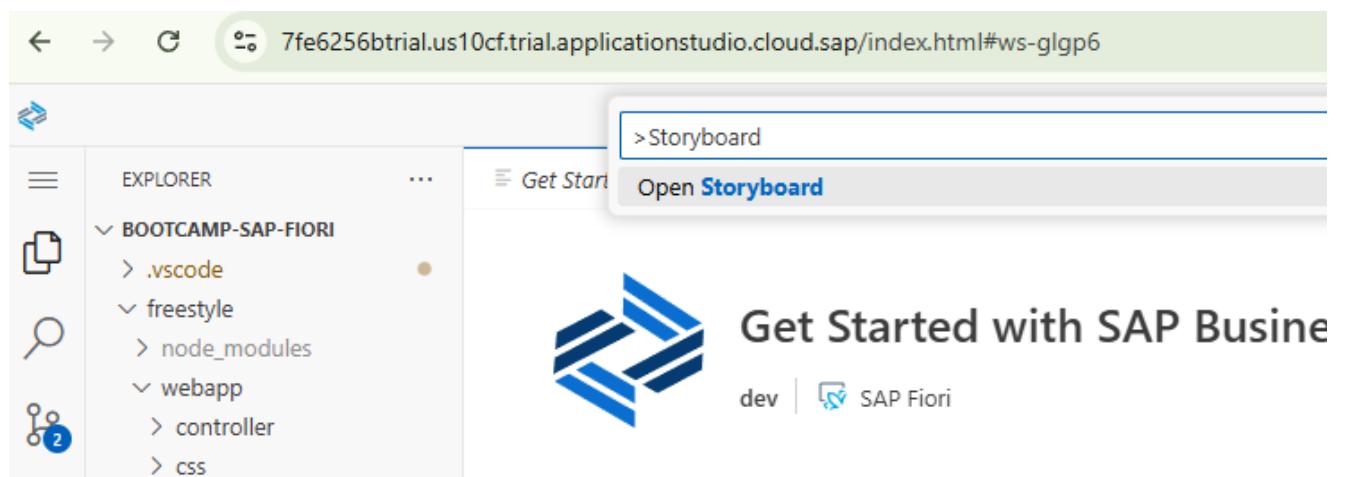


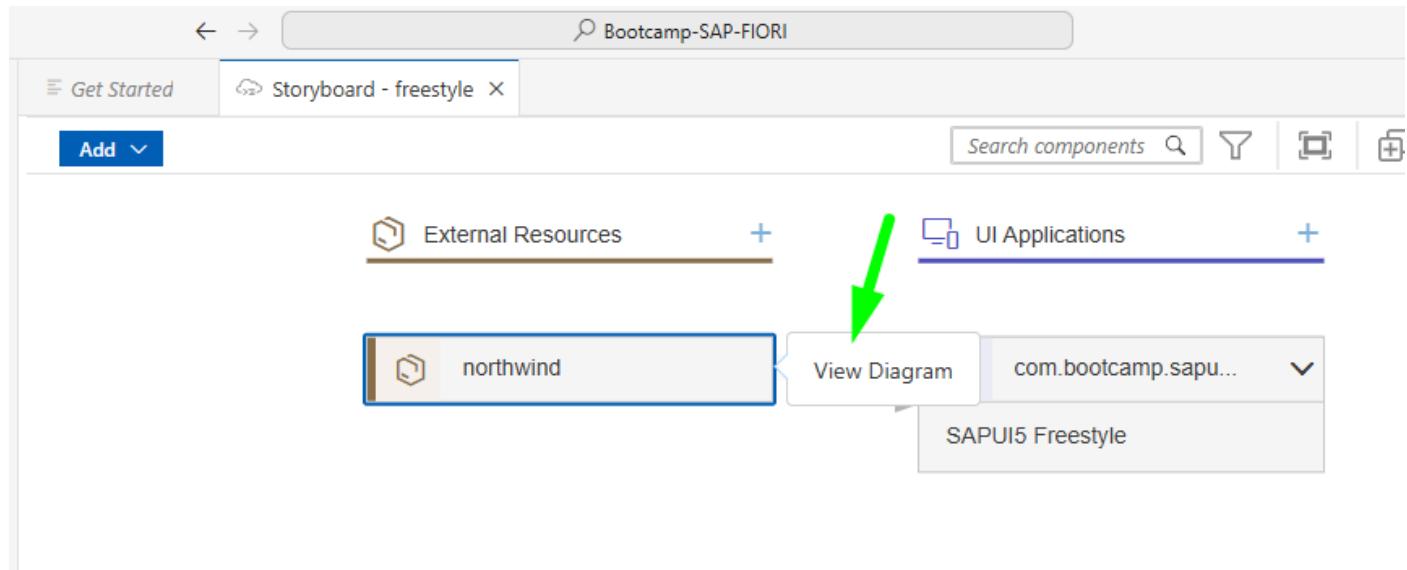
Interactuar con el Modelo del servicio Northwind:  
**Storyboard**

## Working in the Storyboard

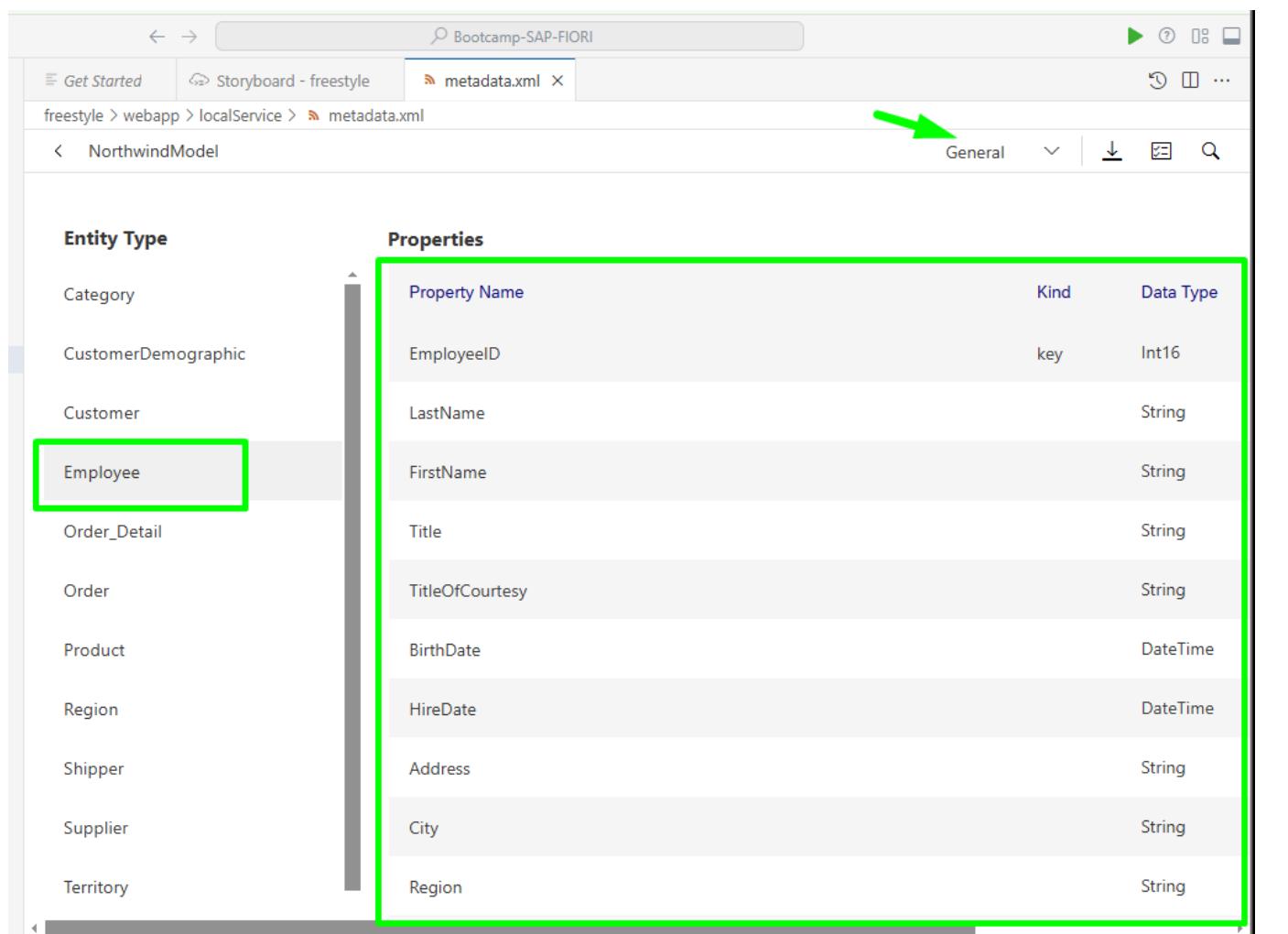
To open the storyboard:

1. Press **F1** to open the command palette.
2. Enter **Storyboard**.
3. Select **Open Storyboard**.
4. Select the project for which you want to open the storyboard.





The screenshot shows the SAP Fiori Launchpad interface. At the top, there are tabs for "Get Started" and "Storyboard - freestyle". Below the tabs, there is a search bar and a toolbar with icons for "Add", "Search components", and others. Two main sections are visible: "External Resources" and "UI Applications". Under "External Resources", there is a box for "northwind". Under "UI Applications", there is a box for "com.bootcamp.sapu... SAPUI5 Freestyle". A green arrow points from the "View Diagram" button next to the "northwind" box towards the "SAPUI5 Freestyle" box.

The screenshot shows the SAP Fiori Modeler interface. The title bar says "Bootcamp-SAP-FIORI". The breadcrumb navigation shows "freestyle > webapp > localService > metadata.xml". The left sidebar lists entity types: Category, CustomerDemographic, Customer, Employee, Order\_Detail, Order, Product, Region, Shipper, Supplier, and Territory. The "Employee" entity is highlighted with a red box. The main area displays the properties of the "Employee" entity:

Property Name	Kind	Data Type
EmployeeID	key	Int16
LastName		String
FirstName		String
Title		String
TitleOfCourtesy		String
BirthDate		DateTime
HireDate		DateTime
Address		String
City		String
Region		String

A green arrow points from the "General" tab in the top right corner of the interface towards the properties table.

Cambiamos la vista a Diagrama:

Screenshot of the SAP Fiori Modeler interface showing the 'Diagram' view.

The top navigation bar shows the project path: freestyle > webapp > localService > metadata.xml. The right-hand toolbar has a dropdown menu with three options: General, Diagram (highlighted with a green border), and Metadata. A large green arrow points from the 'Diagram' option in the toolbar to the diagram area below.

The diagram area displays four entity types:

- Sales\_by\_Category** Entity Type:
 

CategoryID	Integer
CompanyName	String(15)
ProductName	String(40)
ProductSales	Decimal(19,4)
- Sales\_Totals\_by\_Amount** Entity Type:
 

SaleAmount	Decimal(19,4)
OrderID	Integer
CompanyName	String(40)
ShippedDate	DateTime
- Order\_Details\_Extended** Entity Type:
 

OrderID	Integer
ProductID	Integer
ProductName	String(40)
UnitPrice	Decimal(19,4)
Quantity	Integer16
Discount	Single
ExtendedPrice	Decimal(19,4)

Screenshot of the SAP Fiori Modeler interface showing the 'Diagram' view.

The top navigation bar shows the project path: freestyle > webapp > localService > metadata.xml. The right-hand toolbar has a dropdown menu with three options: Diagram (highlighted with a blue border), General, and Metadata.

The diagram area displays two entity types with their associations:

- Customer** Entity Type:
 

CustomerID	String(5)
CompanyName	String(40)
ContactName	String(30)
ContactTitle	String(30)
Address	String(60)
City	String(15)
Region	String(15)
PostalCode	String(10)
Country	String(15)
Phone	String(24)
Fax	String(24)
Orders	many Order
CustomerDemographics	many CustomerDemographic
- Shipper** Entity Type:
 

ShipperID	String(5)
CompanyName	String(40)
Phone	String(24)

Associations are shown as blue lines connecting the entities. The 'Customer' entity has a one-to-many association labeled '1' to the 'Orders' field in the 'Customer' row. The 'Customer' entity also has a many-to-many association labeled 'many CustomerDemographic' to the 'CustomerDemographics' field in the 'Customer' row. The 'Shipper' entity has a many-to-many association labeled 'many Order\_Detail' to the 'Order\_Details' field in the 'Shipper' row. There is also a many-to-many association between 'Customer' and 'Shipper'.