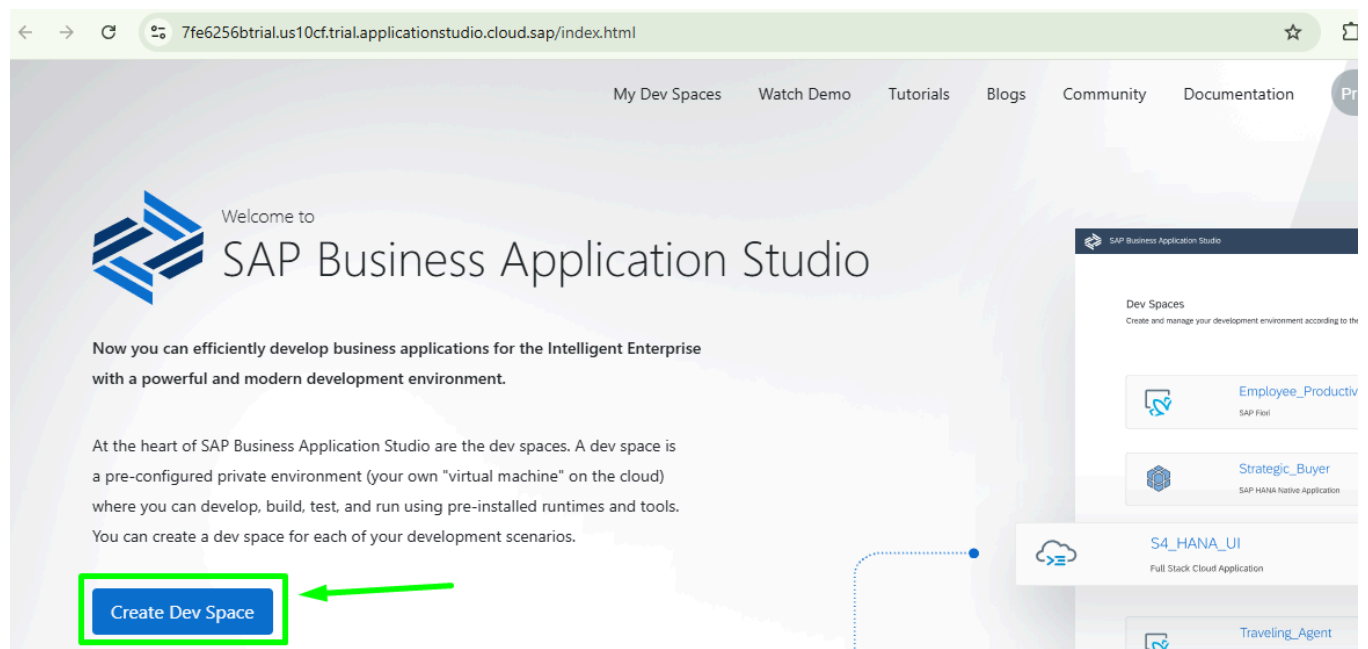
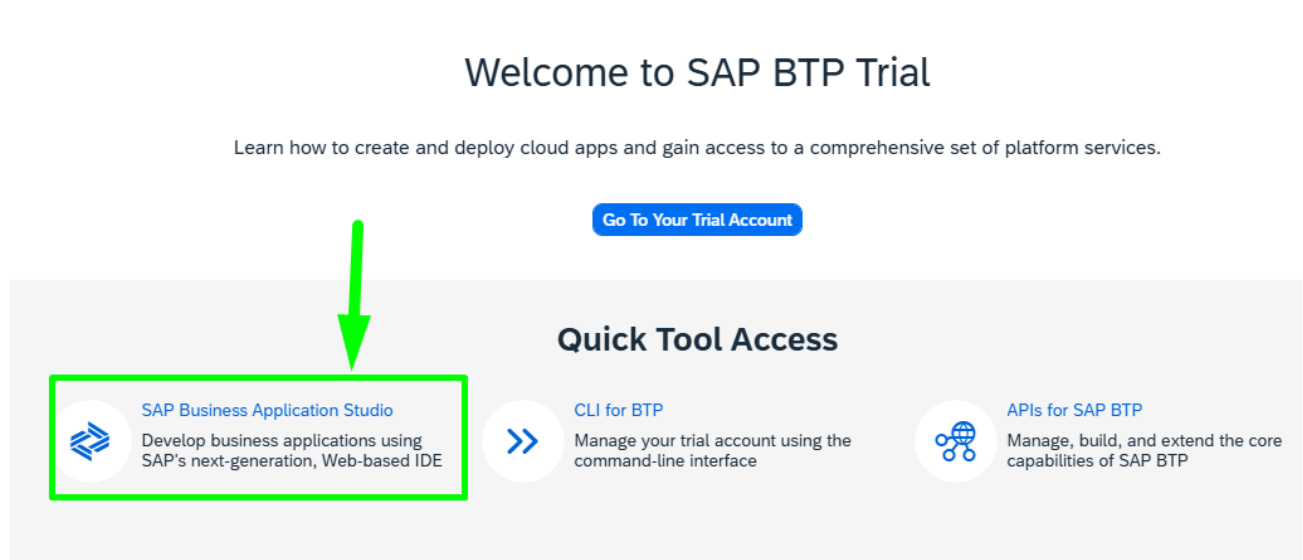


## Configuración ambiente (BTP - Destination)

- [Creación de cuenta Trial](#)
- [Configuración BAS](#)

Log into SAP BTP Trial:

<https://account.hanatrial.ondemand.com/trial/#/home/trial>






SAP Business Application Studio


## Create a New Dev Space


 


What kind of application do you want to create?

☒  SAP Fiori

☐  Full-Stack Application Using Productivity Tools

☐  Full Stack Cloud Application

☐  SAP HANA Native Application

☐  SAP Mobile Application

☐  Basic

## SAP Fiori Dev Space

Develop, test, build, and deploy SAP Fiori freestyle or SAP Fiori elements applications to SAP Business... This dev space contains a comprehensive set of tools, such as best practice templates, code assist, co... modeling, and application modeling and preview.

### SAP Predefined Extensions

The following extensions are enabled by default.



#### Basic Tools

Allows you to perform operations such as build and deployment of multitarget... [more](#)



#### Fiori Freestyle Tools

Allows you to optimize your Fiori-based development. [more](#)



#### HTML5 Runner

Allows you to locally run HTML5 applications. Includes the HTML5... [more](#)



#### MTA Tools

Allows you to create and develop multitarget applications. [more](#)



#### SAP Fiori Tools

Allows you to simplify the development of SAP Fiori elements applications by... [more](#)

### Additional SAP Extensions

Select additional extensions to enhance



#### SAPUI5 Adaptation (Foundry only)

Allows you to extend applications using... [more](#)



#### Docker Image Builder

Allows you to build Docker images. Supports build, lo... [more](#)



#### CDS Graphical Modeler

Allows you to design services models in... [more](#)



#### CAP Tools

Allows you to develop applications using... [more](#)



#### SAP HANA Calculation Editor

Allows you to edit HANA calculation... [more](#)


Cancel






## Dev Spaces

Create and manage your development environment according to the type of applications you want to develop.

You can add extensions and tools to further enhance your development options. For more information, see the [documentation](#).

Create Dev Space

 You're using a trial version. Any dev space that hasn't been running for 30 days will be deleted. See the full list of [restrictions](#).

 <div> <div>dev</div> <div>SAP Fiori</div> </div>	<div>←</div>	<div>RUNNING</div>	<div>Created On</div> <div>02/17/2025 9:12 PM</div>	<div>ID</div> <div>ws-glgp6</div>	<div>Disk Usage</div> <div>44 KB / 3.8 GB</div>	<div></div> <div></div> <div></div> <div></div>
--	--------------	--------------------	---	-----------------------------------	---	---

## Estructura de un Proyecto SAPUI5

### JavaScript

#### myApp/

```

|
├── webapp/
|   ├── controller/      # Controladores de la aplicación
|   ├── view/            # Vistas de la aplicación (XML, HTML, JS)
|   ├── model/           # Modelos de datos (JSON, OData, etc.)
|   ├── i18n/            # Archivos de internacionalización (traducciones)
|   ├── css/             # Archivos de estilo CSS
|   ├── js/              # Archivos JavaScript adicionales
|   ├── index.html       # Archivo HTML principal
|   ├── manifest.json     # Archivo de configuración de la aplicación
|   └── Component.js      # Archivo principal de la aplicación (definición de
componentes)
|
├── test/                # Pruebas del proyecto
|   ├── unit/            # Pruebas unitarias
|   └── integration/     # Pruebas de integración
|
└── package.json          # Dependencias del proyecto y configuración
    
```

## Descripción de las Carpetas y Archivos

1. **webapp/**: Es la carpeta principal donde se encuentra todo el código fuente de la aplicación.
  - **controller/**: Contiene los controladores de la aplicación que gestionan la lógica y el comportamiento. Cada vista suele tener un controlador asociado.
  - **view/**: Aquí se encuentran las vistas de la aplicación, que pueden estar definidas en formatos como XML, HTML o JS.
  - **model/**: Almacena los modelos de datos utilizados en la aplicación, como archivos JSON o configuraciones de OData.
  - **i18n/**: Contiene los archivos de traducción para soportar múltiples idiomas en la aplicación.
  - **css/**: Archivos de estilo CSS para personalizar la apariencia de la aplicación.
  - **js/**: Archivos JavaScript adicionales que no se encuentran en los controladores o vistas.
  - **index.html**: El archivo HTML principal que carga la aplicación.
  - **manifest.json**: Un archivo clave que contiene la configuración de la aplicación, incluyendo rutas, modelos, y dependencias.
  - **Component.js**: Define el componente de la aplicación y se encarga de la inicialización y configuración.
2. **package.json**: Archivo que gestiona las dependencias del proyecto y puede incluir scripts para tareas comunes de desarrollo.

## MVC

MVC ([Modelo-Vista-Controlador](#)) es un patrón de diseño ampliamente utilizado en el desarrollo de aplicaciones, y SAPUI5 adopta este patrón para estructurar la creación de sus aplicaciones web. A continuación, se explica cada uno de los componentes del patrón MVC en el contexto de SAPUI5:

1. **Modelo (Model)**:
  - El modelo es la parte de la aplicación que maneja los datos y la lógica de negocio. En SAPUI5, puedes utilizar diferentes tipos de modelos, como:
    - **JSONModel**: Para manejar datos en formato JSON.
    - **XMLModel**: Para manejar datos en formato XML.
    - **ODataModel**: Para interactuar con servicios OData que permiten operaciones CRUD (Crear, Leer, Actualizar, Eliminar).
  - El modelo se encarga de almacenar el estado de la aplicación y puede ser utilizado para realizar operaciones de lectura y escritura de datos.

## 2. Vista (View):

- La vista es la parte de la aplicación que se encarga de la presentación de la interfaz de usuario. En SAPUI5, las vistas pueden ser creadas utilizando diferentes formatos, como:
  - XML: Formato comúnmente utilizado para definir vistas en SAPUI5.
  - HTML: Se puede utilizar HTML para definir vistas.
  - JS: Se pueden crear vistas programáticamente usando JavaScript.
- La vista presenta los datos del modelo al usuario y puede incluir elementos de interfaz de usuario como botones, tablas, formularios, etc.

## 3. Controlador (Controller):

- El controlador actúa como intermediario entre el modelo y la vista. Su función principal es manejar la lógica de la aplicación y responder a las interacciones del usuario.
- En SAPUI5, cada vista suele tener un controlador asociado que define eventos (como clics en botones) y actualiza el modelo o la vista en consecuencia.
- El controlador puede interactuar con el modelo para obtener o modificar datos y luego actualizar la vista para reflejar esos cambios.

### Alternativa:

MVC (Model-View-Controller) en SAPUI5 es un patrón de arquitectura que separa la lógica de la aplicación en tres componentes:

- **Model (Modelo):** Maneja los datos y la lógica de negocio, puede ser OData, JSON o XML.
- **View (Vista):** Define la interfaz de usuario, generalmente en XML, HTML o JavaScript.
- **Controller (Controlador):** Contiene la lógica de interacción entre la vista y el modelo.

Esta estructura facilita el mantenimiento, reutilización y escalabilidad de las aplicaciones SAPUI5.

Aquí tienes un ejemplo simple de **MVC en SAPUI5**:

### 1. Modelo (Model) – JSON Model en **Component.js**

Define los datos de la aplicación.

```
// Component.js
sap.ui.define(["sap/ui/core/UIComponent", "sap/ui/model/json/JSONModel"],
function (UIComponent, JSONModel) {
    return UIComponent.extend("myApp.Component", {
        metadata: { manifest: "json" },
        init: function () {
            UIComponent.prototype.init.apply(this, arguments);
            var oData = { name: "SAPUI5" };
            var oModel = new JSONModel(oData);
            this.setModel(oModel, "myModel");
        }
    });
});
```

## 2. Vista (View) – Archivo **MainView.view.xml**

Define la interfaz de usuario.

```
<mvc:View xmlns:mvc="sap.ui.core.mvc" xmlns="sap.m" controllerName="myApp.controller.MainView">
    <VBox>
        <Text text="{myModel}/name}" />
        <Button text="Click Me" press=".onPress" />
    </VBox>
</mvc:View>
```

## 3. Controlador (Controller) – Archivo **MainView.controller.js**

Maneja la lógica de interacción.

```
sap.ui.define(["sap/ui/core/mvc/Controller", "sap/m/MessageToast"],
function (Controller, MessageToast) {
    return Controller.extend("myApp.controller.MainView", {
        onPress: function () {
            MessageToast.show("¡Botón presionado!");
        }
    });
});
```

Este ejemplo muestra cómo MVC separa los datos (modelo), la interfaz (vista) y la lógica de la aplicación (controlador).

## Flujo de Trabajo en MVC

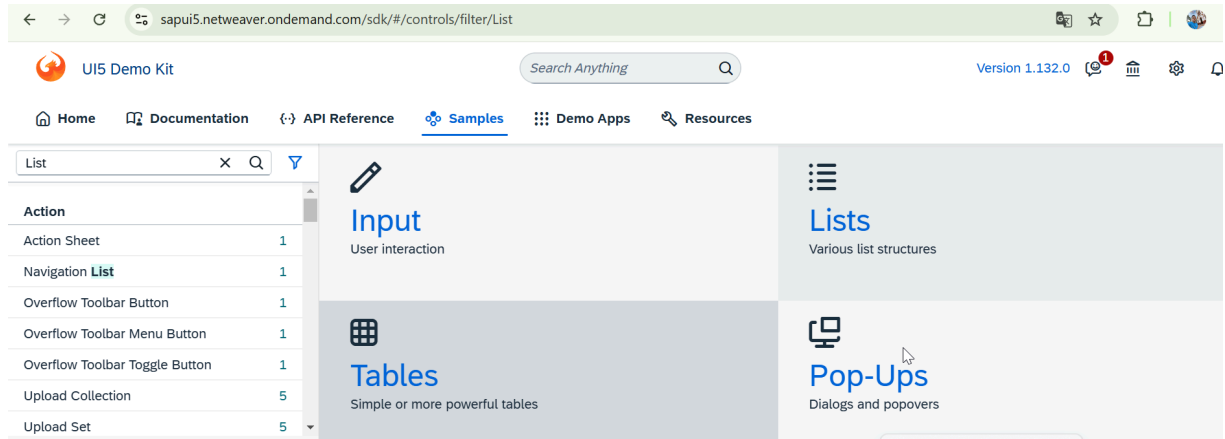
1. **Interacción del Usuario:** El usuario interactúa con la vista (por ejemplo, haciendo clic en un botón).
2. **Evento en el Controlador:** El evento se envía al controlador correspondiente, que maneja la lógica de la aplicación.
3. **Acceso al Modelo:** El controlador puede acceder al modelo para obtener datos o actualizarlo.
4. **Actualización de la Vista:** Una vez que el modelo se actualiza o los datos son recuperados, el controlador puede actualizar la vista para reflejar los cambios.

## Controles y Componentes UI5

SAP nos proporciona un amplio conjunto de controles y ejemplos para hacer más rápido el desarrollo de nuestras aplicaciones.

Ingresando [UI5 Demo Kit](#) podremos encontrar la documentación de los controles y librerías disponibles, teniendo en cuenta la versión de SAPUI5 que estemos utilizando.

La invitación es ingresar y dar un vistazo a esta gran herramienta para que se vayan familiarizando con los diferentes controles que podemos utilizar a lo largo del Bootcamp:



The screenshot shows the SAPUI5 Demo Kit website. The browser address bar displays the URL: `sapui5.netweaver.ondemand.com/sdk/#/controls/filter/List`. The page header includes the SAPUI5 logo, the text "UI5 Demo Kit", a search bar with the placeholder "Search Anything", and the version "Version 1.132.0". The navigation menu contains links for Home, Documentation, API Reference, Samples (which is the active tab), Demo Apps, and Resources. The main content area is divided into a left sidebar and a main grid. The sidebar has a search bar with the text "List" and a list of controls with their counts: Action Sheet (1), Navigation List (1), Overflow Toolbar Button (1), Overflow Toolbar Menu Button (1), Overflow Toolbar Toggle Button (1), Upload Collection (5), and Upload Set (5). The main grid displays four categories of controls: Input (User interaction), Lists (Various list structures), Tables (Simple or more powerful tables), and Pop-Ups (Dialogs and popovers).

Action	Count
Action Sheet	1
Navigation List	1
Overflow Toolbar Button	1
Overflow Toolbar Menu Button	1
Overflow Toolbar Toggle Button	1
Upload Collection	5
Upload Set	5