

Configuración ambiente (BTP - Destination)

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

Log into SAP BTP Trial:

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

Welcome to SAP BTP Trial

Learn how to create and deploy cloud apps and gain access to a comprehensive set of platform services.

[Go To Your Trial Account](#)

Quick Tool Access

- SAP Business Application Studio**
Develop business applications using SAP's next-generation, Web-based IDE
- CLI for BTP**
Manage your trial account using the command-line interface
- APIs for SAP BTP**
Manage, build, and extend the core capabilities of SAP BTP

Welcome to SAP Business Application Studio

Now you can efficiently develop business applications for the Intelligent Enterprise with a powerful and modern development environment.

At the heart of SAP Business Application Studio are the dev spaces. A dev space is a pre-configured private environment (your own "virtual machine" on the cloud) where you can develop, build, test, and run using pre-installed runtimes and tools. You can create a dev space for each of your development scenarios.

[Create Dev Space](#)

SAP Business Application Studio

Create a New Dev Space

dev ✓

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, code 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 your dev space.

SAPUI5 Adaptation (Foundry only)
Allows you to extend SAP Fiori applications using SAPUI5... [more](#)

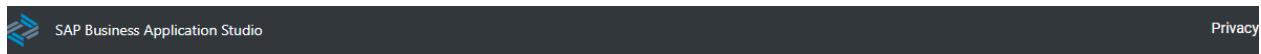
Docker Image Builder
Allows you to build Docker images. Supports build, local testing, and deployment. [more](#)

CDS Graphical Modeler
Allows you to design and analyze SAP CDS services models in a graphical way. [more](#)

CAP Tools
Allows you to develop SAP Fiori applications using SAP CAP... [more](#)

SAP HANA Calculation Editor
Allows you to edit and analyze SAP HANA calculation logic. [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](#).

Icon	Name	Status	Created On	ID	Disk Usage	Action
	dev	RUNNING	02/17/2025 9:12 PM	ws-glfp6	44 KB / 3.8 GB	

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

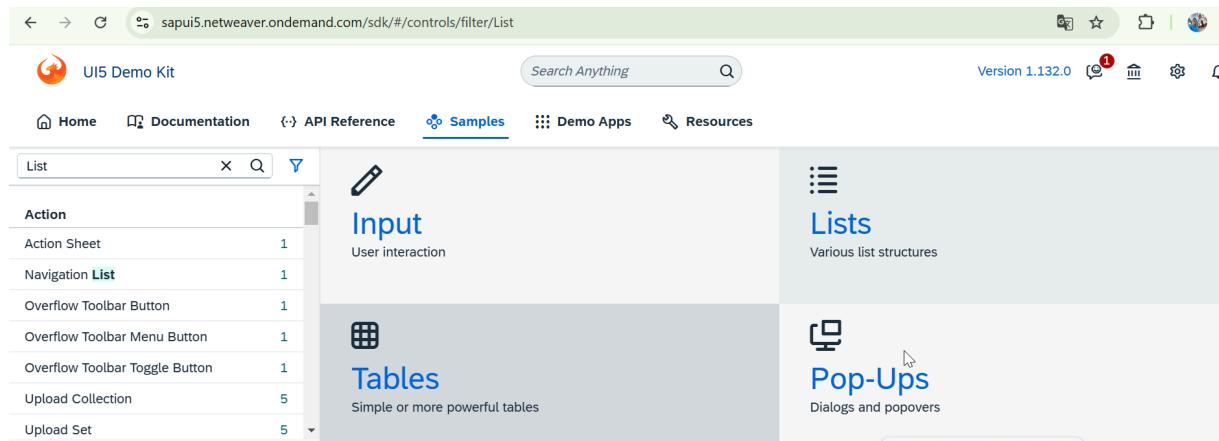
- Interacción del Usuario:** El usuario interactúa con la vista (por ejemplo, haciendo clic en un botón).
- Evento en el Controlador:** El evento se envía al controlador correspondiente, que maneja la lógica de la aplicación.
- Acceso al Modelo:** El controlador puede acceder al modelo para obtener datos o actualizarlo.
- 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 interface at [sapui5.netweaver.ondemand.com/sdk/#/controls/filter/List](http://sapui5.netweaver.ondemand.com/sdk/#/controls/filter>List). The top navigation bar includes links for Home, Documentation, API Reference, Samples (which is selected), Demo Apps, and Resources. A search bar is also present. The main content area displays four sample cards: 'Input' (User interaction), 'Lists' (Various list structures), 'Tables' (Simple or more powerful tables), and 'Pop-Ups' (Dialogs and popovers). On the left, a sidebar lists various UI 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 'Samples' tab is currently active.