



Teoría

UI – Metadata Extensions

ABAP RESTful – Arquitectura Cloud





Contenido

1. UI – Metadata Extensions	3
1.1. Cabecera y Variante	5
1.2. Listado	8
1.3. Campos de Selección Filtros	10
1.4. Ayudas de búsqueda	12
1.5. Disposición del texto	15
1.6. Pestañas – Identification Reference	17
1.7. Acciones	21
1.8. Campos ocultos	25



1. UI – Metadata Extensions

Las Metadata Extensions en CDS (Core Data Services) de SAP permiten la modularización y separación de las definiciones de entidades CDS. Estas extensiones permiten añadir anotaciones semánticas y de interfaz de usuario, facilitando el mantenimiento y la claridad del código. Al separar las definiciones SQL de las anotaciones de la interfaz de usuario en Fiori Elements, se mejora la flexibilidad y comprensión en el desarrollo de aplicaciones empresariales. Las Metadata Extensions son esenciales para crear aplicaciones ricas en funcionalidad y fáciles de mantener.

Travels (2,972)							Accept Travel	Reject Travel	Deduct Discount	Create	Delete	Import	Export	Print	Copy	Reset
	Travel...	Agency ID	Customer ID	Starting Date	End Date	Booking Fee	Total Price	Overall Status								
<input type="checkbox"/>	Travel...	Agency ID	Ryan (600)	Jul 20, 2024	Jul 21, 2024	40.00	EUR	10,098.00	EUR	Accepted						
<input type="checkbox"/>	3000	Fly Low (70027)	Sisko (657)	Jul 20, 2024	Jul 21, 2024	20.00	EUR	5,016.00	EUR	Open						
<input checked="" type="checkbox"/>	2999	Around the World (70019)	Henry (273)	Jul 20, 2024	Jul 21, 2024	40.00	EUR	9,999.00	EUR	Open						
<input type="checkbox"/>	2998	All British Air Planes (70047)	Buchholm (12)	Jul 20, 2024	Jul 21, 2024	60.00	EUR	14,827.00	EUR	Open						
<input type="checkbox"/>	2997	No Return (70020)	Moyano (511)	Jul 20, 2024	Jul 20, 2024	30.00	EUR	7,563.00	EUR	Open						
<input type="checkbox"/>	2996	Pink Panther (70004)	Henry (264)	Jul 20, 2024	Jul 21, 2024	60.00	EUR	14,628.00	EUR	Open						
<input type="checkbox"/>	2995	Special Agency Peru (70021)	Pan (627)	Jul 20, 2024	Jul 20, 2024	10.00	EUR	2,421.00	EUR	Open						
<input type="checkbox"/>	2994	Your Choice (70005)	Simonen (249)	Jul 20, 2024	Jul 21, 2024	20.00	EUR	4,843.00	EUR	Open						
<input type="checkbox"/>	2993	The Ultimate Answer (70042)	Mechler (218)	Jul 20, 2024	Jul 21, 2024	20.00	EUR	4,880.00	EUR	Accepted						
<input type="checkbox"/>	2992	Up 'n' Away (70029)	Leisert (694)	Jul 20, 2024	Jul 21, 2024	40.00	EUR	9,720.00	EUR	Open						
<input type="checkbox"/>	2991	Special Agency Peru (70021)	Goeike (185)	Jul 20, 2024	Jul 21, 2024	40.00	EUR	9,771.00	EUR	Open						
<input type="checkbox"/>	2990	All's Bazar (70016)	Sessler (482)	Jul 20, 2024	Jul 21, 2024	60.00	EUR	14,555.00	EUR	Open						
<input type="checkbox"/>	2989	All's Bazar (70016)	Simonen (258)	Jul 20, 2024	Jul 21, 2024	20.00	EUR	4,915.00	EUR	Open						
<input type="checkbox"/>	2988	Women only (70040)	Koslowski (130)	Jul 20, 2024	Jul 21, 2024	20.00	EUR	4,819.00	EUR	Open						
<input type="checkbox"/>	2987	Asia By Plane (70023)	Pan (627)	Jul 20, 2024	Jul 21, 2024	60.00	EUR	14,516.00	EUR	Open						
<input type="checkbox"/>	2986	Around the World (70019)	Benjamin (449)	Jul 20, 2024	Jul 20, 2024	30.00	EUR	7,274.00	EUR	Open						
<input type="checkbox"/>	2985	Bella Italia (70006)	Dumbach (170)	Jul 20, 2024	Jul 21, 2024	40.00	EUR	9,701.00	EUR	Accepted						
<input type="checkbox"/>	2984	Aussie Travel (70028)														

Habilitar el uso de Metadata Extensions:

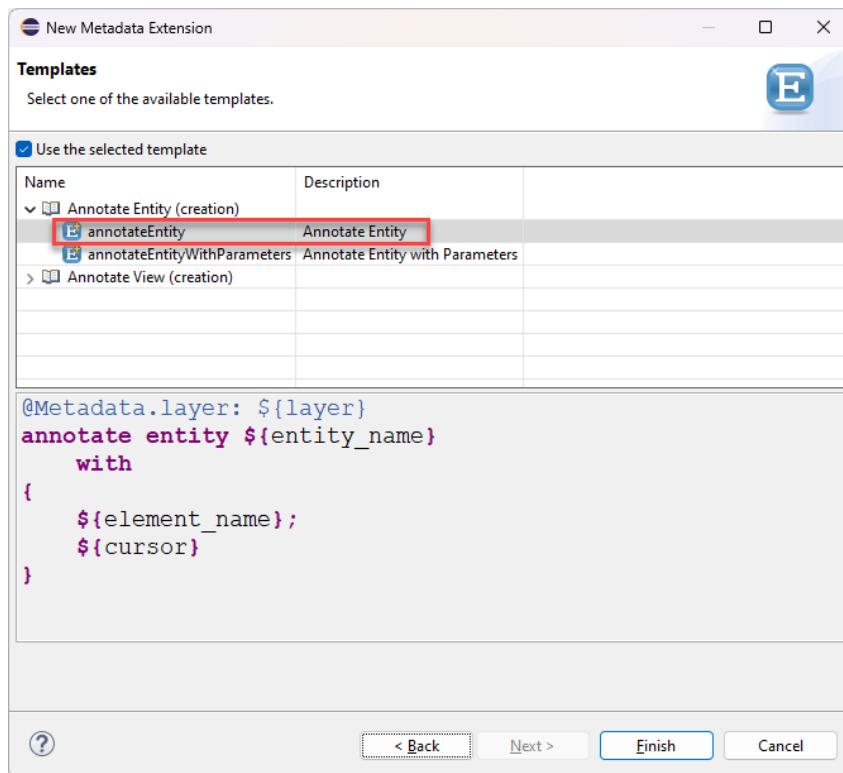
Las Metadata Extensions se crean como objetos separados que se asocian con la entidad CDS original o de consumo, permitiendo mantener las definiciones SQL y las anotaciones de interfaz de usuario en archivos distintos. Para crear Metadata Extensions, primero se define una entidad CDS de consumo que permita la delegación de extensiones mediante la anotación:

- `@Metadata.allowExtensions: true`

Esta anotación se configura a nivel de la cabecera de la entidad CDS de consumo, habilitando las Metadata Extensions que permiten que las aplicaciones basadas en Fiori Elements y SAPUI5 interpreten y rendericen automáticamente las anotaciones definidas en las entidades CDS. Esto facilita la creación de aplicaciones empresariales con interfaces de usuario ricas y funcionales, basadas en datos definidos en el backend.



Para crear un Metadata Extensions, se realiza a través de la carpeta de proyecto con clic derecho en **New** luego en la opción **Other ABAP Repository Object** ubicar la carpeta **Core Data Services** y luego seleccionar **Metadata Extensions** y seleccionar la plantilla **annotateEntity** que se encuentra en la carpeta **Annotate Entity (creation)**.



Luego especificar la entidad o vista CDS a la que se requiere hacer la relación con la instrucción “**annotate entity entity_name_cds with**”.

Sin embargo, existe otra forma de crear un metadata extension, que permite especificar como base de consulta la entidad de consumo, agilizando parte del proceso de creación. Para esto, es necesario ubicar la carpeta **Dictionary** dentro del proyecto y buscar la entidad de consumo requerida dentro de la carpeta **Core Data Service**. Una vez localizada, se hace clic derecho y se selecciona la opción **New MetaData Extension** y seleccionar la plantilla **annotateEntity** que se encuentra en la carpeta **Annotate Entity (creation)**.



Sintaxis:

```
@Metadata.layer: layer
annotate entity entity_name
with
{
    element_name;
}
```

1.1. Cabecera y Variante

En un Metadata Extension se pueden agregar diferentes anotaciones de configuración a nivel de cabecera que permiten interpretar y renderizar la interfaz de usuario de Fiori Elements.

Anotaciones utilizadas a nivel de cabecera:

- **Uso de capas de prioridad:** Estas extensiones se utilizan para determinar la prioridad, cuando hay múltiples extensiones para la misma entidad mediante el concepto de capas. Esta modularidad y priorización facilitan el mantenimiento, la extensibilidad y la organización del código en las aplicaciones SAP RAP, asegurando que las anotaciones correctas se utilicen al renderizar la interfaz de usuario. Estas capas son las siguientes: **#CUSTOMER, #PARTNER, #INDUSTRY, #LOCALIZATION, y #CORE**. Cada capa tiene una prioridad, siendo Customer la más alta y Core la más baja.

Sintaxis:

```
@Metadata.layer: #layer
```

Además, es posible sobrescribir una metadata extension estándar creada por SAP, al crear o configurar una nueva metadata extension con una capa con una prioridad superior.

- **Comportamientos en la interfaz de usuario:** se pueden configurar ciertos comportamientos en los títulos de la interfaz de usuario en Fiori Elements, por medio de la anotación **@UI.option**. Se pueden agrupar varias anotaciones que



comiencen por el mismo concepto, utilizando dos puntos seguidos de llaves. Por ejemplo **@UI: {options}**. Donde para añadir múltiples opciones se deben de separar por una coma. Algunas anotaciones del grupo **@UI**, estudiadas en esta sesión son las siguientes:

- **headerInfo**: este grupo de anotaciones permite configurar los títulos de las interfaces de usuario.

Mediante las opciones:

- **typeName: 'ComponentTitle'**: Al colocar un título de componente se mostrará de esta forma al visualizar un solo registro de una lista de valores en la interfaz de usuario.
- **typeNamePlural: 'ComponentTitle'**: Al colocar un título de componente se mostrará de esta forma al visualizar múltiples registros de una lista de valores en la interfaz de usuario. Ejemplo:

Travels (2,972)		
	Travel ID	Agency ID
<input type="checkbox"/>	3000	Fly Low (70027)
<input checked="" type="checkbox"/>	2999	Around the World (70019)
<input type="checkbox"/>	2998	All British Air Planes (70047)
		Customer ID
		Ryan (600)
		Sisko (657)
		Henry (273)

- **title**: Permite colocar un formato al título mostrado. Mediante las opciones:
 - **type**: que permite seleccionar el tipo de formato en el que se mostrará el título. Los formatos disponibles son los siguientes:
 - **#STANDARD**: se utiliza para definir un título estándar con un solo campo. Y por lo general es el que más se utiliza.
 - **#AS_CONNECTED_FIELDS**: se utiliza para concatenar múltiples campos en el título.
 - **#WITH_INTENT_BASED_NAVIGATION**: se utiliza para habilitar la navegación basada en intenciones en el título.



- **#WITH_NAVIGATION_PATH:** se utiliza para habilitar la navegación directa a una ruta específica en el título.
- **#WITH_URL:** se utiliza para incluir un enlace URL en el título.

```

@ #AS_CONNECTED_FIELDS (annotation)
@ #STANDARD (annotation)
@ #WITH_INTENT_BASED_NAVIGATION (annotation)
@ #WITH_NAVIGATION_PATH (annotation)
@ #WITH_URL (annotation)

```

- **value:** permite especificar un componente de la entidad para colocar como título al visualizar los detalles. El valor de un registro de la lista de valores de la interfaz de usuario. Ejemplo:

2999			
Travel Master Data			
Travel ID: 2999	Customer ID: Sisko (657)	End Date: Jul 21, 2024	Total Price: 5.016,00 EUR
Agency ID: Around the World (70019)	Starting Date: Jul 20, 2024	Booking Fee: 20,00 EUR	Description: Business Trip to Italy

- **presentationVariant:** permite configurar la presentación por defecto de las variantes de selección en una interfaz de usuario. Mediante las opciones:
 - **sortOrder:** permite configurar el tipo de orden que se mostrará una lista de valores en una interfaz de usuario en Fiori Elements. Mediante las opciones:
 - **by:** permite especificar el componente donde se realizará el orden.
 - **direction:** permite establecer el tipo de orden ascendente o descendiente con las opciones **#ASC** y **#DESC**, respectivamente.
 - **visualizations:** permite configurar el formato de los registros en la interfaz de usuario. Mediante la opción:
 - **type:** para seleccionar los diferentes formatos disponibles:



- **#AS_LINEITEM:** Esta anotación se utiliza para visualizar datos en forma de una lista de elementos de línea.
- **#AS_DATAPOINT:** se utiliza para visualizar un punto de datos específico.
- **#AS_CHART:** se utiliza para visualizar datos en forma de una lista de elementos de línea.

```
#AS_CHART (annotation)
#AS_DATAPOINT (annotation)
#AS LINEITEM (annotation)
```

Ejemplo:

```
@Metadata.layer: #CUSTOMER
@UI: { headerInfo: { typeName: 'Component',
                     typeNamePlural: 'Components',
                     title: { type: #STANDARD,
                             value: 'Component' } },
       presentationVariant: [{ sortOrder: [{ by: 'Component',
                                             direction: #DESC }],
                             visualizations: [{ type: #AS_LINEITEM }] }]
     }
annotate entity entity_name
  with
  {
    element_name;
  }
```

1.2. Listado

En esta lección se aborda la configuración de metadata extensions en SAP RAP (Restful ABAP Programming) para representar la interfaz de usuario, específicamente en la disposición y presentación de columnas en un listado o tabla. Utilizando anotaciones, se define cómo se deben ordenar y presentar las columnas en la interfaz de



usuario, asegurando una experiencia óptima y ajustable a diferentes dispositivos.

Anotaciones utilizadas en los elementos o componentes:

Se pueden agregar configuraciones específicas a componentes vinculadas a la entidad de consumo. Mediante el uso de anotaciones para agregar propiedades y mejorar la funcionalidad de la entidad en diferentes contextos, como en la interfaz de usuario. En esta sección se estudiarán las siguientes:

- **@UI.dataFieldDefault: [{ importance: # }]:** Esta anotación se utiliza para definir campos de datos en la interfaz de usuario, especificando la importancia del campo. La importancia determina cómo y cuándo se muestra el campo en diferentes dispositivos y contextos. Se puede ajustar la importancia (#LOW, #MEDIUM y #HIGH) para controlar la visibilidad y relevancia del campo en la interfaz de usuario.
- **@UI.lineItem:** esta anotación se utiliza para definir cómo se presentan las columnas de una entidad en una lista o tabla dentro de la interfaz de usuario. Entre las configuraciones más importantes de esta notación tenemos:
 - **position:** permite especificar la posición de cada columna en la tabla utilizando la propiedad. Se recomienda por buenas prácticas que la posición empiece con el valor de 10, ya que el sistema interpreta las posiciones de manera consecutiva. Esto permite que, si se debe realizar un ajuste en el futuro donde se necesite agregar un elemento adicional, se puedan utilizar las posiciones entre 10 y 20 sin tener que reorganizar todas las posiciones.
 - **importance:** permite definir la relevancia de cada columna, con opciones como #HIGH, #MEDIUM y #LOW. Esto determina en qué dispositivos (móviles, tablets, escritorio) se mostrarán las columnas, optimizando la visualización según el dispositivo.
 - **label:** permite asignar una etiqueta a la columna que se mostrará en la interfaz de usuario. Si no se especifica, la



etiqueta se hereda del elemento de datos definido en la tabla de persistencia.

Ejemplo:

```

@Metadata.layer: #CUSTOMER
@UI: { headerInfo: { typeName: 'Component',
                     typeNamePlural: 'Components',
                     title: { type: #STANDARD,
                               value: 'Component' } },
       presentationVariant: [{ sortOrder: [{ by: 'Component',
                                              direction: #DESC }],
                               visualizations: [{ type: #AS_LINEITEM }]
                           }]
     }
annotate entity entity_name
with
{
  @UI.lineItem: [{ position: 10,
                  importance: #HIGH,
                  label: 'Component' }]
}
Component;
}

```

1.3. Campos de Selección | Filtros

Habilitar la búsqueda avanzada en los componentes de la entidad de consumo permite filtrar datos de manera eficiente y precisa. Y utilizando Metadata Extensions y anotaciones específicas, se puede controlar la inclusión y presentación de estos campos en las búsquedas dentro de la interfaz de usuario.

Travels (5)									
	Travel...	Agency ID	Customer ID	Starting Date	End Date	Booking Fee	Overall Status	Accept Travel	Reject Travel
<input type="checkbox"/>	2958	Asia By Plane (70023)	Montero (489)	Jul 19, 2024	Jul 19, 2024	30.00	SGD Open		
<input type="checkbox"/>	2072	Asia By Plane (70023)	Montero (486)	Jul 19, 2024	Jul 19, 2024	20.00	USD Open		
<input type="checkbox"/>	1190	Asia By Plane (70023)	Montero (484)	Jul 19, 2024	Jul 19, 2024	20.00	SGD Open		
<input type="checkbox"/>	892	Asia By Plane (70023)	Montero (484)	Jul 20, 2024	Jul 20, 2024	3,000	JPY Open		
<input type="checkbox"/>	110	Asia By Plane (70023)	Montero (495)	Jul 16, 2024	Jul 18, 2024	60.00	EUR Open		



Implementación de Campos de Selección:

Se pueden incluir campos de búsqueda adicionales para los componentes vinculados a la entidad raíz que tenga habilitadas las funciones de búsqueda avanzada en la interfaz de usuario. Por medio de la anotación **@UI.selectionField: [{ option }]**. Donde las posibles opciones que se pueden configurar son:

- **element:** Define un elemento específico que debe ser considerado en la pantalla de selección.
- **exclude: true:** Indica que un campo debe ser excluido de la pantalla de selección.
- **position:** Permite asignar posiciones para el orden en el que se mostrarán en la interfaz de usuario. Se recomienda por buenas prácticas que la posición empiece con el valor de 10, ya que el sistema interpreta las posiciones de manera consecutiva. Esto permite que, si se debe realizar un ajuste en el futuro donde se necesite agregar un elemento adicional, se puedan utilizar las posiciones entre 10 y 20 sin tener que reorganizar todas las posiciones.

The screenshot shows a standard SAP Fiori application interface. At the top, there is a search bar with placeholder text "plane montero" and a dropdown menu showing "Editing Status: All". Below the search bar are four input fields labeled "Travel ID:", "Agency ID:", "Customer ID:", and "Overall Status:". The "Travel ID:" field is highlighted with a red box. Below these fields is a table titled "Travels (5)". The table has columns: "Travel ID", "Agency ID", "Customer ID", "Starting Date", "End Date", "Booking Fee", and "Overall Status". The data in the table is as follows:

Travel ID	Agency ID	Customer ID	Starting Date	End Date	Booking Fee	Overall Status
2958	Asia By Plane (70023)	Montero (489)	Jul 19, 2024	Jul 19, 2024	30.00	SGD Open
2072	Asia By Plane (70023)	Montero (486)	Jul 19, 2024	Jul 19, 2024	20.00	USD Open
1190	Asia By Plane (70023)	Montero (484)	Jul 19, 2024	Jul 19, 2024	20.00	SGD Open
892	Asia By Plane (70023)	Montero (484)	Jul 20, 2024	Jul 20, 2024	3,000	JPY Open
110	Asia By Plane (70023)	Montero (495)	Jul 16, 2024	Jul 18, 2024	60.00	EUR Open

- **qualifier:** Proporciona un calificador para diferenciar elementos que pueden tener nombres similares o propiedades compartidas.

```

@element: " (annotation)
@exclude: true (annotation)
@position: (annotation)
@qualifier: " (annotation)

```

Ejemplo



```
@Metadata.layer: #CUSTOMER
@UI: { headerInfo: { typeName: 'Component',
                     typeNamePlural: 'Components',
                     title: { type: #STANDARD,
                             value: 'Component' } },
       presentationVariant: [{ sortOrder: [{ by: 'Component',
                                              direction: #DESC }],
                               visualizations: [{ type: #AS_LINEITEM }] }]
     }
annotate entity entity_name
with
{
  @UI.lineItem: [{ position: 10,
                  importance: #HIGH,
                  label: 'Component' }]}
  @UI.selectionField: [{ position: 20 }]
Component;
  @UI: {lineItem: [{ position: 20,
                     importance: #HIGH,
                     label: 'Travel ID' }],
        selectionField: [{ position: 20 }] }
Component;
}
```

Las anotaciones **@UI.lineItem** y **@UI.selectionField** forman parte del mismo grupo de interfaz de usuario (**@UI**). Por lo que se pueden agrupar para mejorar la legibilidad y organización del código.

1.4. Ayudas de búsqueda

Las ayudas de búsqueda (value helps) en SAP RAP (Restful ABAP Programming) son herramientas que mejoran la eficiencia y precisión en la selección de valores en la interfaz de usuario.





Estas ayudas permiten a los usuarios seleccionar valores predefinidos de listas vinculadas a entidades de datos relevantes en el backend. Además, es posible agregar validaciones, asegurando que los datos introducidos sean correctos y consistentes.

Agency ID

Search and Select **Define Conditions**

Search Go Show Filters

Items (50)

<input type="checkbox"/>	Agency ID	Agency Name	Street	Postal Code	City
<input type="checkbox"/>	70001	Sunshine Travel	134 West Street	54323	Rochester
<input type="checkbox"/>	70002	Fly High	Berliner Allee 11	40880	Duesseldorf
<input type="checkbox"/>	70003	Happy Hopping	Calvinstr. 36	13467	Berlin
<input type="checkbox"/>	70004	Pink Panther	Auf der Schanz 54	65936	Frankfurt
<input type="checkbox"/>	70005	Your Choice	Gustav-Jung-Str. 425	90455	Nuernberg
<input type="checkbox"/>	70006	Bella Italia	Via Marconi 123	00139	Roma
<input type="checkbox"/>	70007	Hot Socks Travel	224 Balnagask Rd	8053	Sydney
<input type="checkbox"/>	70008	Burns Nuclear	14 Science Park Drive	118228	Singapore
<input type="checkbox"/>	70009	Honauer Reisen GmbH	Baumgarten 8	4212	Neumarkt
<input type="checkbox"/>	70010	Travel from Walldorf	Altonaer Str. 24	10557	Berlin
<input type="checkbox"/>	70011	Voyager Enterprises	Gustavslundsvaegen 151	70563	Stockholm
<input type="checkbox"/>	70012	Ben McCloskey Ltd.	74 Court Oak Rd	B17 9TN	Birmingham

No Items or Conditions Selected

Implementación de las Ayudas de búsqueda en la interfaz de usuario por medio de MetaData Extensions:

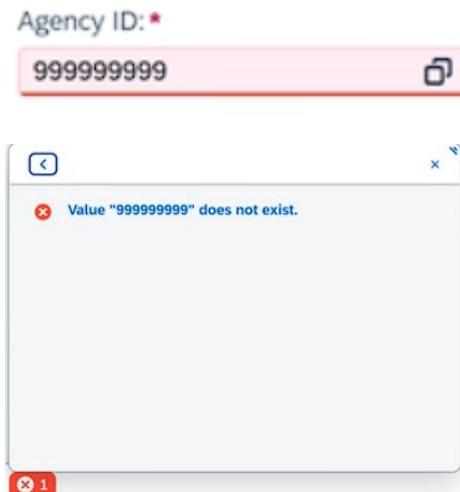
Las ayudas de búsqueda en la interfaz de usuario se implementan mediante Metadata Extensions. Esto se realiza agregando la anotación **@Consumption.valueHelpDefinition**, a los componentes requeridos para utilizar la ayuda de búsqueda, vinculada a la entidad cds correspondiente. Para esto es necesario configurar los siguientes parámetros de la propiedad **entity**:

- **name:** Define el nombre de la entidad CDS que se utilizará como fuente de datos para la ayuda de búsqueda. La entidad CDS que se utilizará debe estar bien configurada para soportar la búsqueda avanzada con la anotación **@Search.searchable: true**, en el encabezado de la entidad.
- **element:** Identifica el elemento o componente específico dentro de la entidad que se utilizará para la búsqueda. Recordando que dentro de la entidad fuente para la ayuda de



búsqueda debe existir el componente que se desea vincular y poseer la anotación `@Search.defaultSearchElement: true`. Para habilitar las búsquedas avanzadas en dichos elementos.

- **useForValidation:** Habilita la validación del valor en la capa del cliente, al establecer la propiedad con el valor true. Lo que asegura que en el caso de que un usuario introduzca un valor incorrecto, se muestre un mensaje de error y el campo reciba el foco.



Algunos ejemplos de entidades CDS utilizadas como fuentes de datos para las ayudas de búsqueda en componentes de la interfaz de usuario son: `/DMO/I_Customer_StdVH`, `/DMO/I_Agency_StdVH`, `/DMO/I_Overall_Status_VH` y `I_CurrencyStdVH`.

Ejemplo:

```
@Metadata.layer: #CUSTOMER
@UI: { headerInfo: { typeName: 'Component',
                     typeNamePlural: 'Components',
                     title: { type: #STANDARD,
                             value: 'Component' } },
       presentationVariant: [{ sortOrder: [{ by: 'Component',
                                              direction: #DESC }],
                             visualizations: [{ type: #AS_LINEITEM }] }]
     }
annotate entity entity_name
with
{
```



```
@Consumption.valueHelpDefinition: [{ entity: { name:  
'entity_name',  
element: 'component_name' },  
useForValidation: true }]  
 @UI: {lineItem: [{ position: 10,  
importance: #HIGH,  
label: 'Travel ID' }],  
selectionField: [{ position: 20 }] }  
Component;  
}
```

1.5. Disposición del texto

La disposición del texto en la interfaz de usuario de SAP RAP (Restful ABAP Programming) permite controlar cómo se presentan los textos asociados a los códigos en la aplicación. Utilizando anotaciones y propiedades específicas, los desarrolladores pueden definir si el texto se muestra antes, después o en lugar del código o valor del componente obtenido de las consultas de las bases de datos por medio de la entidad de consumo. Esta funcionalidad mejora la presentación y comprensión de los datos, proporcionando una experiencia de usuario más intuitiva y amigable. Las configuraciones se aplican tanto en las tablas y pantallas de selección como en los formularios de edición, asegurando coherencia en toda la aplicación.



Implementación de la Disposición del texto en la interfaz de usuario a través de las MetaData Extensions:

Se puede controlar la disposición del texto en la interfaz de usuario para mejorar la presentación y comprensión de los datos. Para implementar las diferentes disposiciones es necesario:

- **Proyección de los textos:** La entidad de consumo debe poseer componentes que tengan la anotación **@ObjectModel.text.element**. La cual permitirá mostrar el texto vinculado a la entidad por medio de una asociación.



- **Configurar el tipo de disposición:** Esto se realiza por medio de la propiedad **textArrangement**, de la anotación **@UI**. La cual puede tomar los siguientes valores:

- **#TEXT_FIRST:** Muestra el texto primero, seguido del código.
- **#TEXT_LAST:** Muestra el código primero, seguido del texto.
- **#TEXT_ONLY:** Muestra solo el texto.
- **#TEXT_SEPARATE:** Muestra el texto y el código por separado.

Ejemplo:

```
@Metadata.layer: #CUSTOMER
@UI: { headerInfo: { typeName: 'Component',
                     typeNamePlural: 'Components',
                     title: { type: #STANDARD,
                               value: 'Component' } },
       presentationVariant: [{ sortOrder: [{ by: 'Component',
                                              direction: #DESC }],
                               visualizations: [{ type: #AS_LINEITEM }] }]
     }
annotate entity entity_name
  with
  {
    @Consumption.valueHelpDefinition: [{ entity: { name:
      'entity_name',
      element: 'component_name' },
                                         useForValidation: true }]
  }
@UI: {lineItem: [{ position: 10,
                  importance: #HIGH,
                  label: 'Travel ID' }],
      selectionField: [{ position: 20 }],
      textArrangement: #TEXT_LAST
    }
Component;
```



{

1.6. Pestañas – Identification Reference

La creación de vistas de detalles y pestañas utilizando identification reference en SAP RAP es una práctica esencial para organizar y presentar datos de manera efectiva en la interfaz de usuario. Mediante el uso de metadata extensions y anotaciones específicas, se puede controlar qué elementos se muestran y cómo se disponen en las vistas de detalles. Esta funcionalidad mejora la usabilidad y experiencia del usuario, permitiendo una navegación clara y una gestión eficiente de los datos.

Travel Master Data				
Travel ID: 2999	Customer ID: Sisko (657)	End Date: Jul 21, 2024	Total Price: 5.016.00 EUR	Status: Open
Agency ID: Around the World	Starting Date: Jul 20, 2024	Booking Fee: 20.00 EUR	Description: Business Trip to Italy	

Creación de Pestañas de Detalle:

Cuando se selecciona un elemento de una lista para ver sus detalles, se puede navegar a una pestaña específica utilizando el concepto de identification reference. En la interfaz de usuario, estas pestañas se presentan como secciones de detalle que permiten a los usuarios ver y editar información detallada sobre el elemento seleccionado.

Se utilizan propiedades específicas dentro de las anotaciones **@UI.facet** y **@UI.identification** para definir y organizar los elementos que aparecerán en la vista de detalles. Y se explican a continuación:

- **@UI.facet**: Esta anotación se agrega en alguno de los componentes del metadata extension permitiendo habilitar la vista de los detalles en la interfaz de usuario. Por lo general se agrega a un componente clave ID ó UUID e incluye propiedades para su configuración como:
 - **id**: Permite colocar un identificador interno de la pestaña utilizando el formato UpperCamelCase. Este



tipo de identificador por lo general se utiliza cuando se van a implementar múltiples pestañas.

- **purpose:** Esta anotación contiene el formato de la pestaña de detalles y por lo general se utiliza la opción **#STANDARD**. Aunque las posibles opciones permitidas son las siguientes:
 - **#FILTER:** Se utiliza para definir una sección de filtros en la interfaz de usuario.
 - **#HEADER:** Define una sección de encabezado en la interfaz de usuario.
 - **#QUICK_CREATE:** Se utiliza para facilitar la creación rápida de registros.
 - **#QUICK_VIEW:** Define una vista rápida para mostrar información esencial.
 - **#STANDARD:** Se utiliza para definir una sección estándar en la interfaz de usuario.

#FILTER (annotation)
#HEADER (annotation)
#QUICK_CREATE (annotation)
#QUICK_VIEW (annotation)
#STANDARD (annotation)

- **type:** Define el tipo de visualización de la pestaña y por lo general se utiliza la opción **#IDENTIFICATION_REFERENCE**. Aunque las posibles opciones permitidas son las siguientes:
 - **#ADDRESS_REFERENCE:** Se utiliza para mostrar direcciones.
 - **#BADGE_REFERENCE:** Se utiliza para mostrar insignias o distintivos.
 - **#CHART_REFERENCE:** Se utiliza para mostrar gráficos.
 - **#COLLECTION:** Se utiliza para mostrar colecciones de elementos.



- **#CONTACT_REFERENCE:** Se utiliza para mostrar información de contacto.
- **#DATAPOINT_REFERENCE:** Se utiliza para mostrar puntos de datos individuales.
- **#FIELDGROUP_REFERENCE:** Se utiliza para agrupar y mostrar campos relacionados.
- **#HEADERINFO_REFERENCE:** Se utiliza para mostrar información de encabezado.
- **#IDENTIFICATION_REFERENCE:** Se utiliza para mostrar detalles de identificación.
- **#LINEITEM_REFERENCE:** Se utiliza para mostrar líneas de ítems en detalle.
- **#NOTE_REFERENCE:** Se utiliza para mostrar notas o comentarios.
- **#PRESENTATIONVARIANT_REFERENCE:** Se utiliza para mostrar variantes de presentación.
- **#SELECTIONPRESENTATIONVARIANT_REFERENCE:** Se utiliza para mostrar variantes de presentación con opciones de selección.
- **#STATUSINFO_REFERENCE:** Se utiliza para mostrar información de estado.
- **#URL_REFERENCE:** Se utiliza para mostrar URLs.

```
#ADDRESS_REFERENCE (annotation)
#BADGE_REFERENCE (annotation)
#CHART_REFERENCE (annotation)
#COLLECTION (annotation)
#CONTACT_REFERENCE (annotation)
#DATAPOINT_REFERENCE (annotation)
#FIELDGROUP_REFERENCE (annotation)
#HEADERINFO_REFERENCE (annotation)
#IDENTIFICATION_REFERENCE (annotation)
#LINEITEM_REFERENCE (annotation)
#NOTE_REFERENCE (annotation)
#PRESENTATIONVARIANT_REFERENCE (annotation)
#SELECTIONPRESENTATIONVARIANT_REFERENCE (annotation)
#STATUSINFO_REFERENCE (annotation)
#URL_REFERENCE (annotation)
```

- **label:** Etiqueta visible para el usuario.



- **position:** Posición de la pestaña en la interfaz de usuario. Este elemento no es obligatorio si se mostrará una sola pestaña.

Para habilitar múltiples pestañas se debe crear otro conjunto de las propiedades mencionadas anteriormente que se encuentran definidas dentro de las llaves {}, y colocar el siguiente conjunto después de una coma.

- **@UI.identification:** Especifica los elementos que aparecerán en la vista de detalles luego de haber incluido el valor #IDENTIFICATION_REFERENCE, en la propiedad type, de la anotación @UI.facet. Incluye propiedades como:
 - **position:** Define la posición del elemento en la vista de detalles.
 - **label:** Etiqueta del elemento, heredada del elemento de datos si no se especifica.

Ejemplo:

```
@Metadata.layer: #CUSTOMER
@UI: { headerInfo: { typeName: 'Component',
                     typeNamePlural: 'Components',
                     title: { type: #STANDARD,
                               value: 'Component' } },
       presentationVariant: [{ sortOrder: [{ by: 'Component',
                                              direction: #DESC }],
                               visualizations: [{ type: #AS_LINEITEM }] }
                             }]
}
annotate entity entity_name
with
{
  @UI.facet: [{ id: 'InternalIdentifier',
                purpose: #STANDARD,
                type: #IDENTIFICATION_REFERENCE, label: 'UserLabel', position:
                10 }, { id, purpose, type, label, position }]
  ComponentID;
```



```
@Consumption.valueHelpDefinition: [{ entity: { name:  
'entity_name',  
element: 'component_name' },  
useForValidation: true }]  
 @UI: {lineItem: [{ position: 10,  
importance: #HIGH,  
label: 'Travel ID' }],  
selectionField: [{ position: 10 }],  
identification: [{ position: 10 }],  
textArrangement: #TEXT_LAST  
}  
Component;  
}
```

1.7. Acciones

La creación y gestión de acciones en la interfaz de usuario de SAP RAP es esencial para proporcionar una experiencia interactiva y eficiente a los usuarios. Mediante la definición de botones en las metadata extensions y la vinculación de estos a métodos ABAP, se pueden realizar diversas operaciones de manera dinámica. La posibilidad de replicar estas acciones en diferentes vistas asegura que los usuarios puedan interactuar con los datos de forma coherente y accesible desde cualquier parte de la aplicación.

Definición de Acciones en Metadata Extensions:

Las acciones en la interfaz de usuario se representan mediante botones. Estos botones se habilitan cuando se selecciona uno o más registros en una lista.

End Date	Booking Fee	Overall Status
Jul 21, 2024	40.00	EUR Accepted
Jul 21, 2024	20.00	EUR Open



La acción de crear (create) siempre está habilitada, permitiendo a los usuarios añadir nuevos registros. Otras acciones estándar incluyen la eliminación (delete), que se habilita al seleccionar un registro.

Accept Travel	Reject Travel	Deduct Discount	Create	Delete	Ctrl+D				
End Date	Booking Fee		Overall Status						
Jul 21, 2024	40.00		EUR	Accepted					
Jul 21, 2024	20.00		EUR	Open					

Las acciones se definen en las metadata extendidas en Eclipse. Por medio de la anotación **@UI.LineItem**, para crear las propiedades que definirán los botones. Estos se agregan en un conjunto aparte del definido en la sección de listado al colocar una coma y colocar las propiedades faltantes agrupándolas entre llaves **{}** separados por una coma. Dichas propiedades son las siguientes:

- **type:** Define el tipo de elemento que se mostrará en la interfaz de usuario dentro de una tabla o lista. La opción **#FOR_ACTION**, es lo que define al botón dentro de la interfaz de usuario.
- **dataAction:** Se utiliza junto con el tipo **#FOR_ACTION** para vincular un botón de acción a un método ABAP específico que define la lógica de la acción. Las acciones están vinculadas a métodos ABAP que se definirán en la clase de implementación del comportamiento (behavior pool). Los métodos deben estar definidos en la behavior definition y luego implementados en la clase ABAP.

Es posible agregar una etiqueta al botón utilizando la propiedad **label** y definir su posición con la propiedad **position**. Estas propiedades ya han sido detalladamente explicadas en la sección de listado de esta unidad.

Un aspecto importante es que las mismas acciones pueden ser replicadas en la vista o pestaña de detalles ampliando las propiedades de la anotación **@UI.identification**, asegurando que los botones estén disponibles en todos los contextos necesarios. Esto se realiza agregando las mismas definiciones de los botones encerradas entre llaves **{}** en la sección correspondiente a la anotación mencionada.



Travel ID:	2997	Customer ID:	Buchholz (12)	End Date:	Jul 21, 2024	Total Price:	14,827.00 EUR	Status:	Open
Agency ID:	No Return	Starting Date:	Jul 20, 2024	Booking Fee:	60.00 EUR	Description:	Business Trip to Italy		

Ejemplo:

```

@Metadata.layer: #CUSTOMER
@UI: { headerInfo: { typeName: 'Component',
                     typeNamePlural: 'Components',
                     title: { type: #STANDARD,
                             value: 'Component' } },
       presentationVariant: [{ sortOrder: [{ by: 'Component',
                                             direction: #DESC }],
                               visualizations: [{ type: #AS_LINEITEM }] }
     ]
}
annotate entity entity_name
with
{
  @UI.facet: [{ id: 'InternalIdentifier',
                purpose: #STANDARD,
                type: #IDENTIFICATION_REFERENCE, label: 'UserLabel', position: 10
              }, { id, purpose, type, label, position }]
  ComponentID;

  @Consumption.valueHelpDefinition: [{ entity: { name:
    'entity_name',
    element: 'component_name' },
    useForValidation: true }]
}

@UI: {lineItem: [{ position: 10,
                  importance: #HIGH,
                  label: 'Travel ID' },
                 { type: #FOR_ACTION,
                   dataAction: 'methodName',
                   label: 'ButtonName',
                   position: 10},
                 { type: #FOR_ACTION,
                   dataAction: 'methodName2',

```



```
    label: 'ButtonName2',
    position: 10}
],
selectionField: [{ position: 10 }],
identification: [{ position: 10 },
{ type: #FOR_ACTION,
  dataAction: 'methodName',
  label: 'ButtonName',
  position: 10},
{ type: #FOR_ACTION,
  dataAction: 'methodName2',
  label: 'ButtonName2',
  position: 10}],
textArrangement: #TEXT_LAST
}
Component;
}
```

1.8. Campos ocultos

La ocultación de campos en la interfaz de usuario de SAP RAP (Restful ABAP Programming) permite a los desarrolladores controlar qué datos se muestran a los usuarios, asegurando una presentación clara y relevante de la información. Utilizando la anotación `@UI.hidden`, se pueden ocultar completamente ciertos campos en la interfaz de usuario, impidiendo que los usuarios puedan visualizarlos o los añadirlos nuevamente a través de personalizaciones de la interfaz. Esta técnica mejora la usabilidad y la eficiencia, presentando solo la información necesaria y pertinente al usuario, adaptando la visualización según el tamaño de la pantalla y las necesidades de la aplicación.

Sintaxis:



```
@UI.hidden: true  
component;
```

Consideraciones para Ocultar Campos:

- Es importante decidir qué información debe estar oculta basándose en la relevancia y sensibilidad de los datos.
- Algunos campos pueden dejarse visibles para que los usuarios decidan si quieren incluirlos en sus vistas personalizadas.
- La ocultación de campos asegura que solo la información necesaria y pertinente se presenta al usuario, mejorando la claridad y eficiencia en el uso de la aplicación.