



# LOGALI

## Teoría

### **Aceleradores RAP**

---

ABAP RESTful – Arquitectura Cloud





## Contenido

<b>1. Aceleradores RAP</b>	<b>3</b>
<b>1.1. Conceptos</b>	<b>3</b>
<b>1.2. Automatización artefactos RAP</b>	<b>4</b>
<b>1.3. Análisis y ajustes</b>	<b>11</b>



## 1. Aceleradores RAP

### 1.1. Conceptos

Como se ha estudiado con anterioridad la creación de una aplicación RAP implica de manera resumida la generación de múltiples artefactos de desarrollo como lo son:

- La definición de las tablas de persistencia y draft que almacenarán los datos.
- Utilizar Core Data Services (CDS) para modelar las relaciones, proyecciones y extensiones de metadatos (metadata extensions).
- Definir las vistas de proyección y las relaciones entre las distintas entidades.
- Creación de la definición de comportamiento (behavior definition) junto con su proyección. Esto incluye la definición de cómo interactúan los datos y cómo se comportan las entidades por medio de la clase Behavior Pool.
- Y configurar los servicios necesarios para la aplicación, permitiendo que los datos sean accesibles a través de diferentes interfaces. Se definen y vinculan los servicios mediante service definition y service binding.

La generación manual de estos artefactos puede ser repetitiva y consumir mucho tiempo. Sin embargo es posible utilizar el proyecto **RAP Generator** que permite agilizar el proceso.

El RAP Generator es una herramienta innovadora desarrollada para facilitar la creación de aplicaciones RAP (RESTful ABAP Programming Model). Donde su principal objetivo es facilitar a los desarrolladores la creación de la pila completa de objetos necesarios para implementar un objeto de negocio RAP. El objetivo es generar la mayoría del código repetitivo (boiler plate coding) para que los desarrolladores puedan comenzar más rápidamente a implementar la lógica de negocio.



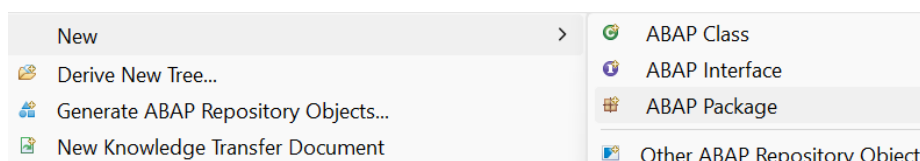
Esta herramienta fue desarrollada y es mantenida por **Andre Fischer**, donde trabaja en la gestión de productos para SAP Gateway desde el lanzamiento del producto en 2011 y ha estado trabajando en temas relacionados con SAP desde 1995. Es un experto en la materia con una vasta experiencia en SAP Gateway, servicios OData y RAP. Es posible consultar otros proyectos desarrollados por él a través de su perfil de GitHub [github.com/anfisc](https://github.com/anfisc).



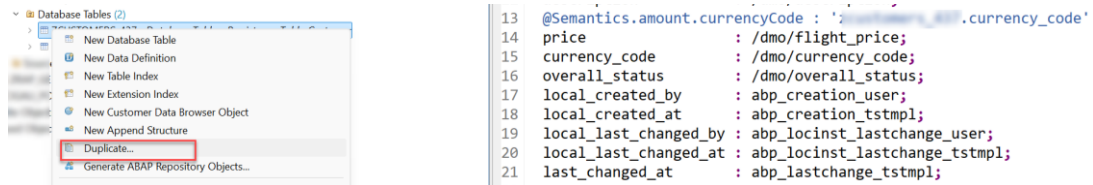
## 1.2. Automatización artefactos RAP

Para la creación automática de un proyecto rap por medio de la herramienta RAP Generator es necesario realizar los siguientes pasos:

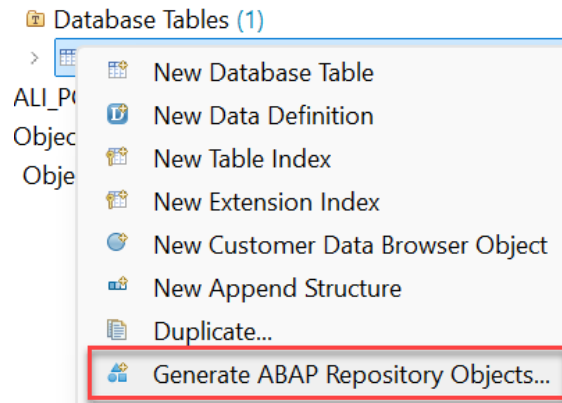
- Crear un nuevo paquete de desarrollo en el entorno de ABAP, Este paquete contendrá todos los artefactos generados por el RAP Generator.



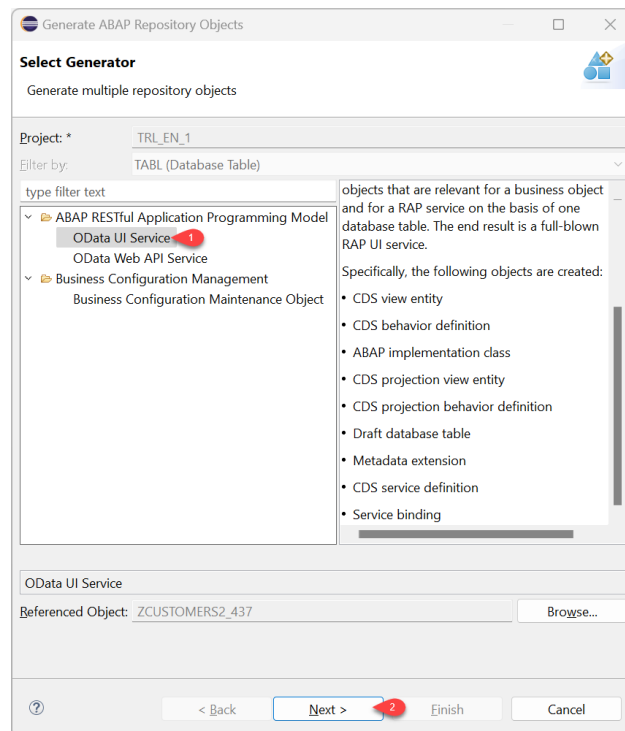
- Realizar una copia de una tabla de persistencia existente que contenga los componentes de auditoría: **local\_created\_by**, **local\_created\_at**, **local\_last\_changed\_by**, **local\_last\_changed\_at** y **last\_changed\_at**. Para luego activarla en el paquete recién creado.



- Seleccionar la opción "Generate Repository Objects" en el entorno de ABAP, se generan los artefactos necesarios a partir de la tabla de persistencia duplicada con clic derecho.



- De la ventana que se mostrará a continuación seleccionar la opción **OData UI Service** y continuar con el proceso con el botón Next.



La opción servicio de UI OData hace posible consumir un servicio RAP con una UI de Fiori Elements u otros clientes de



UI. Y crea toda la información específica de la UI que está anotada en los artefactos RAP en los metadatos del servicio OData. Un servicio de UI puede ser previsualizado con la Vista Previa de la Aplicación de Fiori Elements en el artefacto de vinculación del servicio. Este generador crea todos los objetos de desarrollo que son relevantes para un objeto de negocio y para un servicio RAP sobre la base de una tabla de base de datos. El resultado final es un servicio de UI RAP completo. Específicamente, se crean los siguientes objetos:

- Entidad de vista CDS.
  - Definición de comportamiento CDS.
  - Clase de implementación ABAP.
  - Entidad de vista de proyección CDS.
  - Definición de comportamiento de proyección CDS.
  - Tabla de base de datos de borrador.
  - Extensión de metadatos.
  - Definición de servicio CDS.
  - Vinculación de servicio.
- En la siguiente ventana se debe seleccionar el paquete creado para la generación de la aplicación RAP.

- Luego es posible en la siguiente ventana personalizar el nombre o alias de algunos de los artefactos que serán generados por medio de la herramienta en su respectiva sección:
  - **General:** Cambiar el nombre del proyecto, además de agregar un prefijo o sufijo para los artefactos generados.



**Generate ABAP Repository Objects**

**Configure Generator**  
Specify values for the generator, 'OData UI Service'

**Details for RAP artifacts generation**

**General**

The object on which the root entity of the business object is based upon.

Referenced Object:

Project Name:

Artifacts Prefix:

Artifacts Suffix:

< Back Next > Finish Cancel

- **Data Model:** Cambiar el nombre y alias de la entidad raíz.

**Generate ABAP Repository Objects**

**Configure Generator**  
Specify values for the generator, 'OData UI Service'

**Details for RAP artifacts generation**

**Data Model**

Provide the name of the CDS data definition for the root entity and an alias. CDS data definitions are used to define and organize the data model.

CDS Entity Name: \*

CDS Entity Alias:

< Back Next > Finish Cancel

- **Behavior:** Cambiar el nombre del Behavior Definition y tabla Draft.



- **Service Projection Entity:** Cambiar el nombre de la vista de proyección de consumo.

- **Service Projection Behavior:** Cambiar el nombre de la clase Behavior Pool.





Generate ABAP Repository Objects

**Configure Generator**  
Specify values for the generator, 'OData UI Service'

**Details for RAP artifacts generation**

type filter text

- General
- Business Object
  - Data Model
  - Behavior
- Service Projection
  - Service Projection Entity
  - Service Projection Behavior**
- Business Service
  - Service Definition
  - Service Binding

Service Projection Behavior

Behavior Implementation Class: \*

< Back Next > Finish Cancel

- **Service Definition:** Cambiar el nombre del Service Definition.

Generate ABAP Repository Objects

**Configure Generator**  
Specify values for the generator, 'OData UI Service'

**Details for RAP artifacts generation**

type filter text

- General
- Business Object
  - Data Model
  - Behavior
- Service Projection
  - Service Projection Entity
  - Service Projection Behavior
- Business Service
  - Service Definition**
  - Service Binding

Service Definition

Provide the name for the service definition. It defines the CDS entities that are part of the data model to be exposed as a business service.

Service Definition Name: \*

< Back Next > Finish Cancel

- **Service Binding:** Cambiar el nombre del Service Binding y por defecto viene la opción Binding Type con el valor OData V4 - UI.



**Generate ABAP Repository Objects**

**Configure Generator**  
Specify values for the generator, 'OData UI Service'

**Details for RAP artifacts generation**

type filter text

- General
- Business Object
  - Data Model
  - Behavior
  - Service Projection
  - Service Projection Entity
  - Service Projection Behavior
- Business Service
  - Service Definition
  - Service Binding**

**Service Binding**  
Set the properties of the service binding. It exposes the service definition as a business service using the configured binding type.

Service Binding Name: \*

Binding Type: \* OData V4 - UI

< Back Next > Finish Cancel

- Por último se mostrará una ventana con una lista de los elementos que serán generados. Donde en la parte inferior de esta se podrá ver una previsualización de cualquier elemento seleccionado de la lista. Para finalizar con el proceso asignando la tarea a una orden de transporte.

**Generate ABAP Repository Objects**

**ABAP Artifacts Generation/Modification List**  
List of repository objects that are going to be generated

type filter text

Repository Object Na...	ABAP Object Type	Operation	Notes
Service Artifacts			
Business Service			
Service Definition	Service Definition	CREATE	
Service Binding	Service Binding	CREATE	Preview is not availabl...
Business Object			
Data Definition	Data Definition	CREATE	
Metadata Extension	Metadata Extension	CREATE	
Behavior Definition	Behavior Definition	CREATE	
Class	Class	CREATE	
Business Object			
Data Definition	Data Definition	CREATE	
Behavior Definition	Behavior Definition	CREATE	
Class	Class	CREATE	
Data Base Mode			

**Preview:**

```
@EndUserText.label : 'Service Definition for
@ObjectModel.leadingEntity.name : '
define service provider
contracts odata_v4_ui {
  expose ;
```

< Back Next > Finish Cancel

Después de finalizar el proceso el Service Binding estará activo, aunque se necesitará que se publique de manera manual en el endpoint correspondiente. Esto debido a que cada compañía es



responsable de dar a conocer o exponer la información de su propio sistema.

### 1.3. Análisis y ajustes

El RAP Generator es una herramienta esencial para automatizar la creación de artefactos en aplicaciones RAP, pero requiere un análisis y ajuste posteriores para asegurar su optimización. Mediante la revisión de los objetos generados, la eliminación de campos innecesarios y la mejora de la interfaz de usuario, se puede lograr una aplicación más funcional y centrada en el usuario. Este proceso asegura que los datos se presenten de manera coherente y eficiente, mejorando la experiencia general del usuario final.

#### Algunos puntos importantes a considerar:

- Identificar campos que no son útiles para los usuarios, como el UUID, y eliminarlos de la pantalla de selección a través de las metadata extensiones.
  - Cambiar para una prueba inicial el tipo de control de acceso generado por la herramienta a **#NOT\_REQUIRED**, en la entidad raíz y su vista de proyección. Aunque siempre es posible cambiar el tipo más adelante para agregar los controles de accesos necesarios para la aplicación.
  - Por defecto en el Service Definition se establece con un contrato del tipo **odata\_v4\_ui** aunque siempre es posible cambiar el tipo de contrato por **odata\_v4\_webapi** para relacionarlo en la generación del respectivo Service Binding para las entidades que expone ese servicio.
- 
- Recordando que se puede realizar una previsualización de la aplicación por medio del Service Binding.



**Service Version Details**  
View information on selected service version

Service Information

Service URL:

type filter text

Entity Set and Association

[Create Fiori Project...](#)

[Preview...](#)