



Teoría

## **BIL – Behavior Implementation Language**

---

ABAP RESTful – Arquitectura Cloud





## Contenido

<b>1. BIL – Behavior Implementation Language</b>	<b>3</b>
<b>1.1. Behavior Pool – Clase ABAP</b>	<b>3</b>
<b>1.2. Behavior Handler – Clase Local</b>	<b>6</b>



## 1. BIL – Behavior Implementation Language

### 1.1. Behavior Pool – Clase ABAP

El Behavior Pool es una clase ABAP que contiene la lógica de programación asociada a la manipulación de entidades de negocio. Esta clase es creada y definida a partir de la definición del comportamiento (Behavior Definition), y es responsable de implementar los métodos necesarios para gestionar operaciones como creación, actualización, eliminación, validaciones, determinaciones y otros aspectos importantes de la entidad de negocio.

Por defecto, el nombre de la clase se genera automáticamente en el encabezado del Behavior Definition de la entidad raíz durante su creación. Este sigue la nomenclatura que empieza con un prefijo estándar **ZBP\_**, seguido del nombre del Behavior Definition, por ejemplo, **ZBP\_BEHAVIOR\_NAME**. Sin embargo, este nombre puede ser personalizado según las convenciones de nomenclatura de la organización o preferencias del desarrollador.

**Ejemplo:**

```
managed implementation in class ZBP_BEHAVIOR_NAME unique;
strict ( 2 );
with draft;
define behavior for ZDT_BEHAVIOR_NAME//alias <alias_name>
persistent table database_table_name
draft table draft_table_name
lock master
total etag LastChangedAt
etag master LocalLastChangedAt
authorization master ( instance )
{
...
}
```



La implementación del Behavior Pool puede realizarse de manera centralizada en una sola clase para una sola entidad como se ha explicado anteriormente o mediante clases separadas para múltiples entidades, dependiendo de las necesidades específicas del proyecto y las preferencias del equipo de desarrollo. Ambos enfoques tienen sus propias ventajas y desventajas que deben considerarse cuidadosamente.


**Ejemplo:**

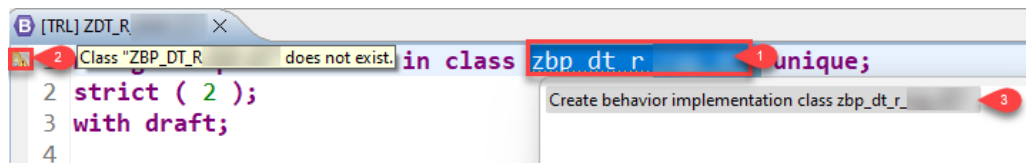
```
managed;  
strict ( 2 );  
with draft;  
define behavior for ZDT_BEHAVIOR_NAME//alias <alias_name>  
implementation in class ZBP_BEHAVIOR_NAME unique  
persistent table database_table_name  
draft table draft_table_name  
lock master  
total etag LastChangedAt  
etag master LocalLastChangedAt  
authorization master ( instance )  
{  
...  
}
```

Aunque es posible utilizar una única clase Behavior Pool para manejar múltiples entidades, se centraliza la lógica de negocio en una sola clase. Esto es útil cuando las entidades están estrechamente relacionadas y comparten mucha lógica de negocio.



## Creación de la Clase Behavior Pool:

Es esencial activar la Behavior Definition antes de proceder con la creación de la clase. Dicha clase Behavior Pool se genera automáticamente desde la definición del comportamiento utilizando un asistente (wizard). Para habilitar el asistente se debe posicionar el cursor en el nombre de la clase del Behavior Pool que se desea crear para luego presionar el icono , para seleccionar en la lista de opciones desplegables **Create behavior implementation class...**



Una vez que la clase ha sido creada, la sección de Local Types se abrirá por defecto. Esta sección contiene los métodos y tipos necesarios para todas las operaciones configuradas en el Behavior Definition de la entidad raíz, permitiendo así su implementación lógica. Es en esta área donde el desarrollador define los detalles específicos de cada operación.



Aunque los métodos son generados automáticamente, la implementación de la lógica específica es responsabilidad del desarrollador. Esto incluye la definición de la lógica de negocio que gobierna las operaciones en la entidad.

El framework RAP gestiona automáticamente muchas operaciones estándar, como create, update, y delete, una vez que la clase Behavior Pool está correctamente configurada.

AL posicionarse en la sección **Global Class**. El asistente configura inicialmente la clase como abstracta y pública, asegurando que no se pueda heredar de ella y que su definición sea global.





## 1.2. Behavior Handler – Clase Local

El Behavior Handler es una clase ABAP que encapsula la lógica de programación relacionada con la manipulación de entidades de negocio. La clase local dentro del Behavior Pool hereda de la clase estándar **CL\_ABAP\_BEHAVIOR\_HANDLER**, la cual proporciona acceso a métodos, constantes y tipos predefinidos necesarios para gestionar las operaciones especificadas en la Behavior Definition de la entidad raíz. Estas operaciones se representan como métodos dentro de la clase local, permitiendo la implementación de lógica de programación específica para cada uno de ellos.

### Ejemplo:

```

CLASS lhc_local_class_name DEFINITION INHERITING FROM
cl_abap_behavior_handler.
PRIVATE SECTION.
METHODS get_instance_features FOR INSTANCE FEATURES
IMPORTING keys REQUEST requested_features FOR database_table_name RESULT
result.
METHODS get_instance_authorizations FOR INSTANCE AUTHORIZATION
IMPORTING keys REQUEST requested_authorizations FOR database_table_name
RESULT result.
METHODS actionName FOR MODIFY
IMPORTING keys FOR ACTION database_table_name~actionName RESULT result.
METHODS actionName2 FOR MODIFY
IMPORTING keys FOR ACTION database_table_name~actionName2 RESULT result.
METHODS determinationName FOR DETERMINE ON MODIFY
IMPORTING keys FOR database_table_name~determinationName.
METHODS determinationName2 FOR DETERMINE ON SAVE
IMPORTING keys FOR database_table_name~determinationName2.
METHODS validationName FOR VALIDATE ON SAVE
IMPORTING keys FOR database_table_name~validationName.
ENDCLASS.

```