1. ¿Qué son las Badis (Business Adds-Ins)?

Son un nuevo tipo de extensión al sistema SAP, basado en la utilización de objetos ABAP para la implementación de validaciones y ampliaciones al código estándar de SAP, sin tener que modificar el código estándar.

Mediante las Badis podremos realizar modificaciones al estándar de modo de poder satisfacer los requerimientos del cliente.

Utilizando Badis podremos definir filtros para la ejecución de forma tal de poder tener diferentes procesos para, por ejemplo, países diferentes. Esto le permite a SAP poder utilizar las Badis para realizar localizaciones del sistema; por eso es que el sistema estándar ya incluye varias implementaciones de Badis

Mediante las implementaciones de las Badis se pueden hacer aplicaciones para negocios específicos como petroleras, papeleras, químicas, etc. Esto hace que las Badis sean muy útiles debido a que poseen una arquitectura Multicapa (SAP, partners, soluciones de clientes, localizaciones, soluciones específicas para industrias, etc.

*Dado que las Badis son un tipo de ampliación de SAP, es importante dejar claro que solo se implementan en el código estándar de SAP, ya que el código Z es decir el codigo que como programadores creamos, el cual no es estandar, puede ser modificado sin inconvenientes ya sea por nosotros o por cualquier otro programador.*

1. Tipos de Badis

**Badis Clásicas:** son las primeras que existieron.

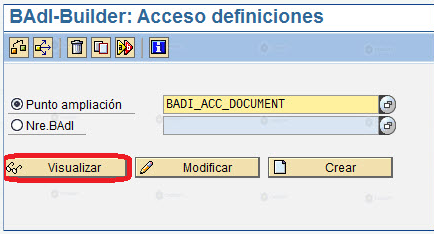
**Badis Nuevas:** también conocidas como **Puntos de ampliación o Enhancement Spot** que incorporan las últimas tecnologías de SAP.

1. **Vistas de una Badi**

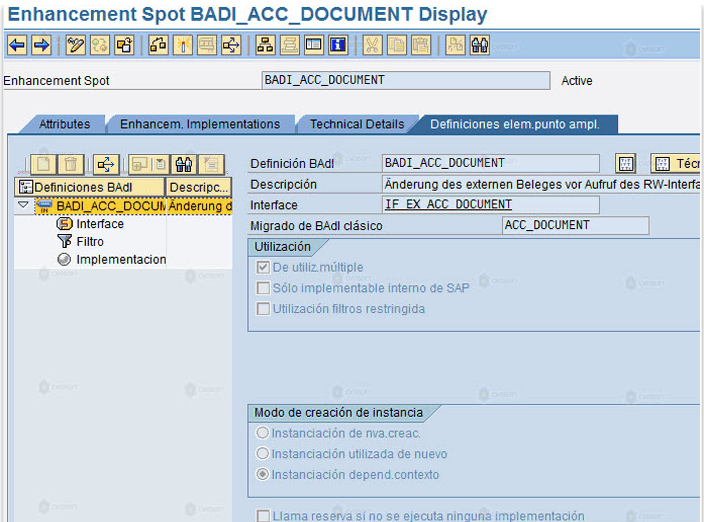
Las Badis tienen dos vistas: la **vista de definición** y la **vista de implementación**. A continuación veremos en detalle estos dos conceptos

1. **Definición de una Badi**

Mediante la transacción SE18 accedemos a la vista de definición de las Badis.

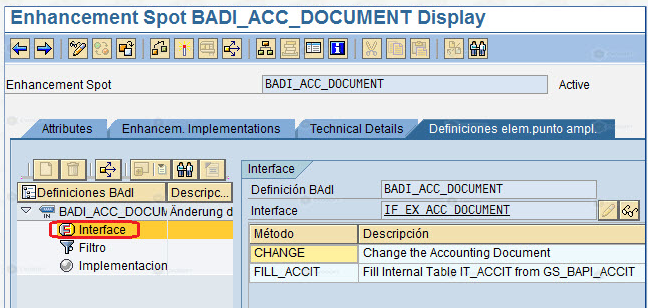


Veamos la configuración correspondiente a un **Punto de ampliación o Enhancement Spot**. En la solapa **Definiciones elem.punto ampl.** veremos las configuraciones generales de la Badi.

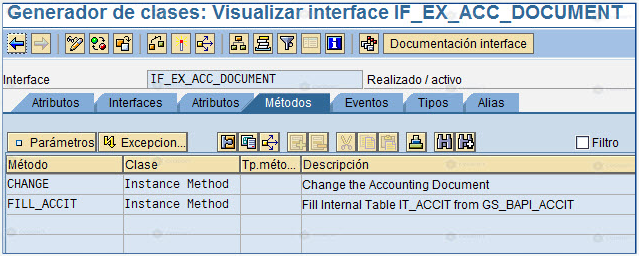


En la solapa **Enhancem.Implementations** veremos si la Badi está implementada

Si hacemos doble clic en **Interface** veremos cuál es la Interface asociada a la Badi y cuáles métodos están asociados a la definición de la Badi

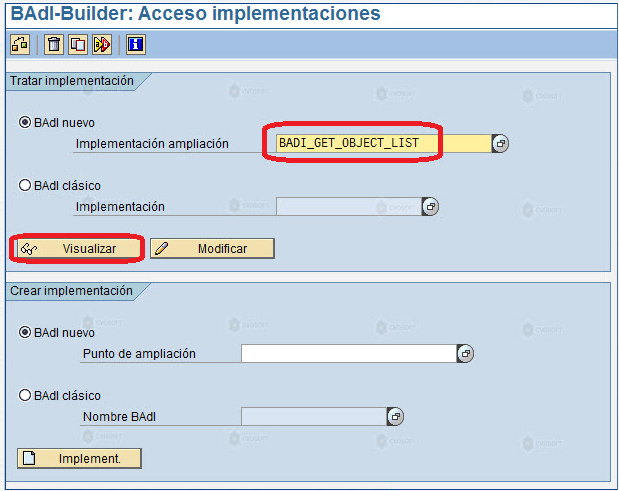


Si hacemos doble clic en un método veremos los parámetros de entrada y salida del método



1. **Implementación de una Badi**

A través de la transacción SE19 se accede a la implementación de las Badis.



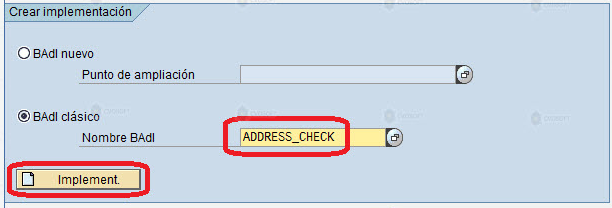
Podemos ver los métodos disponibles en la implementación. Si hacemos doble clic en los métodos, podremos ver la lógica asociada a ellos, por lo tanto se podrá elegir entre utilizar dicha lógica estándar o crear una nueva

1. Paso a paso en la creación de una implementación

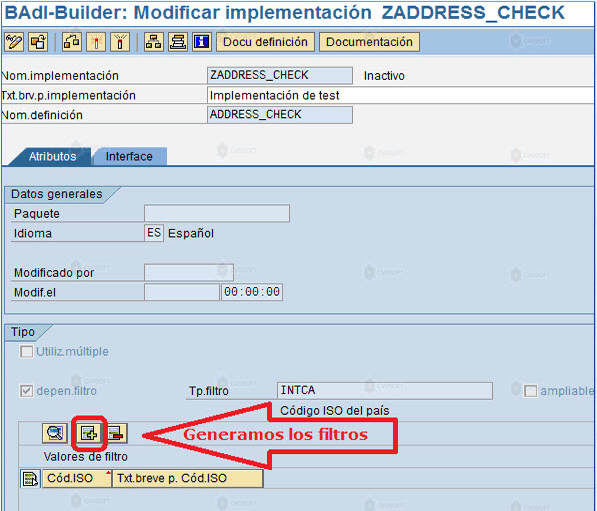
Vamos a implementar una badi que permita mostrar un mensaje cuando se están modificando los datos de dirección de un banco determinado mediante la transacción FI02 (utilizada para la modificación de los datos de los bancos existentes en el sistema SAP).

Implementaremos la badi **ADDRESS\_CHECK**, en el método **IS\_ACTIVE\_FOR\_COUNTRY**, el cual permite ampliar la verificación del país.

1. Primero que haremos es acceder a la transacción SE19, tildamos **"Badi clásico"**, completamos el nombre de la definición y presionamos el botón **Crear Implementación.**



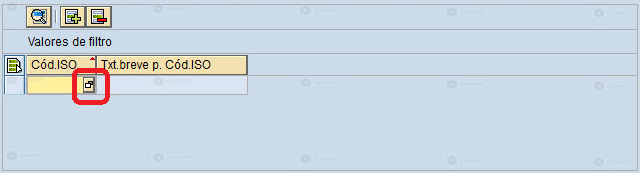
1. Luego introducimos el nombre de la implementación que será **ZADDRESS\_CHECK** y presionamos Continuar
2. En la siguiente pantalla introducimos un texto breve para la implementación. También vamos a generar los filtros con los cuales se va a ejecutar la badi.



*Sin dudas una de las grandes ventajas que presentan las Badis es la posibilidad de tener diferentes implementaciones de la misma Badi en base a los valores de los filtros.*

*Por ejemplo: Supongamos que deseamos implementar una Badi que tiene relación con un proceso contable particular de una empresa multinacional el cual es diferente según el país en donde nos encontremos. Dada esta situación podemos desarrollar una implementación diferente para cada país creando un filtro por código de país en la Badi y asignar el desarrollo de cada implementación a un programa ABAP diferente o al equipo de programación ABAP de cada país. En caso que dos países realicen el proceso contable de igual forma podremos asignar la misma implementación a diferentes valores de los filtros.*

1. Agregamos una nueva entrada en la sección **Valores de filtro**



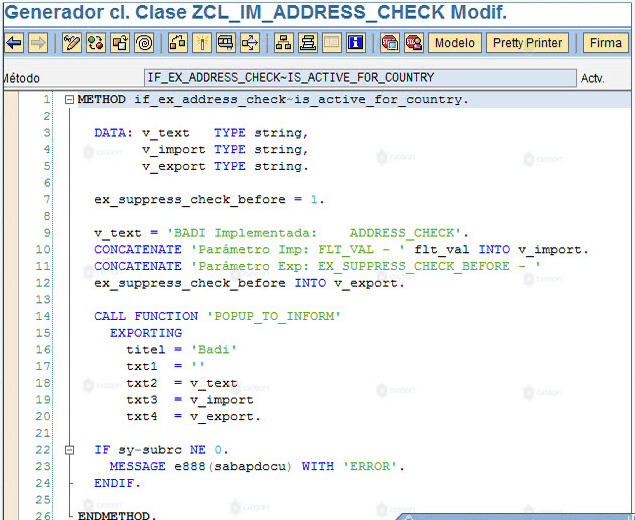
1. Generando una nueva entrada para el código de país **AR**. Es decir, la badi a implementar solo se ejecutará si el país es **AR**. Finalmente guardamos los cambios

La tabla de Valores de filtro está habilitada debido a que en la implementación de la badi está tildada la opción **dependiente filtro**.

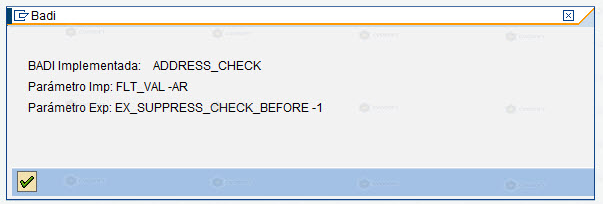
1. Introducimos el código en el método **IS\_ACTIVE\_FOR\_COUNTRY**. Para ello, accedemos a la solapa **Interface** y hacemos doble clic en el método.

Para modificar la implementación de una badi primero debemos desactivarla.

1. Dentro del método introducimos el código ABAP. Para generar el mensaje en pantalla utilizamos la función **POPUP\_TO\_INFORM.**

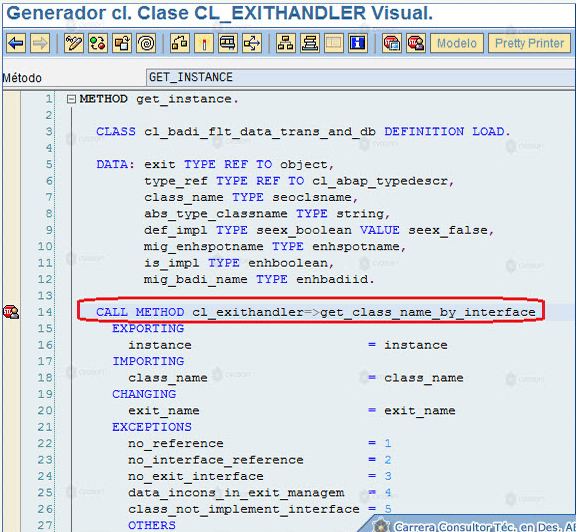
****

1. Guardamos y activamos
2. Para probar como funciona la implementación accedemos a la transacción FI02, introducimos el país del banco **AR** (Argentina), la clave de banco **014** y presionamos enter
3. En la siguiente pantalla seleccionamos el botón **Dirección o Address.**
4. Luego en el campo País elegimos **AR** y presionamos el botón Aceptar.
5. Vemos el mensaje en pantalla



**7. Método para encontrar una Badi**

Ponemos un BREAKPOINT de sesión dentro del método **GET\_INSTANCE** de la clase **CL\_EXITHANDLER.** Para acceder a una clase utilizaremos el Class Builder que es la transacción **SE24.**



Luego se ejecuta la transacción o programa que queremos verificar y el programa se detendrá en el BREAKPOINT, en la variable **EXIT\_NAME** se mostrarán los nombres de las Badis que implementa el programa.

Este método puede resultar un poco engorroso, ya que se detiene muchas veces la ejecución, pero nos aseguraremos que si una Badi existe se detendrá en ese punto.

Resumen: Ponemos un BREAKPOINT en el momento en que SAP verifica si una definición de Badi tiene alguna implementación activa; por lo tanto, aunque no exista ninguna implementación activa para esa Badi particular el método detectará que existe una Badi en ese punto

8. Diferencias entre las BADIS y los Customer Exits

* Pueden existir varias implementaciones activas en el mismo sistema de una Badi pero solo una de un Customer Exit.
* Mientras que los Customer Exits utilizan módulos de funciones, las Badis utilizan métodos de objetos que se disparan en determinados momentos de los procesos.
* Las Badis poseen una arquitectura Multicapa (SAP, partners, soluciones de clientes, localizaciones, soluciones específicas para industrias, etc.) mientras que los Customer Exits son doble capa únicamente (SAP y soluciones de cliente).