

UNIDAD

14

## DIPLOMATURA EN PROGRAMACION ABAP

MÓDULO 14: BAPI

---

BAPI

## Business Application Programming Interface (BAPIs) de SAP R/3

Este módulo introduce los conceptos relacionados con la librería de aplicaciones de negocios de **SAP**, conocida como *BAPI*.

### Introducción

A través de la transacción BAPI, accederemos al browser de Bapis disponibles en el Sistema. La arquitectura abierta basada en componentes mediante la cual los componentes software de Sap R/3 interactúan y se integran con otros componentes de software de terceros fabricantes está confirmándose como una alternativa con grandes posibilidades.

Los Objetos de Negocios de Sap R/3 conforman una parte esencial del marco de trabajo del mundo de los negocios. Estos objetos cubren un amplio rango de los procesos de datos y negocio de Sap R/3 y pueden ser implementados usando métodos estables y estandarizados, las BAPIs o Business Application Programming Interfaces.

La aparición de BAPIs en Sap R/3 fue a partir de la versión 3.1 y se hizo para permitir la integración de software entre Sap R/3 y otros fabricantes de software. En número total de BAPIs se ha ido incrementando con la liberación de nuevas versiones de Sap R/3. Los conceptos técnicos de las BAPIs se circunscriben a los de la Programación Orientada a Objetos, por esta razón al sumergirnos en el entorno de BAPIs escucharemos con mucha frecuencia los términos siguientes:

- Objeto
- Propiedad
- Atributo
- Método
- Evento

- Herencia
- Polimorfismo
- Instancia

### ***Repositorio de Objetos de Negocio***

La totalidad de objetos de negocios de Sap R/3 están identificados y descritos en el Repositorio de Objetos de Negocio, este repositorio fue originalmente introducido con la versión 3.0 en el mismo momento en que se introdujeron los Objetos de Negocio y el SAP Business Workflow.

El Repositorio de Objetos de Negocio incluye dos categorías de tipos de objetos:

- Tipos de Objetos de Negocios
- Tipos de Objetos Técnicos

Con el surgimiento de las BAPIs en la versión 3.1, el Repositorio de Objetos de Negocios asumió un nuevo e importante rol. Ahora es el punto central de acceso a los objetos de negocio y sus BAPIs para las aplicaciones externas. En este sentido el Repositorio de Objetos de Negocio sirve para dos propósitos esenciales:

- Identificar y describir los Objetos de Negocio de Sap y sus BAPIs disponibles
- Crear las instancias de objetos

El runtime del Repositorio recibe peticiones para crear objetos en tiempo de ejecución, desde los programas de aplicación y crea las instancias correspondientes.

### ***BAPI***

Los objetos contenidos en el repositorio encapsulan los datos y procesos. De esta forma el acceso a los datos y procesos, por agentes externos, es permitido solo mediante métodos específicos; representados estos por su BAPI correspondiente. De esto último se desprende que una BAPI no es más que un método de un Objeto de Negocio de Sap R/3.

Por ejemplo, la funcionalidad implementada para el objeto tipo "Customer" incluye una verificación para la existencia del cliente. De esta forma el tipo de objeto "Customer" posee una BAPI llamada "Customer.CheckExistence".

Para que un programa de aplicación sea capaz de usar la BAPI de un método, solo necesita saber cómo invocar el método; esto, es necesario conocer la definición de la interface del método. La interface de una BAPI se define por:

- Parámetros de Entrada( Import Parameters): Contiene datos a ser pasados desde el programa llamante a la BAPI
- Parámetros de Salida( Export Parameters): Contiene los datos que la BAPI pasa al programa invocante.
- Tablas de Entrada/Salida( Import/Export Table Parameters)

En la actualidad las BAPIs del sistema Sap R/3 se implementan como módulos de función que se tratan con el Function Builder del Workbench ABAP. Cada módulo de función que da soporte a una BAPI debe cumplir los requisitos siguientes:

- Soportar el Protocolo Remote Function Call( RFC )
- Haber sido asignada como un método a un Objeto de Negocio Sap
- Debe procesarse sin retornar ventanas de diálogo al programa llamante

### ***Ventajas de usar BAPI***

Las BAPIs son métodos estandarizados de los Objetos de Negocio Sap R/3, que permiten la integración de los componentes de software de los clientes y de terceros. Algunas de las ventajas de usar BAPIs son:

- Es un Estándar de Negocio
- Es un Estándar Consensuado
- Garantiza estabilidad y compatibilidad futura
- Garantiza orientación a objetos
- Independencia de entorno( apertura )

### ***BAPIs en programas de aplicación***

Actualmente una BAPI puede ser invocada de dos formas diferentes:

- Llamar la BAPI en el repositorio
- Hacer llamada RFC a la función que implementa la BAPI

### ***Acceso a una BAPI mediante llamadas a Módulos de Función***

Se pueden hacer llamadas RFC directas a los módulos de función sobre los que las BAPIs se basan. Las llamadas pueden ser hechas desde programas ABAP o desde plataformas de desarrollo externas que permitan el uso de C/C++ RFC Class Library.

Para incluir BAPIs en nuestros programas de aplicación es necesario realizar las tareas siguientes:

- Identificar los objetos y BAPIs
- Determinar la información de parámetros en torno a la interface de la BAPI
- Incluir la llamada a BAPI o Función y la declaración de parámetros

### ***BAPIs Estándares***

Existe un conjunto de BAPIs que ofrecen un servicio similar a los distintos objetos de negocio Sap, la función exacta de la BAPI la determina el objeto al cual pertenece. Ejemplos de estas BAPIs son las siguientes:

- GetList
- GetDetail
- CreateFromData
- Create
- Change
- Delete

### **Restricciones de las BAPIs**

- Garantizar la consistencia en la Base de Datos
- Jamás retornar ventanas de diálogo
- Verificar niveles de autorización
- Las BAPIs retornan los importes con cuatro decimales
- Las BAPIs retornan las fechas en formato YYYYMMDD
- Generalmente se procesan de forma sincrónica
- Programas que hacen llamadas RFC no pueden asignar valores a los atributos

### **BAPI Browser**

Los Objetos y BAPIs de Sap R/3 son manejados en el Repositorio de Objetos mediante una estructura jerárquica que se corresponde con la jerarquía de las aplicaciones de negocio de SAP R/3.

El BAPI Browser muestra los objetos en la jerarquía de las aplicaciones. Para acceder al BAPI Browser se debe seguir la ruta: Tools-Business Framework-BAPI Browser o por la transacción BAPI o si se conoce el método de las BAPIs a través de la SE37.

### **Información de Parámetros Requeridos**

- Key Fields
- Parámetros( Import, Export o Import/Export )
- Nombre del módulo de función, si se van a hacer llamadas RFC

Una forma, de las varias existentes, para obtener toda la información necesaria para usar las BAPIs es la siguiente:

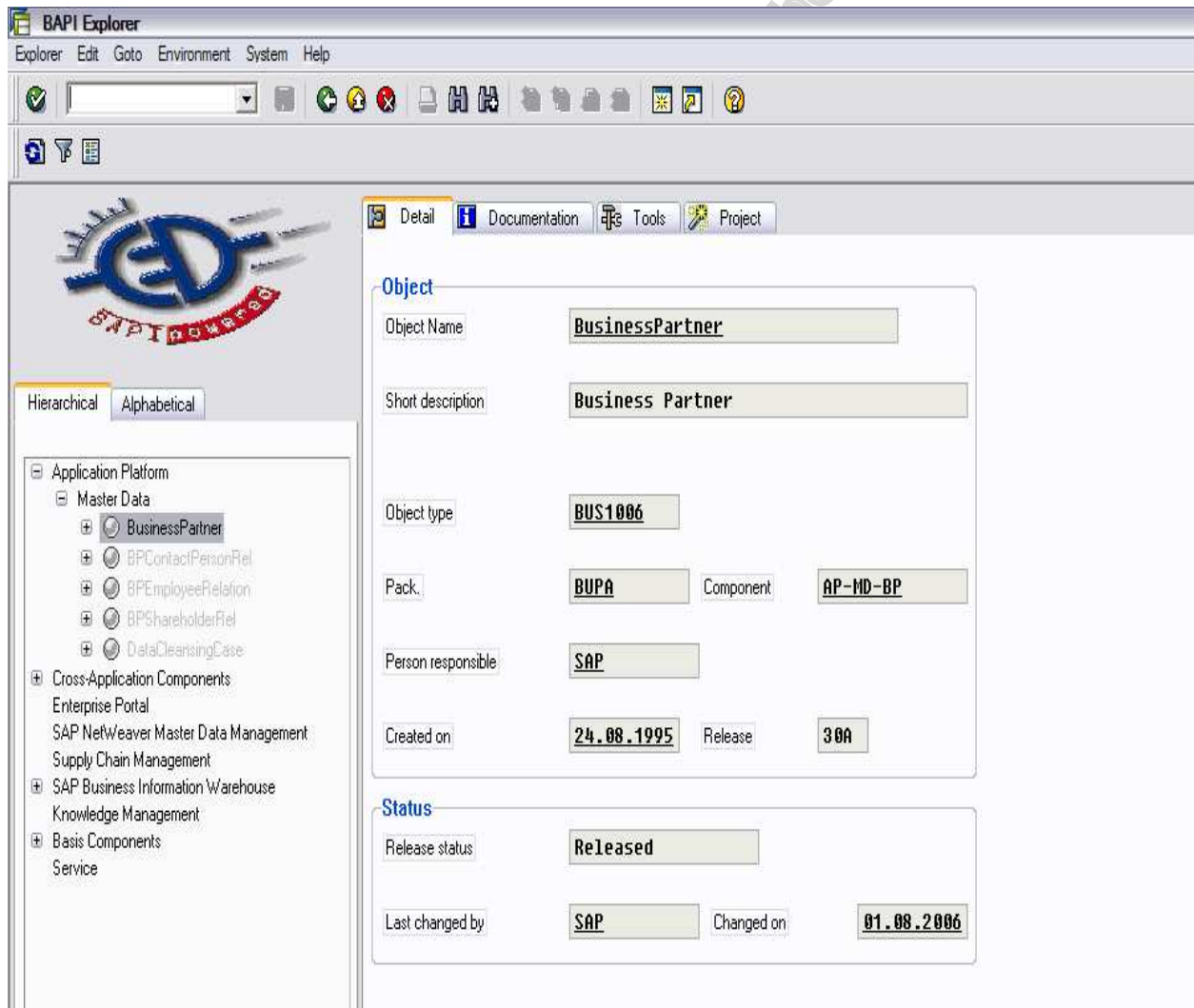
1. Acceder al Workbench ABAP
2. Seleccionar la opción Business Object Builder
3. Indicar el Nombre del Objeto, por ejemplo BUS2032 para SalesOrder, o usar el botón de comando Business Object Repository
4. Poner el puntero del ratón sobre el método que se desea analizar
5. Pulsar el botón de comando Parámetros

6. Pulsar doble-clic sobre el parámetro para ver la tabla o campo de referencia

### Documentación de las BAPIs

Cada una de las BAPIs posee su propia documentación, en la que se describe la funcionalidad y se enumeran los parámetros. Para acceder a la documentación de la BAPI se procede de la manera siguiente:

1. Abrir el objeto de negocio
2. Se expande el nodo de los métodos
3. Se pulsa sobre el icono de información



### **Modificaciones de cliente**

Las modificaciones a las BAPIs pueden ser hechas, solo, en el rango de nombres permitidos a los usuarios. Por lo que se permite:

- Crear un tipo de objeto nuevo
- Crear una BAPI nueva

### **Tipos de interfaces y BAPIs**

Los interfaces sirven para conectar un sistema SAP con otro sistema cualquiera (también puede ser SAP). Un interface es una utilidad para transferir información de forma segura y automatizada. Hay varios tipos de interfaces: Batch input, Bapis, Idocs, RFCs y Guis.

Un Batch Input es simular mediante un proceso Batch la introducción de datos en el sistema vía transacción online.

Una Bapi (Business Application Programming Interface) es una función estándar de SAP que ya hace el interface, con lo que el programador tan sólo tiene que hacer una llamada a esa función.

Al entrar un Idoc al sistema, dispara una función que hace la entrada de datos automáticamente.

Una RFC (Remote Function Call) es una llamada en un sistema (servidor) desde otro sistema (cliente), es decir, desde SAP llamamos a un trozo de código en el sistema que queremos conectar con SAP que ejecuta, dentro de SAP, la entrada de datos.

Las BAPIs, Business Application Programming Interfaces (BAPIs) son el núcleo de las herramientas de desarrollo de SAP Business Object Repository (BOR). Las BAPIs se introdujeron en ABAP a partir de la versión 3.x pero cobraron importancia en cuanto a número y funcionalidad a partir de la versión 4.x.

SAP implementa BAPIs a través de RFCs. Las BAPIs son un ejemplo perfecto de la potencia de las RFCs. Llegarán a ser más y más importantes para los programadores en la medida en que la integración de SAP con otros sistemas sea más común.

Una BAPI se define como un método de SAP Business Object. Las BAPIs posibilitan un método cómodo de acceso a los datos y funciones de SAP desde programas tanto externos como internos.



El programa que llama a la BAPI no requiere conocer cómo SAP almacena los datos o el funcionamiento interno de SAP; solo necesita conocer la interfaz de la BAPI. Esto implica conocer el significado y configuración de los parámetros y tablas que son exportadas e importadas desde la BAPI

SAP basa la tecnología actual de las BAPI en RFCs, existiendo algunas diferencias entre el módulo de una función normal RFC y un módulo de función BAPI RFC, tales como:

- Las BAPIs RFC se registran como métodos dentro del BOR.
- Las BAPIs RFC no pueden contener pantallas o módulos de diálogo.
- Las BAPIs RFC no contienen el comando ABAP “COMMIT WORK”
- Las BAPIs RFC deben seguir un estricto conjunto de reglas que los nombres de las propias BAPIs, los nombres de los parámetros, las estructuras y la documentación de éstas. Esto fuerza la consistencia entre todas las BAPIs.

Las BAPIs se integran en SAP a través del BOR. Esta definición se puede encontrar en la transacción de código BAPI.

Expandiendo la estructura de árbol para los campos claves y los métodos podemos dirigir las asociaciones entre las BAPIs y sus parámetros asociados

El mundo de las BAPIs está cambiando muy rápidamente, por lo que uno de los mejores caminos para recoger la última información acerca del uso de las BAPIs es Internet y las revistas técnicas. Las siguientes secciones describen dos recursos donde podemos encontrar información.

### ***SAP's BAPI Network Home Page***

En la web de SAP América, <http://www.sap.com/> podemos encontrar una serie de documentos relacionados con las BAPIs, tanto a nivel introductorio como a nivel avanzado, acerca del uso y desarrollo de estas. Además podemos darnos de alta en una lista de correo que proporciona información acerca de las cosas que suceden en el mundo de las SAP BAPI.

En la web de SAP Technical, localizada en: <http://www.saptechnical.com>, también podemos encontrar gran cantidad de información técnica acerca de las BAPIs.

### **Ejemplo de RFC usando BAPIs**

Los siguiente son ejemplos de cómo realizar llamadas a funciones remotas BAPI. Una de las ventajas de las BAPI RFCs es que ellas no necesitan ser usadas como los RFCs. Si queremos ejecutar una función que tiene construida una llamada BAPI, podemos usar ésta directamente en el código ABAP.

Ejemplo 1:

- **Obteniendo datos del Maestro de Materiales.**  
Este ejemplo recupera datos para un número de material específico.

Código:

```
DATA: WS_MATNR LIKE BAPIMATDET-MATNR,  
STR_RETVAL LIKE BAPIMATDOA,  
BAPI_RC LIKE BAPIRETRUN,  
WS_MESS(255).  
WS_MATNR = 'TSTMAT'.  
CALL FUNCTION 'BAPI_MATERIAL_GET_DEATIL'  
DESTINATION 'PLUTO'  
EXPORTING  
MATERIAL = WS_MATNR  
IMPORTING  
MATERIAL_GENERAL_DATA = STR_RETVAL  
RETURN = BAPI_RC  
EXCEPTIONS  
SYSTEM_FAILURE = 1  
MESSAGE WS_MESS  
COMMUNICATIONS_FAILURE = 2  
MESSAGE WS_MESS.  
IF SY-SUBRC EQ 1.  
WRITE: /001 'RFC System Error',  
/001 WS_MESS.  
EXIT.  
ENDIF.  
IF SY-SUBRC EQ 2.  
WRITE: /001 'RFC Communications Error',  
/001 WS_MESS.  
EXIT.  
ENDIF.  
WRITE: /001 STR_RETVAL-MATNR,  
020 STR_RETVAL-SPART,  
025 STR_RETVAL-MAKTX.
```

Ejemplo 2:

- **Obteniendo una lista de Materiales.**

Código:

```
DATA:INT_MATLST LIKE BAPIMATLST OCCURS 50 WITH HEADER LINE,
INT_MATSEL LIKE BAPIMATRAM OCCURS 10 WITH HEADER LINE,
INT_BAPI_RET LIKE BAPIRET2 OCCURS 10 WITH HEADER LINE,
WS_MESS(255).
INT_MATSEL-LOW = 'B*'.
INT_MATSEL-SIGN = 'I'.
INT_MATSEL-OPTION = 'CP'.
APPEND INT_MATSEL.
CALL FUNCTION 'BAPI_MATERIAL_GETLIST'
DESTINATION 'PLUTO'
EXPORTING
MAXROWS = '20'
TABLES
MATNRSELECTION = INT_MATSEL
MATNRLIST = INT_MATLST
RETURN = INT_BAPI_RET
EXCEPTIONS
SYSTEM_FAILURE = 1
MESSAGE WS_MESS
CUMMINICATIONS_FAILURE = 2
MESSAGE WS_MESS.
IF SY-SUBRC EQ 1.
WRITE: /001 'RFC System Error',
/001 WS_MESS.
EXIT.
ENDIF.
IF SY-SUBRC EQ 2.
WRITE: /001 'RFC Communications Error',
/001 WS_MESS.
EXIT.
ENDIF.
LOOP AT INT_MATLST.
WRITE: /001 INT_MATLST-MATNR,
/020 INT_MATLST-MAKTX.
ENDLOOP.
LOOP AT INT_BAPI_RET.
WRITE: /001 INT_BAPI_RET-MESSAGE.
ENDLOOP.
```

-----